Road Map

- Basics of Probability and Statistical Estimation
- Bayesian Networks
- Markov Networks (briefly; we'll come back to this)
- **Inference**
- Learning
- Semi-supervised Learning, Hidden Markov Models
- Papers on active learning

Inference: Variable Elimination

Doug Downey
Northwestern EECS 395/495 Fall 2011

Inference: Answering Queries

- Given:
  - A probability model
  - Subsets of random variables
    - ***Y*** (query) and
    - ***E*** (evidence) with assignments ***e*** to ***E***
- Find P(***Y*** | ***E*** **=** ***e***)
- E.g.,
  - P(*Battery* | *Starts* = false)
  - P(*Disease* | *Symptoms* = ***e***)
  - P(*StockMarketCrash* | *RecentPriceActivity* = ***e***)

What else can we do with queries?

- Prioritizing info gathering
  - Which additional evidence would be most informative?
- Explanation
  - Why do I need a new fan belt?
- Sensitivity Analysis
  - Which variable values are most critical?

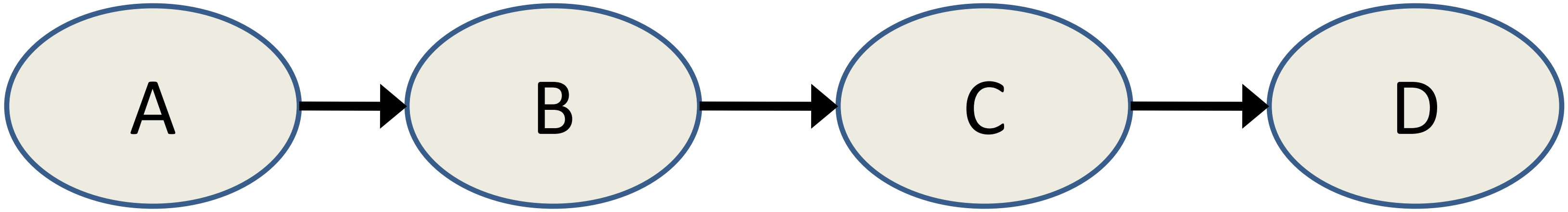
Gee, it's easy

- $P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e}) = \dfrac{P(\boldsymbol{Y}, \boldsymbol{e})}{P(\boldsymbol{e})}$

- Given joint $P(\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{w})$, we can compute r.h.s. by summing out $\boldsymbol{w}$, $\boldsymbol{y}$

But…

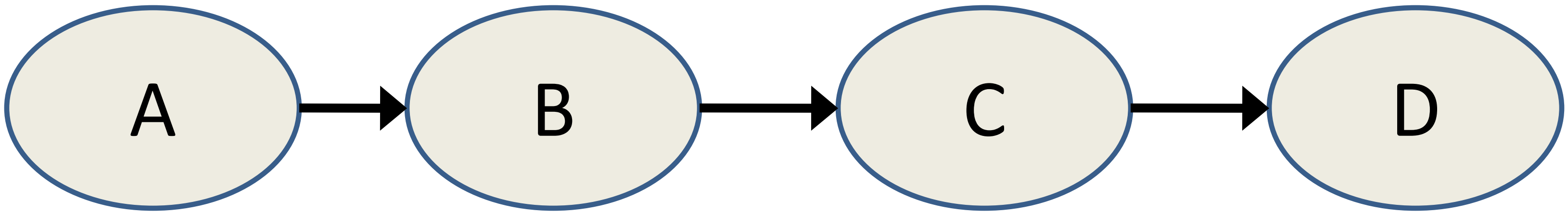
- Naïve summing is costly



  - $P(A, B, C, D) = P(A)\,P(B|A)\,P(C|B)\,P(D|C)$

- $P(D) = \Sigma_A \Sigma_B \Sigma_C\, P(A)\,P(B|A)\,P(C|B)\,P(D|C)$
  - 8 combinations, 8*3 = **24** multiplications
  - **Exponential** in # of variables

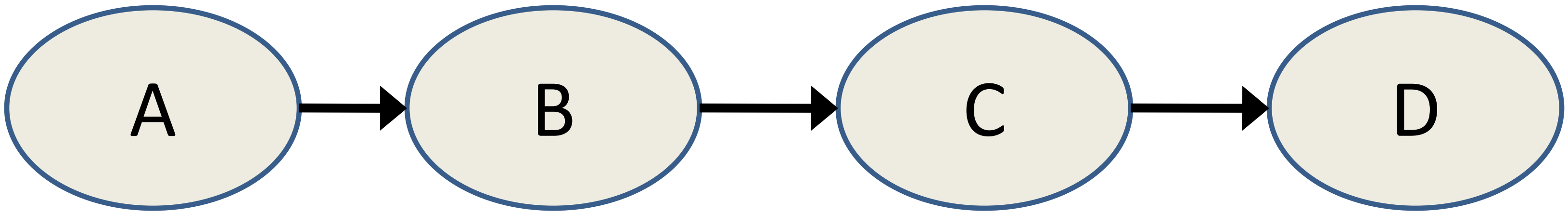
Variable Elimination



$$P(D) = \Sigma_A \Sigma_B \Sigma_C P(A)\, P(B|A)\, P(C|B)\, P(D|C)$$

$$= \Sigma_C P(D|C)\, \Sigma_B P(C|B) \underbrace{\Sigma_A P(B|A)\, P(A)}_{P(B)}$$

Variable Elimination



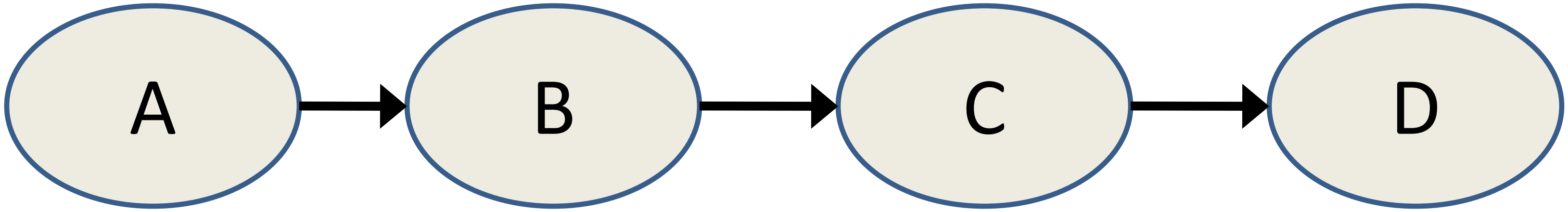
$$P(D) = \Sigma_A \Sigma_B \Sigma_C P(A)\, P(B|A)\, P(C|B)\, P(D|C)$$

$$= \Sigma_C P(D|C) \Sigma_B P(C|B) \Sigma_A P(B|A)\, P(A)$$

Has 4+4+4=**12** multiplications (vs. 24)

– For *n*-edge binary chain, only ***4n*** multiples

With evidence



$$P(D \mid A{=}a) = \Sigma_B \, \Sigma_C \, P(B \mid A{=}a) \, P(C \mid B) \, P(D \mid C)$$

$$= \Sigma_C \, P(D \mid C) \, \Sigma_B \, P(C \mid B) \, P(B \mid A{=}a)$$

Variable Elimination

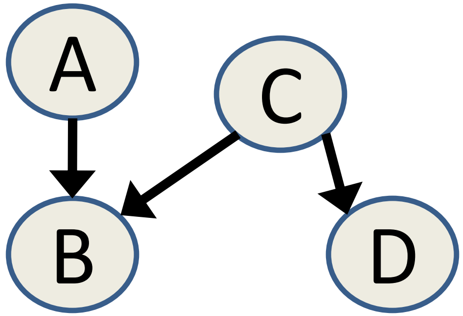
- Two steps:
  - Push summations as far as possible to right (assuming some ordering of variables)
  - Compute the sum

$$P(D|A=a) = \Sigma_B \Sigma_C P(D|C)\, P(C|B)\, P(B|A=a)$$

$$= \Sigma_C P(D|C)\, \Sigma_B P(C|B)\, P(B|A=a)$$

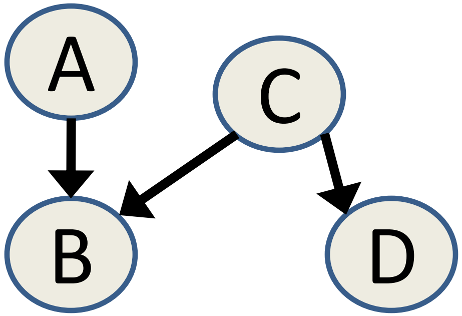
“Factors”

- $P(A, B, C, D)$
  $= \underbrace{P(A)}_{\phi_1}\ \underbrace{P(C)}_{\phi_2}\ \underbrace{P(B \mid A, C)}_{\phi_3}\underbrace{P(\mathrm{D} \mid \mathrm{C})}_{\phi_4}$



- Scope [ϕ_4] = {D, C}
- Variable Elimination: write out joint as factors
  - factor ϕ_i out of sum over X when $X \notin$ scope [ϕ_i]

Discarding non-Ancestors



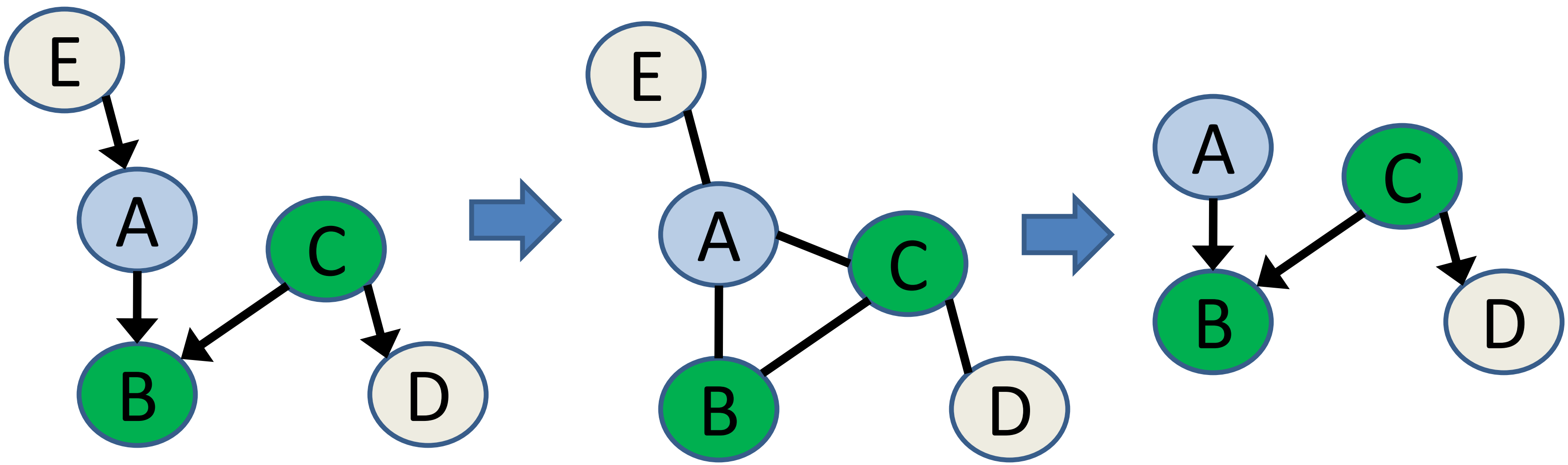
- $P(A, B, C, D)$
  $= P(A)\, P(C)\, P(B \mid A, C) P(D \mid C)$
- Query: $P(B, C \mid A\text{=a})$

  $= \Sigma_{\text{D}}\, P(\text{A=a})\, P(C)\, P(B \mid A\text{=a}, C) P(D \mid C)$

  $= P(\text{A=a})\, P(C)\, P(B \mid A\text{=a}, C)\, \Sigma_{\text{D}}\, P(D \mid C)$
- $\Sigma_{\text{D}}\, P(D \mid C) = 1$ for all C, we can ignore it
- In general: when computing $\text{P}(\boldsymbol{Y} \mid \boldsymbol{E})$ we can ignore nodes not in $Ancestors(\boldsymbol{Y}, \boldsymbol{E})$

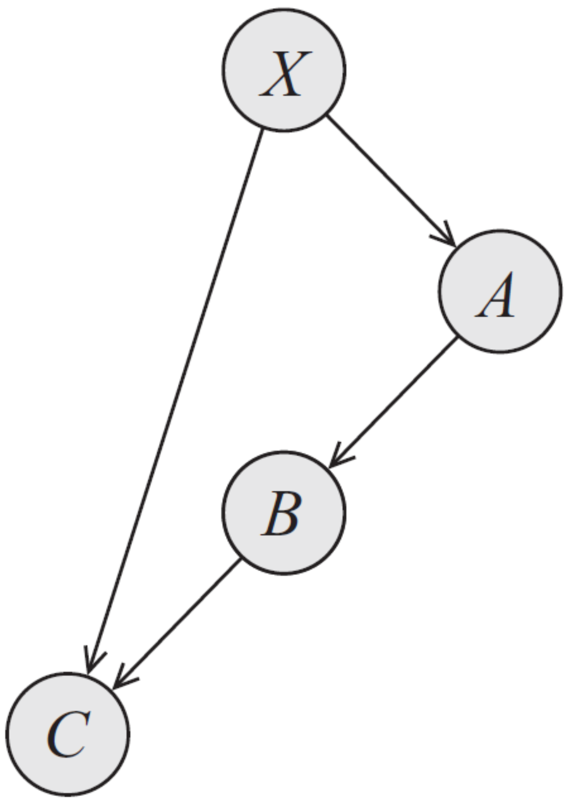
Discard by separation in Markov Network

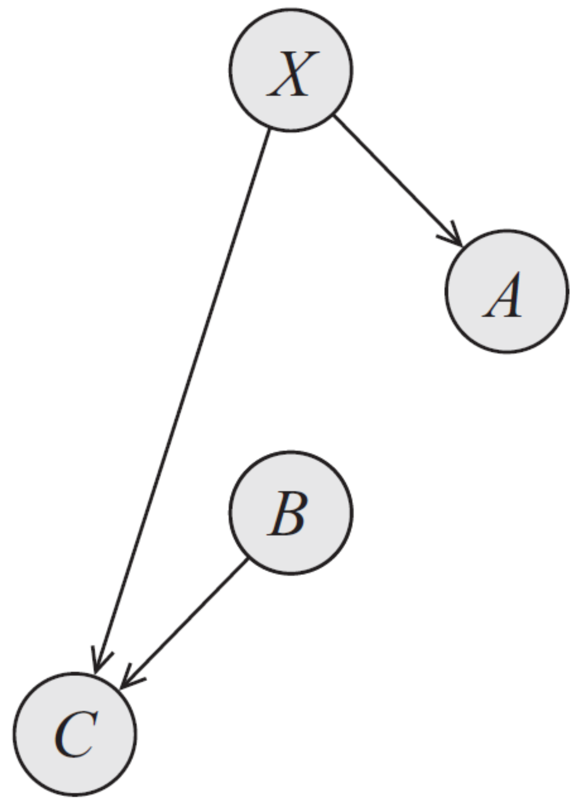
- $P(A, B, C, D, E)$
  $= P(E)\,P(A|E)\,P(C)\,P(B \mid A, C)P(D \mid C)$
- Query: $P(B, C \mid A{=}a)$
  - Throw out variables separated from query by evidence in moral graph



Semantics of summed-out factors

- Sums don't always correspond to simple conditional probabilities





Complexity of Inference

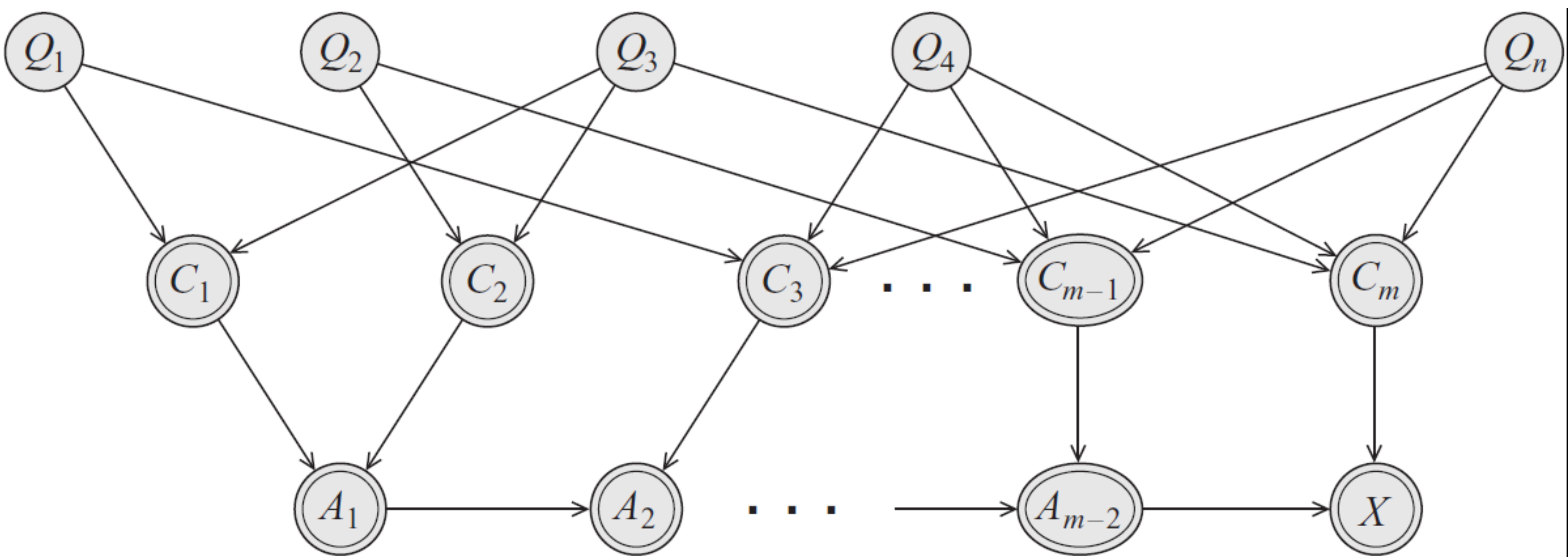
- What does variable elimination buy us?
- It depends on the network
  - If the distribution doesn’t factor well, elimination won’t help
- Generally, Bayesian Inference is hard
- NP-complete problems can be reduced to it

Reduction to Boolean Satisfiability (1)

- Boolean Satisfiability
  - Given a boolean formula in 3-CNF, e.g.:
    `(x1 v -x3 v x7) ^ (x4 v x5 v -x6) ^` …
    Is there an assignment to variables (i.e. xi = true|false) make the formula true?

Reduction to Boolean Satisfiability (2)

- **(x1 v -x3 v x7) ^ (x4 v x5 v -x6)**
  - Let Q_i **= xi**
  - C_i **=** clauses (e.g. **(x1 v -x3 v x7)**)
  - **X =** 1 *iff* all C_i are true, A_i = "and" variables



Inference complexity details

- Actually #P-complete
  - Asking for probability like counting number of satisfying assignments
- Even approximation is NP-hard
- (see book)