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
- Language models

Inference: Variable Elimination

Doug Downey
Northwestern EECS 395/495 Fall 2014

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?

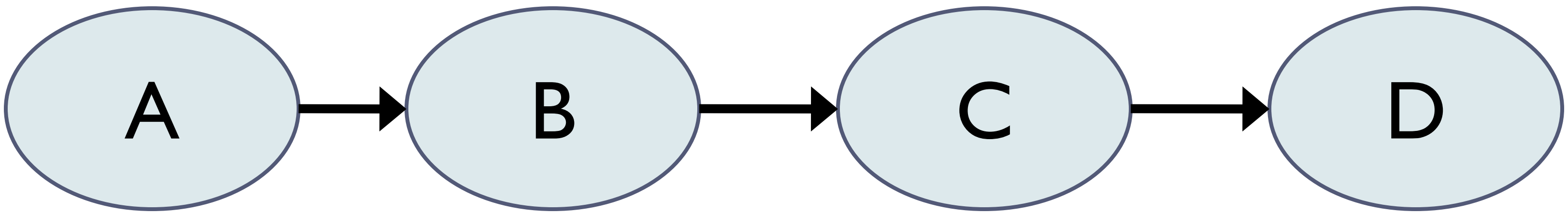
Gee, it's easy

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

- Given joint P(***y***, **e**, ***w***), we can compute r.h.s. by summing out ***w***, ***y***

But…

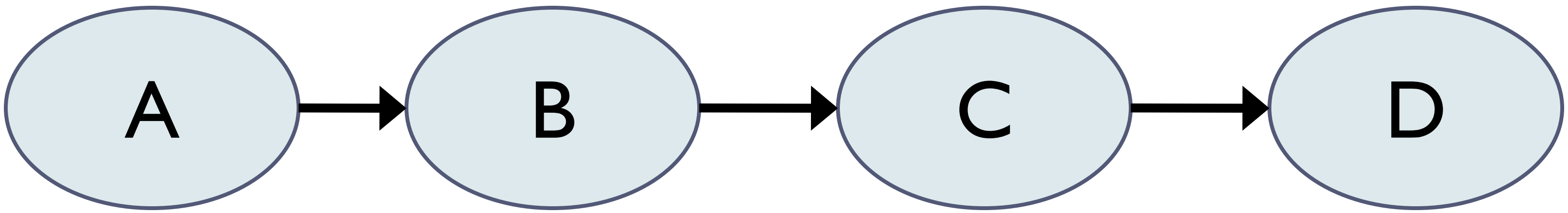
- Naïve summing is costly



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

- $P(D) = \Sigma_A \Sigma_B \Sigma_C\ P(A)\ P(B|A)\ P(C|B)\ P(D|C)$
  - $2^3 = 8$ combinations, $8*3 =$ **24** multiplications
  - **Exponential** in # of variables

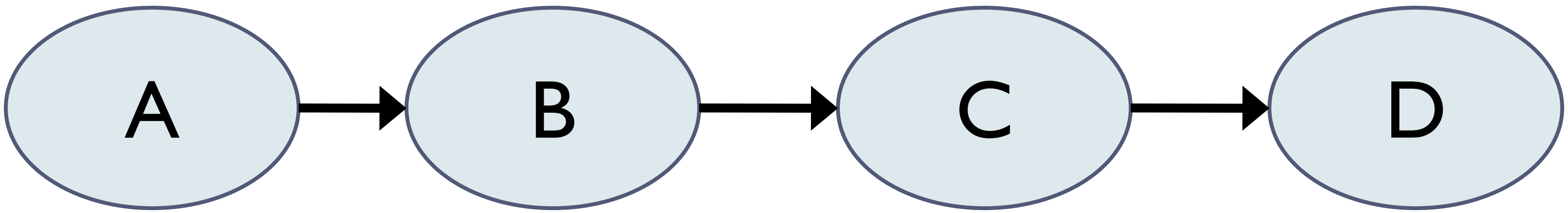
Variable Elimination



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

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

Variable Elimination



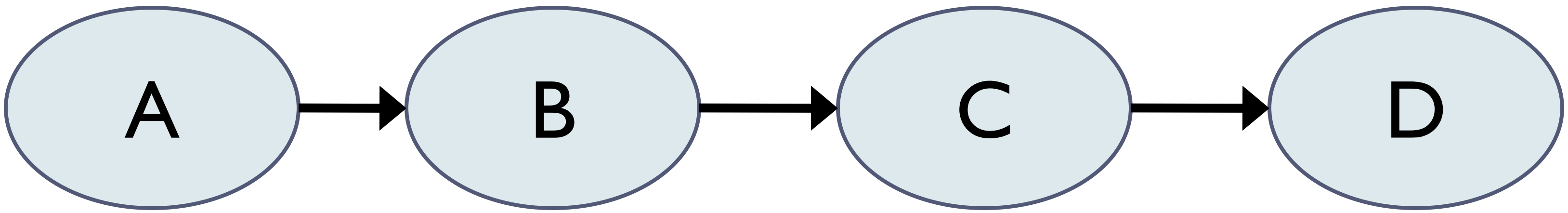
$$P(D) = \Sigma_A \Sigma_B \Sigma_C P(A) P(B|A) P(C|B) P(D|C)$$

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

Has 2+2+2=**6** multiplications (vs. 24)

- For *n*-edge binary chain, only ***2n*** multiples

With evidence



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

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

Variable Elimination

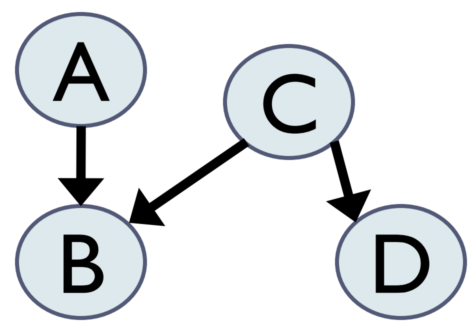
- Two steps:
  - Push summations as far as possible to right (assuming some ordering of variables)
  - Compute the sum

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

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

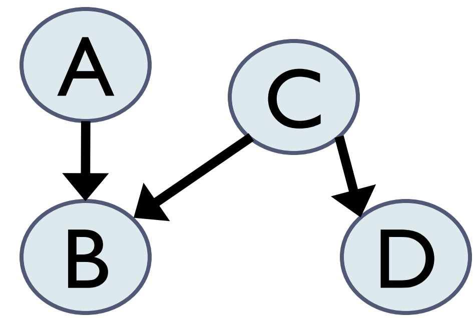
"Factors"

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



- Scope $[\phi_4] = \{D, C\}$
- Variable Elimination: write out joint as factors
  - factor ϕ_i out of sum over X when $X \notin$ scope $[\phi_i]$

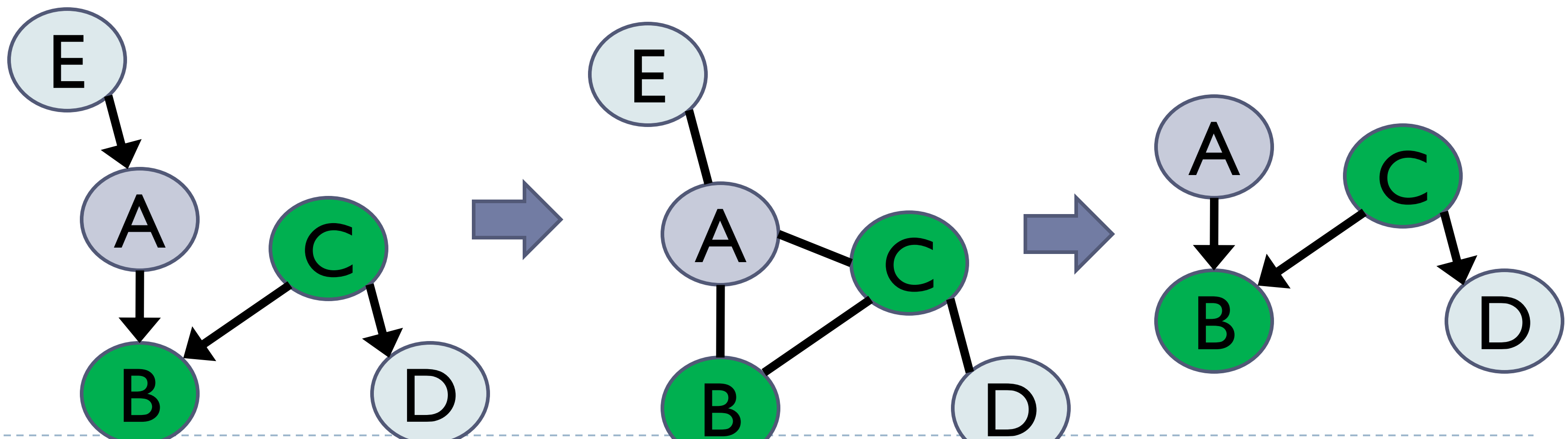
Discarding non-Ancestors



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

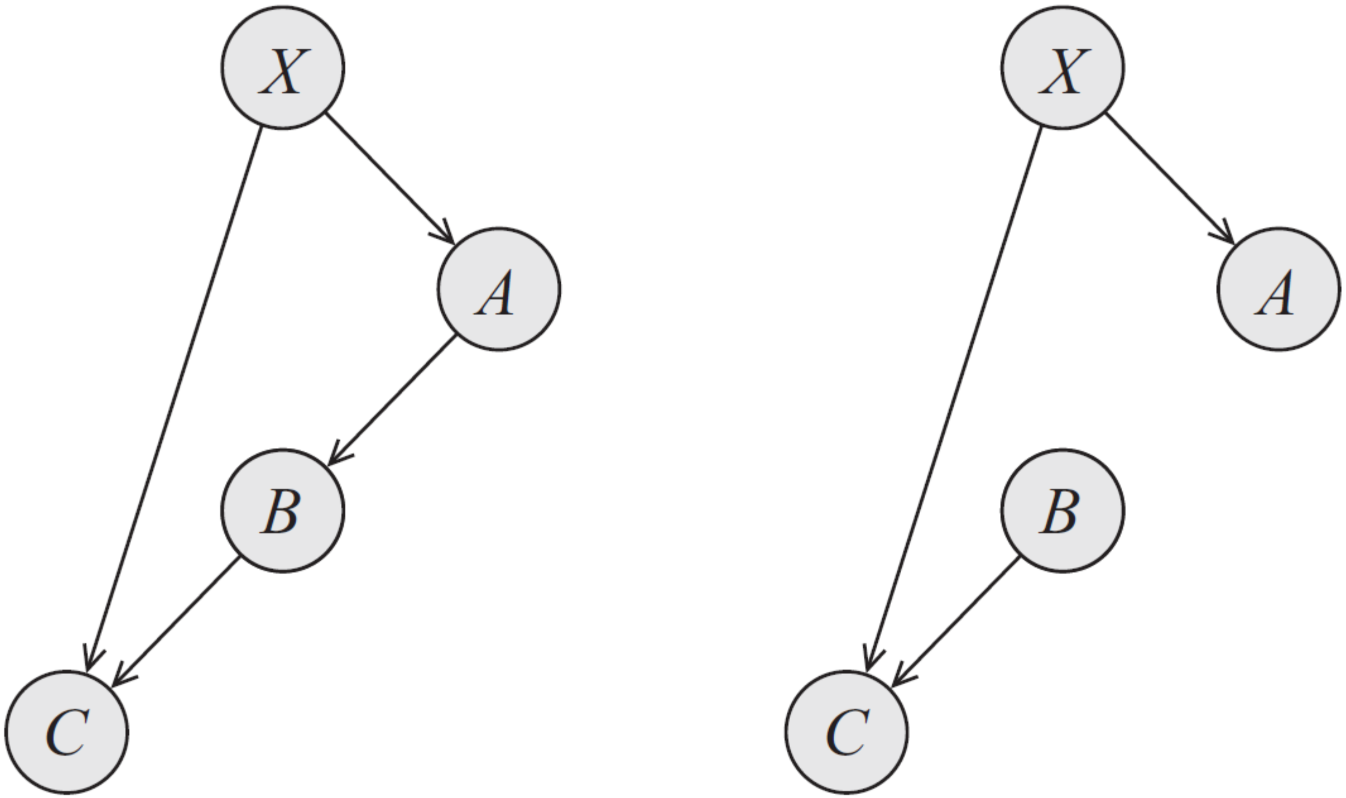
Discard by separation in Markov Network

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

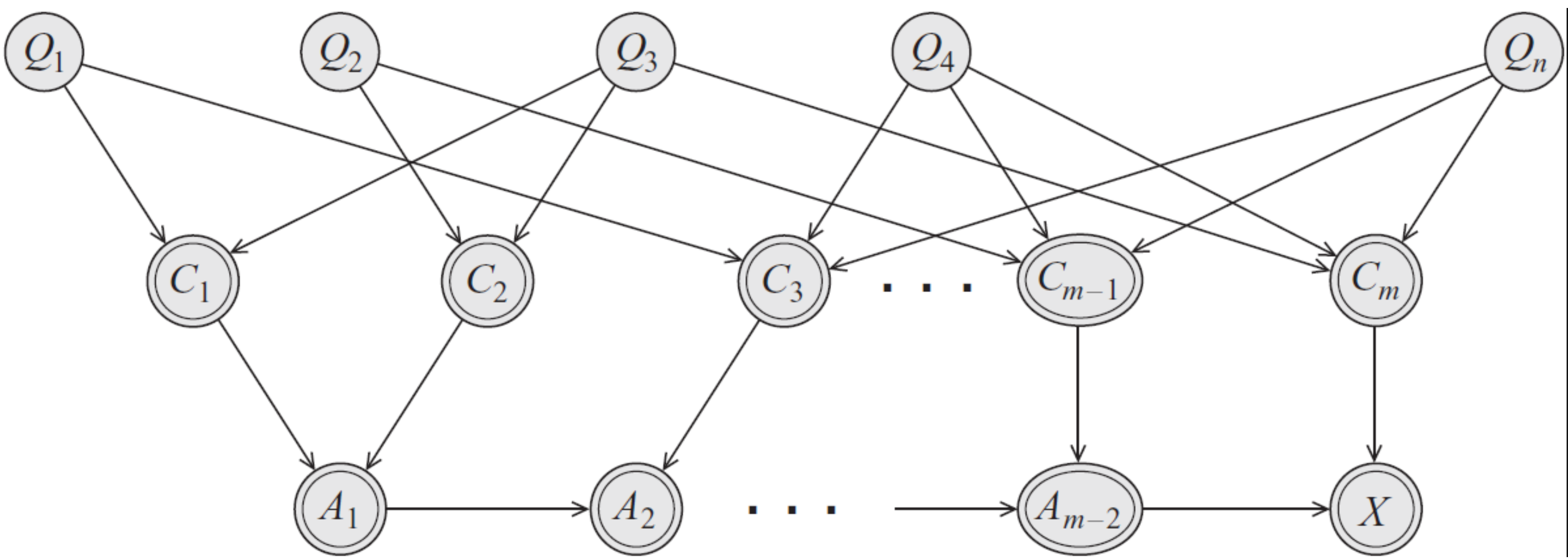
- Ordering heuristics:
  - Min neighbors (weighted)
  - Min fill (weighted)

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) that makes the formula true?

Reduction to Boolean Satisfiability (2)

- `(x1 v -x3 v x7) ^ (x4 v x5 v -x6)`
  - Let `Qi = xi`
  - `Ci =` clauses (e.g. `(x1 v -x3 v x7)`)
  - `X =` true *iff* all `Ci` are true, A_i's are "and" variables



Inference complexity details

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