Reading:
Chapters 5&6 of Koller&Friedman

BN Semantics 3 – Now it's personal! Exact inference & Variable elimination

Graphical Models – 10708

Carlos Guestrin
Carnegie Mellon University

September 21st, 2005

Announcements

- Let’s talk about the Waiting List
  - Who wants to be registered?
  - Send us an email ASAP if you want in!!!!

Perfect maps (P-maps)

- I-maps are not unique and often not simple enough
- Define "simplest" *G* that is I-map for *P*
  - A BN structure *G* is a **perfect map** for a distribution *P* if I(*P*) = I(*G*)
- Our goal:
  - Find a perfect map!
  - Must address equivalent BNs

Inexistence of P-maps 1

- XOR (this is a hint for the homework)

A, B, C binary : $C = A \text{ xor } B$

A, B uniform (50/50)

$(A \perp B)$

$(A \perp C)$

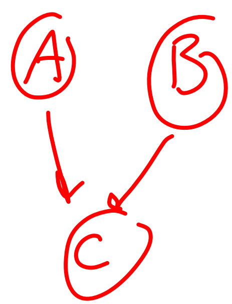
$(B \perp C)$

not ind:

$(A \perp B \mid C)$

$(A \perp C \mid B)$

$(B \perp C \mid A)$

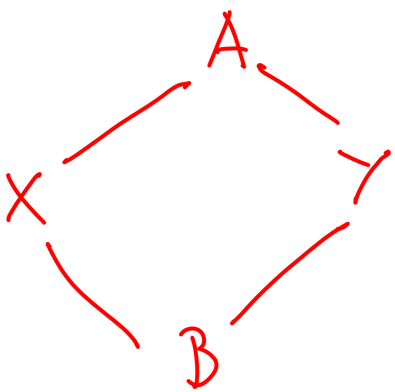
I maps	P-maps
 	NO!

Inexistence of P-maps 2

- (Slightly un-PC) swinging couples example

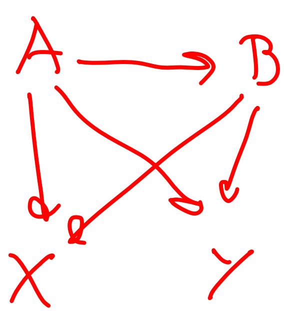
men A, B

women X, Y



$(A \perp B \mid X, Y)$

$(X \perp Y \mid A, B)$

I-map	P-map
	No!

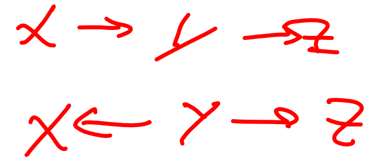
Obtaining a P-map

- Given the independence assertions that are true for *P*

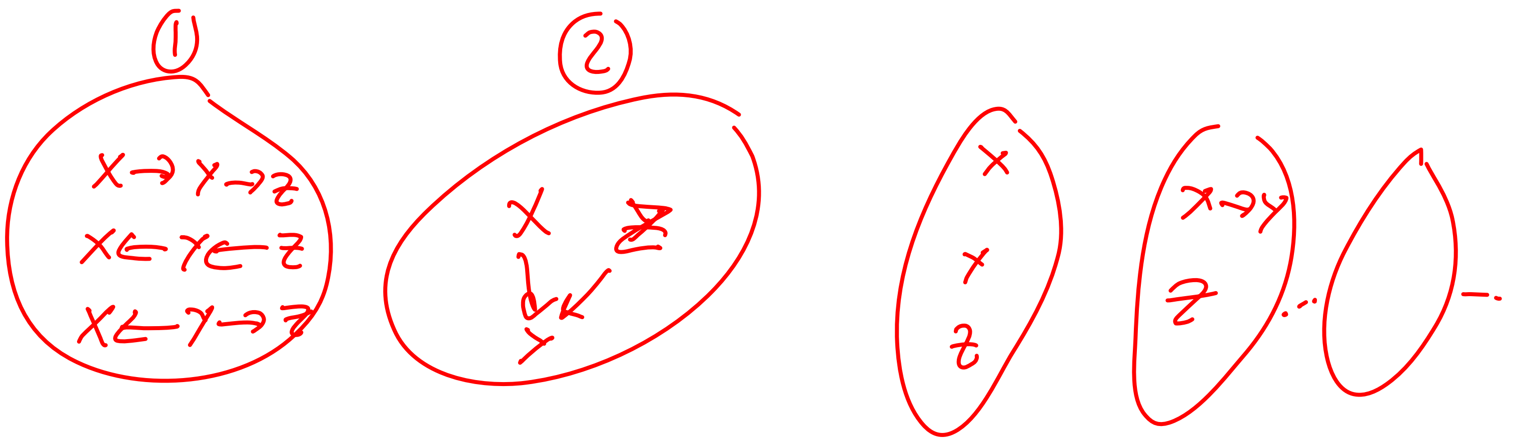
- Assume that there exists a perfect map G^*
  - Want to find G^*

- Many structures may encode same independencies as G^*, when are we done?
  - Find all equivalent structures simultaneously!

I-Equivalence



- Two graphs G_1 and G_2 are **I-equivalent** if $I(G_1) = I(G_2)$
- **Equivalence class** of BN structures
  - Mutually-exclusive and exhaustive partition of graphs

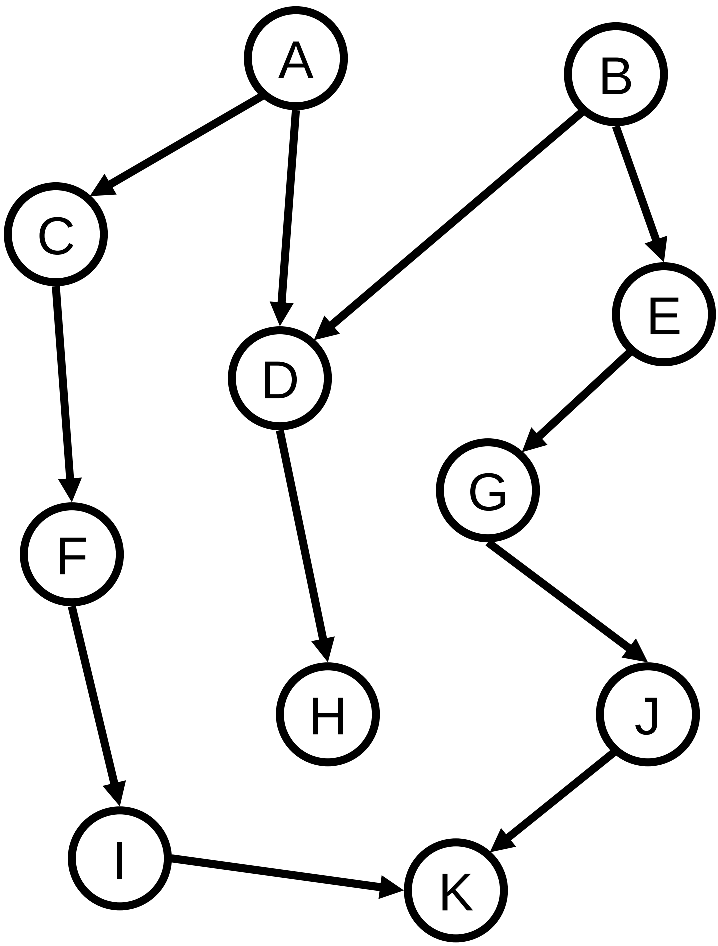


- How do we characterize these equivalence classes?

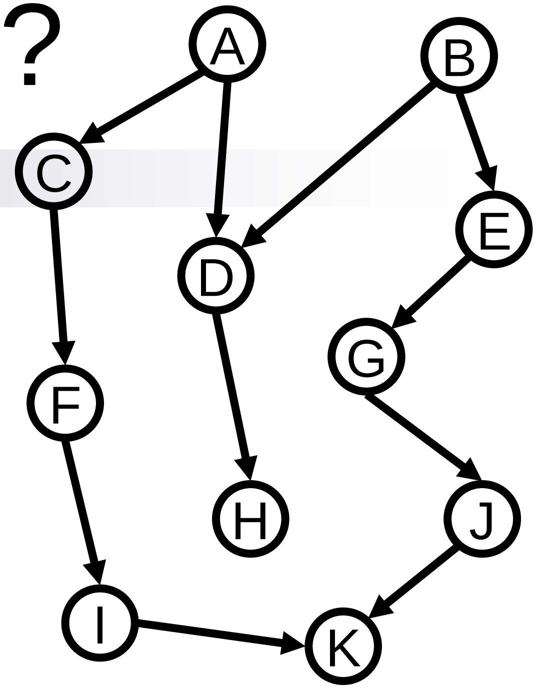
Skeleton of a BN

- **Skeleton** of a BN structure *G* is an **undirected graph** over the same variables that has an edge X–Y for every X→Y or Y→X in *G*
- (Little) **Lemma:** Two I-equivalent BN structures must have the same skeleton

counter example



What about V-structures?



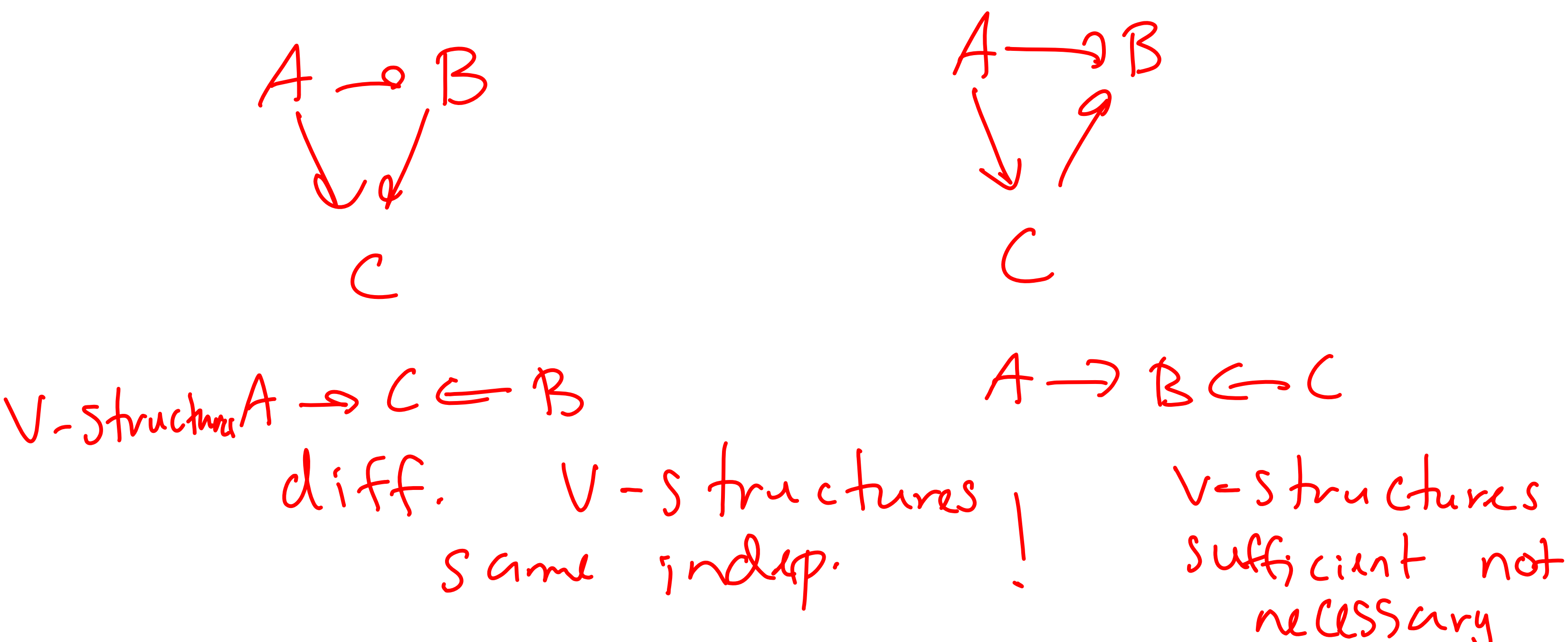
- **V-structures are key property of BN structure**

- **Theorem:** If G_1 and G_2 have the same skeleton and V-structures, then G_1 and G_2 are I-equivalent

Same V-structures not necessary

- **Theorem:** If G_1 and G_2 have the same skeleton and V-structures, then G_1 and G_2 are I-equivalent

- Though sufficient, same V-structures not necessary



Immoralities & I-Equivalence

- Key concept not V-structures, but “immoralities” (unmarried parents ☺)
  - $X \rightarrow Z \leftarrow Y$, with no arrow between X and Y
  - Important pattern: X and Y independent given their parents, but not given Z
  - (If edge exists between X and Y, we have *covered* the V-structure)
- **Theorem:** G_1 and G_2 have the same skeleton and immoralities if and only if G_1 and G_2 are I-equivalent

Obtaining a P-map

- Given the independence assertions that are true for *P*
  - Obtain skeleton
  - Obtain immoralities
- From skeleton and immoralities, obtain every (and any) BN structure from the equivalence class

Identifying the skeleton 1

- When is there an edge between X and Y?

- When is there no edge between X and Y?

Identifying the skeleton 2

- Assume d is max number of parents (d could be n)

- For each X_i and X_j
  - $E_{ij} \leftarrow$ true
  - For each $\mathbf{U} \subseteq \mathbf{X} - \{X_i, X_j\}$, $|\mathbf{U}| \leq 2d$
    - Is $(X_i \perp X_j \mid \mathbf{U})$?
      - $E_{ij} \leftarrow$ true
  - If E_{ij} is true
    - Add edge X – Y to skeleton

Identifying immoralities

- Consider X – Z – Y in skeleton, when should it be an immorality?

- Must be $X \rightarrow Z \leftarrow Y$ (immorality):
  - When X and Y are **never independent** given **U,** if $Z \in$ **U**

- Must **not** be $X \rightarrow Z \leftarrow Y$ (not immorality):
  - When there exists **U** with $Z \in$ **U**, such that X and Y are **independent** given **U**

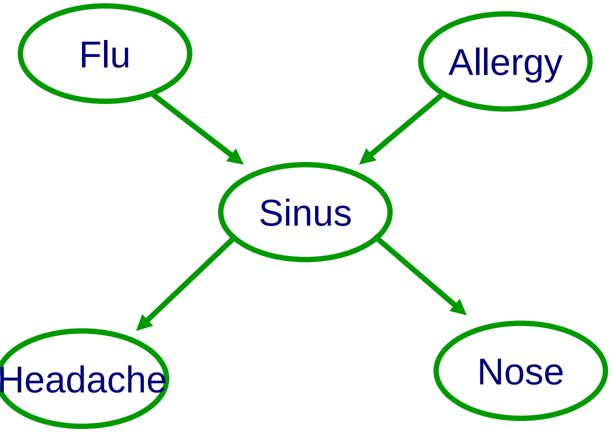
From immoralities and skeleton to BN structures

- Representing BN equivalence class as a **partially-directed acyclic graph** (PDAG)
- Immoralities force direction on other BN edges
- Full (polynomial-time) procedure described in reading

What you need to know about BN semantics

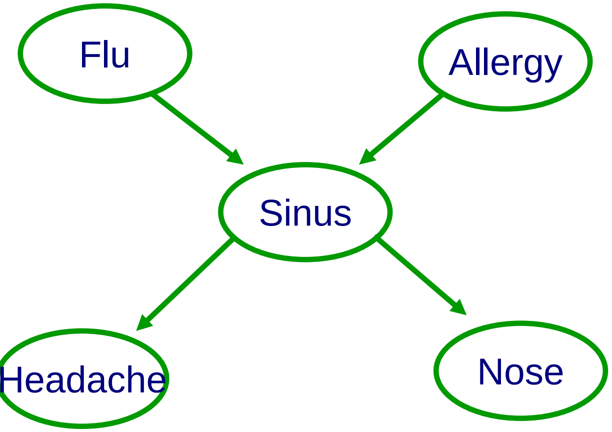
- Definition of a BN
- Local Markov assumption
- The representation theorem: *G* is an I-map for *P* if and only if *P* factorizes according to *G*
- d-separation – sound and complete procedure for finding independencies
  - (almost) all independencies can be read directly from graph without looking at CPTs
- Minimal I-map
  - every *P* has one, but usually many
- Perfect map
  - better choice for BN structure
  - not every *P* has one
  - can find one (if it exists) by considering I-equivalence
  - Two structures are I-equivalent if they have same skeleton and immoralities

Inference in graphical models: Typical queries 1



- Conditional probabilities
  - Distribution of some var(s). given evidence

Inference in graphical models: Typical queries 2 – Maximization



- Most probable explanation (MPE)
  - Most likely assignment to all hidden vars given evidence
- Maximum a posteriori (MAP)
  - Most likely assignment to some var(s) given evidence

Complexity of conditional probability queries 1

- How hard is it to compute P(X|**E**=**e**)?

Reduction – 3-SAT

$$(\overline{X}_1 \vee X_2 \vee X_3) \wedge (\overline{X}_2 \vee X_3 \vee X_4) \wedge \ldots$$

Complexity of conditional probability queries 2

- How hard is it to compute P(X|**E**=**e**)?
  - At least NP-hard, but even harder!

Inference is #P-hard, hopeless?

- Exploit structure!
- Inference is hard in general, but easy for many (real-world relevant) BN structures

What about the maximization problems? First, most probable explanation (MPE)

- What's the complexity of MPE?

What about maximum a posteriori?

- At least, as hard as MPE!

- Actually, much harder!!! NP^{PP}-complete!

Can we exploit structure for maximization?

- For MPE

- For MAP

Exact inference is hard, what about approximate inference?

- Must define approximation criterion!
- Relative error of $\varepsilon>0$
- Absolute error of $\varepsilon>0$

Hardness of approximate inference

- Relative error of ε

- Absolute error of ε

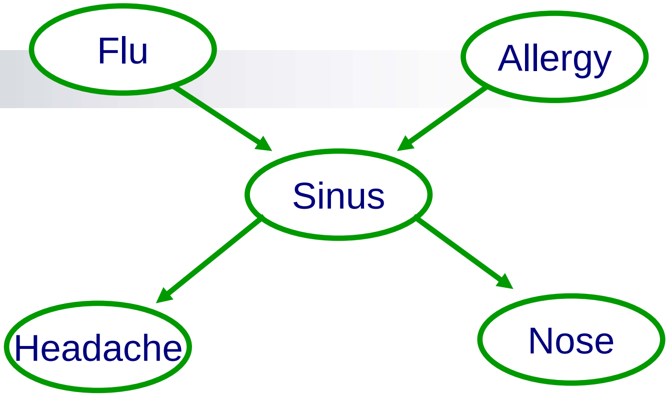
Inference in BNs hopeless?

- In general, yes!
  - Even approximate!

- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)

- For now, we'll talk about exact inference
  - Approximate inference later this semester

General probabilistic inference



- Query: $P(X \mid e)$

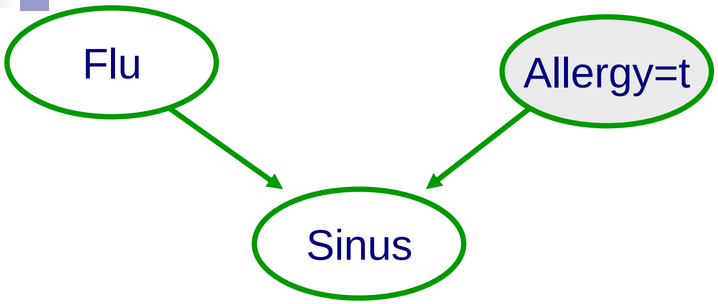
- Using Bayes rule:

$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

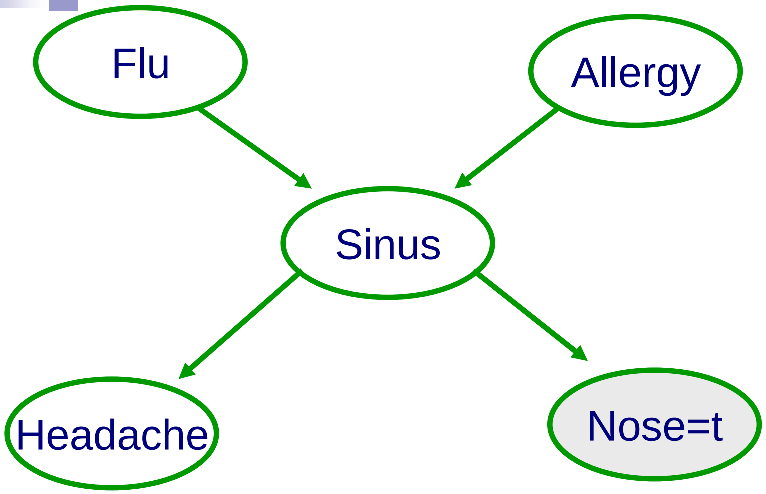
- Normalization:

$$P(X \mid e) \propto P(X, e)$$

Marginalization

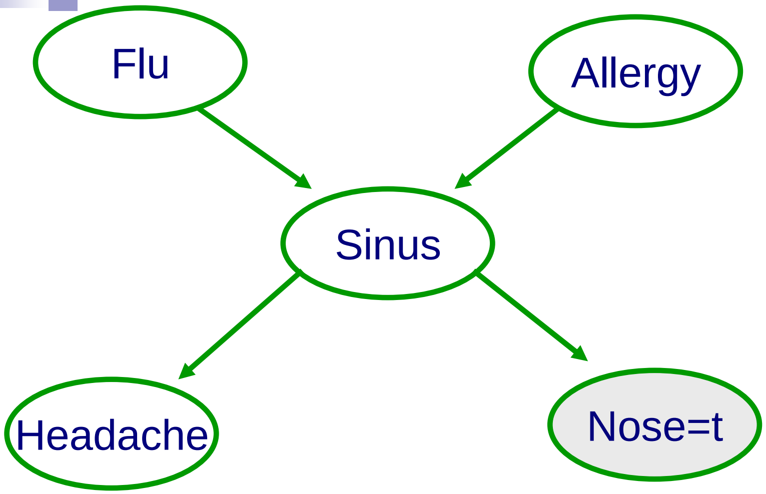


Probabilistic inference example



Inference seems exponential in number of variables!

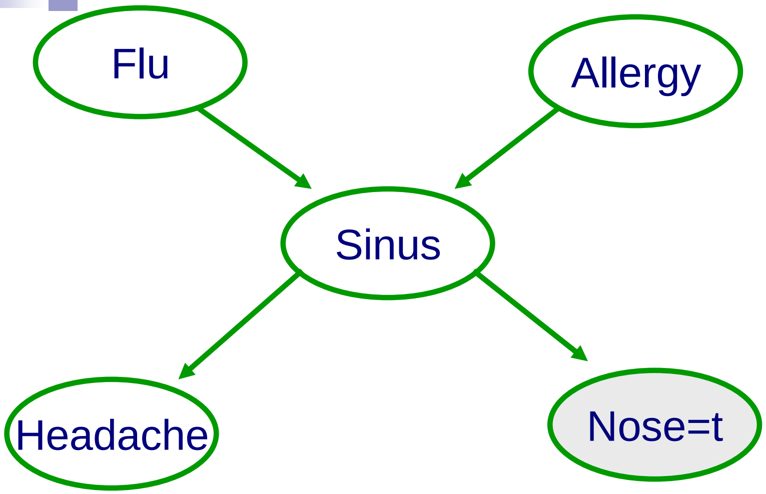
Fast probabilistic inference example – Variable elimination



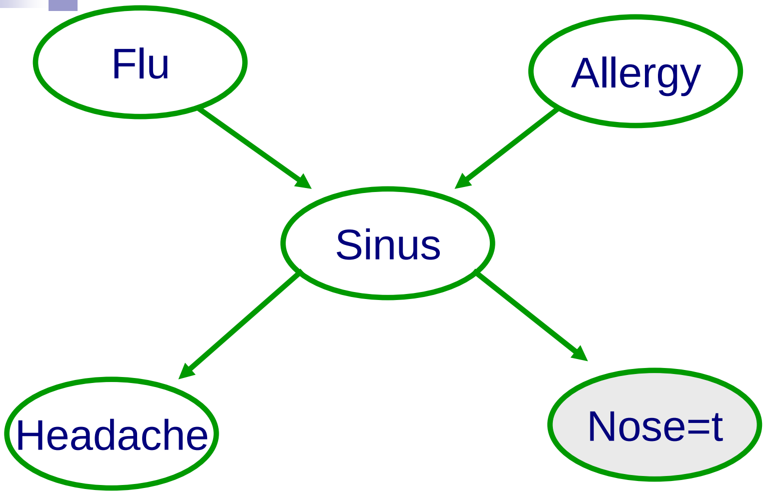
(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference

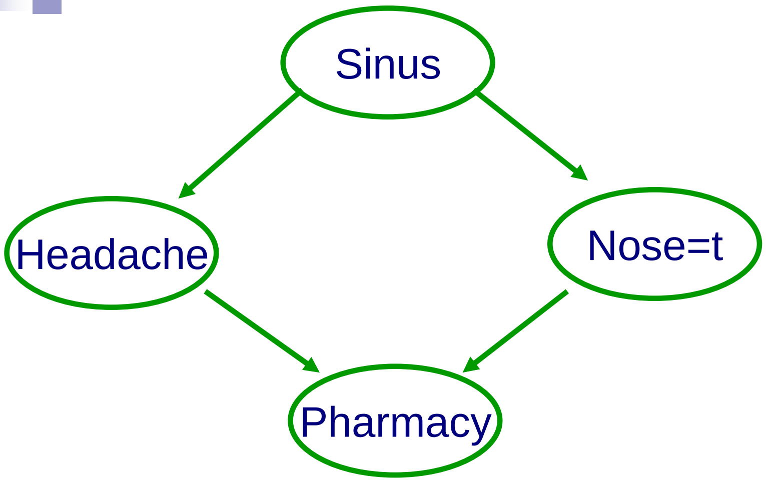


Understanding variable elimination – Intermediate results

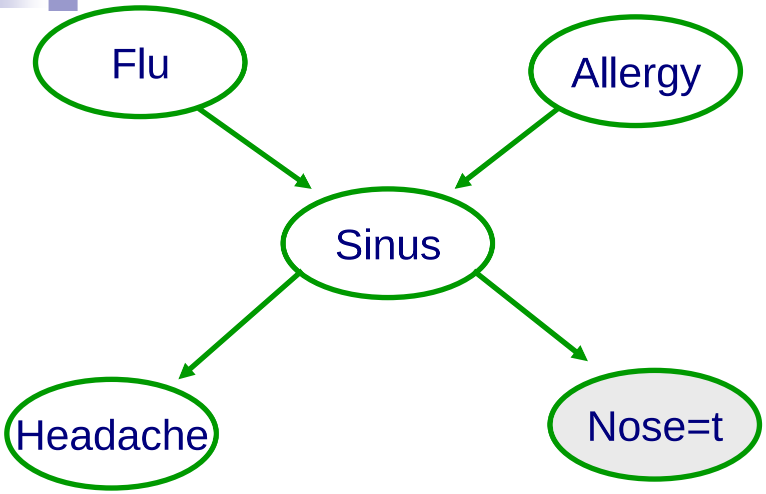


Intermediate results are probability distributions

Understanding variable elimination – Another example



Pruning irrelevant variables



Prune all non-ancestors of query variables

More generally: prune all nodes not on active trail between evidence and query vars

Variable elimination algorithm

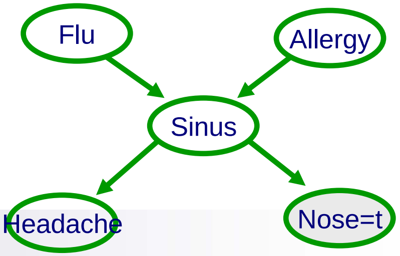
- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence **e**
- Prune non-active vars for {X,e}
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- Initial *factors* $\{f_1,\ldots,f_n\}$: $f_i = P(X_i|\mathbf{Pa}_{Xi})$ (CPT for X_i)
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!
- Normalize P(X,e) to obtain P(X|e)

IMPORTANT!!!

Operations on factors

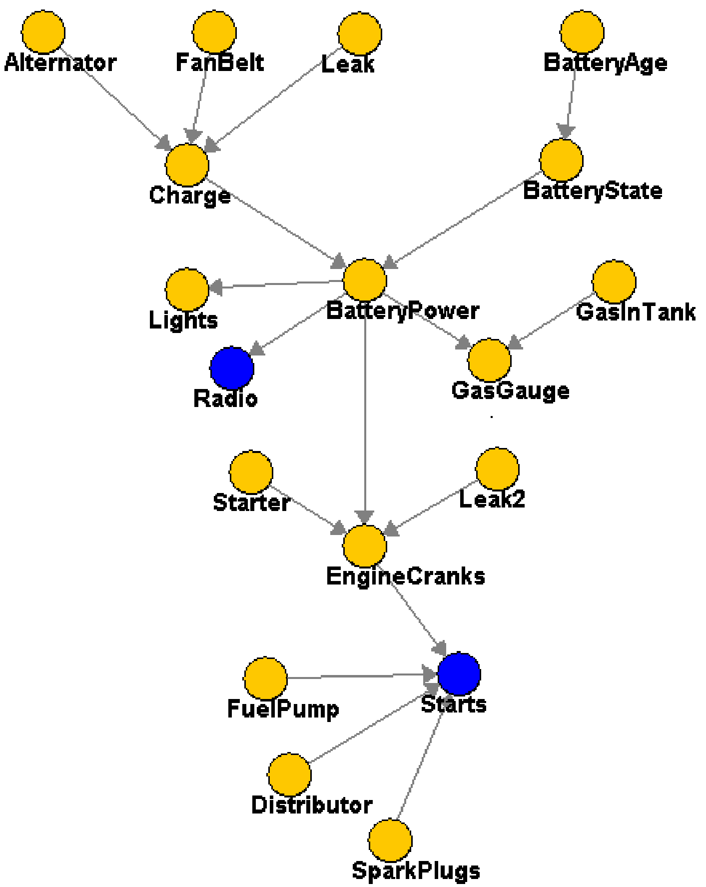


$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

Multiplication:

Marginalization:

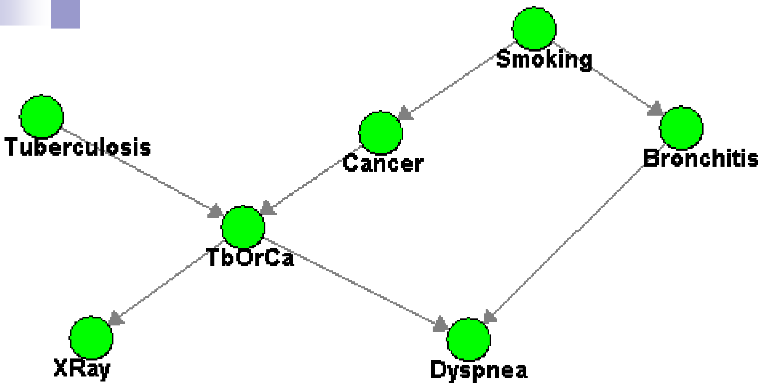
Complexity of variable elimination – (Poly)-tree graphs



Variable elimination order:
Start from “leaves” in –
- Start from skeleton!
- Choose a “root”, any node
- Find topological order for root
- Eliminate variables in reverse order

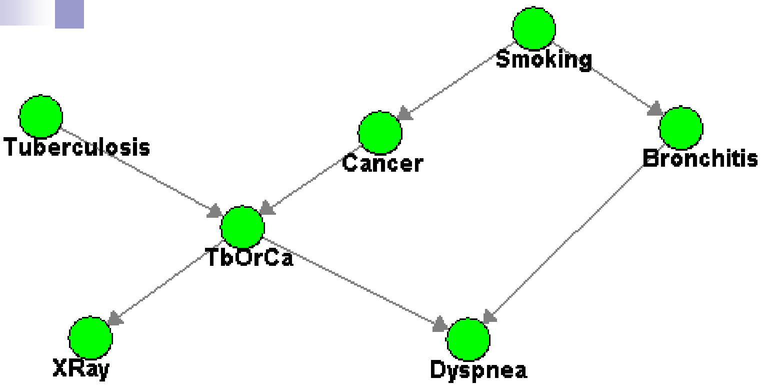
Linear in CPT sizes!!! (versus exponential)

Complexity of variable elimination – Graphs with loops



Exponential in number of variables in largest factor generated

Complexity of variable elimination – Tree-width



Moralize graph:
Connect parents
into a clique and
remove edge directions

Complexity of VE elimination:
(“Only”) exponential in tree-width

Example: Large tree-width with small number of parents

Compact representation $\not\Rightarrow$ Easy inference ☹

Choosing an elimination order

- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can't beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive