EECS 391: Introduction to AI

Soumya Ray

Website: http://vorlon.case.edu/~sray/eecs391_sp12/index.html

Email: sray@case.edu

Office: Olin 516

Office hours: Tue 2:30-4:00 or by appointment

Announcements

- PA4 online later today
- Extra meeting tomorrow 3-4pm here

Today

- Bayesian networks (Ch 14)
- Dynamic Bayesian networks (Ch 15.5.1, 15.5.3)

Bayesian network Assumption

- For an arbitrary DAG,

$$\Pr(x_1,\ldots,x_n)$$

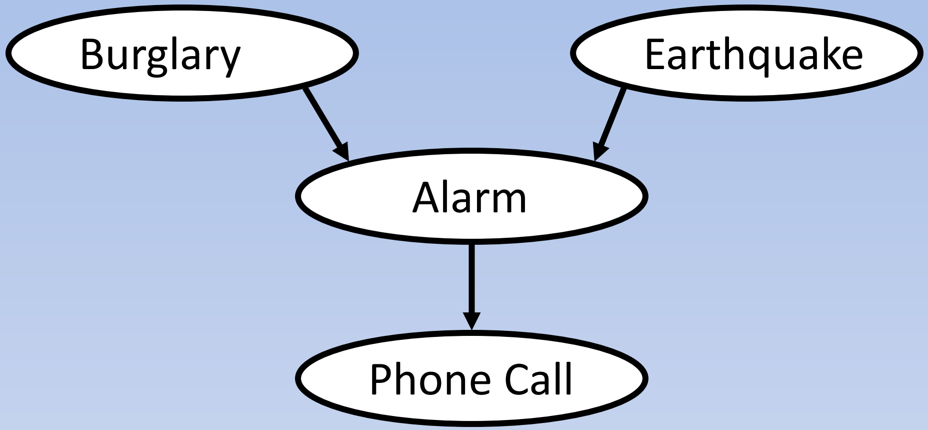
$$=\Pr(x_n)\prod_{i=1}^{n-1}\Pr(x_i\mid\{x_j\}_{j=i+1}^{n})$$

$$\boxed{=\prod_{i=1}^{n}\Pr(x_i\mid Pa(x_i))\qquad\text{By BNA}}$$

Example (J. Pearl et al.)

- My house has a sensitive burglar alarm which occasionally also goes off if there is an earthquake. If the alarm goes off, my neighbor might call and tell me about it.
- How to describe this with a BN?

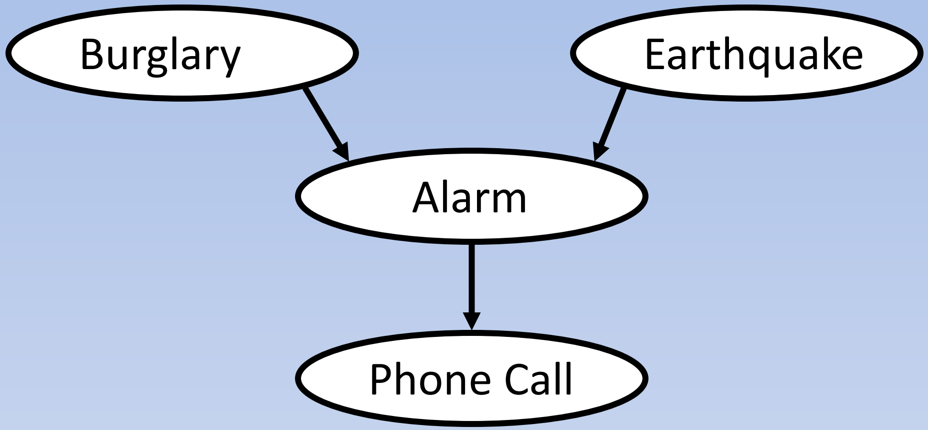
Alarm network



Pr(B,E,A,P)=?

Example: Wumpus World

Explaining Away

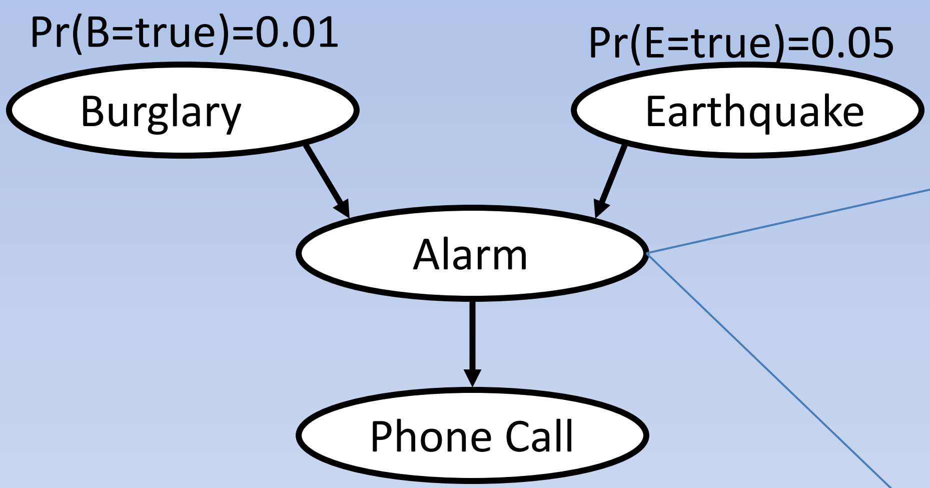


- When Alarm is unknown, Burglary and Earthquake are independent
- But if the Alarm goes off, then they become dependent because they “compete” to explain the Alarm

Representing Probabilities

- How to represent $Pr(x_i|Pa(x_i))$?
  - (Assume all the r.v.’s are discrete)
- Often represented as a table, the “**conditional probability table**”
- For each value of x_i and $Pa(x_i)$, write down the probability

CPTs Example



Burglary	Earthquake	A=true
False	False	0.0001
False	True	0.05
True	False	0.8
True	True	0.9

Alarm	P=true
False	0.1
True	0..75

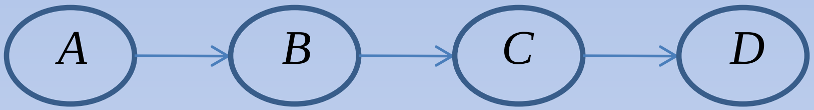
Inference in Bayesian Networks

- Two kinds of algorithms:
  - Exact
    - Always returns exact answer, but may take a long time
  - Approximate
    - Returns approximate answer. More time=better answers

Exact Inference in BNs

Variable Elimination

- Suppose we had the BN:



- And we want $Pr(D)$
- Well, we know $Pr(A)$
- We can calculate $Pr(B)$ using $Pr(B|A)$ and $Pr(A)$: $\Pr(B=b)=\sum_a \Pr(B=b \mid A=a)\Pr(A=a)$
- Then we calculate $Pr(C)$ and $Pr(D)$ like this as well

Variable Elimination

- Or:

$$\Pr(D) = \sum_{A,B,C} \Pr(A, B, C, D)$$

$$= \sum_{A,B,C} \Pr(A)\Pr(B \mid A)\Pr(C \mid B)\Pr(D \mid C)$$

$$= \sum_{C} \Pr(D \mid C) \sum_{B} \Pr(C \mid B) \sum_{A} \Pr(B \mid A)\Pr(A)$$

Each term here is a table, called a “factor”. A factor may not be a probability distribution (though in this case it is). Notice that factors are computed by eliminating variables. The efficiency of VE comes from “pushing in” the sums as far as possible.

Variable Elimination

- Basic exact inference algorithm---inference by enumeration, but taking advantage of the graph structure
- Order the variables in the network with the variable(s) in the query coming last
- At each step, we will “sum out” one of the variables and cache the result
- The efficiency of this procedure depends on the order of the variables
  - Finding an optimal order is NP-complete

Summing out a variable

- Multiply all the tables involving this variable
- Then add the rows where this variable is the only one changing and the others are fixed
- This gives a new table over the other variables alone
  - This new table may not be a probability distribution
  - Sometimes called “factor” or “potential”

Incorporating evidence

- If we know the value of a variable, just select that value instead of summing out

Approximate Inference in BNs

Approximate Inference

- Sometimes a BN can be very complex
- Sometimes we don’t really need the exact probabilities

- In these cases, we can use sampling methods to answer queries
  - Often very fast, very easy to implement
  - Convergence is only asymptotic in general

Approximate Inference 1 (Monte Carlo)

- Idea: Topologically sort the variables according to the graph structure
- Sample each according to the conditional distribution (well-defined due to the sorting)
- Count the samples with desired values
- Easy!
  - Right?

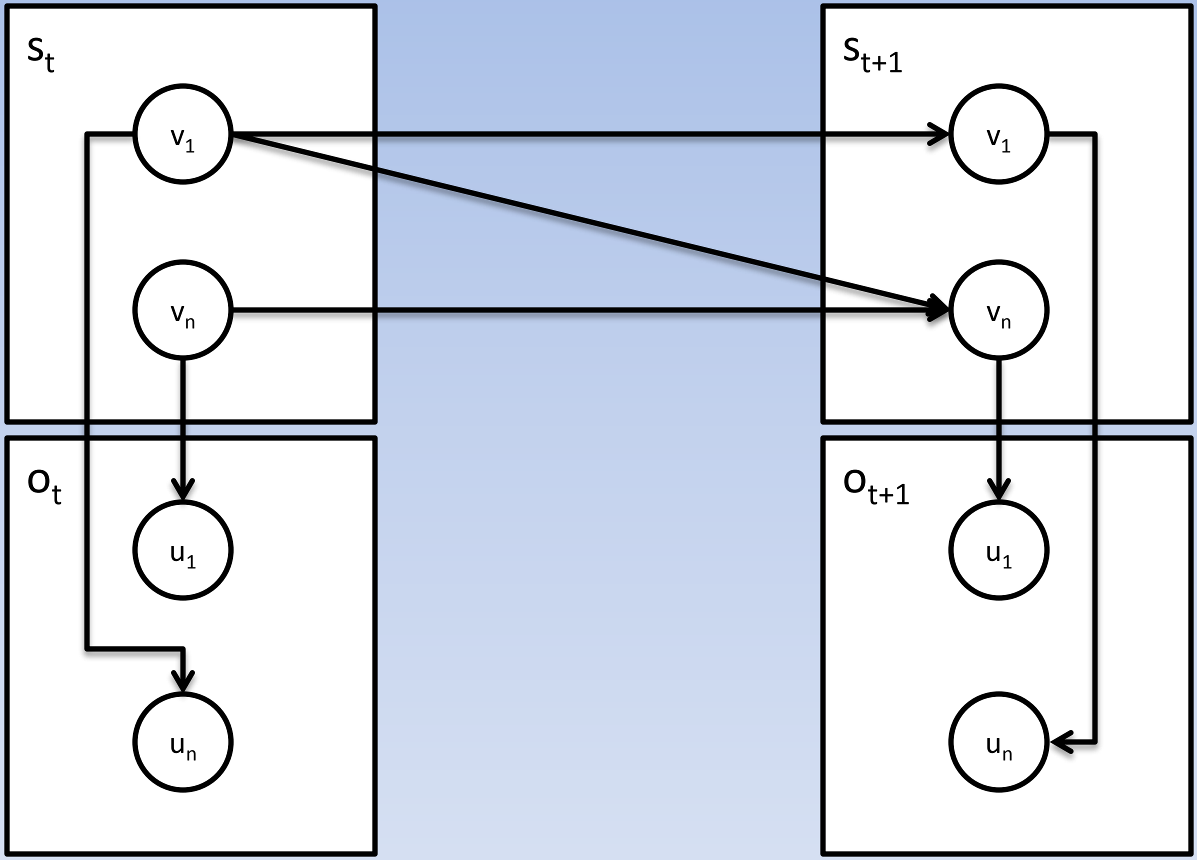
Approximate Inference 2: Incorporating Evidence

- What if we have evidence?
- Well, let’s just throw away the samples that have the evidence variables wrong
  - “Rejection Sampling”

Reasoning Over Time (Ch 15.5)

- In many problems, agents need to reason about how the states of the world evolve over time
- Suppose at time t, the agent is at state s_t and observes o_t
  - At time $t+1$, the state changes to s_{t+1} (possibly because of the agent's actions) and the agent observes o_{t+1}
- This situation can be represented with *dynamic Bayesian networks*

Dynamic Bayesian Networks



A “first order” model

Filtering in DBNs

- A key problem in temporal probabilistic inference is “filtering”: given all my previous observations, what is the probability distribution over my current state?
  - Find $\Pr(s_t|o_1,\ldots,o_t)$
- This problem is elegantly solved by a widely used and powerful approximate inference algorithm called “particle filtering”
  - “Sequential Monte Carlo” algorithm

Particle Filtering (15.5.3)

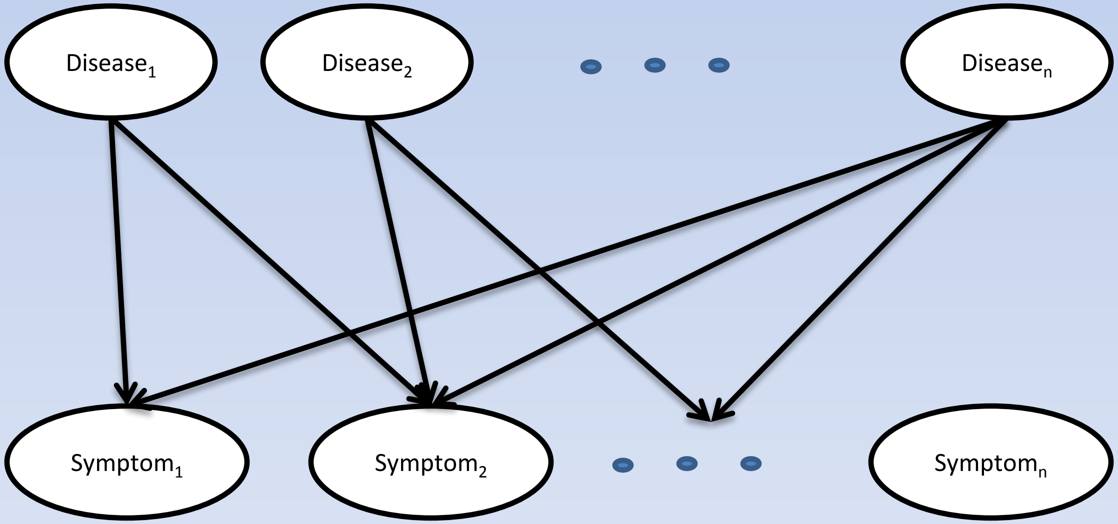
- Sample a set of N "particles" from the initial state distribution Pr(s_0)
- Propagate each particle s^i_t forward: s^i_{t+1} is a sample from Pr(s_{t+1}| s^i_t)
- Each new sample s^i_{t+1} is weighted by Pr(o_{t+1}| s^i_{t+1})
- A new population is sampled from the set of N particles (with replacement), with the probabilities according to the weights

Properties of Particle Filtering

- Particle filtering is consistent
  - With enough “particles” it converges to a true representation of $\Pr(s_t|o_1,\ldots,o_t)$
- Empirically found to be efficient
  - Not too many particles needed to model the next state distribution in general

Application 1: Medical Diagnosis

- The QMR-DT network (Shwe et al. 1991)
  - Two layer bipartite Bayesian network
  - 600 diseases, 4000 symptoms



Application 2: Intelligent Vehicles

- Particle filters are widely used in autonomous vehicles for localization, object detection and tracking

- Also widely used in object recognition in video

Summary

- Exact Inference in BNs through variable elimination
- Approximate inference through Monte Carlo and rejection sampling