CSC384h: Intro to Artificial Intelligence

Reasoning Under Uncertainty

- This material is covered in chapters 13 and 14. Chapter 13 gives some basic background on probability from the point of view of AI. Chapter 14 talks about Bayesian Networks, exact reasoning in Bayes Nets as well as approximate reasoning, which will be main topics for us.

Fahiem Bacchus, University of Toronto

Uncertainty

- In search we viewed actions as being deterministic.
  - If you are in state S_1 and you execute action A you arrive at state S_2.
- Furthermore, there was a fixed initial state S_0. So with deterministic actions after executing any sequence of actions we know exactly what state we have arrived at.
  - Always know what state one is in.

- These assumptions are sensible in some domains
- But in many domains they are not true.

Uncertainty

- We might not know exactly what state we start off in
  - E.g., we can't see our opponents cards in a poker game
  - We don't know what a patient's ailment is.
- We might not know all of the effects of an action
  - The action might have a random component, like rolling dice.
  - We might not know all of the long term effects of a drug.
  - We might not know the status of a road when we choose the action of driving down it.

Uncertainty

- In such domains we still need to act, but we can't act solely on the basis of known true facts. We have to "gamble".
- E.g., we don't know for certain what the traffic will be like on a trip to the airport.

Uncertainty

- But how do we gamble **rationally**?
  - If we must arrive at the airport at 9pm on a week night we could "safely" leave for the airport ½ hour before.
    - Some probability of the trip taking longer, but the probability is low.
  - If we must arrive at the airport at 4:30pm on Friday we most likely need 1 hour or more to get to the airport.
    - Relatively high probability of it taking 1.5 hours.

Uncertainty

- To act rationally under uncertainty we must be able to evaluate how likely certain things are.
- By weighing likelihoods of events (probabilities) we can develop mechanisms for acting rationally under uncertainty.

Probability over Finite Sets. **(Review)**

- Probability is a function defined over a set of **atomic events U.**
  - **The universe of events**.
- It assigns a value Pr(e) to each event $e \in U$, in the range [0,1].
- It assigns a value to every set of events **F** by summing the probabilities of the members of that set.

  $$\Pr(\mathbf{F}) = \sum_{e \in F} \Pr(e)$$
- $\Pr(\mathbf{U}) = 1$, i.e., sum over all events is 1.
- Therefore: $\Pr(\{\}) = 0$ and
  $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$

Probability in General **(Review)**

- Given a set **U** (universe), a probability function is a function defined over the subsets of **U** that maps each subset to the real numbers and that satisfies the Axioms of Probability
  1. **Pr(U) = 1**
  2. **Pr(A) ∈ [0,1]**
  3. **Pr(A ∪ B) = Pr(A) + Pr(B) – Pr(A ∩ B)**

  Note if A ∩ B = {} then Pr(A ∪ B) = Pr(A) + Pr(B)

Probability over Feature Vectors

- We will work with universes of events each of which is a vectors of feature values.
- Like CSPs, we have
  1. a set of variables $V_1, V_2, \ldots, V_n$
  2. a finite domain of values for each variable, $Dom[V_1], Dom[V_2], \ldots, Dom[V_n]$.

- The universe of events U is the set of all vectors of values for the variables

  $\langle d_1, d_2, \ldots, d_n \rangle$: $d_i \in Dom[V_i]$

Probability over Feature Vectors

- This event space has size
  Π_i | Dom[V_i] |
  i.e., the product of the domain sizes.

- E.g., if | Dom[V_i] | = 2 we have 2^n distinct atomic events. (Exponential!)

Probability over Feature Vectors

- Asserting that some subset of variables have particular values allows us to specify a useful collection of subsets of U.
- E.g.
  - **$\{V_1 = a\}$ = set of all events where $V_1 = a$**
  - **$\{V_1 = a, V_3 = d\}$ = set of all events where $V_1 = a$ and $V_3 = d$.**
  - **…**
- **E.g.**
  - **$\Pr(\{V_1 = a\}) = \sum_{x \in \text{Dom}[V_3]} \Pr(\{V_1 = a, V_3 = x\})$.**

Probability over Feature Vectors

- If we had Pr of every atomic event (full instantiation of the variables) we could compute the probability of any other set (including sets can cannot be specified some set of variable values).
- E.g.
  - **$\{V_1 = a\}$ = set of all events where $V_1 = a$**
    - **Pf($\{V_1 = a\}$) =**

$$\sum_{x_2 \in \mathbf{Dom}[V_2]} \sum_{x_3 \in \mathbf{Dom}[V_3]} \cdots \sum_{x_n \in \mathbf{Dom}[V_n]}$$

$$\mathbf{Pr}(V_1=a, V_2=x_2, V_3=x_3, \ldots, V_n=x_n)$$

Probability over Feature Vectors

- Problem:
  - This is an exponential number of atomic probabilities to specify.
  - Requires summing up an exponential number of items.

- **For evaluating the probability of sets containing a particular subset of variable assignments we can do much better. Improvements come from the use of probabilistic independence, especially conditional independence.**

Conditional Probabilities. **(Review)**

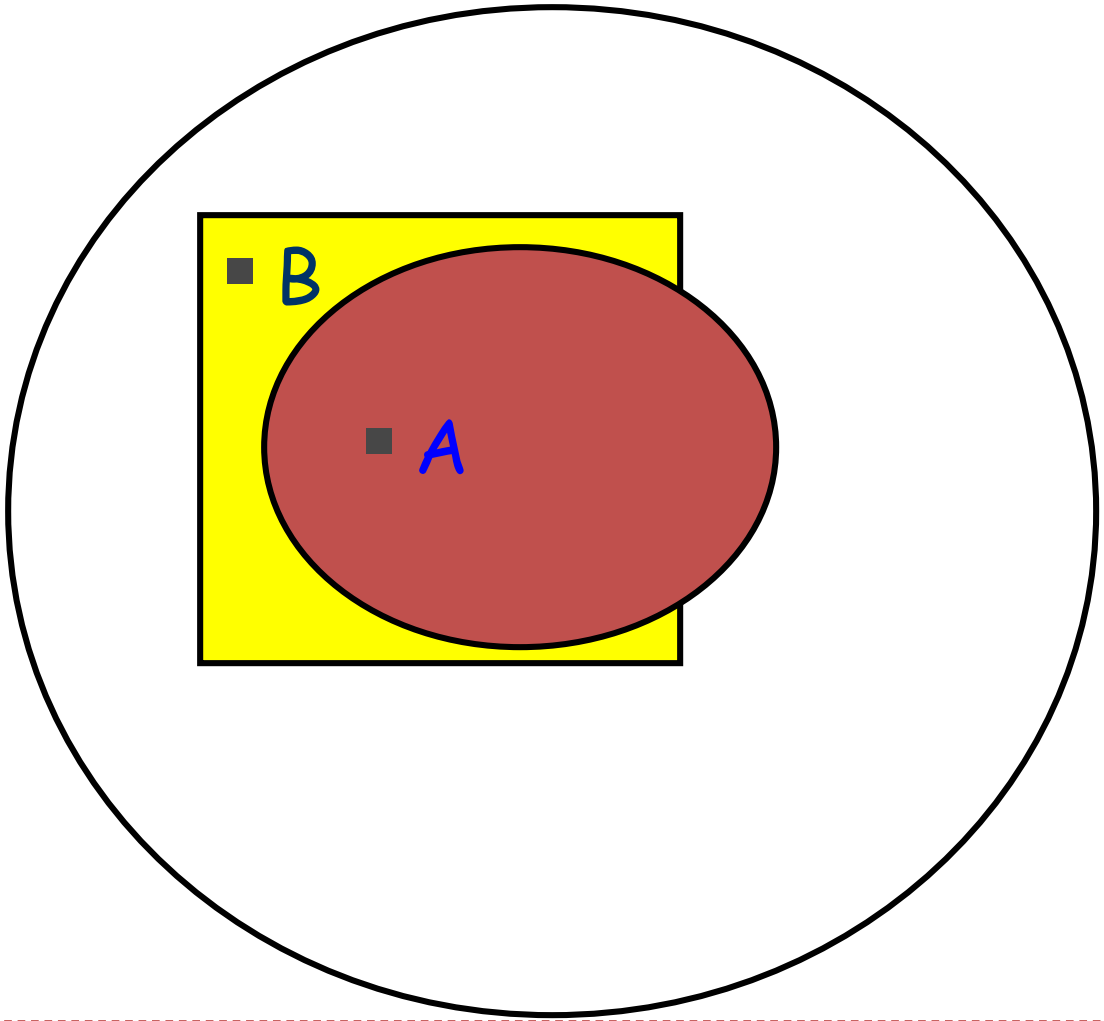
- With probabilities one can capture conditional information by using conditional probabilities.
- Conditional probabilities are essential for both representing and reasoning with probabilistic information.

Conditional Probabilities **(Review)**

- Say that A is a set of events such that $Pr(A) > 0$.
- Then one can define a conditional probability wrt the event A:

$$Pr(B \mid A) = Pr(B \cap A)/Pr(A)$$

Conditional Probabilities **(Review)**



- B covers about 30% of the entire space, but covers over 80% of A.

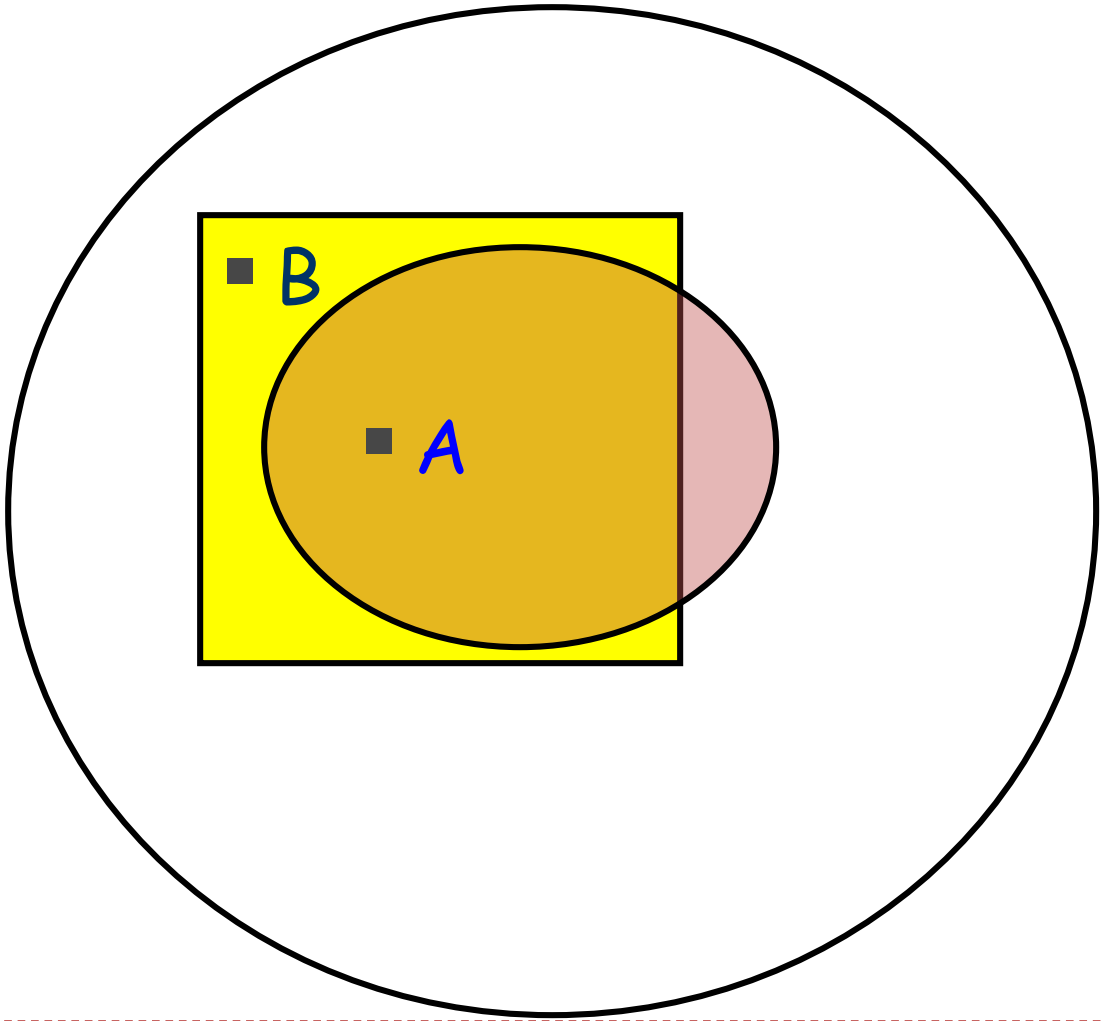
Conditional Probabilities **(Review)**

- Conditioning on A, corresponds to restricting one's attention to the events in A.
- We now consider A to be the whole set of events (a new universe of events):

$$\Pr(A \mid A) = 1.$$

- Then we assign all other sets a probability by taking the probability mass that "lives" in A ($\Pr(B \wedge A)$), and normalizing it to the range [0,1] by dividing by $\Pr(A)$.

Conditional Probabilities **(Review)**



- B's probability in the new universe A is 0.8.

Conditional Probabilities **(Review)**

- A conditional probability is a probability function, but now over A instead of over the entire space.

  - **$Pr(A \mid A) = 1$**
  - **$Pr(B \mid A) \in [0,1]$**
  - **$Pr(C \cup B \mid A) = Pr(C \mid A) + Pr(B \mid A) - Pr(C \cap B \mid A)$**

Summing out rule

- Useful fact about probabilities

- Say that $B_1, B_2, \ldots, B_k$ form a **partition** of the universe **U**.
  1. $B_i \cap B_j = \varnothing \quad i \neq j$ (mutually exclusive)
  2. $B_1 \cup B_2 \cup B_3 \ldots \cup B_k = \mathbf{U}$ (exhaustive)
- In probabilities:
  1. $Pr(B_i \cap B_j) = 0$
  2. $Pr(B_1 \cup B_2 \cup B_3 \ldots \cup B_k) = 1$

Summing out rule

- Given any other set of events A we have that

  $$Pr(A) = Pr(A \cap B_1) + Pr(A \cap B_2) + \ldots + Pr(A \cap B_k)$$

- In conditional probabilities:

  $$Pr(A) = Pr(A \mid B_1)Pr(B_1) + Pr(A \mid B_2)Pr(B_2) + \ldots + Pr(A \mid B_k)Pr(B_k)$$

  $$Pr(A \mid B_i)Pr(B_i) = Pr(A \cap B_i)/Pr(B_i) * Pr(B_i) = Pr(A \cap B_i)$$

- Often we know $Pr(A \mid B_i)$, so we can compute Pr(A) this way.

Properties and Sets

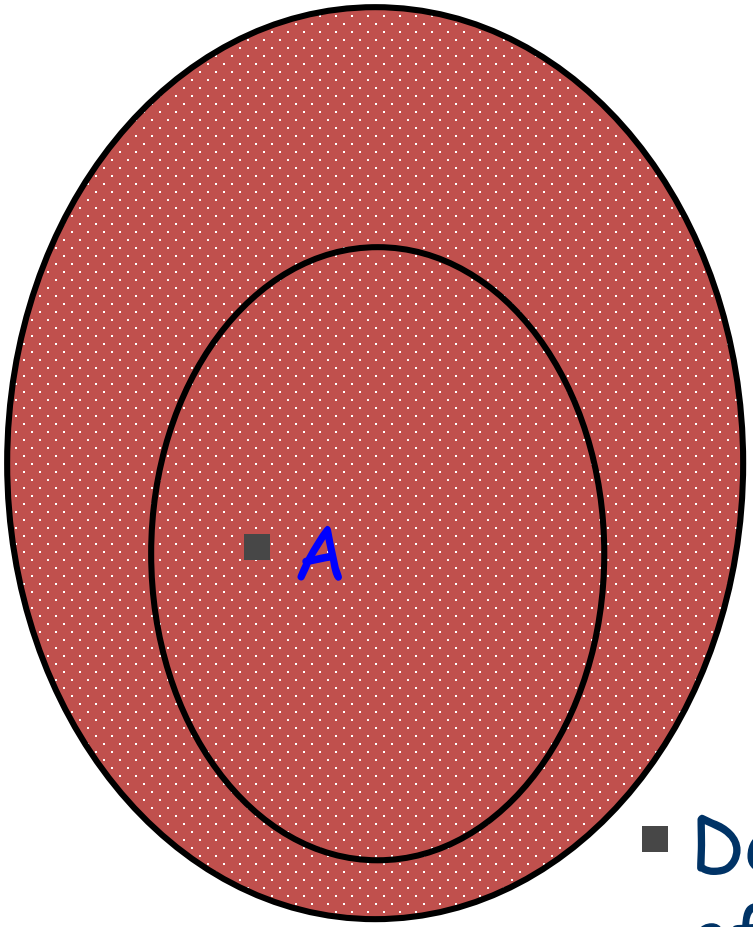
- Any set of events A can be interpreted as a property: the set of events with property A.
- Hence, we often write
  - $A \vee B$ to represent the set of events with either property A or B: the set $A \cup B$
  - $A \wedge B$ to represent the set of events both property A and B: the set $A \cap B$
  - $\neg A$ to represent the set of events that do not have property A: the set U-A (i.e., the complement of A wrt the universe of events U)

Independence **(Review)**

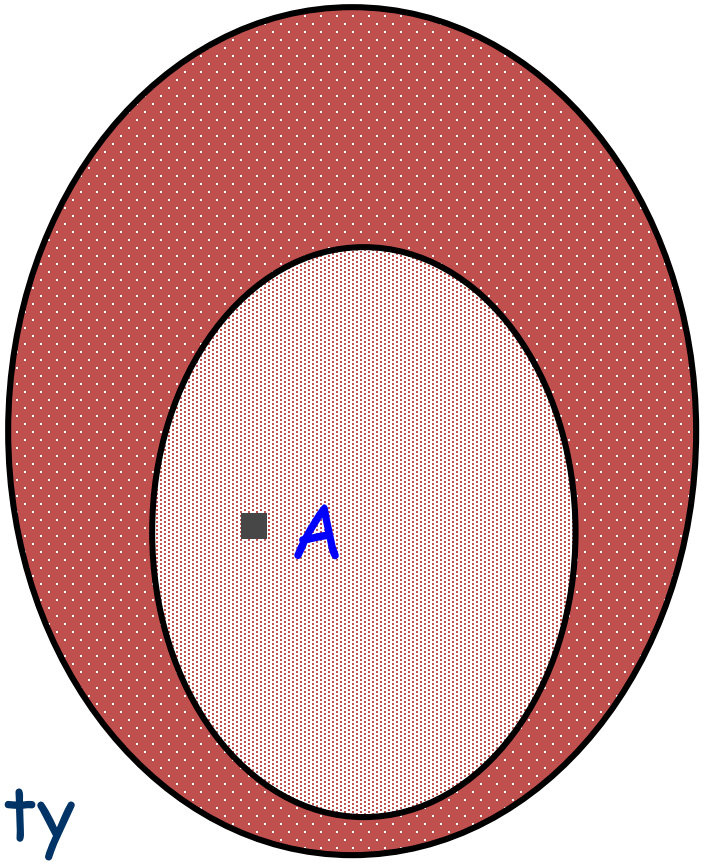
- It could be that the density of B on A is identical to its density on the entire set.
  - Density: pick an element at random from the entire set. How likely is it that the picked element is in the set B?
- Alternately the density of B on A could be much different that its density on the whole space.
- In the first case we say that B is **independent** of A. While in the second case B is dependent on A.

Independence **(Review)**

- Independent



- Dependent



- Density of B

Independence Definition **(Review)**

A and B are independent properties

$$Pr(B \mid A) = Pr(B)$$

A and B are dependent.

$$Pr(B \mid A) \neq Pr(B)$$

Implications for Knowledge

- Say that we have picked an element from the entire set. Then we find out that this element has property A (i.e., is a member of the set A).
  - Does this tell us anything more about how likely it is that the element also has property B?
  - If B is independent of A then we have learned nothing new about the likelihood of the element being a member of B.

Independence

- E.g., we have a feature vector, we don't know which one. We then find out that it contains the feature V_1=a.
  - I.e., we know that the vector is a member of the set $\{V_1 = a\}$.
  - Does this tell us anything about whether or not V_2=a, V_3=c, …, etc?
  - This depends on whether or not these features are independent/dependent of V_1=a.

Conditional Independence

- Say we have already learned that the randomly picked element has property A.
- We want to know whether or not the element has property B:

  $\Pr(B \mid A)$ expresses the probability of this being true.

- Now we learn that the element also has property C. Does this give us more information about B-ness?

  $\Pr(B \mid A \wedge C)$ expresses the probability of this being true under the additional information.

Conditional Independence

- If

$$\Pr(B \mid A \wedge C) = \Pr(B \mid A)$$

  then we have not gained any additional information from knowing that the element is also a member of the set C.

- In this case we say that B is conditionally independent of C given A.

- That is, once we know A, additionally knowing C is irrelevant (wrt to whether or not B is true).
  - Conditional independence is independence in the conditional probability space $\Pr(\bullet \mid A)$.
  - Note we could have $\Pr(B \mid C) \neq \Pr(B)$. But once we learn A, C becomes irrelevant.

Computational Impact of Independence

- We will see in more detail how independence allows us to speed up computation. But the fundamental insight is that

  If A and B are independent properties then

  $$\Pr(A \wedge B) = \Pr(B) * \Pr(A)$$

  Proof:

  $\Pr(B \mid A) = \Pr(B)$ independence

  $\Pr(A \wedge B)/\Pr(A) = \Pr(B)$ definition

  $\Pr(A \wedge B) = \Pr(B) * \Pr(A)$

Computational Impact of Independence

- This property allows us to "break" up the computation of a conjunction "Pr(A∧B)" into two separate computations "Pr(A)" and "Pr(B)".

- Dependent on how we express our probabilistic knowledge this yield great computational savings.

Computational Impact of Independence

- Similarly for conditional independence.

  $\Pr(B \mid C \wedge A) = \Pr(B \mid A) \Rightarrow$

  $\Pr(B \wedge C \mid A) = \Pr(B \mid A) * \Pr(C \mid A)$

Proof:

$\Pr(B \mid C \wedge A) = \Pr(B \mid A)$ independence

$\Pr(B \wedge C \wedge A)/\Pr(C \wedge A) = \Pr(B \wedge A)/\Pr(A)$ defn.

$\Pr(B \wedge C \wedge A)/\Pr(A) = \Pr(C \wedge A)/\Pr(A) * \Pr(B \wedge A)/\Pr(A)$

$\Pr(B \wedge C \mid A) = \Pr(B \mid A) * \Pr(C \mid A)$ defn.

Computational Impact of Independence

- Conditional independence allows us to break up our computation onto distinct parts

  $$\Pr(B \wedge C \mid A) = \Pr(B \mid A) * \Pr(C \mid A)$$

- And it also allows us to ignore certain pieces of information

  $$\Pr(B \mid A \wedge C) = \Pr(B \mid A)$$

Bayes Rule **(Review)**

- Bayes rule is a simple mathematical fact. But it has great implications wrt how probabilities can be reasoned with.

  - $Pr(Y \mid X) = Pr(X \mid Y)Pr(Y)/Pr(X)$

    $$\begin{aligned} Pr(Y \mid X) &= Pr(Y \wedge X)/Pr(X) \\ &= Pr(Y \wedge X)/Pr(X) * P(Y)/P(Y) \\ &= Pr(Y \wedge X)/Pr(Y) * Pr(Y)/Pr(X) \\ &= Pr(X \mid Y)Pr(Y)/Pr(X) \end{aligned}$$

Bayes Rule

- Bayes rule allows us to use a supplied conditional probability in both directions.

- E.g., from treating patients with heart disease we might be able to estimate the value of

  Pr(high_Cholesterol | heart_disease)

- With Bayes rule we can turn this around into a predictor for heart disease

  Pr(heart_disease | high_Cholesterol)

- Now with a simple blood test we can determine "high_Cholesterol" use this to help estimate the likelihood of heart disease.

Bayes Rule

- For this to work we have to deal with the other factors as well

  Pr(heart_disease | high_Cholesterol)
  = Pr(high_Cholesterol | heart_disease)
  * Pr(heart_disease)/Pr(high_Cholesterol)

- We will return to this later.

Bayes Rule Example

- Disease ∈ {malaria, cold, flu}; Symptom = fever
  - Must compute Pr(D | fever) to prescribe treatment
- Why not assess this quantity directly?
  - Pr(mal | fever) is not natural to assess; Pr(mal | fever) does not reflects the underlying "causal" mechanism fever ➔ malaria
  - Pr(mal | fever) is not "stable": a malaria epidemic changes this quantity (for example)
- So we use Bayes rule:
  - Pr(mal | fever) = Pr(fever | mal) Pr(mal) / Pr(fever)

Bayes Rule

- Pr(mal | fever) = Pr(fever | mal)Pr(mal)/Pr(fever)
- Pr(mal)?
  - This is the prior probability of Malaria, i.e., before you exhibited a fever, and with it we can account for other factors, e.g., a malaria epidemic, or recent travel to a malaria risk zone.
  - E.g., The center for disease control keeps track of the rates of various diseases.
- Pr(fever | mal)?
  - This is the probability a patient with malaria exhibits a fever.
  - Again this kind of information is available from people who study malaria and its effects.

Bayes Rule

- Pr(fever)?
  - This is typically not known, but it can be computed!
  - We eventually have to divide by this probability to get the final answer: Pr(mal | fever) = Pr(fever | mal)Pr(mal)/Pr(fever)
- First, we find a set of mutually exclusive and exhaustive causes for fever:
  - Say that in our example, mal, cold and flu are only possible causes of fever and they are mutually exclusive.
  - Pr(fev | $\neg$ mal $\wedge$ $\neg$ cold $\wedge$ $\neg$ flu) = 0 ➔ **Fever can't happen with one of these causes.**
  - Pr(mal $\wedge$ cold) = Pr(mal $\wedge$ flu) = Pr(cold $\wedge$ flu) = 0 ➔ **these causes can't happen together.** (Note that our example is not very realistic!)
- Second, we compute

  Pr(fever | mal)Pr(mal),
  Pr(fever | cold)Pr(cold).
  Pr(fever | flu)Pr(flu).

  We know Pr(fever | cold) and Pr(fever | flu), along with Pr(cold) and Pr(flu) from the same sources as Pr(fever | mal) and Pr(mal)

Bayes Rule

- Since flu, cold and malaria are exclusive, {flu, cold, malaria, $\neg$mal $\wedge$ $\neg$cold $\wedge$ $\neg$flu} forms a partition of the universe. So

$$\begin{aligned}\Pr(\text{fever}) = {} & \Pr(\text{fever} \mid \text{mal})*\Pr(\text{mal}) + \Pr(\text{fever} \mid \text{cold})*\Pr(\text{cold}) \\ & + \Pr(\text{fever} \mid \text{flu})*\Pr(\text{flu}) \\ & + \Pr(\text{fever} \mid \neg\text{mal} \wedge \neg\text{cold} \wedge \neg\text{flu})*\Pr(\neg\text{mal} \wedge \neg\text{cold} \wedge \neg\text{flu})\end{aligned}$$

  - The last term is zero as fever is not possible unless one of malaria, cold, or flu is true.
- So to compute the trio of numbers, Pr(mal | fever), Pr(cold | fever), Pr(flu | fever), we compute the trio of numbers Pr(fever | mal)*Pr(mal), Pr(fever | cold)*Pr(cold), Pr(fever | flu)*Pr(flu)
- And then we divide these three numbers by Pr(fever).
  - That is we divide these three numbers by their sum: This is called **normalizing** the numbers.
- Thus we never need actually compute Pr(fever) (unless we want to).

Normalizing

- If we have a vector of k numbers, e.g., <3, 4, 2.5, 1, 10, 21.5> we can **normalize** these numbers by dividing each number by the sum of the numbers:
  - 3 + 4 + 2.5 +1 +10 + 21.5 = 42
  - Normalized vector
    = <3/42, 4/42, 2.5/42, 1/42, 10/42, 21.5/42>
    = <0.071, 0.095, 0.060, 0.024, 0.238, 0.512>
- After normalizing the vector of numbers sums to 1
  - Exactly what is needed for these numbers to specify a probability distribution.

Chain Rule **(Review)**

- $Pr(A_1 \wedge A_2 \wedge \ldots \wedge A_n) =$
  $Pr(A_1 \mid A_2 \wedge \ldots \wedge A_n) * Pr(A_2 \mid A_3 \wedge \ldots \wedge A_n)$
  $* \ldots * Pr(A_{n-1} \mid A_n) * Pr(A_n)$

Proof:

$Pr(A_1 \mid A_2 \wedge \ldots \wedge A_n) * Pr(A_2 \mid A_3 \wedge \ldots \wedge A_n)$
$* \ldots * Pr(A_{n-1} \mid A_n)$
$= Pr(A_1 \wedge A_2 \wedge \ldots \wedge A_n) / Pr(A_2 \wedge \ldots \wedge A_n) *$
$Pr(A_2 \wedge \ldots \wedge A_n) / Pr(A_3 \wedge \ldots \wedge A_n) * \ldots *$
$Pr(A_{n-1} \wedge A_n) / Pr(A_n) * Pr(A_n)$

Variable Independence

- Recall that we will be mainly dealing with probabilities over feature vectors.
- We have a set of variables, each with a domain of values.
- It could be that $\{V_1=a\}$ and $\{V_2=b\}$ are independent:

$$Pr(V_1=a \wedge V_2=b) = Pr(V_1=a)*Pr(V_2=b)$$

- It could also be that $\{V_1=b\}$ and $\{V_2=b\}$ are not independent:

$$Pr(V_1=b \wedge V_2=b) \neq Pr(V_1=b)*Pr(V_2=b)$$

Variable Independence

- However we will generally want to deal with the situation where we have variable independence.
  - Two ***variables*** X and Y are conditionally independent ***given variable Z*** iff
    $\forall$ x,y,z. $x \in Dom(X) \wedge y \in Dom(Y) \wedge z \in Dom(Z)$
    $\rightarrow$ X=x is conditionally independent of Y=y given Z = z
    $\equiv Pr(X=x \wedge Y=y \mid Z=z)$
    $= Pr(X=x \mid Z=z) * Pr(Y=y \mid Z=z)$
  - Also applies to sets of more than two variables
  - Also to unconditional case (X,Y independent)

Variable Independence

- If you know the value of Z (*whatever* it is), learning Y's value (whatever it is) does not influence your beliefs about any of X's values.
  - these definitions differ from earlier ones, which talk about particular sets of events being independent. Variable independence is a concise way of stating a number of individual independencies.

What does independence buys us?

- Suppose (say, boolean) variables X_1, X_2,…, X_n are mutually independent (i.e., every subset is variable independent of every other subset)
  - we can specify full joint distribution (probability function over all vectors of values) using only n parameters (linear) instead of 2^n -1 (exponential)
- How? Simply specify $Pr(X_1)$, … $Pr(X_n)$ (i.e., $Pr(X_i=true)$ for all i)
  - from this I can recover probability of any primitive event easily (or any conjunctive query).
    e.g. $Pr(X_1 \neg X_2 X_3 X_4) = Pr(X_1)\ (1-Pr(X_2))\ Pr(X_3)\ Pr(X_4)$
  - we can condition on observed value X_k (or $\neg X_k$) trivially
    $Pr(X_1 \neg X_2 \mid X_3) = Pr(X_1)\ (1-Pr(X_2))$

The Value of Independence

- Complete independence reduces both *representation of joint* and *inference* from $O(2^n)$ to $O(n)$!
- Unfortunately, such complete mutual independence is very rare. Most realistic domains do not exhibit this property.
- Fortunately, most domains do exhibit a fair amount of conditional independence. And we can exploit conditional independence for representation and inference as well.
- **Bayesian networks** do just this

An Aside on Notation

- Pr(X) for variable X (or set of variables) refers to the *(marginal) distribution* over X.
  - It specifies Pr(X=d) for all $d \in Dom[X]$
- Note

  $\sum_{d \in Dom[X]} Pr(X=d) = 1$

  (every vector of values must be in one of the sets $\{X=d\}$ $d \in Dom[X]$)
- Also

  $Pr(X=d_1 \wedge X=d_2) = 0$ for all $d_1, d_2 \in Dom[X]$ $d_1 \neq d_2$

  (no vector of values contains two different values for X).

An Aside on Notation

- Pr(X | Y) refers to family of conditional distributions over X, one for each $y \in Dom(Y)$.
  - For each $d \in Dom[Y]$, Pr(X | Y) specifies a distribution over the values of X:
    $Pr(X=d_1 | Y=d), Pr(X=d_2 | Y=d), \ldots, Pr(X=d_n | Y=d)$
    for $Dom[X] = \{d_1, d_2, \ldots, d_n\}$.

- Distinguish between Pr(X)—which is a distribution–and $Pr(x_i)$ ($x_i \in Dom[X]$)-- which is a number. Think of Pr(X) as a function that accepts any $x_i \in Dom(X)$ as an argument and returns $Pr(x_i)$.
- Similarly, think of Pr(X | Y) as a function that accepts any $x_i \in Dom[X]$ and $y_k \in Dom[Y]$ and returns $Pr(x_i \mid y_k)$. Note that Pr(X | Y) is not a single distribution; rather it denotes the family of distributions (over X) induced by the different $y_k \in Dom(Y)$

Exploiting Conditional Independence

- Let's see what conditional independence buys us

- Consider a story:
  - If Craig woke up too early E, Craig probably needs coffee C; if C, Craig needs coffee, he's likely angry A. If A, there is an increased chance of an aneurysm (burst blood vessel) B. If B, Craig is quite likely to be hospitalized H.

E – Craig woke too early A – Craig is angry H – Craig hospitalized
C – Craig needs coffee B – Craig burst a blood vessel

Cond'l Independence in our Story

- If you learned any of E, C, A, or B, your assessment of Pr(H) would change.
  - E.g., if any of these are seen to be true, you would increase Pr(h) and decrease Pr(~h).
  - So H is *not independent* of E, or C, or A, or B.
- But if you knew value of B (true or false), learning value of E, C, or A, would not influence Pr(H). Influence these factors have on H is mediated by their influence on B.
  - Craig doesn't get sent to the hospital because he's angry, he gets sent because he's had an aneurysm.
  - So H is *independent* of E, and C, and A, *given* B

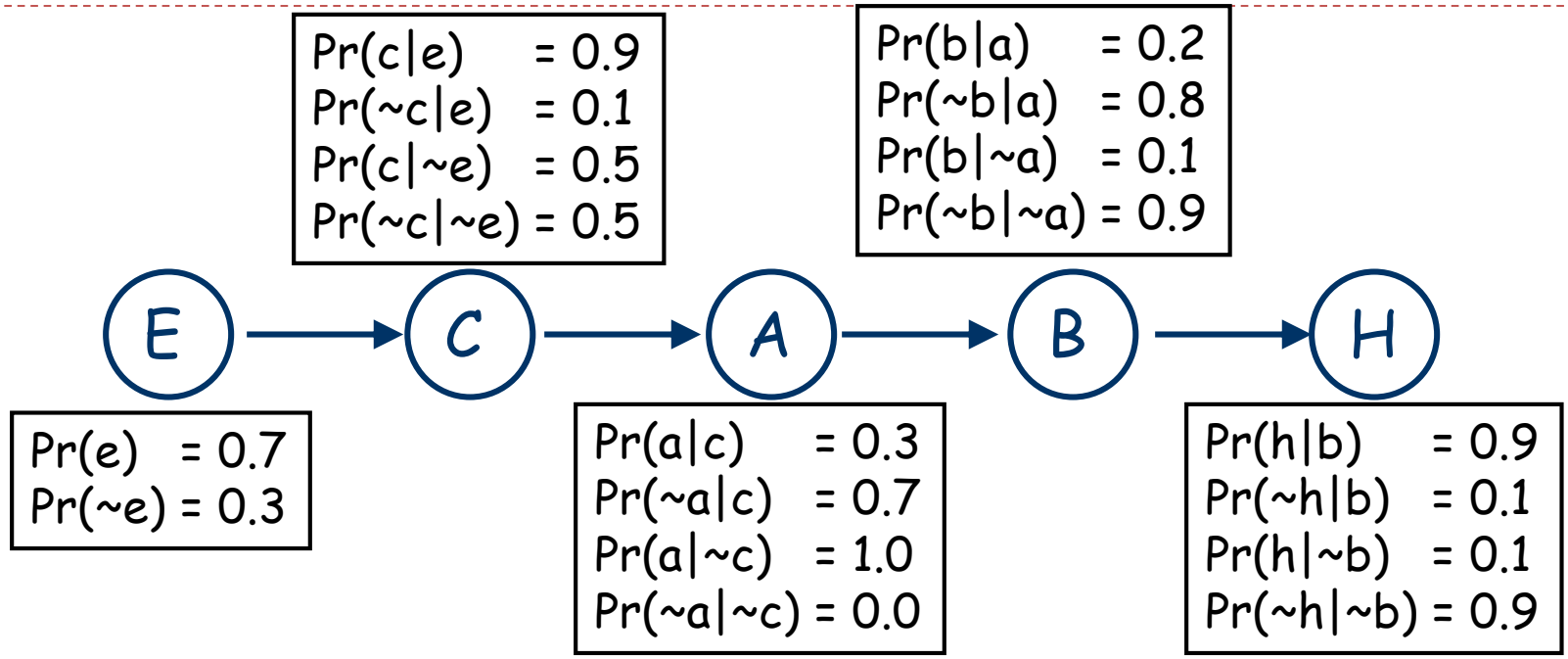
Cond'l Independence in our Story

- Similarly:
  - B is *independent* of E, and C, *given* A
  - A is *independent* of E, *given* C
- This means that:
  - Pr(H | B, {A,C,E}) = Pr(H|B)
    - i.e., for any subset of {A,C,E}, this relation holds
  - Pr(B | A, {C,E}) = Pr(B | A)
  - Pr(A | C, {E}) = Pr(A | C)
  - Pr(C | E) and Pr(E) don't "simplify"

Cond'l Independence in our Story

- By the chain rule (for any instantiation of H…E):
  - Pr(H,B,A,C,E) =
    Pr(H | B,A,C,E) Pr(B | A,C,E) Pr(A | C,E) Pr(C | E) Pr(E)
- By our independence assumptions:
  - Pr(H,B,A,C,E) =
    Pr(H | B) Pr(B | A) Pr(A | C) Pr(C | E) Pr(E)
- We can specify the full joint by specifying five *local conditional distributions*: Pr(H | B); Pr(B | A); Pr(A | C); Pr(C | E); and Pr(E)

Example Quantification



- Specifying the joint requires only 9 parameters (if we note that half of these are "1 minus" the others), instead of 31 for explicit representation
  - linear in number of vars instead of exponential!
  - linear generally if dependence has a chain structure

Inference is Easy

- Want to know P(a)? Use summing out rule:
  - Note the set of events $C=c_i$ for $c_i \in Dom(C)$ is a **partition**.

$$P(a) = \sum_{c_i \in Dom(C)} \Pr(a \mid c_i) \Pr(c_i)$$

$$= \sum_{c_i \in Dom(C)} \Pr(a \mid c_i) \sum_{e_i \in Dom(E)} \Pr(c_i \mid e_i) \Pr(e_i)$$

These are all terms specified in our local distributions!

Inference is Easy

- Computing P(a) in more concrete terms:
  - P(c) = P(c | e)P(e) + P(c | ~e)P(~e)
    = 0.8 * 0.7 + 0.5 * 0.3 = 0.78
  - P(~c) = P(~c | e)P(e) + P(~c | ~e)P(~e) = 0.22
    - P(~c) = 1 – P(c), as well
  - P(a) = P(a | c)P(c) + P(a | ~c)P(~c)
    = 0.7 * 0.78 + 0.0 * 0.22 = 0.546
  - P(~a) = 1 – P(a) = 0.454

Bayesian Networks

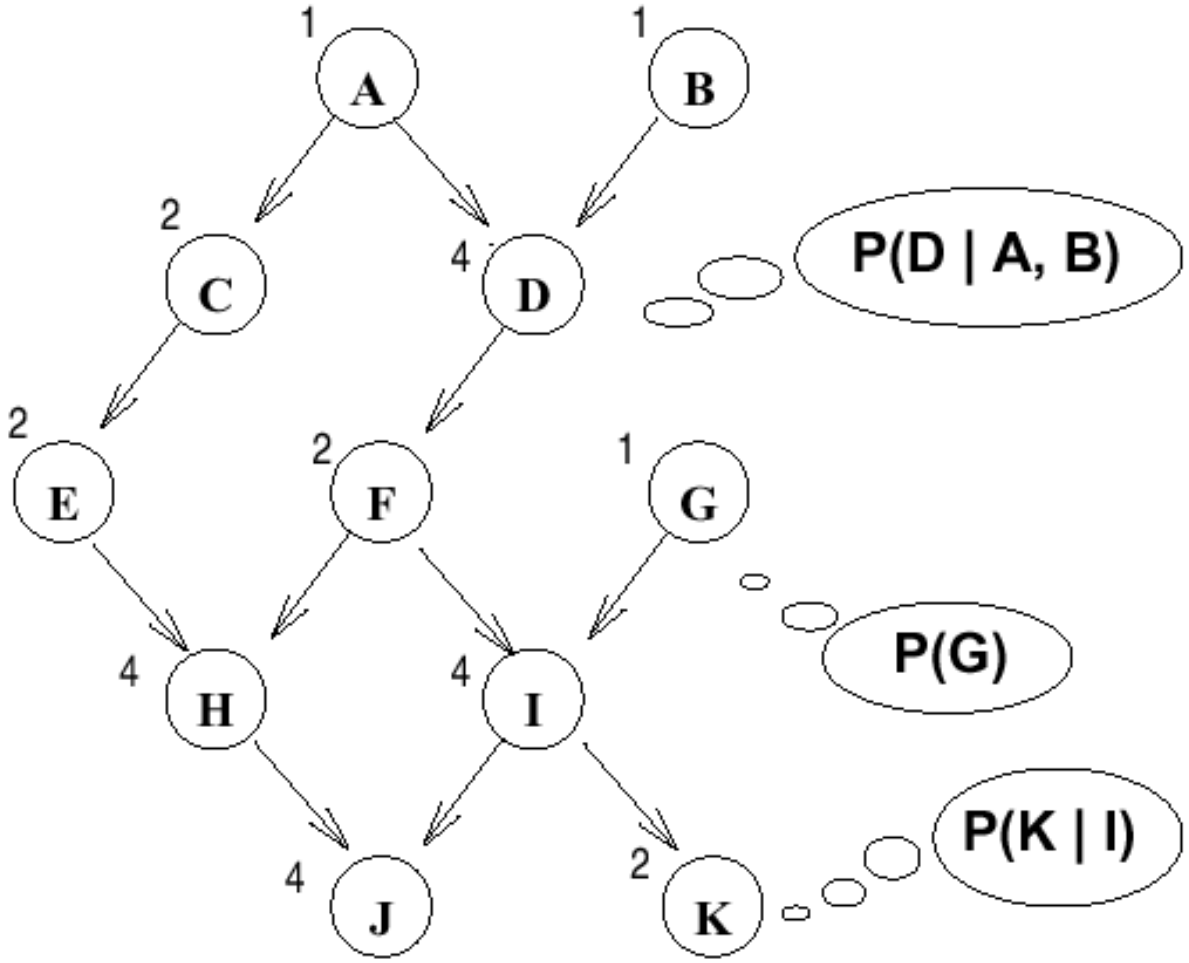
- The structure above is a *Bayesian network*. A BN is a *graphical representation* of the direct dependencies over a set of variables, together with a set of *conditional probability tables* quantifying the strength of those influences.

- Bayes nets generalize the above ideas in very interesting ways, leading to effective means of representation and inference under uncertainty.

Bayesian Networks

- A BN over variables $\{X_1, X_2, \ldots, X_n\}$ consists of:
  - a DAG (directed acyclic graph) whose nodes are the variables
  - a set of **CPTs** (conditional probability tables) $Pr(X_i \mid Par(X_i))$ for each X_i
- Key notions (see text for defn's, all are intuitive):
  - parents of a node: $Par(X_i)$
  - children of node
  - descendents of a node
  - ancestors of a node
  - family: set of nodes consisting of X_i and its parents
    - CPTs are defined over families in the BN

Example (Binary valued Variables)



- A couple of the CPTS are "shown"

Semantics of Bayes Nets.

- A Bayes net specifies that the joint distribution over the variable in the net can be written as the following product decomposition.
- $\Pr(X_1, X_2, \ldots, X_n)$
  $= \Pr(X_n \mid \text{Par}(X_n)) * \Pr(X_{n-1} \mid \text{Par}(X_{n-1}))$
  $* \ldots * \Pr(X_1 \mid \text{Par}(X_1))$
- This equation hold for any set of values $d_1, d_2, \ldots, d_n$ for the variables $X_1, X_2, \ldots, X_n$.

Semantics of Bayes Nets.

- E.g., say we have X_1, X_2, X_3 each with domain $Dom[X_i] = \{a, b, c\}$ and we have

$$Pr(X_1,X_2,X_3) = P(X_3 \mid X_2)\, P(X_2)\, P(X_1)$$

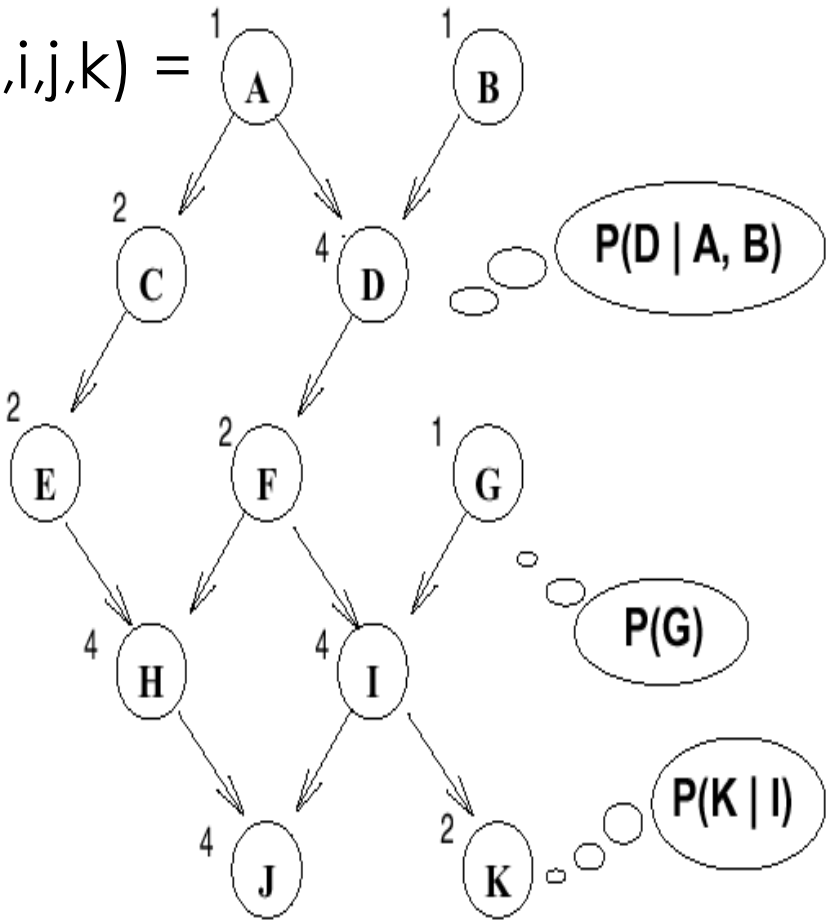
Then

$$Pr(X_1=a,X_2=a,X_3=a) = P(X_3=a \mid X_2=a)\, P(X_2=a)\, P(X_1=a)$$
$$Pr(X_1=a,X_2=a,X_3=b) = P(X_3=b \mid X_2=a)\, P(X_2=a)\, P(X_1=a)$$
$$Pr(X_1=a,X_2=a,X_3=c) = P(X_3=c \mid X_2=a)\, P(X_2=a)\, P(X_1=a)$$
$$Pr(X_1=a,X_2=b,X_3=a) = P(X_3=a \mid X_2=b)\, P(X_2=b)\, P(X_1=a)$$
$$\ldots$$

Example (Binary valued Variables)

Pr(a,b,c,d,e,f,g,h,i,j,k) =

Pr(a)
x Pr(b)
x Pr(c | a)
x Pr(d | a,b)
x Pr(e | c)
x Pr(f | d)
x Pr(g)
x Pr(h | e,f)
x Pr(i | f,g)
x Pr(j | h,i)
x Pr(k | i)



- Explicit joint requires 2^{11} -1 =2047 parmeters
- BN requires only 27 parmeters (the number of entries for each CPT is listed)

Semantics of Bayes Nets.

- Note that this means we can compute the probability of any setting of the variables using only the information contained in the CPTs of the network.

Constructing a Bayes Net

- It is always possible to construct a Bayes net to represent any distribution over the variables X_1, X_2,…, X_n, using any ordering of the variables.

- Take any ordering of the variables (say, the order given). From the chain rule we obtain.

$$\Pr(X_1,\ldots,X_n) = \Pr(X_n|X_1,\ldots,X_{n-1})\Pr(X_{n-1}|X_1,\ldots,X_{n-2})\ldots\Pr(X_1)$$

- Now for each Xi go through its conditioning set $X_1,\ldots,X_{i-1}$, and iteratively remove all variables X_j such that X_i is conditionally independent of X_j given the remaining variables. Do this until no more variables can be removed.
- The final product will specify a Bayes net.

Constructing a Bayes Net

- The end result will be a product decomposition/Bayes net
  $\Pr(X_n \mid Par(X_n))\ \Pr(X_{n-1} \mid Par(X_{n-1}))\ldots\ \Pr(X_1)$

- Now we specify the numeric values associated with each term $\Pr(X_i \mid Par(X_i))$ in a CPT.
- Typically we represent the CPT as a table mapping each setting of $\{X_i, Par(X_i)\}$ to the probability of X_i taking that particular value given that the variables in $Par(X_i)$ have their specified values.
- If each variable has d different values.
  - We will need a table of size $d^{|\{X_i, Par(X_i)\}|}$.
  - That is, exponential in the size of the parent set.

- Note that the original chain rule
  $\Pr(X_1,\ldots,X_n) = \Pr(X_n \mid X_1,\ldots,X_{n-1})\Pr(X_{n-1} \mid X_1,\ldots,X_{n-2})\ldots\Pr(X_1)$
  requires as much space to represent as specifying the probability of each individual event.

Causal Intuitions

- The BN can be constructed using an arbitrary ordering of the variables.
- However, some orderings will yield BN's with very large parent sets. This requires exponential space, and (as we will see later) exponential time to perform inference.
- Empirically, and conceptually, a good way to construct a BN is to use an ordering based on causality. This often yields a more natural and compact BN.

Causal Intuitions

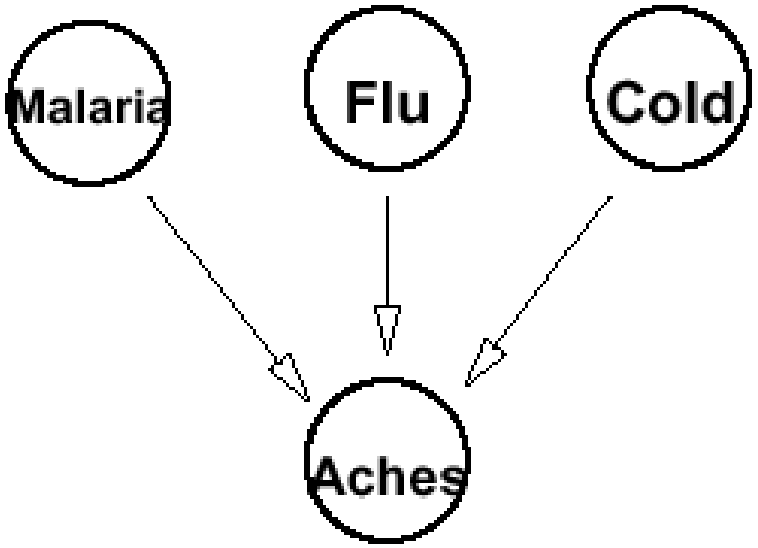
- Malaria, the flu and a cold all "cause" aches. So use the ordering that causes come before effects
  Malaria, Flu, Cold, Aches

  $Pr(M,F,C,A) = Pr(A \mid M,F,C)\ Pr(C \mid M,F)\ Pr(F \mid M)\ Pr(M)$

- Each of these disease affects the probability of aches, so the first conditional probability does not change.
- It is reasonable to assume that these diseases are independent of each other: having or not having one does not change the probability of having the others. So $Pr(C \mid M,F) = Pr(C)$
  $Pr(F \mid M) = Pr(F)$

Causal Intuitions

- This yields a fairly simple Bayes net.
- Only need one big CPT, involving the family of "Aches".



Causal Intuitions

- Suppose we build the BN for distribution P using the opposite ordering
  - i.e., we use ordering Aches, Cold, Flu, Malaria

    $Pr(A,C,F,M) = Pr(M \mid A,C,F)\ Pr(F \mid A,C)\ Pr(C \mid A)\ Pr(A)$

  - We can't reduce $Pr(M \mid A,C,F)$. Probability of Malaria is clearly affected by knowing aches. What about knowing aches and Cold, or aches and Cold and Flu?
    - Probability of Malaria is affected by both of these additional pieces of knowledge

      Knowing Cold and of Flu lowers the probability of Aches indicating Malaria since they "explain away" Aches!

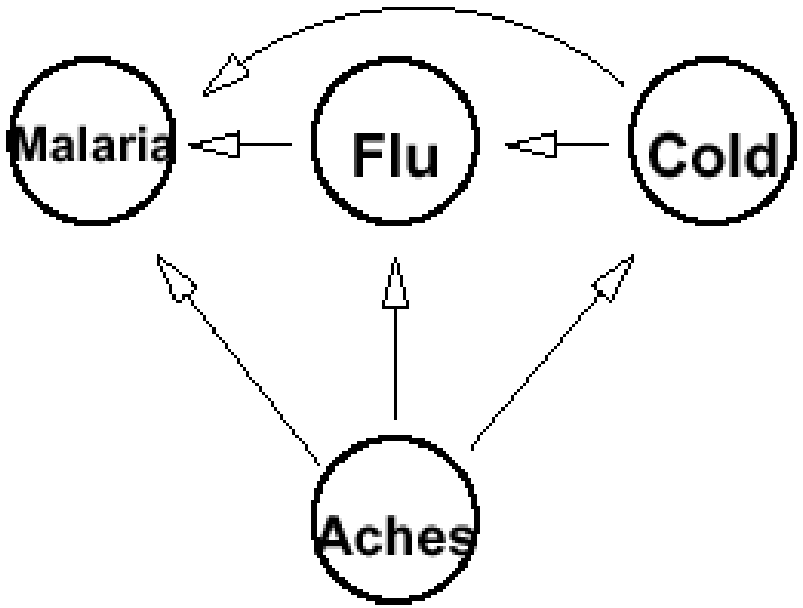
Causal Intuitions

$$\Pr(A,C,F,M) = \Pr(M \mid A,C,F)\ \Pr(F \mid A,C)\ \Pr(C \mid A)\ \Pr(A)$$

- Similarly, we can't reduce $\Pr(F \mid A,C)$.
- $\Pr(C \mid A) \neq \Pr(C)$

Causal Intuitions

- Obtain a much more complex Bayes net. In fact, we obtain no savings over explicitly representing the full joint distribution (i.e., representing the probability of every atomic event).



Bayes Net Examples

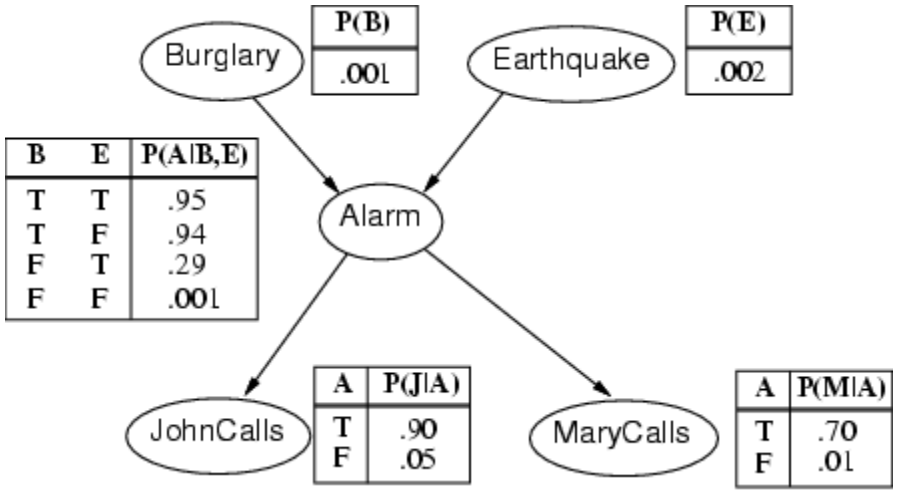
- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

- Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

- Network topology reflects "causal" knowledge:
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Burglary Example

- A burglary can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call



- # of Params: $1 + 1 + 4 + 2 + 2 = 10$ (vs. $2^5 - 1 = 31$)

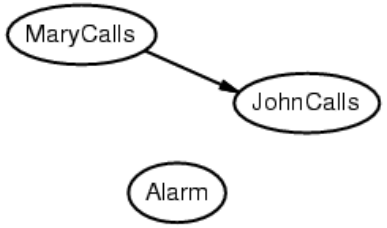
Example of Constructing Bayes Network

- Suppose we choose the ordering *M, J, A, B, E*
-

$\boldsymbol{P}(J \mid M) = \boldsymbol{P}(J)$?

Example continue…

- Suppose we choose the ordering *M, J, A, B, E*



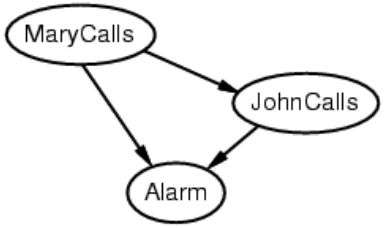
***P**(J | M) = **P**(J)?*

No

***P**(A | J, M) = **P**(A | J)? **P**(A | J, M) = **P**(A)?*

Example continue…

- Suppose we choose the ordering *M, J, A, B, E*



***P**(J | M) = **P**(J)?*

No

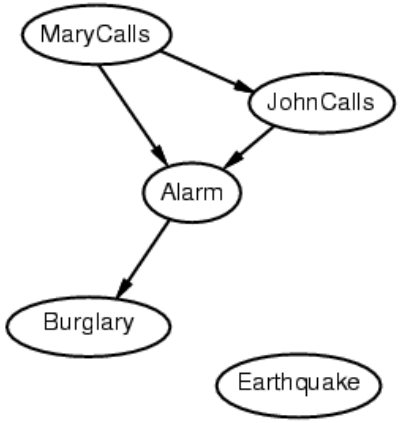
***P**(A | J, M) = **P**(A | J)?* ***P**(A | J, M) = **P**(A)?* **No**

***P**(B | A, J, M) = **P**(B | A)?*

***P**(B | A, J, M) = **P**(B)?*

Example continue…

- Suppose we choose the ordering M, J, A, B, E



P(*J* | *M*) = ***P***(*J*)? **No**

P(*A* | *J*, *M*) = ***P***(*A* | *J*)? ***P***(*A* | *J*, *M*) = ***P***(*A*)? **No**

P(*B* | *A*, *J*, *M*) = ***P***(*B* | *A*)? **Yes**

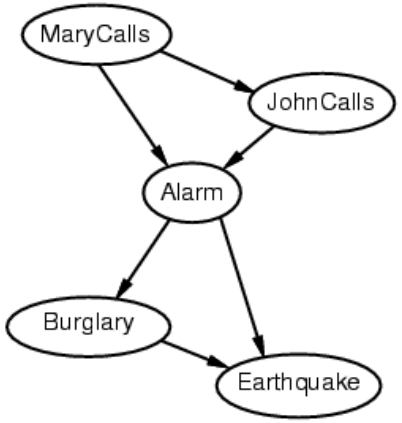
P(*B* | *A*, *J*, *M*) = ***P***(*B*)? **No**

P(*E* | *B*, *A* ,*J*, *M*) = ***P***(*E* | *A*)?

P(*E* | *B*, *A*, *J*, *M*) = ***P***(*E* | *A*, *B*)?

Example continue…

- Suppose we choose the ordering M, J, A, B, E
-



P(*J* | *M*) = ***P***(*J*)? **No**

P(*A* | *J*, *M*) = ***P***(*A* | *J*)? ***P***(*A* | *J*, *M*) = ***P***(*A*)? **No**

P(*B* | *A*, *J*, *M*) = ***P***(*B* | *A*)? **Yes**

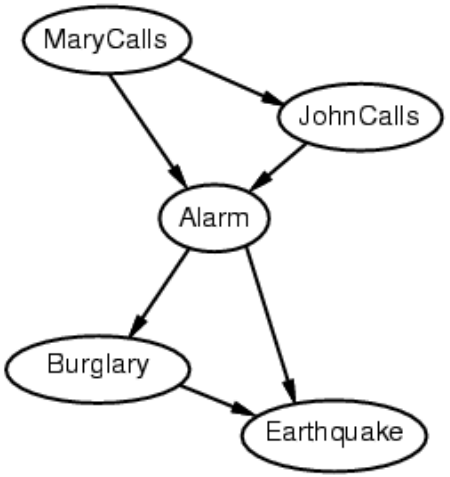
P(*B* | *A*, *J*, *M*) = ***P***(*B*)? **No**

P(*E* | *B*, *A* ,*J*, *M*) = ***P***(*E* | *A*)? **No**

P(*E* | *B*, *A*, *J*, *M*) = ***P***(*E* | *A*, *B*)? **Yes**

Example continue…

- Deciding conditional independence **is hard** in non-causal directions!
- (Causal models and conditional independence seem hardwired for humans!)
- Network is **less compact**: 1 + 2 + 4 + 2 + 4 = 13 numbers needed



Inference in Bayes Nets

- Given a Bayes net

  $Pr(X_1, X_2, \ldots, X_n)$

  $= Pr(X_n \mid Par(X_n)) * Pr(X_{n-1} \mid Par(X_{n-1}))$

  $* \ldots * Pr(X_1 \mid Par(X_1))$

- And some evidence E = {a set of values for some of the variables} we want to compute the new probability distribution

  $Pr(X_k \mid E)$

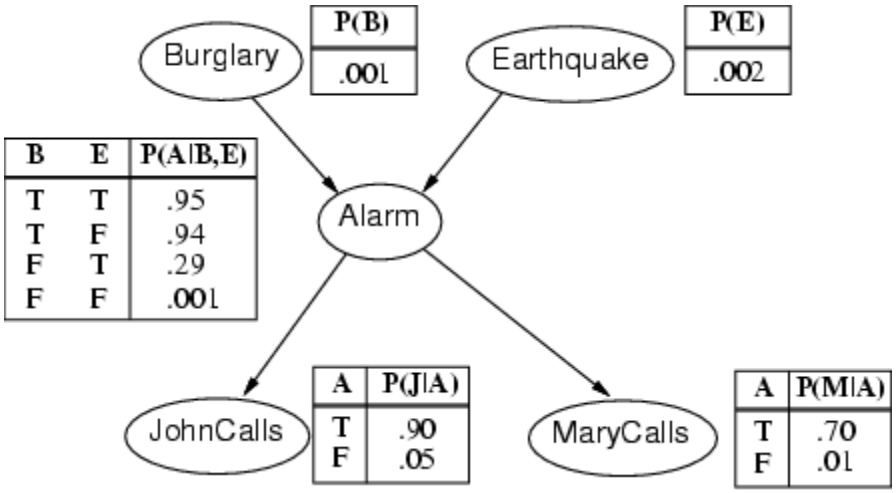
- That is, we want to figure our Pr(X_k = d | E) for all d$\in$ Dom[X_k]

Inference in Bayes Nets

- Other types of examples are, computing probability of different diseases given symptoms, computing probability of hail storms given different metrological evidence, etc.
- In such cases getting a good estimate of the probability of the unknown event allows us to respond more effectively (gamble rationally)

Inference in Bayes Nets

- In the Alarm example:



- Pr(Burglary,Earthquake, Alarm, JohnCalls, MaryCalls) =
  Pr(Earthquake) * Pr(Burglary) *
  Pr(Alarm | Earthquake,Burglary) *
  Pr(JohnCalls | Alarm) * Pr(MaryCalls | Alarm)
- And, e.g., we want to compute things like Pr(Burglary=True | MaryCalls=false, JohnCalls=true)

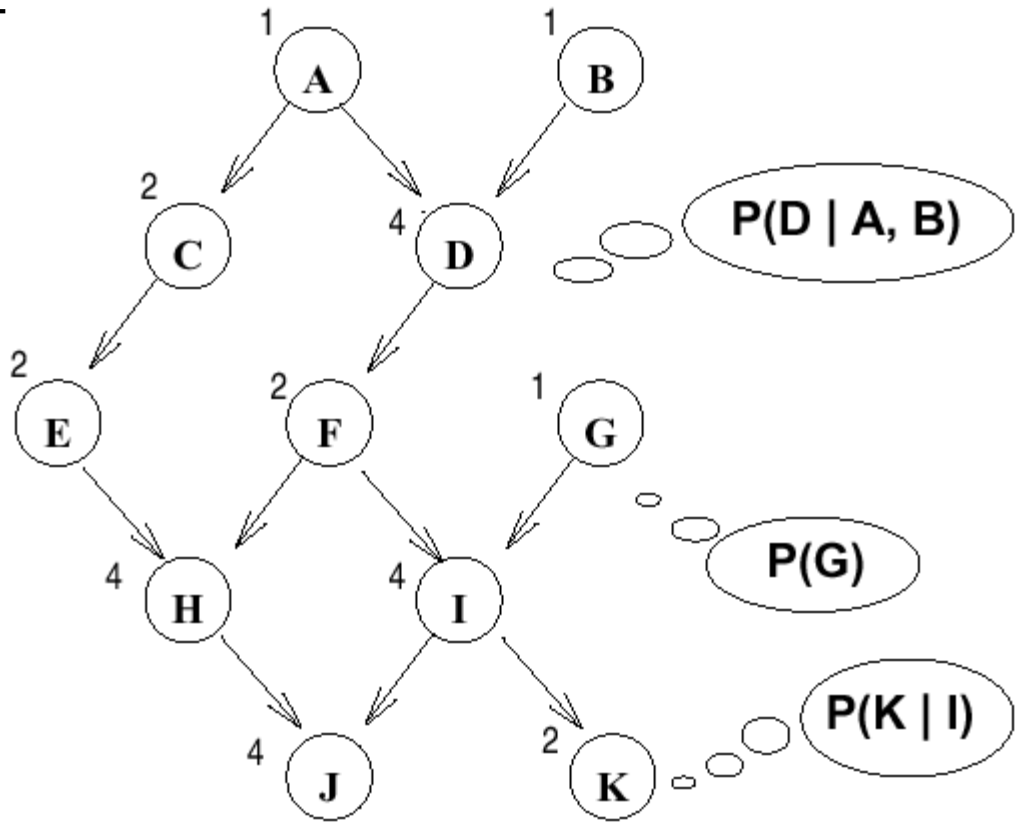
Variable Elimination

- Variable elimination uses the product decomposition that defines the Bayes Net and the summing out rule to compute posterior probabilities from the information (CPTs) already in the network.

Example (Binary valued Variables)

Pr(A,B,C,D,E,F,G,H,I,J,K) =

Pr(A)
x Pr(B)
x Pr(C | A)
x Pr(D | A,B)
x Pr(E | C)
x Pr(F | D)
x Pr(G)
x Pr(H | E,F)
x Pr(I | F,G)
x Pr(J | H,I)
x Pr(K | I)



Example

$$\Pr(A,B,C,D,E,F,G,H,I,J,K) = \Pr(A)\Pr(B)\Pr(C \mid A)\Pr(D \mid A,B)\Pr(E \mid C)\Pr(F \mid D)\Pr(G)\Pr(H \mid E,F)\Pr(I \mid F,G)\Pr(J \mid H,I)\Pr(K \mid I)$$

Say that E = {H=true, I=false}, and we want to know $\Pr(D \mid h,i)$ (h: H is true, -h: H is false)

1. Write as a sum for each value of D

$$\sum_{A,B,C,E,F,G,J,K} \Pr(A,B,C,d,E,F,h,-i,J,K) = \Pr(d,h,-i)$$

$$\sum_{A,B,C,E,F,G,J,K} \Pr(A,B,C,-d,E,F,h,-i,J,K) = \Pr(-d,h,-i)$$

Example

2. $Pr(d,h,-i) + Pr(-d,h,-i) = Pr(h,-i)$
3. $Pr(d \mid h,-i) = Pr(d,h,-i)/Pr(h,-i)$
   $Pr(-d \mid h,-i) = Pr(-d,h,-i)/Pr(h,-i)$

So we only need to compute $Pr(d,h,-i)$ and $Pr(-d,h,-i)$ and then normalize to obtain the conditional probabilities we want.

Example

$Pr(d,h,-i) = \sum_{A,B,C,E,F,G,J,K} Pr(A,B,C,d,E,F,h,-i,J,K)$

Use Bayes Net product decomposition to rewrite summation:

$$\sum_{A,B,C,E,F,G,J,K} Pr(A,B,C,d,E,F,h,-i,J,K)$$
$$= \sum_{A,B,C,E,F,G,J,K} Pr(A)Pr(B)Pr(C \mid A)Pr(d \mid A,B)Pr(E \mid C) Pr(F \mid d)Pr(G)Pr(h \mid E,F)Pr(-i \mid F,G)Pr(J \mid h,-i) Pr(K \mid -i)$$

Now rearrange summations so that we are not summing over that do not depend on the summed variable.

Example

$$= \sum_A \sum_B \sum_C \sum_E \sum_F \sum_G \sum_J \sum_K \Pr(A)\Pr(B)\Pr(C \mid A)\Pr(d \mid A,B)\Pr(E \mid C) \Pr(F \mid d)\Pr(G)\Pr(h \mid E,F)\Pr(-i \mid F,G)\Pr(J \mid h,-i) \Pr(K \mid -i)$$

$$= \sum_A \Pr(A) \sum_B \Pr(B) \sum_C \Pr(C \mid A)\Pr(d \mid A,B) \sum_E \Pr(E \mid C) \sum_F \Pr(F \mid d) \sum_G \Pr(G)\Pr(h \mid E,F)\Pr(-i \mid F,G) \sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i)$$

$$= \sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C) \sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G) \sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i)$$

Example

- Now start computing.

$$\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

$$\sum_K \Pr(K \mid -i) = \Pr(k \mid -i) + \Pr(-k \mid -i) = c_1$$

$$\sum_J \Pr(J \mid h,-i)\ c_1 = c_1 \sum_J \Pr(J \mid h,-i)$$
$$= c_1\ (\Pr(j \mid h,-i) + \Pr(-j \mid h,-i))$$
$$= c_1 c_2$$

Example

$$\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

$$c_1 c_2 \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$= c_1 c_2 (\Pr(g)\Pr(-i \mid F,g) + \Pr(-g)\Pr(-i \mid F,-g))$$

!!But $\Pr(-i \mid F,g)$ depends on the value of F, so this is not a single number.

Example

- Try the other order of summing.

$$\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

=

$$\Pr(a) \sum_B \Pr(B) \Pr(d \mid a,B) \sum_C \Pr(C \mid a) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

+

$$\Pr(-a) \sum_B \Pr(B) \Pr(d \mid -a,B) \sum_C \Pr(C \mid -a) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

Example

$$
\begin{aligned}
&= \\
&\Pr(a)\Pr(b)\ \Pr(d \mid a,b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(\neg i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,\neg i) \\
&\qquad \textstyle\sum_K \Pr(K \mid \neg i) \\
&+ \\
&\Pr(a)\Pr(\neg b)\ \Pr(d \mid a,\neg b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(\neg i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,\neg i) \\
&\qquad \textstyle\sum_K \Pr(K \mid \neg i) \\
&+ \\
&\Pr(\neg a)\Pr(b)\ \Pr(d \mid \neg a,b)\ \textstyle\sum_C \Pr(C \mid \neg a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(\neg i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,\neg i) \\
&\qquad \textstyle\sum_K \Pr(K \mid \neg i) \\
&+ \\
&\Pr(\neg a)\Pr(\neg b)\ \Pr(d \mid \neg a,\neg b)\ \textstyle\sum_C \Pr(C \mid \neg a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(\neg i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,\neg i) \\
&\qquad \textstyle\sum_K \Pr(K \mid \neg i)
\end{aligned}
$$

Example

=

Yikes! The size of the sum is doubling as we expand each variable (into –v and v). This approach has exponential complexity.

But let's look a bit closer.

Example

$$
\begin{aligned}
&= \\
&\Pr(a)\Pr(b)\ \Pr(d \mid a,b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \\
&+ \\
&\Pr(a)\Pr(-b)\ \Pr(d \mid a,-b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \\
&+ \\
&\Pr(-a)\Pr(b)\ \Pr(d \mid -a,b)\ \textstyle\sum_C \Pr(C \mid -a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \\
&+ \\
&\Pr(-a)\Pr(-b)\ \Pr(d \mid -a,-b)\ \textstyle\sum_C \Pr(C \mid -a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i)
\end{aligned}
$$

- Repeated subterm

- Repeated subterm

Dynamic Programming

- If we store the value of the subterms, we need only compute them once.

Dynamic Programming

$$
\begin{aligned}
&= \\
&\Pr(a)\Pr(b)\ \Pr(d \mid a,b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \qquad \blacksquare\ f_1 \\
&+ \\
&\Pr(a)\Pr(-b)\ \Pr(d \mid a,-b)\ \textstyle\sum_C \Pr(C \mid a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \\
&+ \\
&\Pr(-a)\Pr(b)\ \Pr(d \mid -a,b)\ \textstyle\sum_C \Pr(C \mid -a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i) \qquad \blacksquare\ f_2 \\
&+ \\
&\Pr(-a)\Pr(-b)\ \Pr(d \mid -a,-b)\ \textstyle\sum_C \Pr(C \mid -a)\ \sum_E \Pr(E \mid C) \\
&\qquad \textstyle\sum_F \Pr(F \mid d)\ \Pr(h \mid E,F) \sum_G \Pr(G)\ \Pr(-i \mid F,G) \\
&\qquad \textstyle\sum_J \Pr(J \mid h,-i) \\
&\qquad \textstyle\sum_K \Pr(K \mid -i)
\end{aligned}
$$

- $= c_1 f_1 + c_2 f_1 + c_3 f_2 + c_3 f_2$
- $c_1 = \Pr(a)\Pr(b)\ \Pr(d|a,b)$
- $c_2 = \Pr(a)\Pr(-b)\ \Pr(d|a,-b)$
- $c_3 = \Pr(a)\Pr(-b)\ \Pr(d|a,-b)$
- $c_4 = \Pr(a)\Pr(-b)\ \Pr(d|a,-b)$

Dynamic Programming

$$f_1 = \sum_C \Pr(C \mid a) \sum_E \Pr(E \mid C) \sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G) \sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i)$$

$$= \Pr(c \mid a) \sum_E \Pr(E \mid c) \sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G) \sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i)$$

$$+ \Pr(-c \mid a) \sum_E \Pr(E \mid -c) \sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G) \sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i)$$

- Repeated subterm

Dynamic Programming

- So within the computation of the subterms we obtain more repeated smaller subterms.
- The core idea of dynamic programming is to remember all "smaller" computations, so that they can be reused.
- This can convert an exponential computation into one that takes only polynomial time.
- Variable elimination is a dynamic programming technique that computes the sum from the bottom up (starting with the smaller subterms and working its way up to the bigger terms).

Relevant (return to this later)

- A brief aside is to also note that in the sum
  $\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$
  $\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$
  $\sum_J \Pr(J \mid h,-i)$
  $\sum_K \Pr(K \mid -i)$

  we have that $\sum_K \Pr(K \mid -i) = 1$ (Why?), thus
  $\sum_J \Pr(J \mid h,-i) \sum_K \Pr(K \mid -i) = \sum_J \Pr(J \mid h,-i)$

  Furthermore $\sum_J \Pr(J \mid h,-i) = 1$.

  So we could drop these last two terms from the computation---J and K are not relevant given our query D and our evidence –i and –h. For now we keep these terms.

Variable Elimination (VE)

- VE works from the inside out, summing out K, then J, then G, …, as we tried to before.
- When we tried to sum out G

$$\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$$
$$\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$\sum_J \Pr(J \mid h,-i)$$
$$\sum_K \Pr(K \mid -i)$$

$$c_1 c_2 \sum_G \Pr(G) \Pr(-i \mid F,G)$$
$$= c_1 c_2 (\Pr(g)\Pr(-i \mid F,g) + \Pr(-g)\Pr(-i \mid F,-g))$$

we found that $\Pr(-i \mid F,-g)$ depends on the value of F, it wasn't a single number.

- However, we can still continue with the computation by computing two different numbers, one for each value of F (-f, f)!

Variable Elimination (VE)

- $t(-f) = c_1 c_2 \sum_G \Pr(G) \Pr(-i \mid -f, G)$

  $t(f) = c_1 c_2 (\sum_G \Pr(G) \Pr(-i \mid f, G)$

  - $t(-f) = c_1 c_2 (\Pr(g)\Pr(-i \mid -f, g) + \Pr(-g)\Pr(-i \mid -f, -g))$
  - $t(-f) = c_1 c_2 (\Pr(g)\Pr(-i \mid f, g) + \Pr(-g)\Pr(-i \mid f, -g))$

- Now we sum out F

Variable Elimination (VE)

- $\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d \mid A,B) \sum_C \Pr(C \mid A) \sum_E \Pr(E \mid C)$
  $\sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$
  $\sum_J \Pr(J \mid h,-i)$
  $\sum_K \Pr(K \mid -i)$

$c_1 c_2 \sum_F \Pr(F \mid d) \Pr(h \mid E,F) \sum_G \Pr(G) \Pr(-i \mid F,G)$

$= c_1 c_2 (\Pr(f \mid d) \Pr(h \mid E,f) (\sum_G \Pr(G) \Pr(-i \mid f,G))$
$+ \Pr(-f \mid d) \Pr(h \mid E,-f) (\sum_G \Pr(G) \Pr(-i \mid -f,G))$

$= c_1 c_2 \sum_F \Pr(F \mid d) \Pr(h \mid E,F) t(F)$

$t(f)$, $t(-f)$

Variable Elimination (VE)

- $c_1 c_2 (\Pr(f \mid d)\ \Pr(h \mid E,f) t(f)$
  $+\Pr(-f \mid d) \Pr(h \mid E,-f) t(-f)$

- This is a function of E, so we obtain two new numbers

  $s(e) = c_1 c_2 (\Pr(f \mid d)\ \Pr(h \mid e,f) t(f)$
  $+\Pr(-f \mid d) \Pr(h \mid e,-f) t(-f)$

  $s(-e) = c_1 c_2 (\Pr(f \mid d)\ \Pr(h \mid -e,f) t(f)$
  $+\Pr(-f \mid d) \Pr(h \mid -e,-f) t(-f)$

Variable Elimination (VE)

- On summing out E we obtain two numbers, or a function of C. Then a function of B, then a function of A. On finally summing out A we obtain the single number we wanted to compute which is Pr(d,h,-i).
- Now we can repeat the process to compute Pr(-d,h,-i).
- But instead of doing it twice, we can simply regard D as an variable in the computation.
- This will result in some computations depending on the value of D, and we obtain a different number for each value of D.
- Proceeding in this manner, summing out A will yield a function of D. (I.e., a number for each value of D).

Variable Elimination (VE)

- In general, at each stage VE will be compute a table of numbers: one number for each different instantiation of the variables that are in the sum.
- The size of these tables is exponential in the number of variables appearing in the sum, e.g.,

$$\sum_F Pr(F \mid D)\ Pr(h \mid E,F) t(F)$$

depends on the value of D and E, thus we will obtain $|Dom[D]| * |Dom[E]|$ different numbers in the resulting table.

Factors

- we call these tables of values computed by VE factors.
- Note that the original probabilities that appear in the summation, e.g., P(C | A), are also tables of values (one value for each instantiation of C and A).
- Thus we also call the original CPTs factors.

- Each factor is a function of some variables, e.g., P(C | A) = f(A,C): it maps each value of its arguments to a number.
  - A tabular representation is exponential in the number of variables in the factor.

Operations on Factors

- If we examine the inside-out summation process we see that various operations occur on factors.

- Notation: $f(\underline{\mathbf{X}},\underline{\mathbf{Y}})$ denotes a factor over the variables $\underline{\mathbf{X}} \cup \underline{\mathbf{Y}}$ (where $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ are sets of variables)

The Product of Two Factors

- Let f(**X**,**Y**) & g(**Y**,**Z**) be two factors with variables **Y** in common
- The *product* of f and g, denoted h = f x g (or sometimes just h = fg), is defined:

$$h(\mathbf{X},\mathbf{Y},\mathbf{Z}) = f(\mathbf{X},\mathbf{Y}) \times g(\mathbf{Y},\mathbf{Z})$$

f(A,B)		g(B,C)		h(A,B,C)			
ab	0.9	bc	0.7	abc	0.63	ab~c	0.27
a~b	0.1	b~c	0.3	a~bc	0.08	a~b~c	0.02
~ab	0.4	~bc	0.8	~abc	0.28	~ab~c	0.12
~a~b	0.6	~b~c	0.2	~a~bc	0.48	~a~b~c	0.12

Summing a Variable Out of a Factor

- Let f(X,**Y**) be a factor with variable X (**Y** is a set)
- We *sum out* variable X from f to produce a new factor h = $\sum_X$ f, which is defined:

$$h(\mathbf{Y}) = \sum_{x \in Dom(X)} f(x, \mathbf{Y})$$

f(A,B)		h(B)	
ab	0.9	b	1.3
a~b	0.1	~b	0.7
~ab	0.4		
~a~b	0.6		

Restricting a Factor

- Let f(X,**Y**) be a factor with variable X (**Y** is a set)
- We *restrict* factor f *to* X=a by setting X to the value x and "deleting" incompatible elements of f's domain . Define $h = f_{X=a}$ as: h(**Y**) = f(a,**Y**)

f(A,B)		h(B) = $f_{A=a}$	
ab	0.9	b	0.9
a~b	0.1	~b	0.1
~ab	0.4		
~a~b	0.6		

Variable Elimination the Algorithm

Given query var Q, evidence vars **E** (set of variables observed to have values **e**), remaining vars **Z**. Let F be original CPTs.

1. Replace each factor $f \in F$ that mentions a variable(s) in **E** with its restriction $f_{E=e}$ (this might yield a "constant" factor)
2. For each Z_j—in the order given—eliminate $Z_j \in \mathbf{Z}$ as follows:
   (a) Compute new factor $g_j = \sum_{Z_j} f_1 \times f_2 \times \ldots \times f_k$, where the f_i are the factors in F that include Z_j
   (b) Remove the factors f_i (that mention Z_j) from F and add new factor g_j to F
3. The remaining factors refer only to the query variable Q. Take their product and normalize to produce Pr(Q|E)

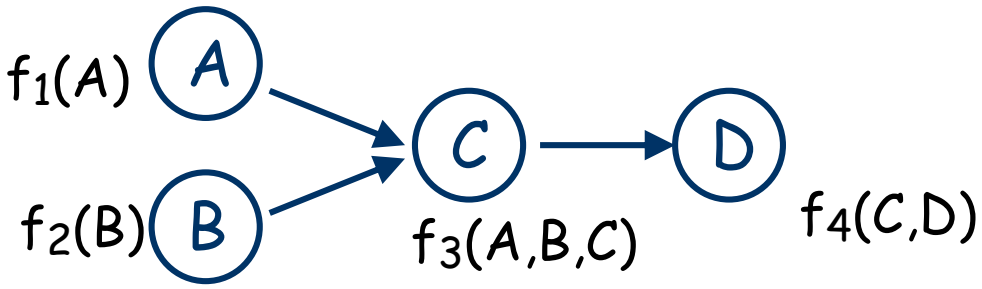
VE: Example

Factors: $f_1(A)$ $f_2(B)$ $f_3(A,B,C)$ $f_4(C,D)$

Query: P(A)?

Evidence: D = d

Elim. Order: C, B



Restriction: replace $f_4(C,D)$ with $f_5(C) = f_4(C,d)$

Step 1: Compute & Add $f_6(A,B) = \Sigma_C f_5(C)\ f_3(A,B,C)$

Remove: $f_3(A,B,C)$, $f_5(C)$

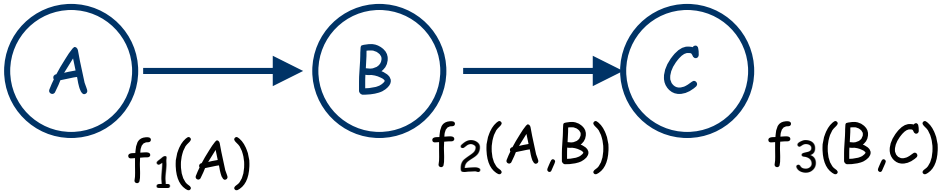
Step 2: Compute & Add $f_7(A) = \Sigma_B f_6(A,B)\ f_2(B)$

Remove: $f_6(A,B)$, $f_2(B)$

Last factors: $f_7(A)$, $f_1(A)$. The product $f_1(A) \times f_7(A)$ is (unnormalized) posterior. So… $P(A \mid d) = \alpha\ f_1(A) \times f_7(A)$ where $\alpha = 1/\Sigma_A\ f_1(A)f_7(A)$

Numeric Example

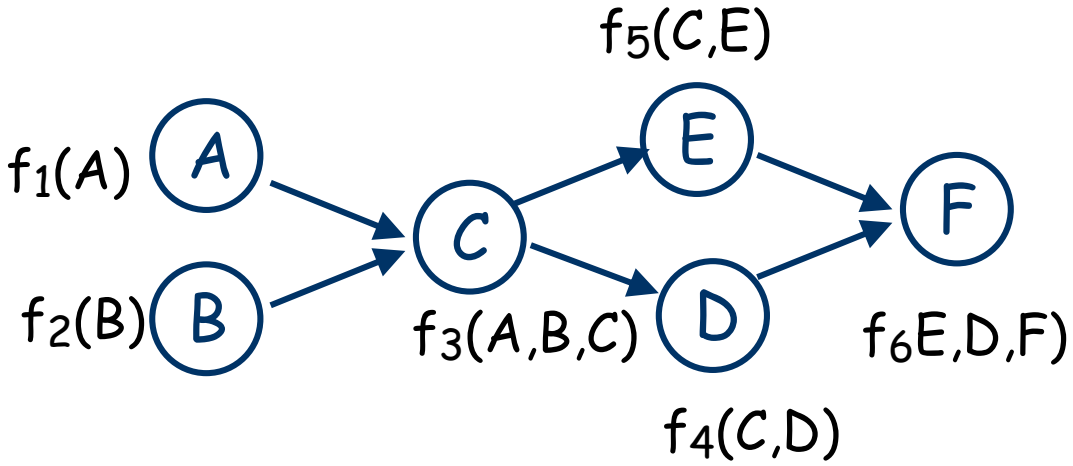
- Here's the example with some numbers



$f_1(A)$		$f_2(A,B)$		$f_3(B,C)$		$f_4(B)$ $\Sigma_A f_2(A,B)f_1(A)$		$f_5(C)$ $\Sigma_B f_3(B,C) f_4(B)$	
a	0.9	ab	0.9	bc	0.7	b	0.85	c	0.625
~a	0.1	a~b	0.1	b~c	0.3	~b	0.15	~c	0.375
		~ab	0.4	~bc	0.2				
		~a~b	0.6	~b~c	0.8				

VE: Buckets as a Notational Device

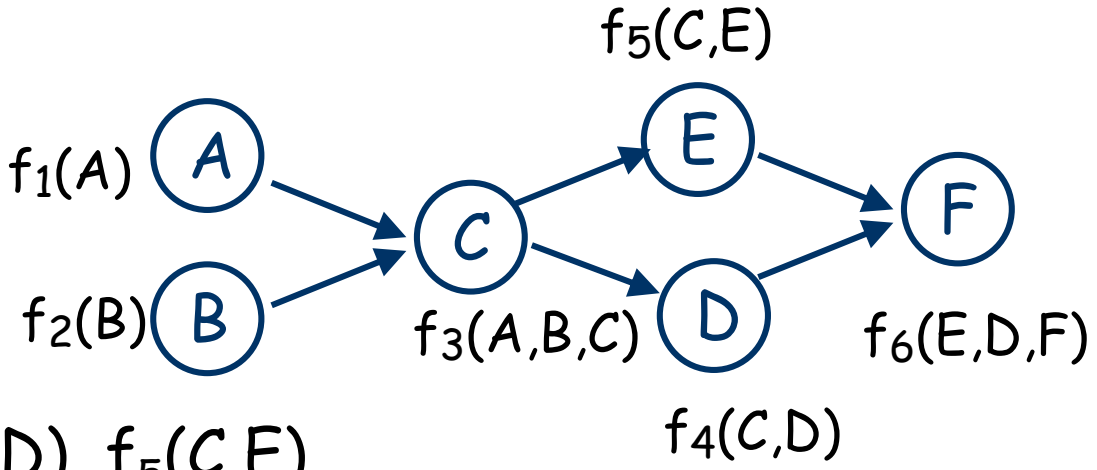
Ordering:
C,F,A,B,E,D



1. C:
2. F:
3. A:
4. B:
5. E:
6. D:

VE: Buckets—Place Original Factors in first applicable bucket.

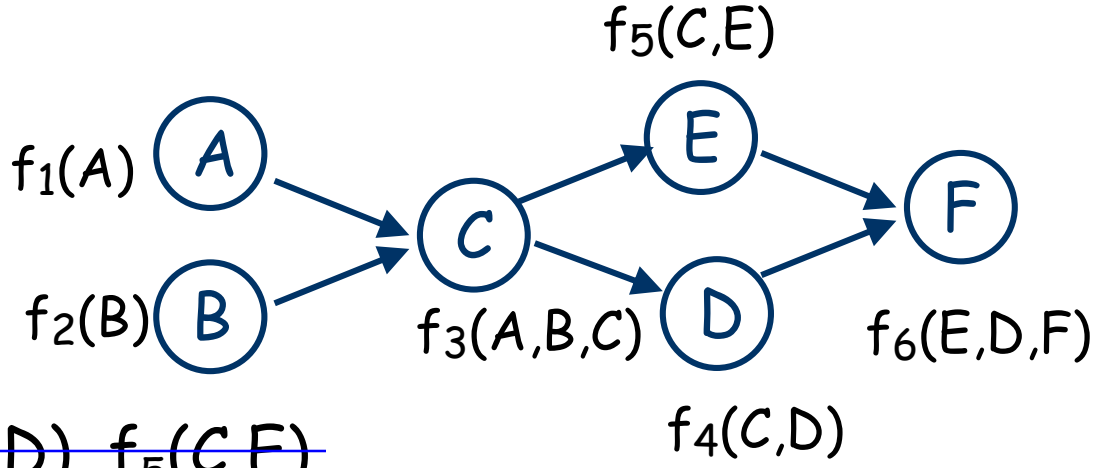
Ordering:
C,F,A,B,E,D



1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

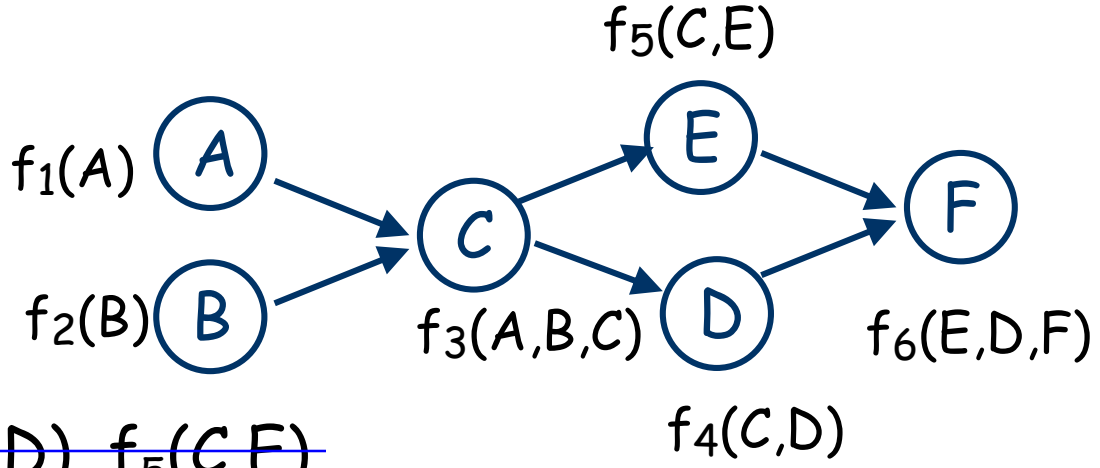


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

1. Σ_C $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
   = $f_7(A,B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

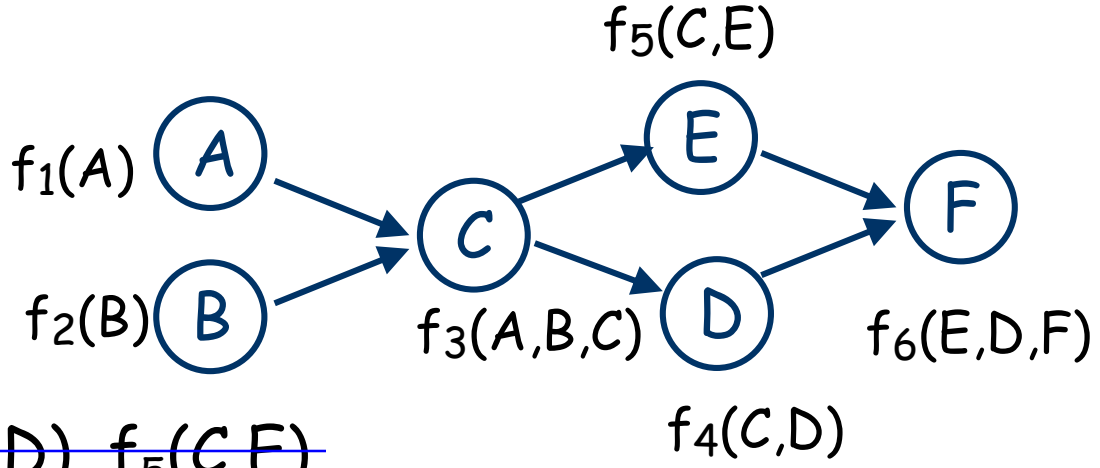


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

2. $\Sigma_F f_6(E,D,F) = f_8(E,D)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

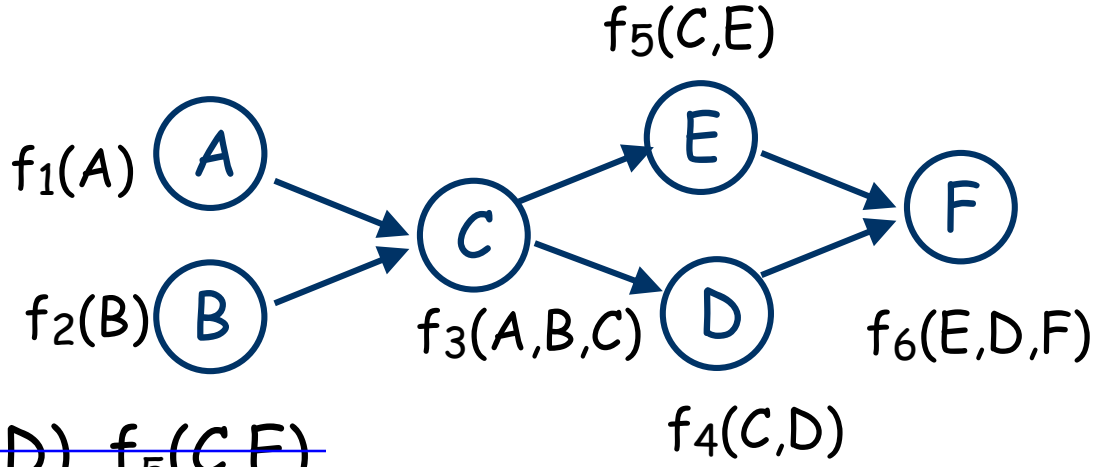


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. B: $f_2(B)$, $f_9(B,D,E)$
5. E: $f_8(E,D)$
6. D:

3. $\Sigma_A f_1(A), f_7(A,B,D,E) = f_9(B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

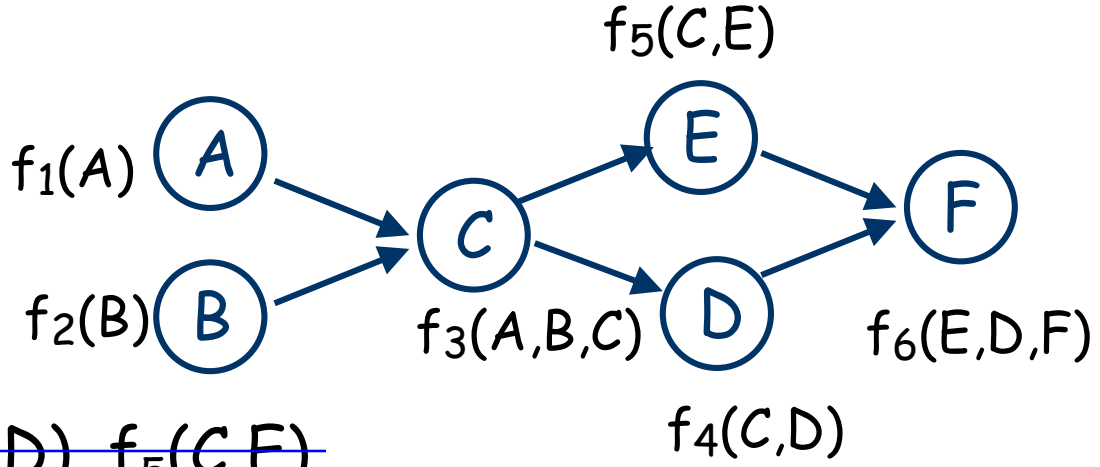


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

4. Σ_B $f_2(B)$, $f_9(B,D,E)$
   = $f_{10}(D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D



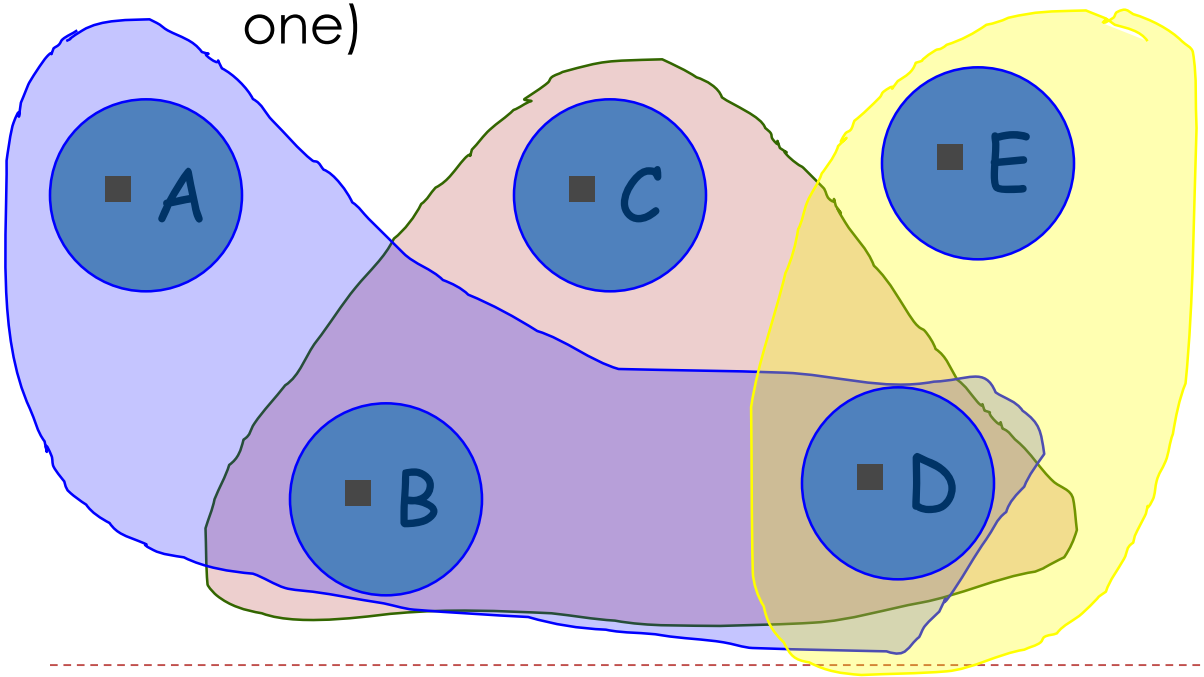
1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. ~~E: $f_8(E,D)$, $f_{10}(D,E)$~~
6. D: $f_{11}(D)$

5. Σ_E $f_8(E,D)$, $f_{10}(D,E)$
   = $f_{11}(D)$

f_{11} is he final answer, once we normalize it.

Complexity of Variable Elimination

- Hypergraph of Bayes Net.
  - Hypergraph has vertices just like an ordinary graph, but instead of edges between two vertices X↔Y it contains **hyperedges**.
    - A hyperedge is a set of vertices (i.e., potentially more than one)



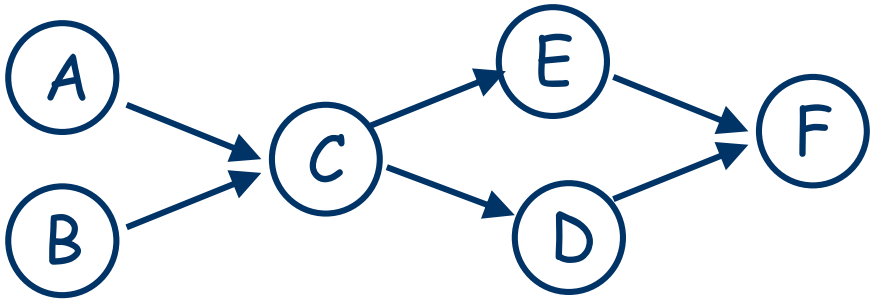
- {A,B,D}
{B,C,D}
{E,D}

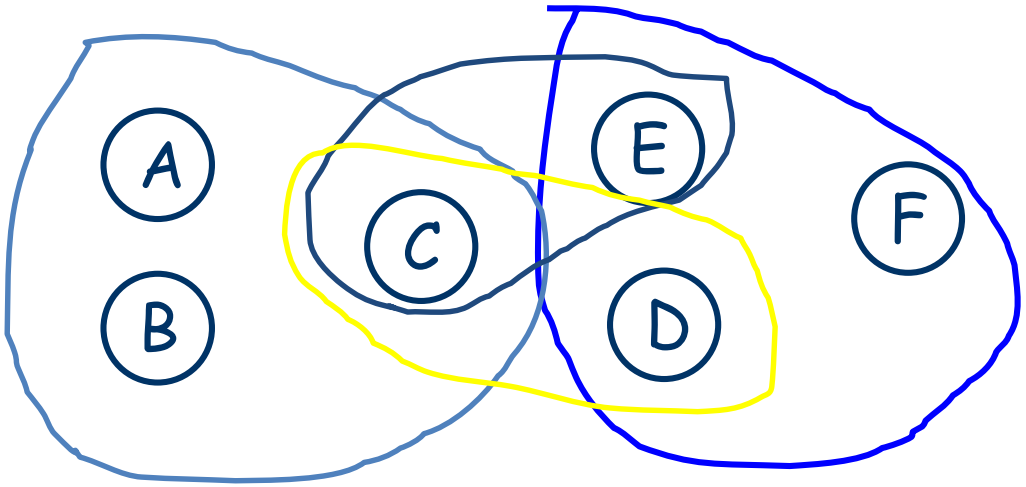
Complexity of Variable Elimination

- Hypergraph of Bayes Net.
  - The set of vertices are precisely the nodes of the Bayes net.
  - The hyperedges are the variables appearing in each CPT.
    - $\{X_i\} \cup Par(X_i)$

Complexity of Variable Elimination

- Pr(A,B,C,D,E,F) =
  Pr(A)Pr(B)
  X Pr(C | A,B)
  X Pr(E | C)
  X Pr(D | C)
  X Pr(F | E,D).

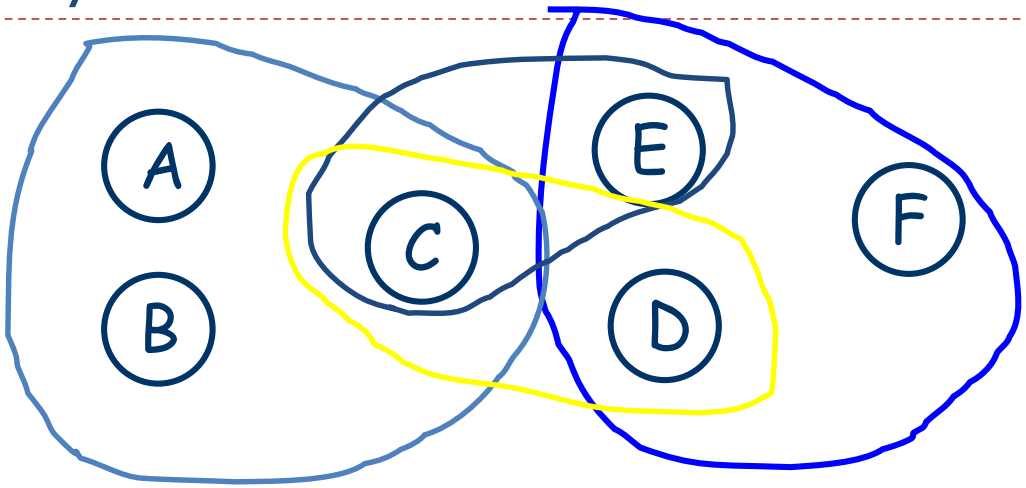




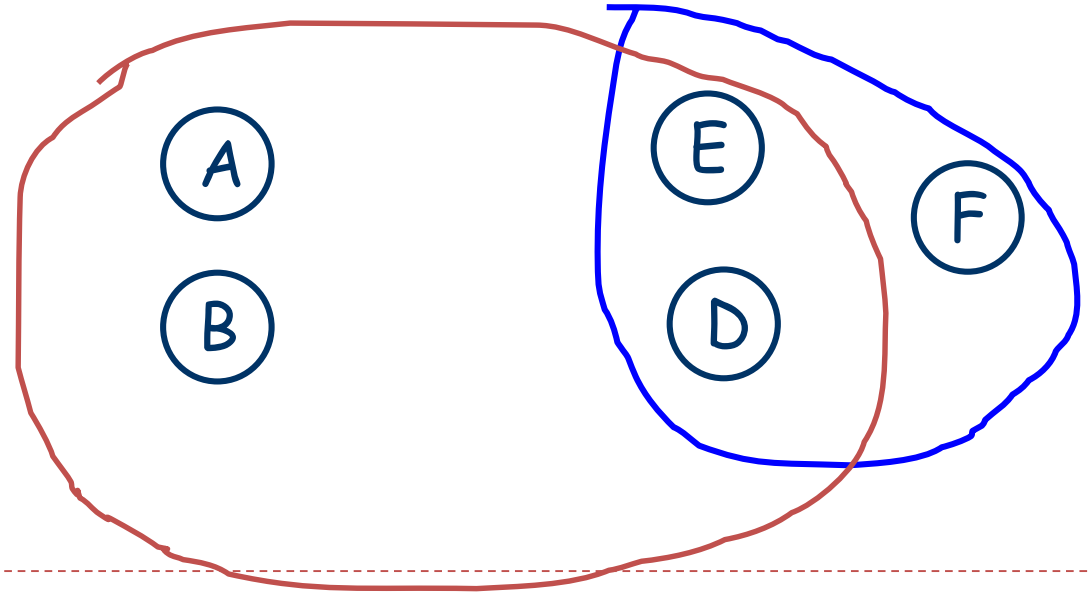
Variable Elimination in the HyperGraph

- To eliminate variable X_i in the hypergraph we
  - we remove the vertex X_i
  - Create a new hyperedge H_i equal to the union of all of the hyperedges that contain X_i minus X_i
  - Remove all of the hyperedges containing X from the hypergraph.
  - Add the new hyperedge H_i to the hypergraph.

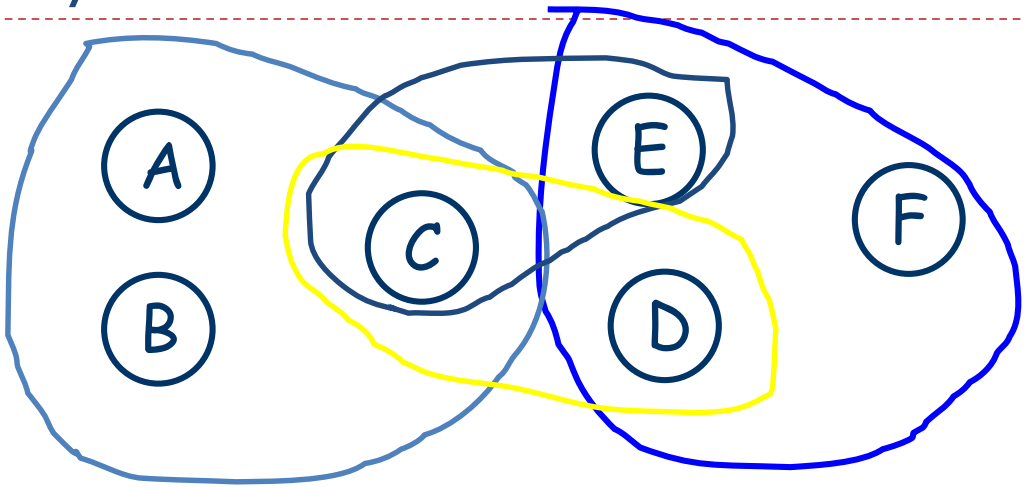
Complexity of Variable Elimination



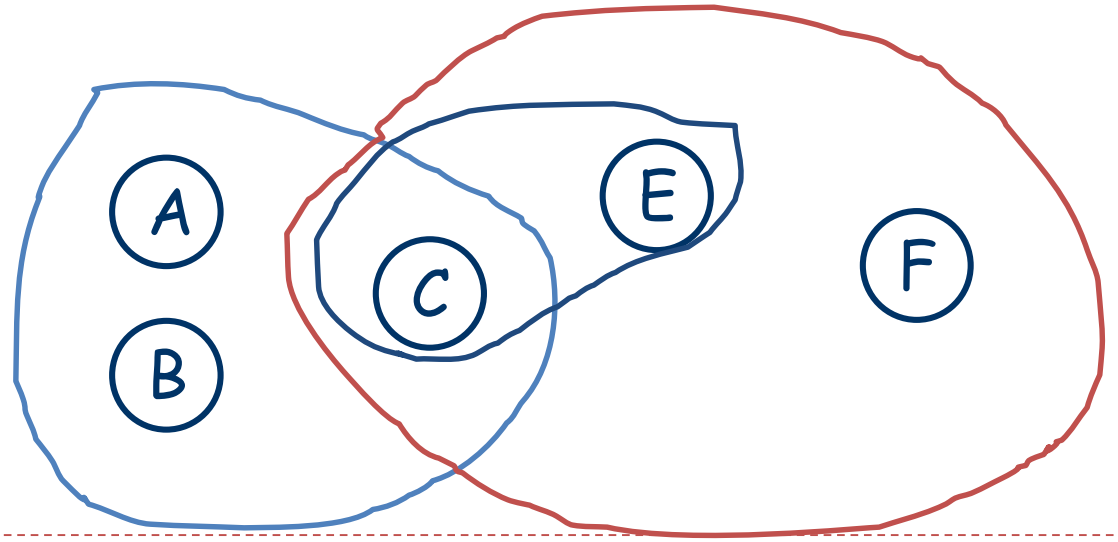
- Eliminate C



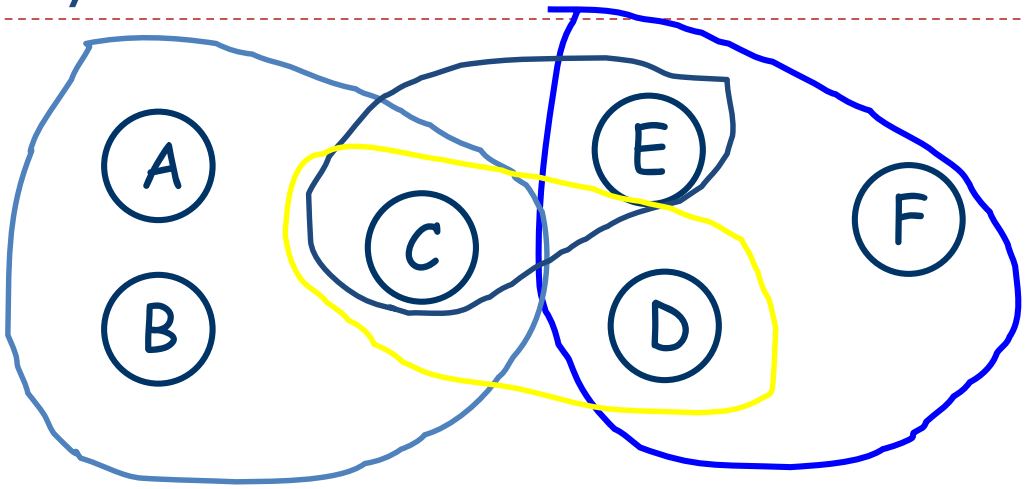
Complexity of Variable Elimination



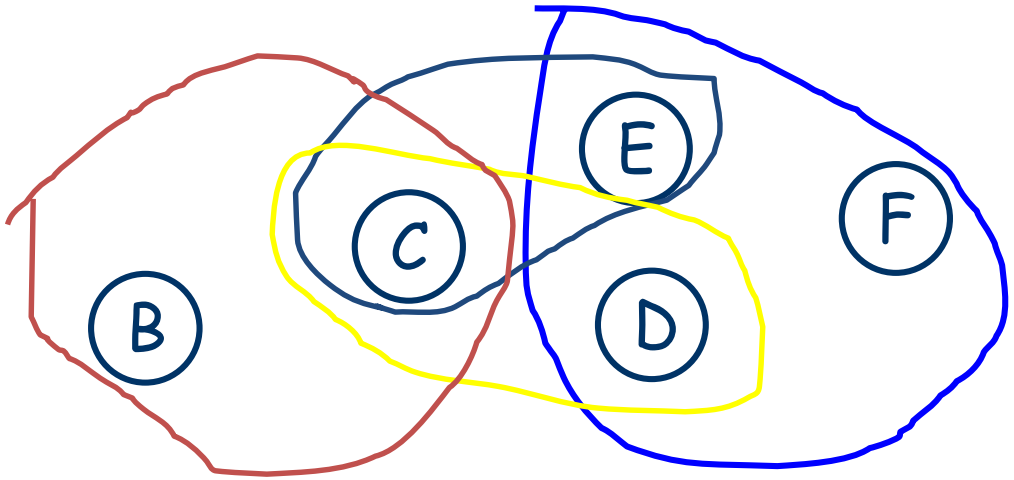
- Eliminate D



Complexity of Variable Elimination



- Eliminate A



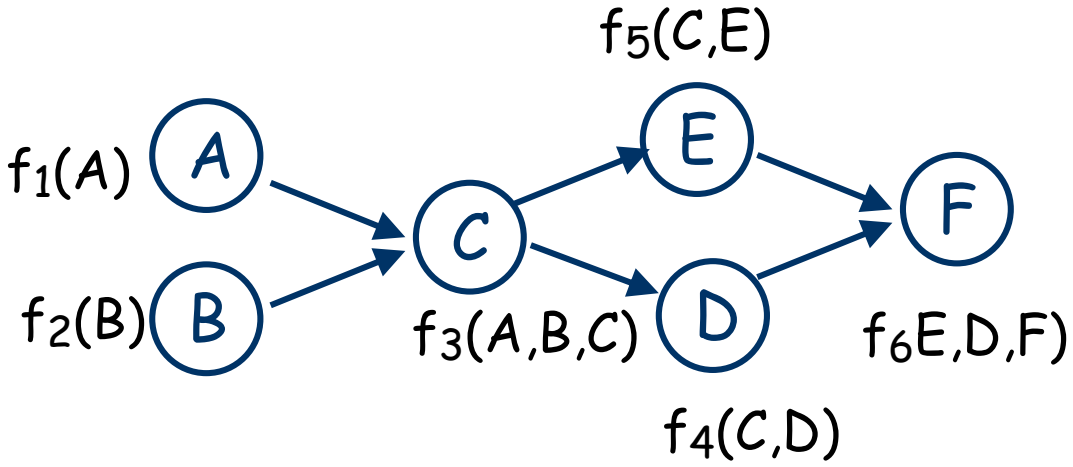
Variable Elimination

- Notice that when we start VE we have a set of factors consisting of the reduced CPTs. The unassigned variables for the vertices and the set of variables each factor depends on forms the hyperedges of a hypergraph H_1.

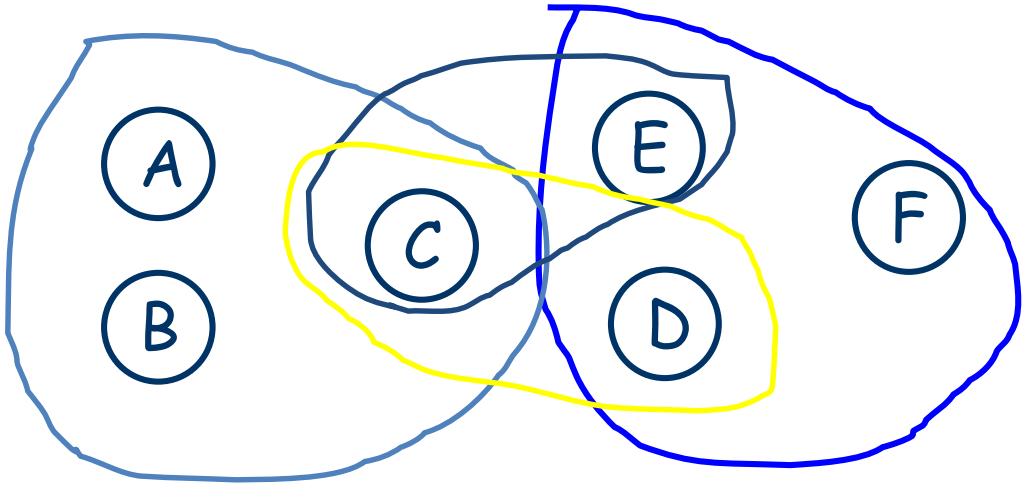
- If the first variable we eliminate is X, then we remove all factors containing X (all hyperedges) and add a new factor that has as variables the union of the variables in the factors containing X (we add a hyperdege that is the union of the removed hyperedges minus X).

VE Factors

Ordering:
C,F,A,B,E,D

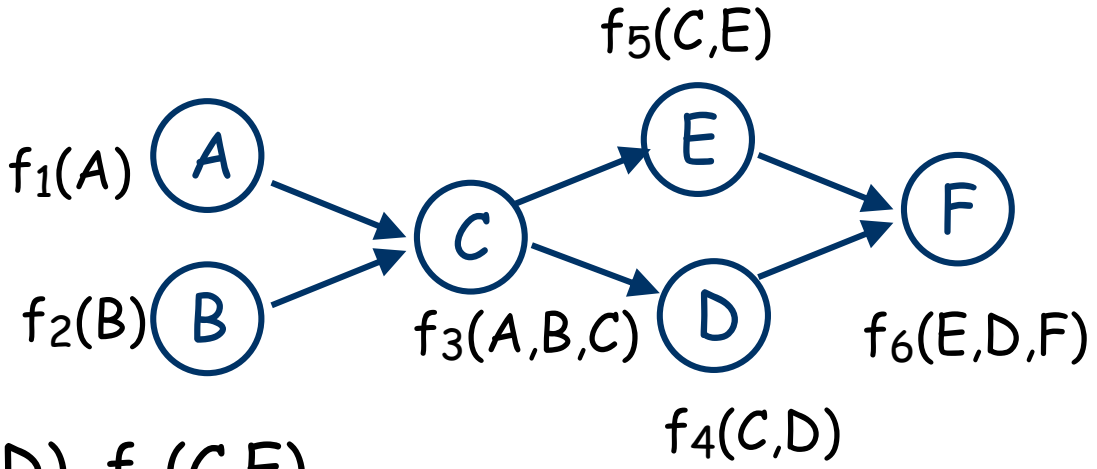


1. C:
2. F:
3. A:
4. B:
5. E:
6. D:



VE: Place Original Factors in first applicable bucket.

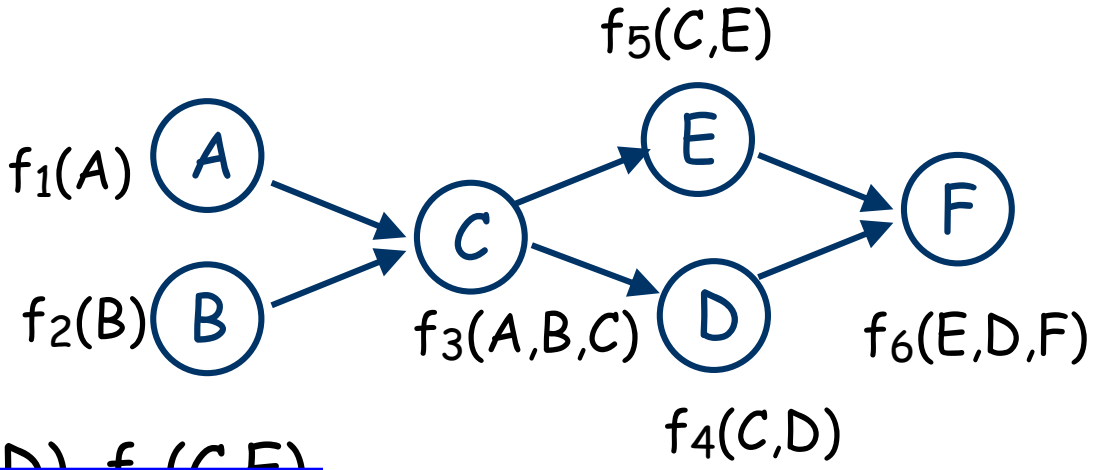
Ordering:
C,F,A,B,E,D



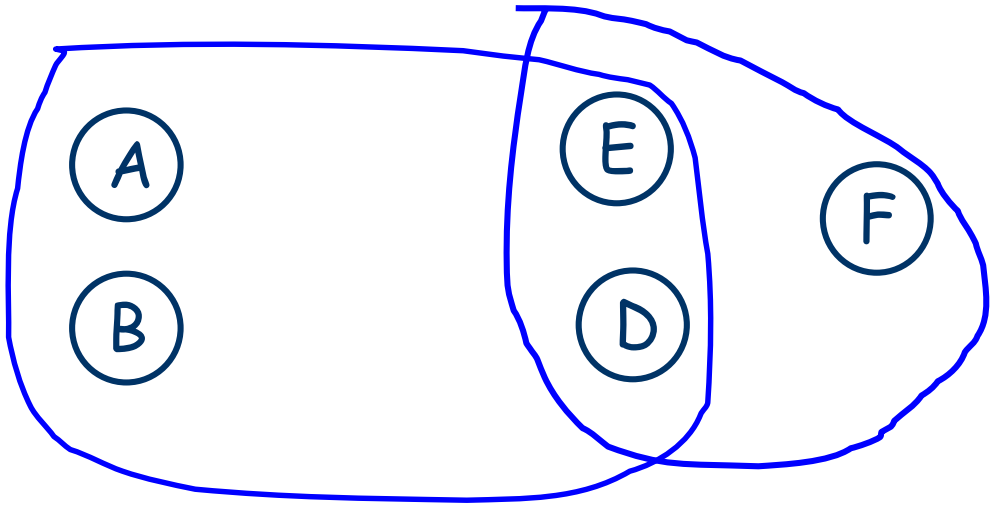
1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

VE: Eliminate C, placing new factor f7 in first applicable bucket.

Ordering:
C,F,A,B,E,D

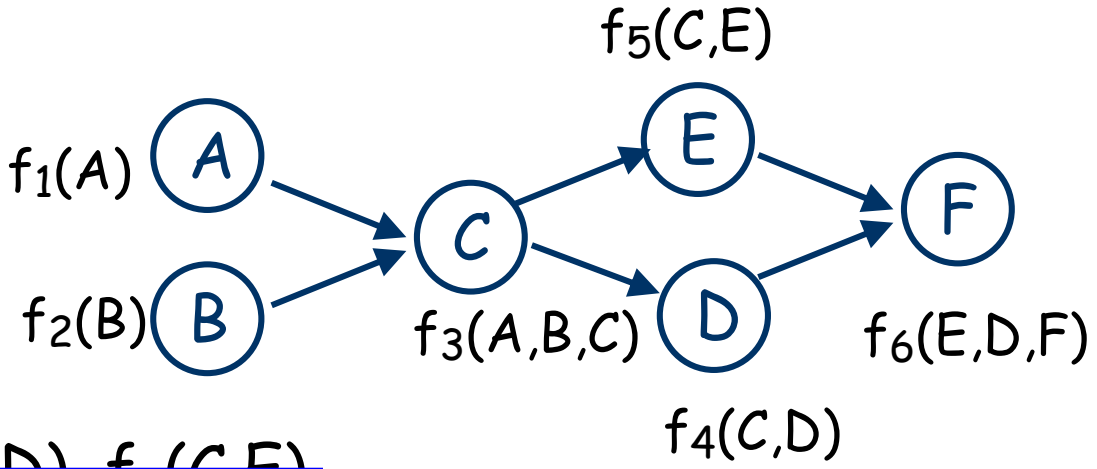


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

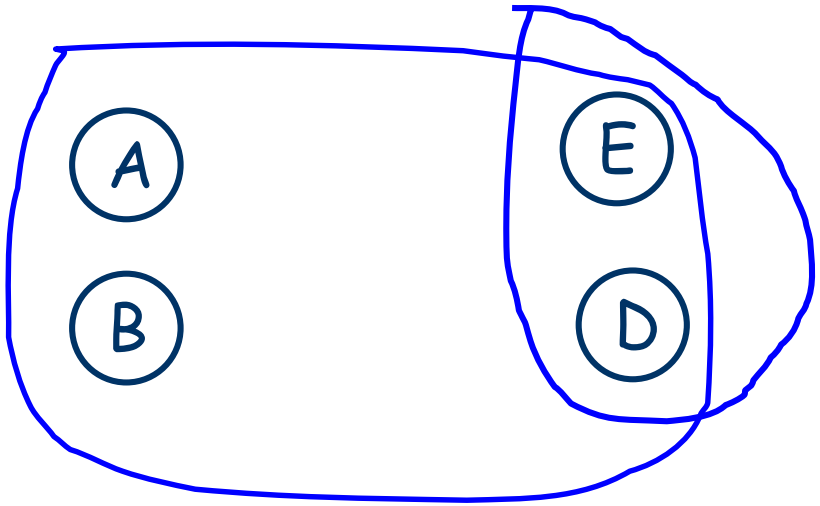


VE: Eliminate F, placing new factor f8 in first applicable bucket.

Ordering:
C,F,A,B,E,D

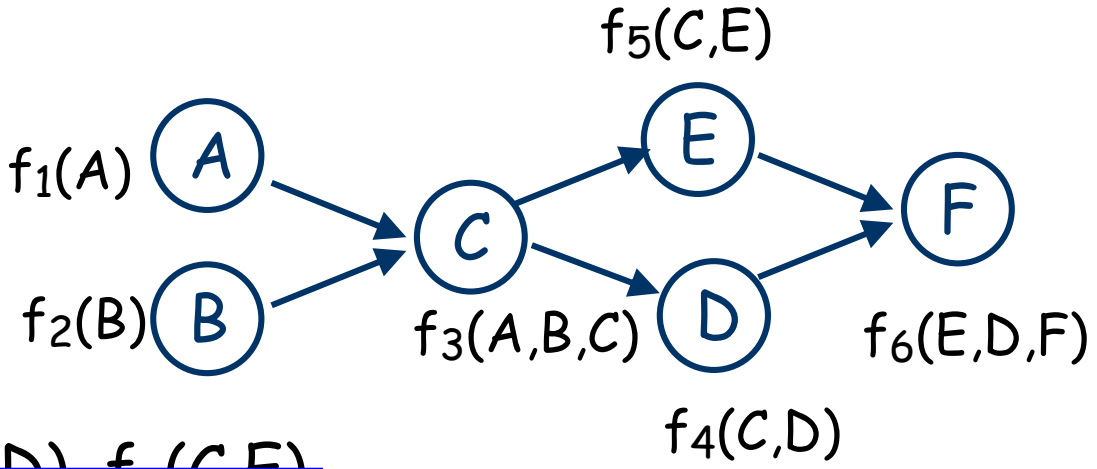


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

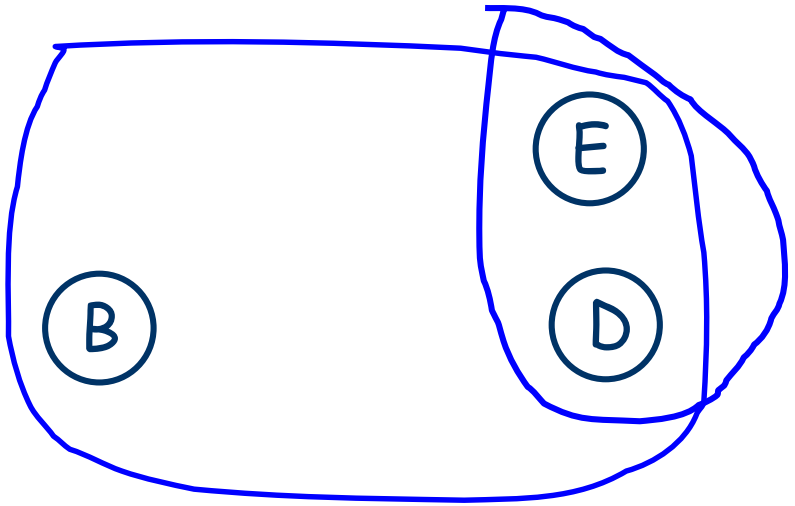


VE: Eliminate A, placing new factor f9 in first applicable bucket.

Ordering:
C,F,A,B,E,D

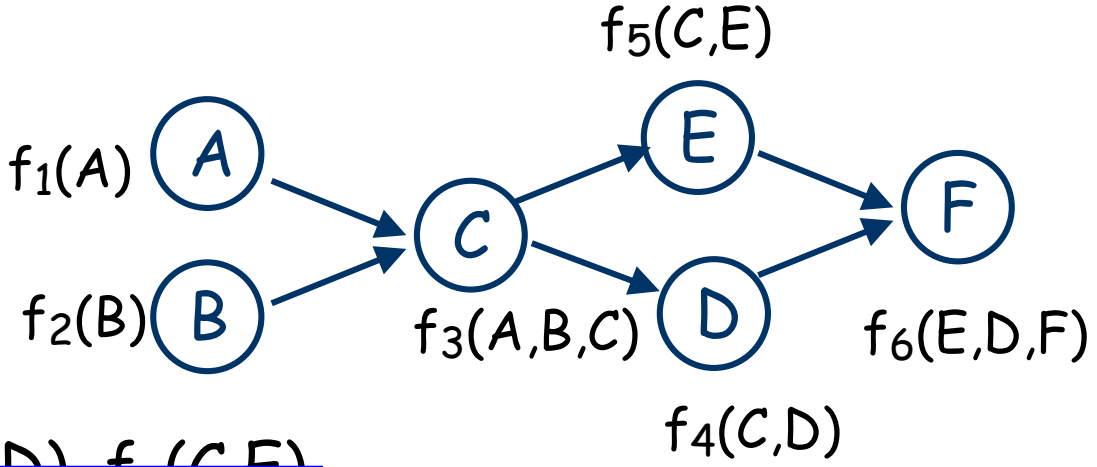


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. B: $f_2(B)$, $f_9(B,D,E)$
5. E: $f_8(E,D)$
6. D:

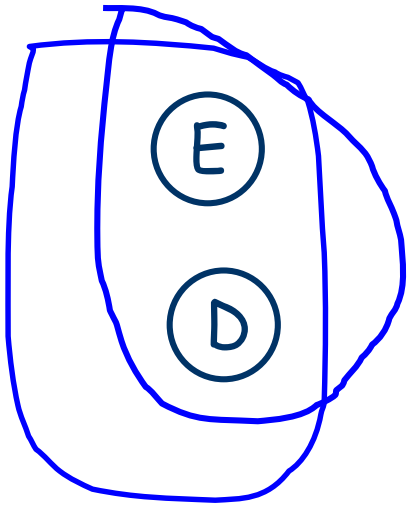


VE: Eliminate B, placing new factor f10 in first applicable bucket.

Ordering:
C,F,A,B,E,D

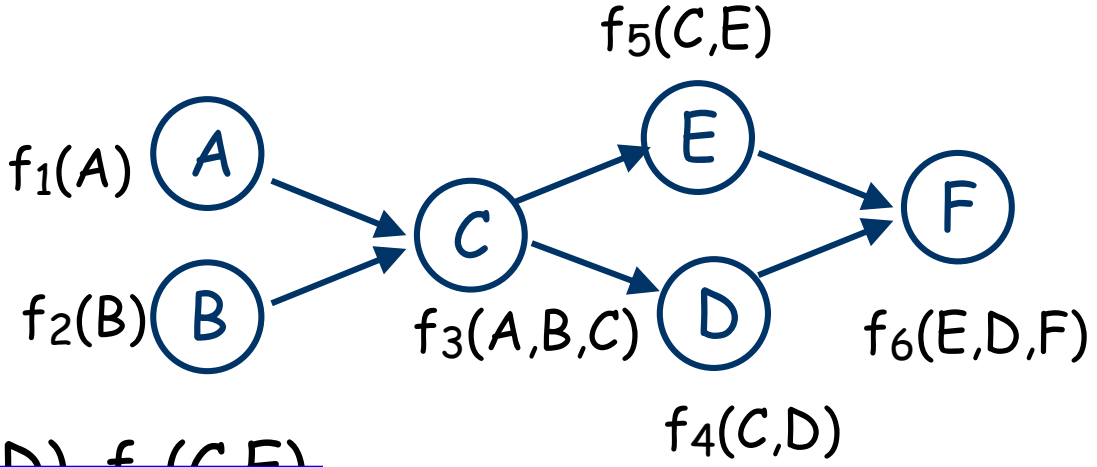


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

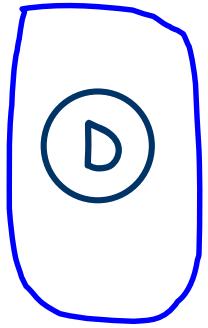


VE: Eliminate E, placing new factor f11 in first applicable bucket.

Ordering:
C,F,A,B,E,D



1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. ~~E: $f_8(E,D)$, $f_{10}(D,E)$~~
6. D: $f_{11}(D)$



Elimination Width

- Given an ordering π of the variables and an initial hypergraph $\mathcal{H}$ eliminating these variables yields a sequence of hypergraphs

  $\mathcal{H} = H_0, H_1, H_2, \ldots, H_n$

- Where H_n contains only one vertex (the query variable).

- The elimination width π is the maximum size (number of variables) of any hyperedge in any of the hypergraphs $H_0, H_1, \ldots, H_n$.
- The elimination width of the previous example was 4 ({A,B,E,D} in H_1 and H_2).

Elimination Width

- If the elimination width of an ordering π is k, then the complexity of VE using that ordering is $2^{O(k)}$
- Elimination width k means that at some stage in the elimination process a factor involving k variables was generated.
- That factor will require $2^{O(k)}$ space to store
  - space complexity of VE is $2^{O(k)}$
- And it will require $2^{O(k)}$ operations to process (either to compute in the first place, or when it is being processed to eliminate one of its variables).
  - Time complexity of VE is $2^{O(k)}$
- NOTE, that k is the elimination width of this particular ordering.

Tree Width

- Given a hypergraph $\mathcal{H}$ with vertices $\{X_1, X_2, \ldots, X_n\}$ the **tree width** of $\mathcal{H}$ is the MINIMUM elimination width of **any of the *n!*** different orderings of the X_i minus *1*.

- Thus VE has best case complexity of $2^{O(\omega)}$ where ω is the TREE WIDTH of the initial Bayes Net.

- In the worst case the tree width can be equal to the number of variables.

Tree Width

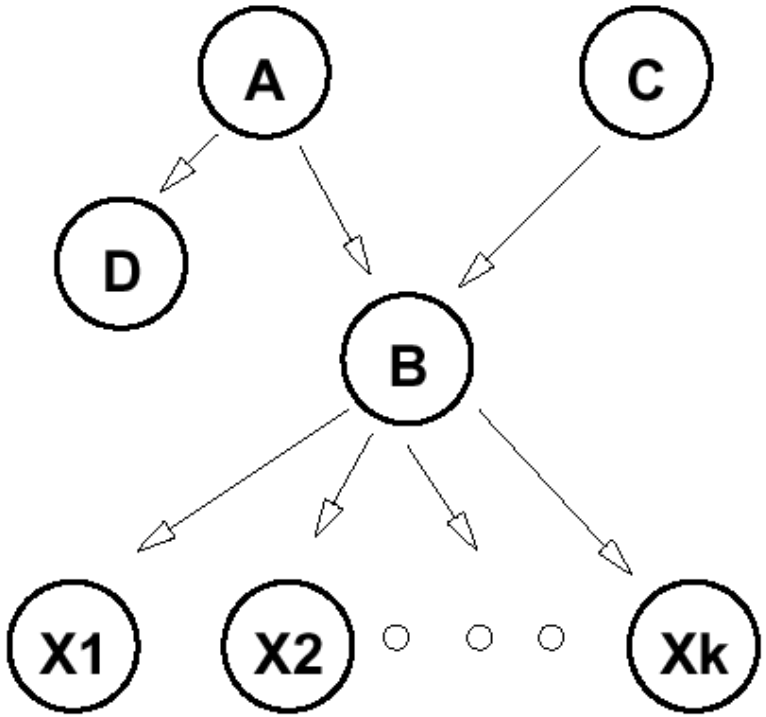
- Exponential in the tree width is the best that VE can do.
  - Finding an ordering that has elimination width equal to tree width is NP-Hard.
    - so in practice there is no point in trying to speed up VE by finding the best possible elimination ordering.
  - Heuristics are used to find orderings with good (low) elimination widths.
  - In practice, this can be very successful. Elimination widths can often be relatively small, 8-10 even when the network has 1000s of variables.
    - Thus VE can be much!! more efficient than simply summing the probability of all possible events (which is exponential in the number of variables).
    - Sometimes, however, the treewidth is equal to the number of variables.

Finding Good Orderings

- A *polytrees* is a singly connected Bayes Net: in particular there is only one path between any two nodes.
- A node can have multiple parents, but we have no cycles.
- Good orderings are easy to find for polytrees
  - At each stage eliminate ***a singly connected node***.
  - Because we have a polytree we are assured that a singly connected node will exist at each elimination stage.
  - The size of the factors in the tree never increase.

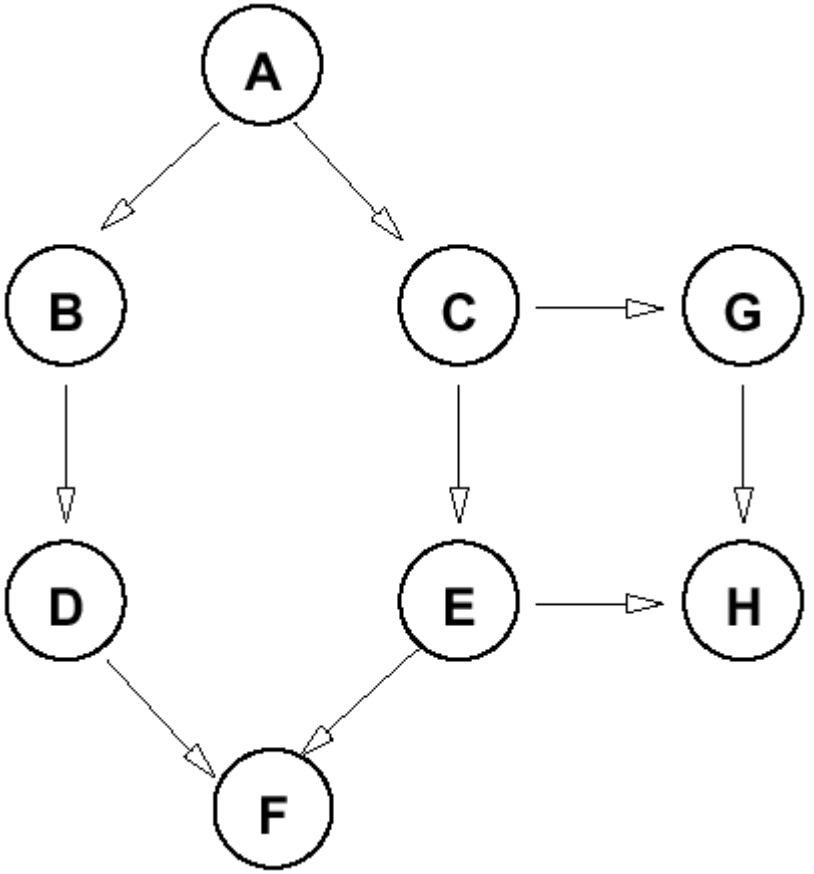
Elimination Ordering: Polytrees

- Treewidth of a polytree is 1!
- Eliminating singly connected nodes allows VE to run in time linear in size of network
  - e.g., in this network, eliminate D, A, C, X1,…; or eliminate X1,… Xk, D, A C; or mix up…
  - result: no factor ever larger than original CPTs
  - eliminating B before these gives factors that include all of A,C, X1,… Xk !!!



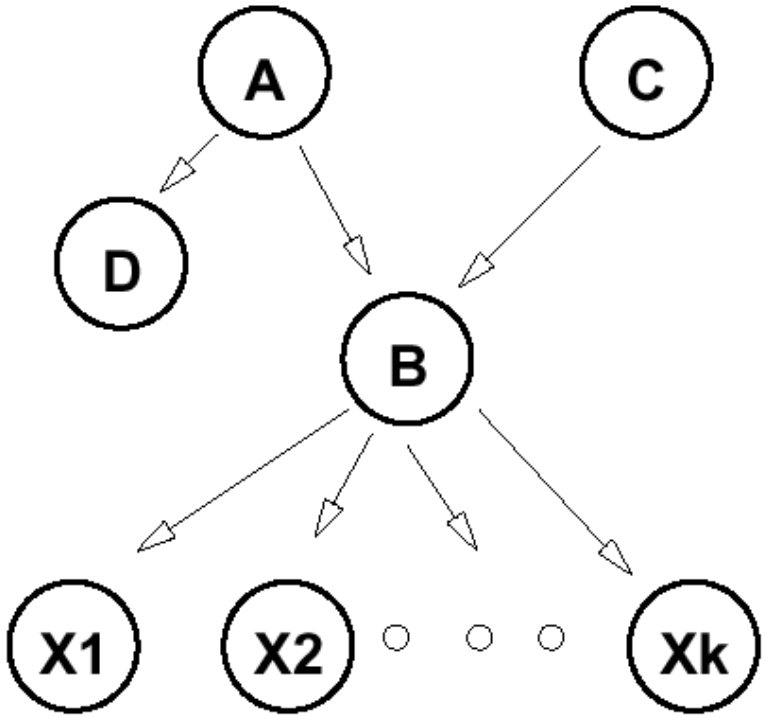
Effect of Different Orderings

- Suppose query variable is D. Consider different orderings for this network (not a polytree!)
  - A,F,H,G,B,C,E:
    - good
  - E,C,A,B,G,H,F:
    - bad



Min Fill Heuristic

- A fairly effective heuristic is always eliminate next the variable that creates the smallest size factor.
- This is called the min-fill heuristic.
- B creates a factor of size k+2
- A creates a factor of size 2
- D creates a factor of size 1



- The heuristic always solves polytrees in linear time.

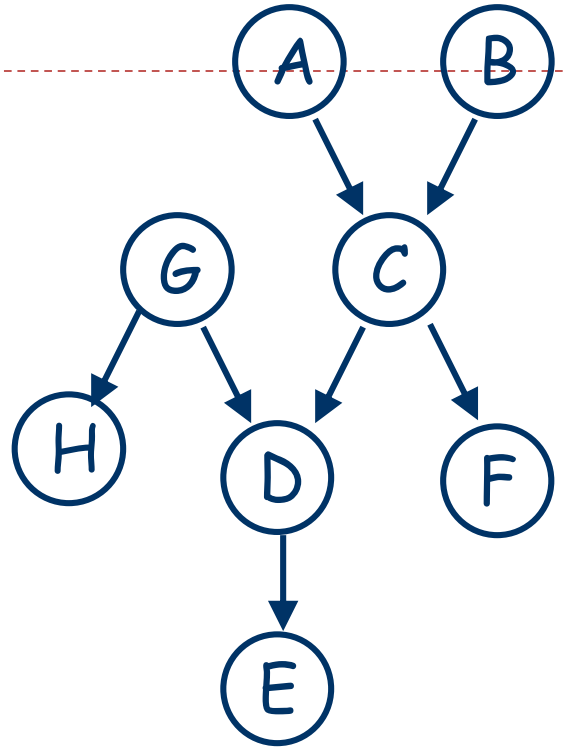
Relevance

- Certain variables have no impact on the query. In network ABC, computing Pr(A) with no evidence requires elimination of B and C.
  - But when you sum out these vars, you compute a trivial factor (whose value are all ones); for example:
  - eliminating C: $f_4(B) = \sum_C f_3(B,C) = \sum_C Pr(C \mid B)$
  - 1 for any value of B (e.g., $Pr(c \mid b) + Pr(\sim c \mid b) = 1$)
- No need to think about B or C for this query

Relevance

- Can restrict attention to *relevant* variables. Given query q, evidence **E**:
  - q itself is relevant
  - if any node **Z** is relevant, its parents are relevant
  - if e∈**E** is a descendent of a relevant node, then E is relevant
- We can restrict our attention to the *subnetwork comprising only relevant variables* when evaluating a query Q

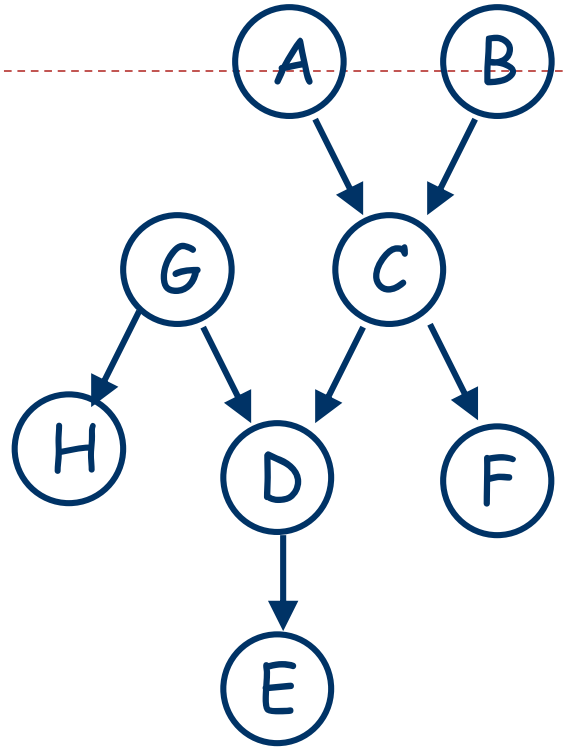
Relevance: Examples



- Query: P(F)
  - relevant: F, C, B, A
- Query: P(F | E)
  - relevant: F, C, B, A
  - **also: E, hence D, G**
  - intuitively, we need to compute P(C|E) to compute P(F | E)
- Query: P(F | H)
  - relevant F,C,A,B.

Pr(A)Pr(B)Pr(C | A,B)Pr(F | C) Pr(G)Pr(h | G)Pr(D | G,C)Pr(E | D)
= … Pr(G)Pr(h | G)Pr(D | G,C) $\sum_E$Pr(E | D) = a table of 1's
= … Pr(G)Pr(h | G) $\sum_D$ Pr(D | G,C) = a table of 1's
= [Pr(A)Pr(B)Pr(C | A,B)Pr(F | C)] [Pr(G)Pr(h | G)]
[Pr(G)Pr(h | G)] ≠ 1 but irrelevant
once we normalize, multiplies each value of F equally

Relevance: Examples



- Query: P(F | E,C)
  - algorithm says all vars except H are relevant; but really none except C, F (since C cuts of all influence of others)
  - algorithm is overestimating relevant set

Independence in a Bayes Net

- Another piece of information we can obtain from a Bayes net is the "structure" of relationships in the domain.
- The structure of the BN means: every X_i is *conditionally independent of all of its nondescendants given it parents*:

$$\Pr(X_i \mid S \cup \text{Par}(X_i)) = \Pr(X_i \mid \text{Par}(X_i))$$

for any subset $S \subseteq \text{NonDescendents}(X_i)$

More generally…

- Many conditional independencies hold in a given BN.
- These independencies are useful in computation, explanation, etc.
- Some of these independencies can be detected using a graphical condition called **D-Separation.**

Approximate Inference in Bayes Nets

- Often the Bayes net is not solvable by Variable Elimination: under any ordering of the variables we end up with a factor that is too large to compute (or store).
- Since we are trying to compute a probability (which only predicts the likelihood of an event occurring) it is natural to consider approximating answer.

Sampling Techniques

- **Direct Sampling** from the **prior** distribution.
- Every Bayes net specifies the probability of every atomic event:
  - Each atomic event is a particular assignment of values to all of the variables in the Bayes nets.
  - Let $V_1, \ldots, V_n$ be the variables in the Bayes net.
  - Let $d_1, \ldots, d_n$ be values for these variables (d_i is the value variable V_i takes).
  - The Bayes net specifies that

$$\Pr(V_1 = d_1, V_2 = d_2, \cdots, V_n = d_n) = \prod_{i=1}^{n} \Pr\left(V_i = d_i \mid ParVals(V_i)\right)$$

  where ParVals(V_i) is the set of assignments $V_k = d_k$ for each $V_k \in Par(V_i)$

Sampling Techniques

- So we want to sample atomic events in such a ways that the probability we select event **e** is equal to Pr(**e**)

1. Select an unselected variable V_i such that all parents of V_i in the Bayes Net have already been selected.
2. Let $[P_1, P_2, \ldots, P_k]$ be the parents of V_i in the Bayes net. Let $[b_1, \ldots, b_k]$ be the values that have already been selected for these parents ($P_i=b_i$).
3. Set V_i to the value $d \in Dom[V_i]$ with probability

$$Pr(V_i = d \mid P_1=b_1, P_2=b_2, \ldots, P_k=b_k)$$

Sampling Techniques

- Note that the probabilities

  $\Pr(V_i = d \mid P_1=b_1, P_2=b_2, \ldots, P_k=b_k)$

  are specified in V_i's CPT in the Bayes net.

- Each variable is given a value by a separate random selection so the probability one obtains a particular atomic event **e** (a setting of all of the variables) via this algorithm is exactly **Pr(e)** as specified by the Bayes Net.

$$\Pr(\mathbf{e} = [V_1 = d_1, V_2 = d_2, \cdots, V_n = d_n]) = \prod_{i=1}^{n} \Pr(V_i = d_i \mid ParVals(V_i))$$

Sampling Techniques

- Say we want to evaluate $Pr(V_1 = d_3)$
- We select **N** random samples of atomic events via this method
- Then we compute the proportion of these N events in which $V_1 = d_3$
- This proportion

  (Number of Events where $V_1 = d_3$)/**N**

  is an estimate of $Pr(V_1 = d_3)$.
- The estimate gets better as **N** gets larger, and by the law of large numbers as **N** approaches infinity the estimate converges (becomes closer and closer) to the exact $Pr(V_1 = d_3)$

Sampling Techniques

- If we want to compute a conditional probability like $Pr(V_1 = d_3 \mid V_4 = d_1)$, then we can
  - Discard all atomic events in which $V_4 \neq d_1$
  - This gives a new smaller set of **N'** sampled atomic events.
  - From those **N'** we compute the proportion in which $V_1 = d_3$
  - This proportion

    (Number of Events where $V_1 = d_3$ from the remaining samples)**/N'**

    is an estimate of $Pr(V_1 = d_3 \mid V_4 = d_1)$
  - This is called **Rejection Sampling**

Sampling Techniques

- **Problem**, almost all samples might be rejected if $V_4 = d_1$ has very low probability.
- The accuracy of the estimate depends on the size of **N'** (the samples that remain after rejection).
- So if very few are left our estimate is not good.
- E.g., if $Pr(V_4 = d_1) = 0.0000001$, then if we generate $1/ 0.0000001 = 10{,}000{,}000$ samples we expect to reject 9,999,999 of them. In that case our estimate of $Pr(V_1 = d_3 \mid V_4 = d_1)$ will be 1 or 0! (Either our sole remaining sample has $V_1 = d_3$ or it doesn't).
- In most cases we want to compute **posterior** probabilities, i.e., probabilities conditioned on the **evidence**. So this is a major problem.

Sampling Techniques

- **Likelihood Weighting** tries to address this issue.
- Force all samples to be compatible with the conditioning event.
- Don't select a value for a variable whose value is specified in the evidence that we are conditioning on.
- Weigh each sample by its probability—some samples count more than others in computing the estimate.

Sampling Techniques

1. Set **w = 1**, let the evidence be a set of variables whose values are already given.
2. **While there are unselected variables**
   1. Select an unselected variable V_i such that all parents of V_i in the Bayes Net have already been selected.
   2. Let $[P_1, P_2, \ldots, P_k]$ be the parents of V_i in the Bayes net. Let $[b_1, \ldots, b_k]$ be the values that have already been selected for these parents ($P_i=b_i$).
   3. **If** V_i's value is specified in the **evidence** and d is the value specified then
      $$w = w * Pr(V_i = d \mid P_1=b_1, P_2=b_2, \ldots, P_k=b_k)$$
   4. **Else** set V_i to the value $d \in Dom[V_i]$ with probability
      $$Pr(V_i = d \mid P_1=b_1, P_2=b_2, \ldots, P_k=b_k)$$

Sampling Techniques

- If we want to compute a conditional probability like $Pr(V_1 = d_3 \mid V_4 = d_1)$, then we can
  - Generate a collection **N** of likelihood weighted samples using the evidence $V_4 = d_1$
  - Each sample (atomic event) **e** has a weight **w.**
  - We compute the sum of the weights of the samples in **N** in $V_1 = d_3$ and divide this by the sum of the weights of all samples in **N.**
  - This number

    (Sum of weights of samples in **N** where $V_1 = d_3$)/(sum of weights of samples in **N**)

    is an estimate of $Pr(V_1 = d_3 \mid V_4 = d_1)$

Sampling Techniques

- **Problem**, many samples might have very low weight. Some might even have zero weight.
  - Zero weight occurs when we have selected the parents of an evidence variable in such a way that
    $Pr(V_i = d \mid P_1=b_1, P_2=b_2, \ldots, P_k=b_k)$
    is zero (this is multiplied into the sample weight).
- The accuracy of the estimate increases as the total weight of the samples increases, so if each sample has very low weight, we may need a very large number of weights.

Sampling Techniques

- Markov Chain Monte Carlo (MCMC) methods some of these problems.
- The book gives a description of Gibbs Sampling (a form of MCMC.