Lecture 21

Bayesian Networks: Structure and Variable Elimination

Lecture Overview

- ➡ Recap
- Final Considerations on Network Structure
- Variable Elimination
- Factors
- Algorithm (time permitting)

Belief (or Bayesian) networks

Def. A Belief network consists of

- a directed, acyclic graph (DAG) where each node is associated with a random variable X_i
- A domain for each variable X_i
- a set of conditional probability distributions for each node X_i given its parents $Pa(X_i)$ in the graph

$$P(X_i \mid Pa(X_i))$$

- The parents $Pa(X_i)$ of a variable X_i are those X_i directly depends on
- A Bayesian network is a compact representation of the JDP for a set of variables ($X_1, \ldots, X_n$)

$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i))$$

How to build a Bayesian network

1. Define a total order over the random variables: $(X_1, \dots, X_n)$

2. Apply the chain rule

Predecessors of X_i in the total order defined

$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \dots, X_{i-1})$$ over the variables

3. For each X_i, , select the smallest set of predecessors $Pa(X_i)$ such that

X_i is conditionally independent from all its

$$P(X_i \mid X_1, \dots, X_{i-1}) = P(X_i \mid Pa(X_i))$$ other predecessors given $Pa(X_i)$

4. Then we can rewrite

$P(X_1, \ldots, X_n) = \prod^{n}_{i=1} P(X_i \mid Pa(X_i))$

- This is a compact representation of the initial JPD • factorization of the JPD based on existing conditional independencies among the variables

How to build a Bayesian network (cont'd)

5. Construct the Bayesian Net (BN)

- Nodes are the random variables
- Draw a directed arc from each variable in $Pa(X_i)$ to X_i
- Define a conditional probability table (CPT) for each variable X_i:

- $P(X_i \mid Pa(X_i))$

Example for BN construction: Fire Diagnosis

You want to diagnose whether there is a fire in a building

- You can receive reports (possibly noisy) about whether everyone is leaving the building
- If everyone is leaving, this may have been caused by a fire alarm
- If there is a fire alarm, it may have been caused by a fire or by tampering
- If there is a fire, there may be smoke

Start by choosing the random variables for this domain, here all are Boolean:

- Tampering (T) is true when the alarm has been tampered with
- Fire (F) is true when there is a fire
- Alarm (A) is true when there is an alarm

- Smoke (S) is true when there is smoke
- Leaving (L) is true if there are lots of people leaving the building
- Report (R) is true if the sensor reports that lots of people are leaving the building

Next apply the procedure described earlier

Example for BN construction: Fire Diagnosis

1. Define a total ordering of variables:
   - Let's chose an order that follows the causal sequence of events
   - Fire (F), Tampering (T), Alarm, (A), Smoke (S) Leaving (L) Report (R)
2. Apply the chain rule

P(F,T,A,S,L,R) =

P(F)P (T | F) P (A | F,T) P (S | F,T,A) P (L | F,T,A,S) P (R | F,T,A,S,L)

We will do steps 3, 4 and 5 together, for each element $P(X_i \mid X_1, \ldots, X_{i-1})$ of the factorization

3. For each variable (X_i), choose the parents Parents(X_i)by evaluating conditional independencies, so that

$$P(X_i \mid X_1, \ldots, X_{i-1}) = P\,(X_i \mid \text{Parents}\,(X_i))$$

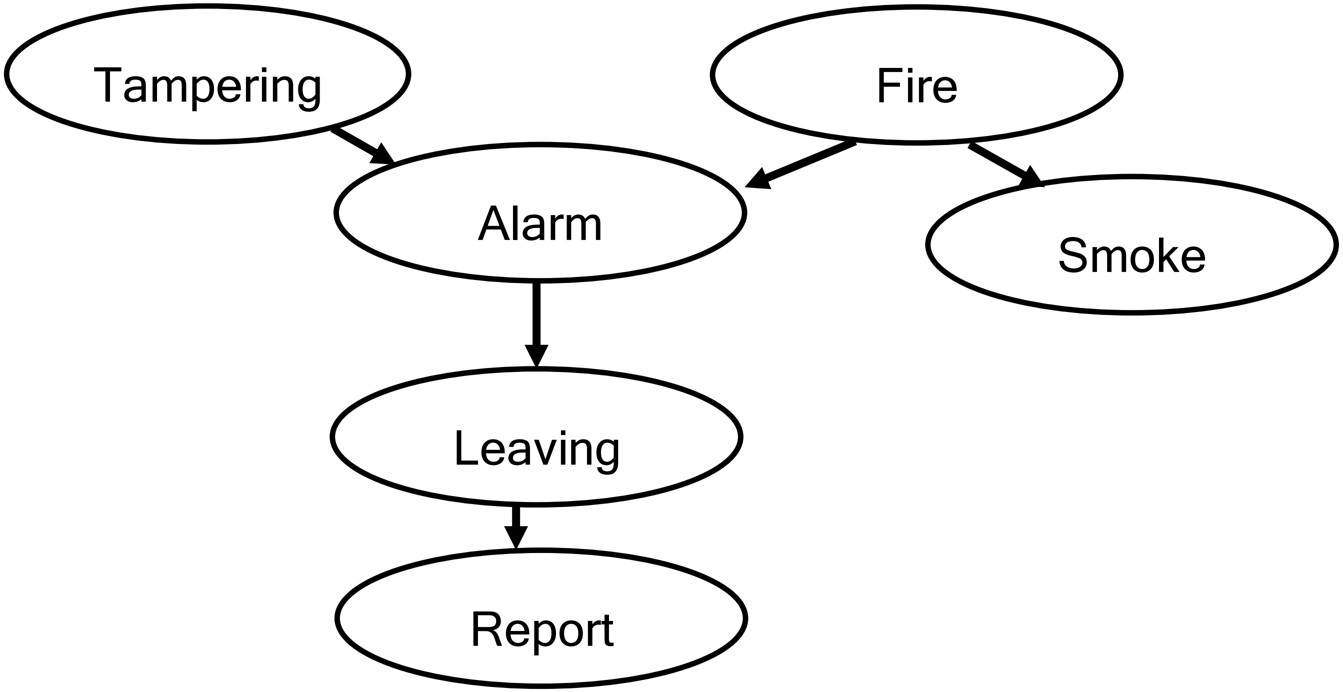
4. Rewrite

$$P(X_1, \ldots, X_n) = \prod_{ni=1} P\,(X_i \mid \text{Parents}\,(X_i))$$

5. Construct the Bayesian network

Fire Diagnosis Example

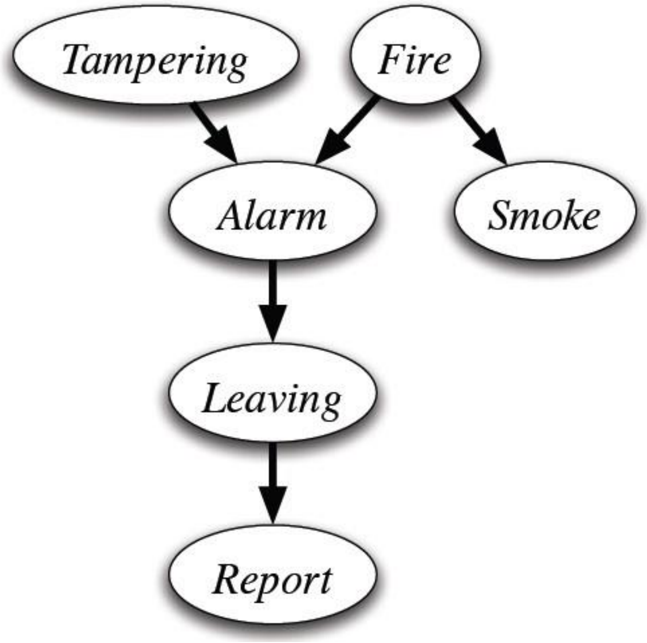
$P(F)P\,(T\,)\,P\,(A \mid F,T)\,P\,(S \mid F)\,P\,(L \mid A)\,P\,(R \mid L)$



The result is the Bayesian network above, and its corresponding, very compact factorization of the original JPD

P(F,T,A,S,L,R)= P(F)P (T) P (A | F,T) P (S | F) P (L | A) P (R | L)

Defining CPTs



- We are not done yet: must specify the Conditional Probability Table (CPT) for each variable. All variables are Boolean.
- How many probabilities do we need to specify for this Bayesian network?

- For instance, how many probabilities do we need to explicitly specify for Fire?

Only P(Fire): 1 probability -> P(Fire = T)
Because P(Fire = F) = 1 - P(Fire = T)

Fire F	*P(Smoke=t |F)*
t	0.9
f	0.01

P(Tampering=t)
0.02

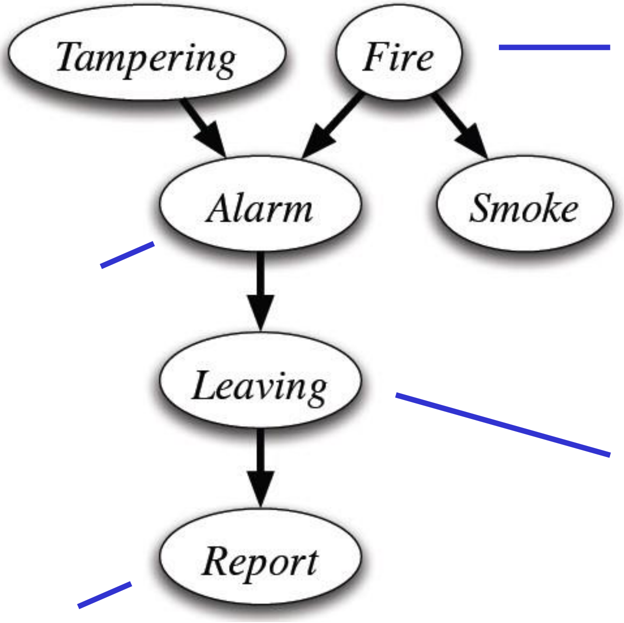
P(Fire=t)
0.01

Tampering T	*Fire F*	*P(Alarm=t|T,F)*
t	t	0.5
t	f	0.85
f	t	0.99
f	f	0.0001

Alarm	*P(Leaving=t|A)*
t	0.88
f	0.001

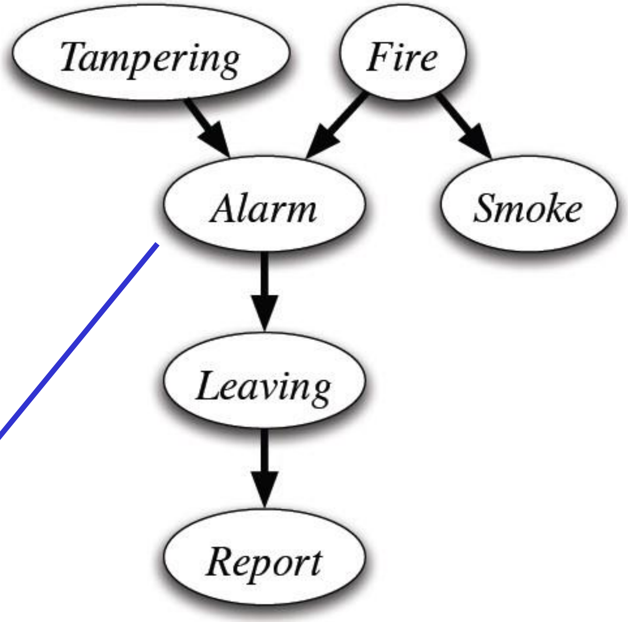
Leaving	*P(Report=t|L)*
t	0.75
f	0.01

Specifying CPTs



- We need to explicitly specify 12 probabilities in total, compared to the 2^6 -1= 63 of the JPD for P(T,F,A,S,L,R)
- Each row in each CPT is a probability distribution.
- The tables above are all missing the column for *P(X=f |Pa(X)).*
- *Values for these columns are derivable as 1 - P(X= t |Pa(X)).*

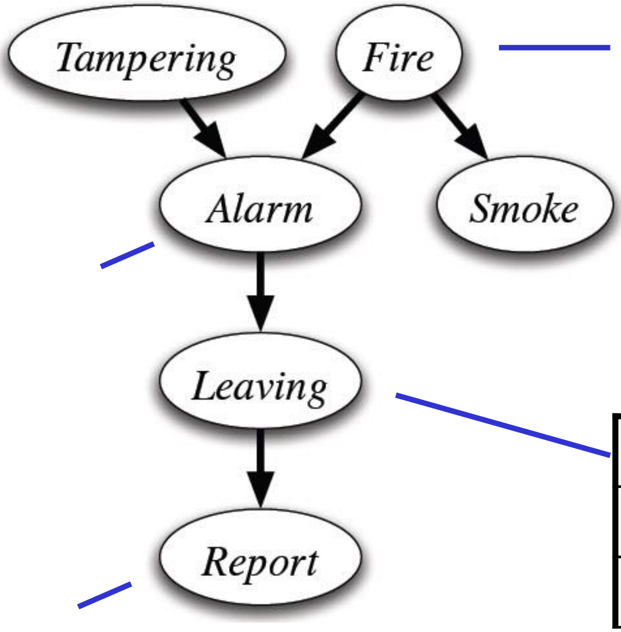
Example for P(Alarm Fire, Tampering)



Tampering T	*Fire F*	*P(Alarm=t|T,F)*	*P(Alarm=f|T,F)*
t	t	0.5	0.5
t	f	0.85	0.15
f	t	0.99	0.01
f	f	0.0001	0.9999

We don’t need to speficy explicitly P(Alarm=f|T,F) since probabilities in each row must sum to 1

Each row of this table is a conditional probability distribution



P(Fire=t)
0.01

Fire F	P(Smoke=t |F)
t	0.9
f	0.01

Alarm	P(Leaving=t|A)
t	0.88
f	0.001

Computing JPD entries

P(Tampering=t)
0.02

Tampering T	Fire F	P(Alarm=t|T,F)

t	t	0.5
t	f	0.85
f	t	0.99
f	f	0.0001

Leaving	*P(Repor*
t	0.7
f	0.0

Once we have the CPTs in the network, we can compute any entry of the JPD

P(Tampering=t, Fire=f, Alarm=t, Smoke=f, Leaving=t, Report=t) =

P(Tampering=t) x P(Fire=f)xP(Alarm=t| Tampering=t, Fire=f)xP(Smoke=f| Fire = f)xP(Leaving=t| Alarm=t) x P(Report=t|Leaving=t) =

= 0.02 x (1-0.01) x 0.85 x (1-0.01) x 0.88 x 0.75 = 0.126

In Summary

- In a Belief network, the JPD of the variables involved is defined as the product of the local conditional distributions

$$P\ (X_1,\ \dots\ ,X_n) = \prod_i P(X_i \mid X_1,\ \dots\ ,X_{i-1}) = \prod{}_i P(X_i \mid \text{Parents}(X_i))$$

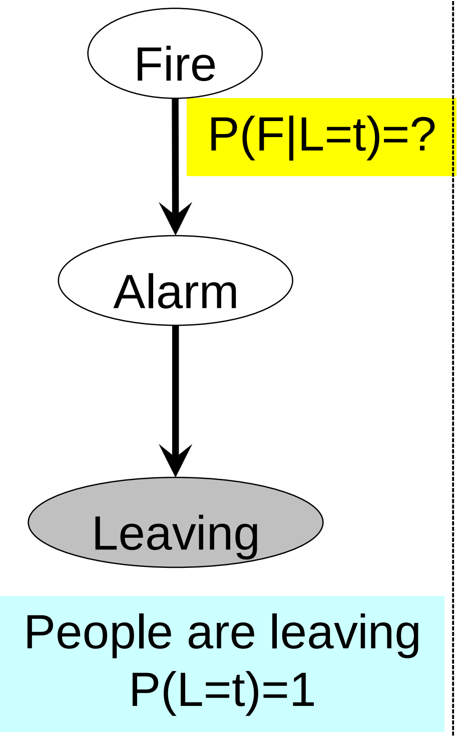
- Any entry of the JPD can be computed given the

  CPTs in the network

Once we know the JPD, we can answer any query about any subset of the variables - (see Inference by Enumeration topic)

Thus, a Belief network allows one to answer any query on any subset of the variables

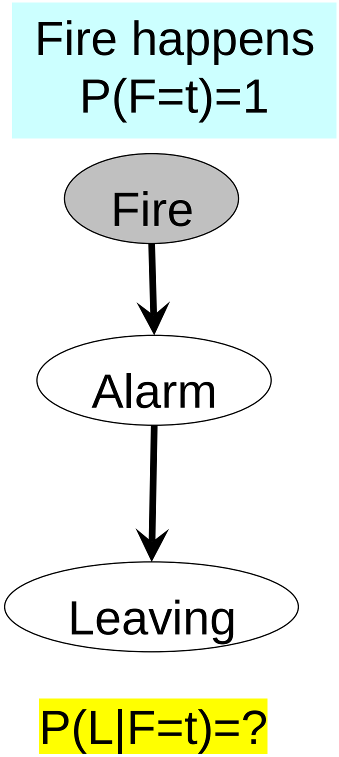
Diagnostic



P(F|L=t)=?

People are leaving
P(L=t)=1

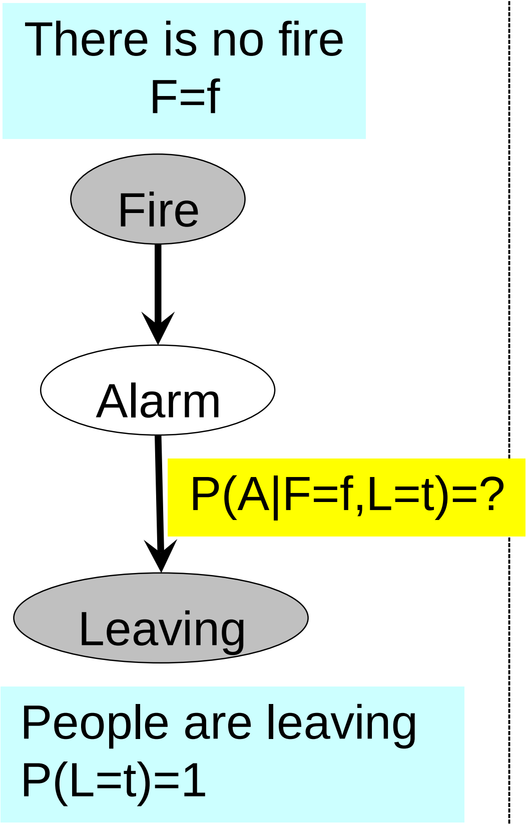
Predictive



Fire happens
P(F=t)=1

P(L|F=t)=?

Mixed

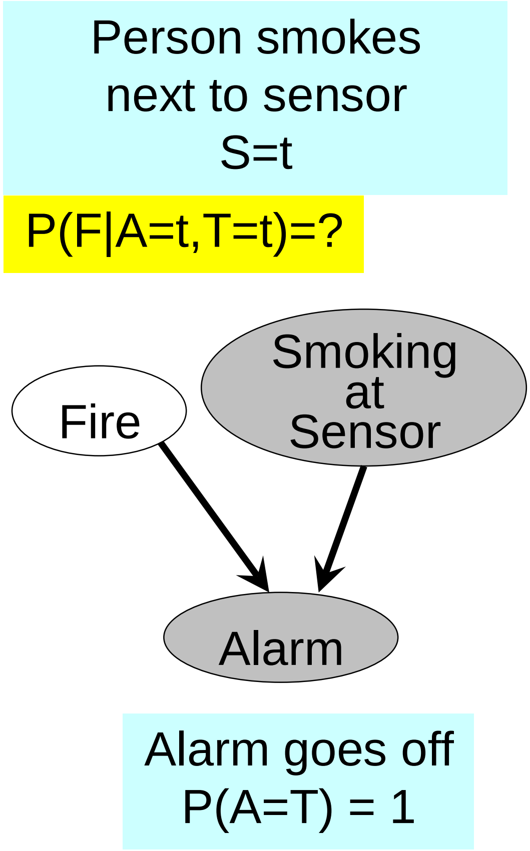


There is no fire
F=f

P(A|F=f,L=t)=?

People are leaving
P(L=t)=1

Intercausal



Person smokes
next to sensor
S=t

P(F|A=t,T=t)=?

Alarm goes off
P(A=T) = 1

There are algorithms that leverage the Bnet structure to perform query answer efficiently

- For instance variable elimination, which we will cover soon
- First, however, we will think a bit more about network structure

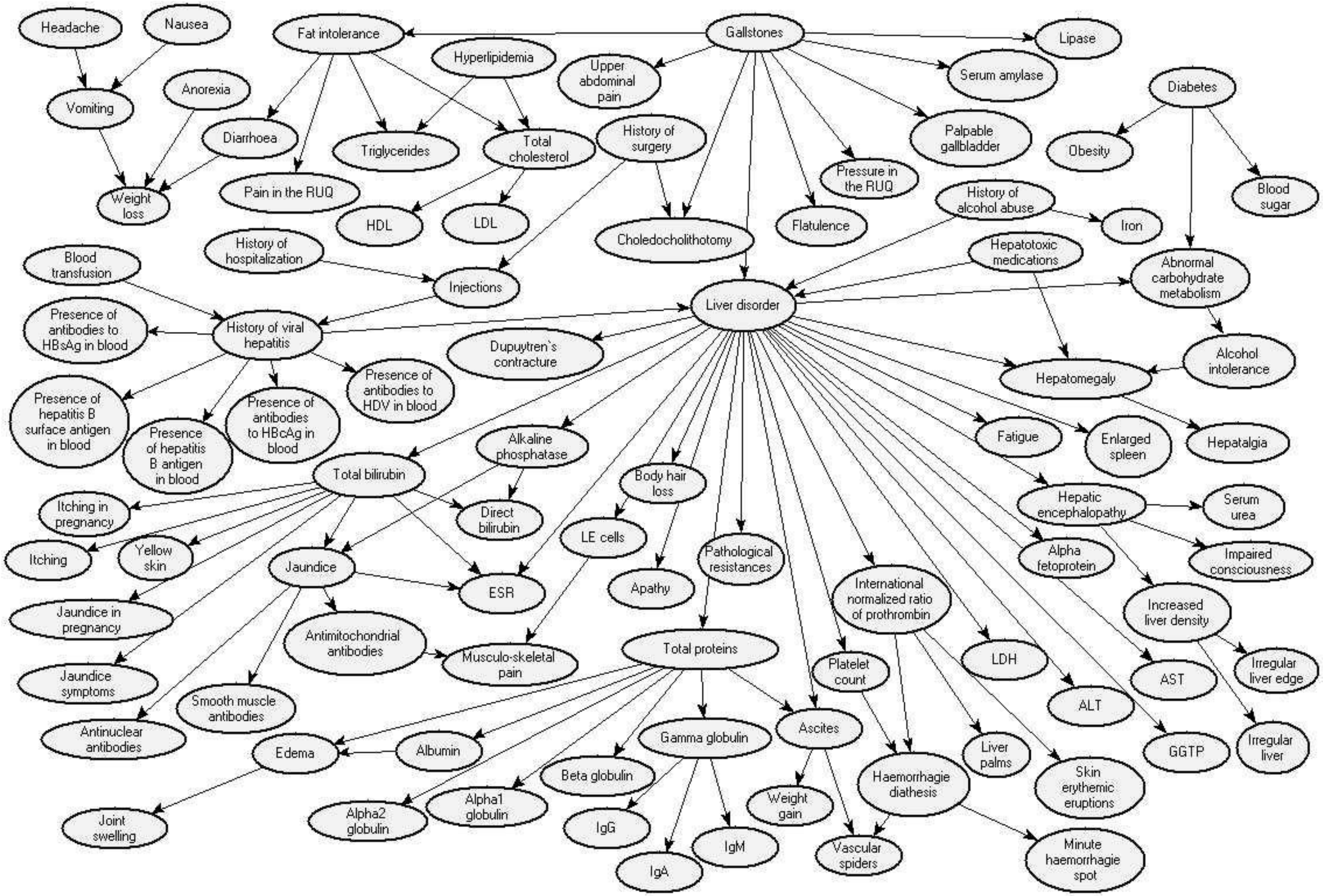
Compactness

- A CPT for a Boolean variable $X_i$with kBoolean parents has 2^krows for the combinations of parent values
- Each row requires one number $p$for X_i= true
  (the number for X_i= falseis just 1-p)
- If each variable has no more than $k$parents, the complete network requires to specify $O(n \cdot 2^k)$ numbers
- For k<< n, this is a substantial improvement,
- the numbers required grow linearly with n, vs. $O(2^n)$for the full joint distribution

- E.g., if we have a Bnets with 30 boolean variables, each with 5 parents
- Need to specify $30*2^5$probability
- But we need 2^{30} for JPD

Realistic BNet: Liver Diagnosis

Source: Onisko et al., 1999



~ 60 nodes, max 4 parents per node

Need ~ 60 x 2^4 = 15 x 2^6 probabilities instead of 2^{60} probabilities for the JPD

Compactness

- What happens if the network is fully connected?
- Or k ≈ n
- Not much saving compared to the numbers needed to specify the full JPD
- Bnets are useful in sparse(or locally structured) domains
- Domains in with each component interacts with (is related to) a small fraction of other components

- What if this is not the case in a domain we need to reason about?

May need to make simplifying assumptions to reduce the dependencies in a domain

"Where do the numbers (CPTs) come from?"

From experts

- Tedious
- Costly
- Not always reliable

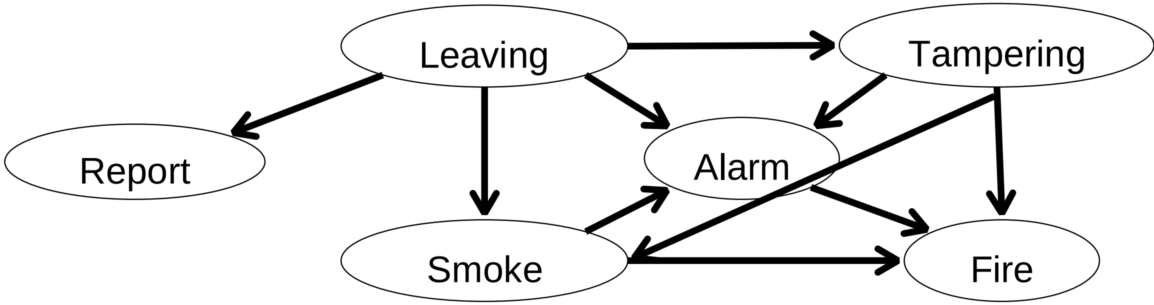
From data => Machine Learning

- There are algorithms to learn both structures and numbers (CPSC 340, CPSC 422)
- Can be hard to get enough data

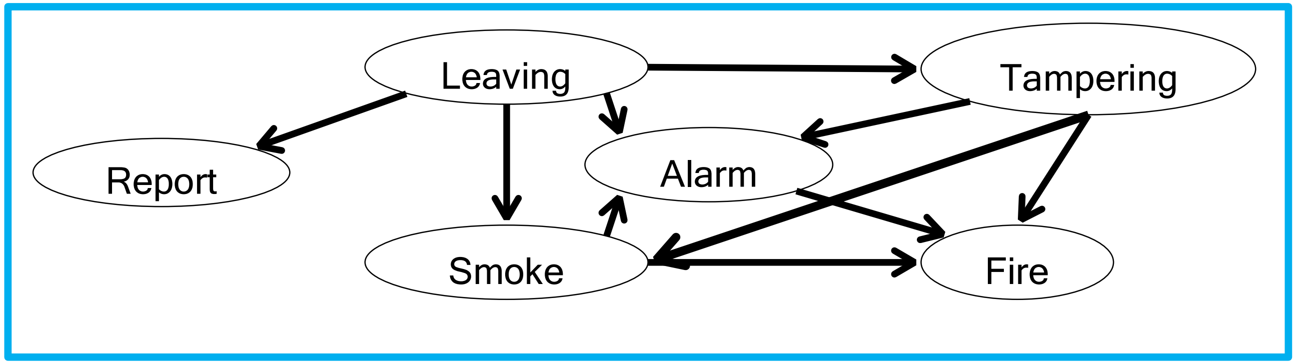
Still, usually better than specifying the full JPD

What if we use a different ordering?

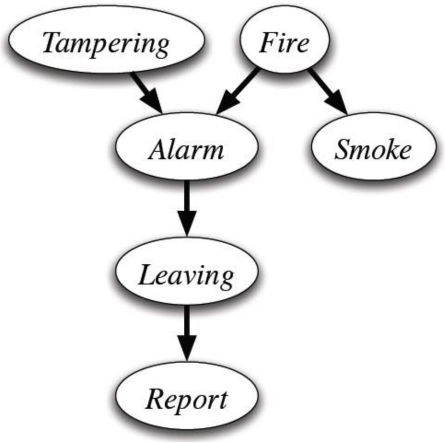
- What happens if we use the following order:
- Leaving; Tampering; Report; Smoke; Alarm; Fire.



- We end up with a completely different network structure! (try it as an exercise)



Which Structure is Better?



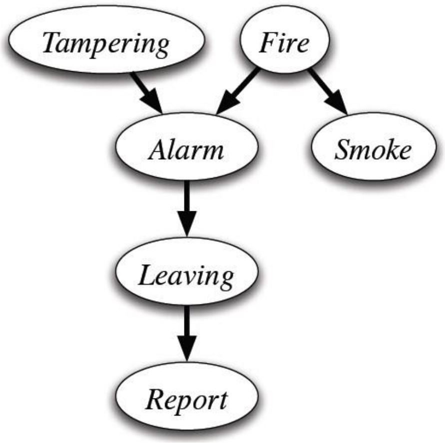
- Non-causal network is less compact: 1+2+2+4+8+8 = 25 numbers needed

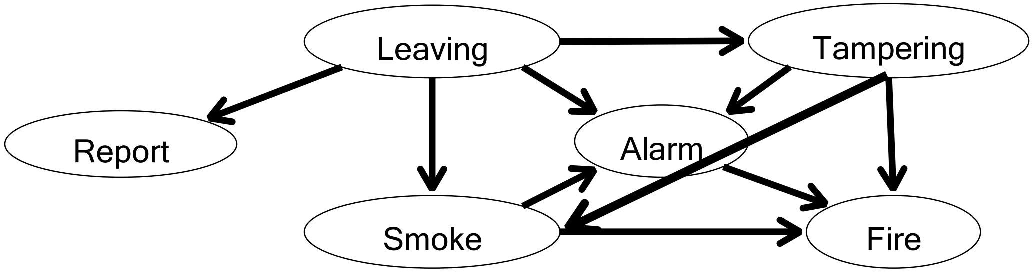
- Deciding on conditional independence is hard in non-causal directions • Causal models and conditional independence seem hardwired for humans!

- Specifying the conditional probabilities may be harder than in causal direction
- For instance, we have lost the direct dependency between alarm and one of its causes, which essentially describes the alarm’s reliability (info often provided by the maker)

Example contd.

- Other than that, our two Bnets for the Alarm problem are equivalent as long as they represent the same probability distribution





Variable ordering: L,T,R,S,A,F

Variable ordering: T,F,A,S,L,R

P(T,F,A,S,L,R) = P (T) P (F) P (A | T,F) P (L | A) P (R|L) =

= P(L)P(T|L)P(R|L)P(S|L,T)P(A|S,L,T) P(F|S,A,T)

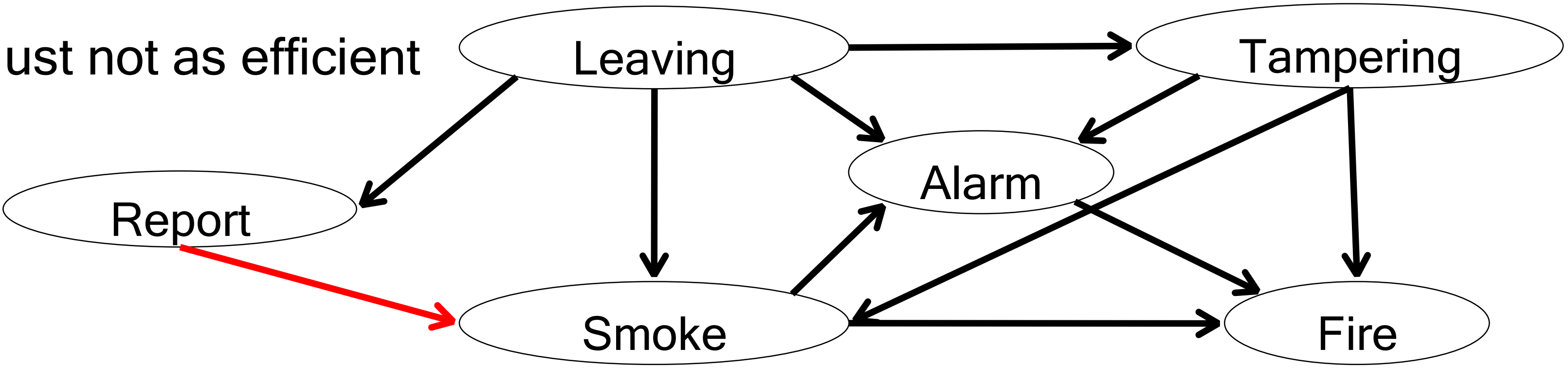
i.e., they are equivalent if the corresponding CPTs are specified so that they satisfy the equation above

Lecture Overview

- Recap
- Final Considerations on Network Structure
- Variable Elimination
- Factors
- Algorithm
- VE example

Are there wrong network structures?

- Given an order of variables, a network with arcs in excess to those required by the direct dependencies implied by that order are still ok
  - Just not as efficient



P (L)P(T|L)P(R|L) P(S|L,R,T) P(A|S,L,T) P(F|S,A,T) =
P (L)P(T|L)P(R|L)P(S|L,T)P(A|S,L,T) P(F|S,A,T)

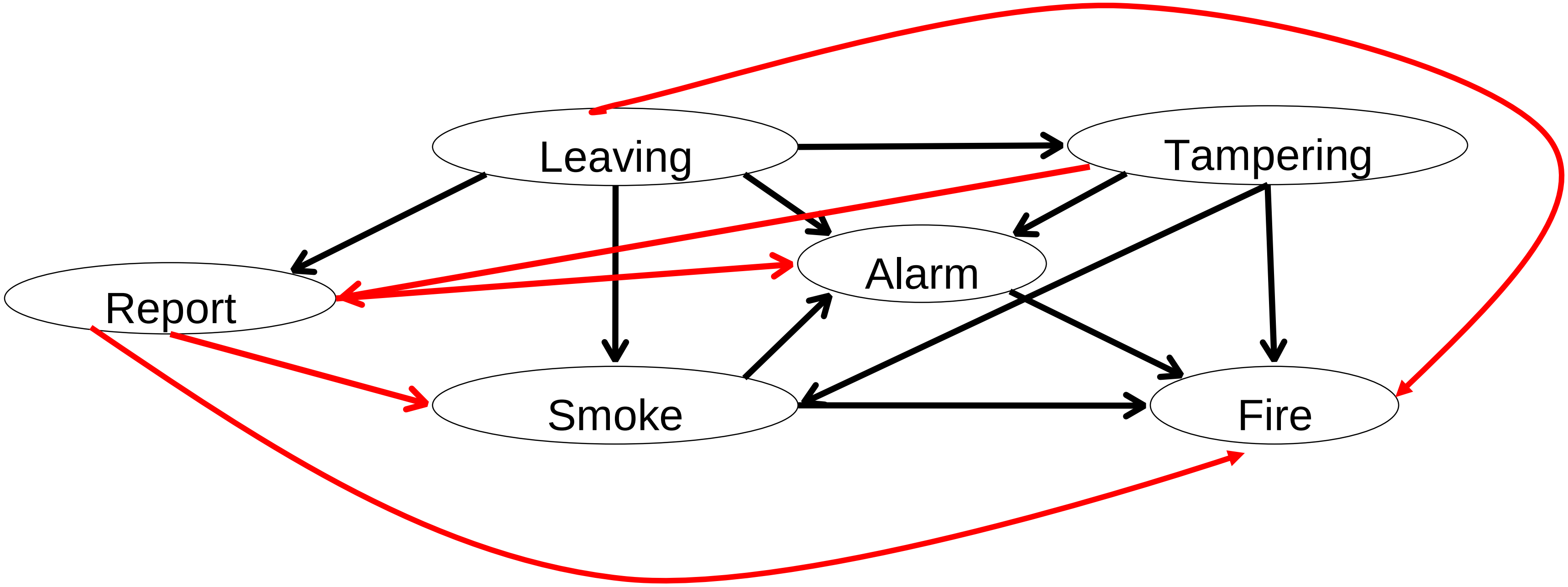
- One extreme: the fully connected network is always correct but rarely the best choice

Are there wrong network structures?

- It corresponds to just applying the chain rule to the JDP, without leveraging conditional independencies to simplify the factorization

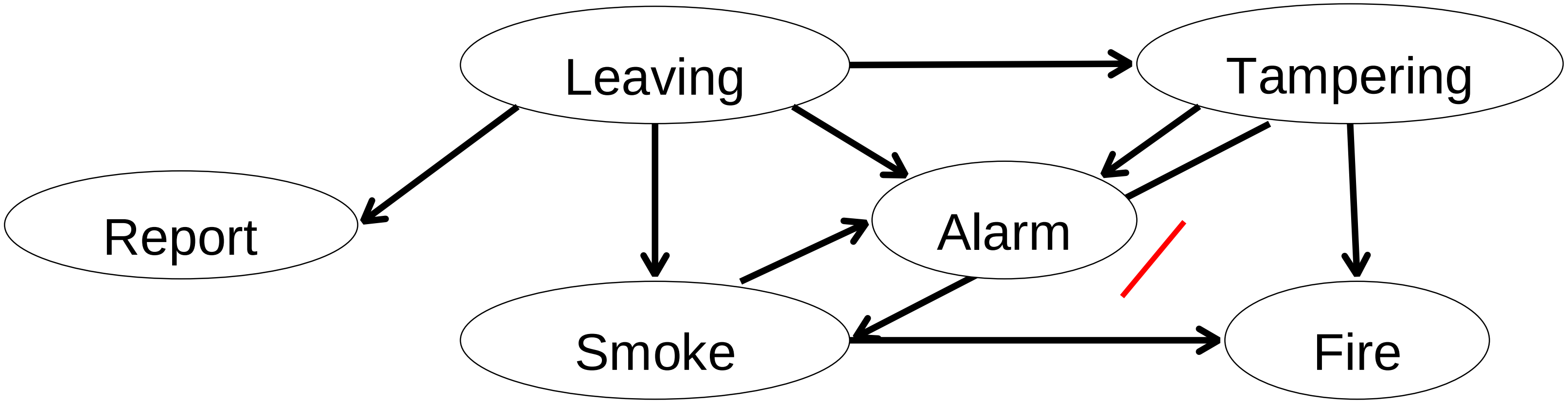
P(L,T,R,S,A,L)= P (L)P(T|L)P(R|L,T)P(S|L,T,R)P(A|S,L,T,R) P(F|S,A,T,L,R)

P(L,T,R,S,A,L)= P (L)P(T|L)P(R|L,T)P(S|L,T,R)P(A|S,L,T,R) P(F|S,A,T,L,R)



Are there wrong network structures?

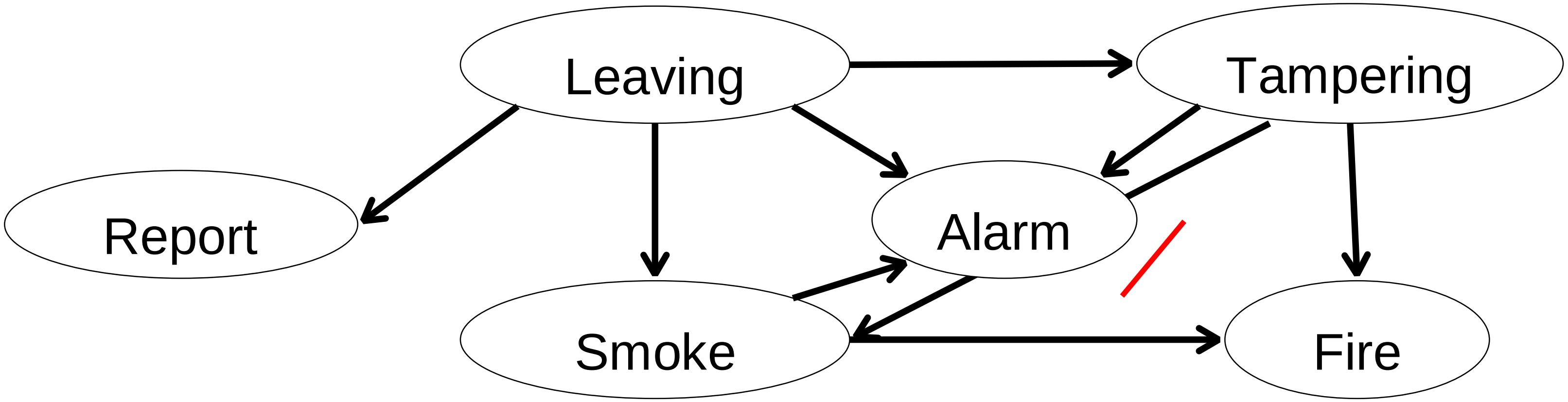
- It corresponds to just applying the chain rule to the JDP, without leveraging conditional independencies to simplify the factorization
- How can a network structure be wrong?
- If it misses directed edges that are required
- E.g. an edge is missing below, making Fire conditionally independent of Alarm given Tampering and Smoke



Are there wrong network structures?

But they are not:
for instance, P(Fire = t| Smoke = f, Tampering = F, Alarm = T) should be
higher than P(Fire = t| Smoke = f, Tampering = f),

- How can a network structure be wrong?
- If it misses directed edges that are required • E.g. an edge is missing below: Fire is not conditionally independent of Alarm | {Tampering, Smoke}

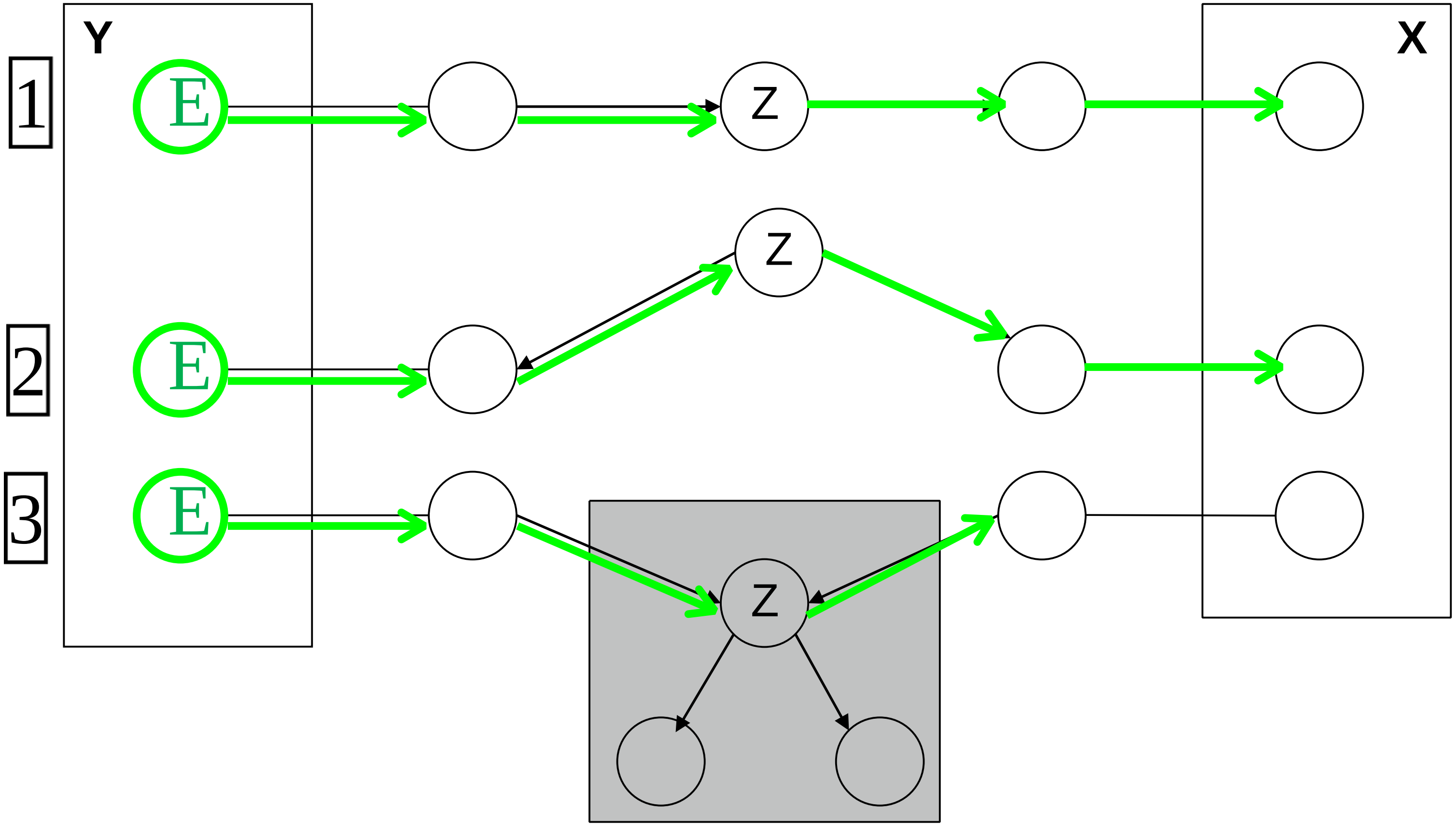


Are there wrong network structures?

But remember what we said a few slides back. Sometimes we may need to make simplifying assumptions - e.g. assume conditional independence when it does not actually hold – in order to reduce complexity

Summary of Dependencies in a Bayesian Network

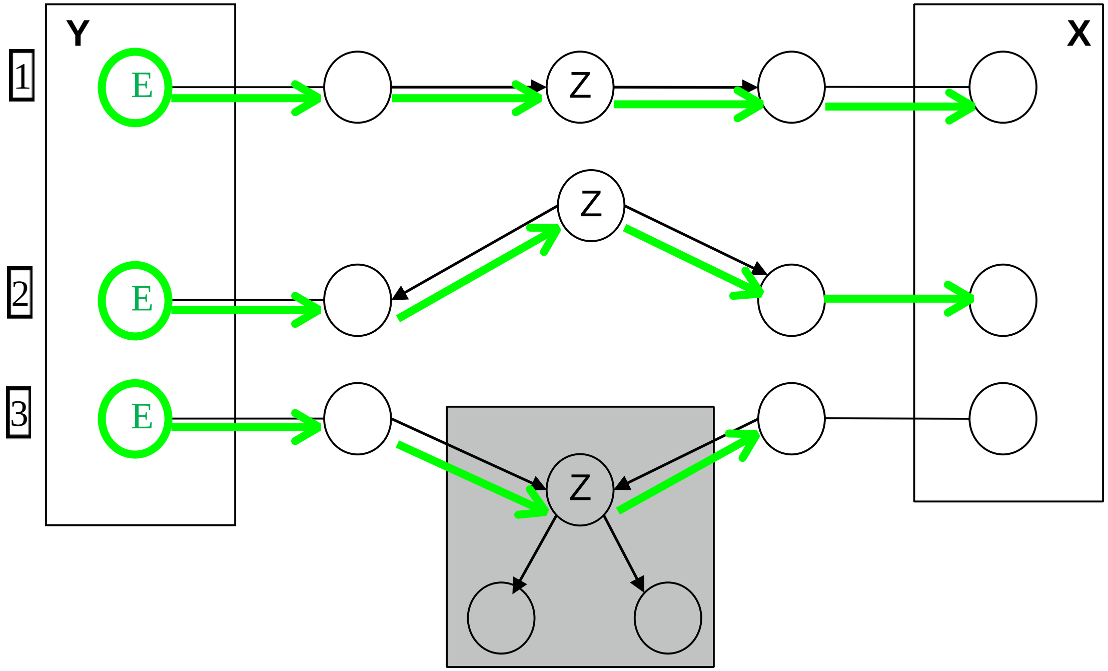
In 1, 2 and 3, X and Y are dependent (grey areas represent existing evidence/observations)

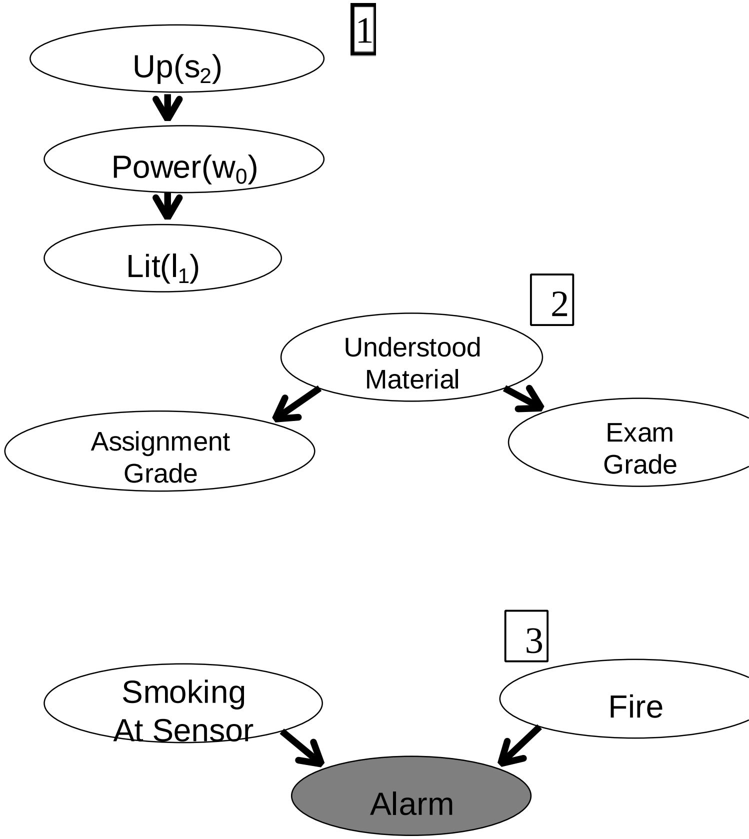


- In 3, X and Y become dependent as soon as there is evidence on Z or on any of its descendants.
- This is because knowledge of one possible cause given evidence of the effect explains away the other cause

Dependencies in a Bayesian Network: summary

In 1, 2 and 3, X and Y are dependent (grey areas represent existing evidence/observations)



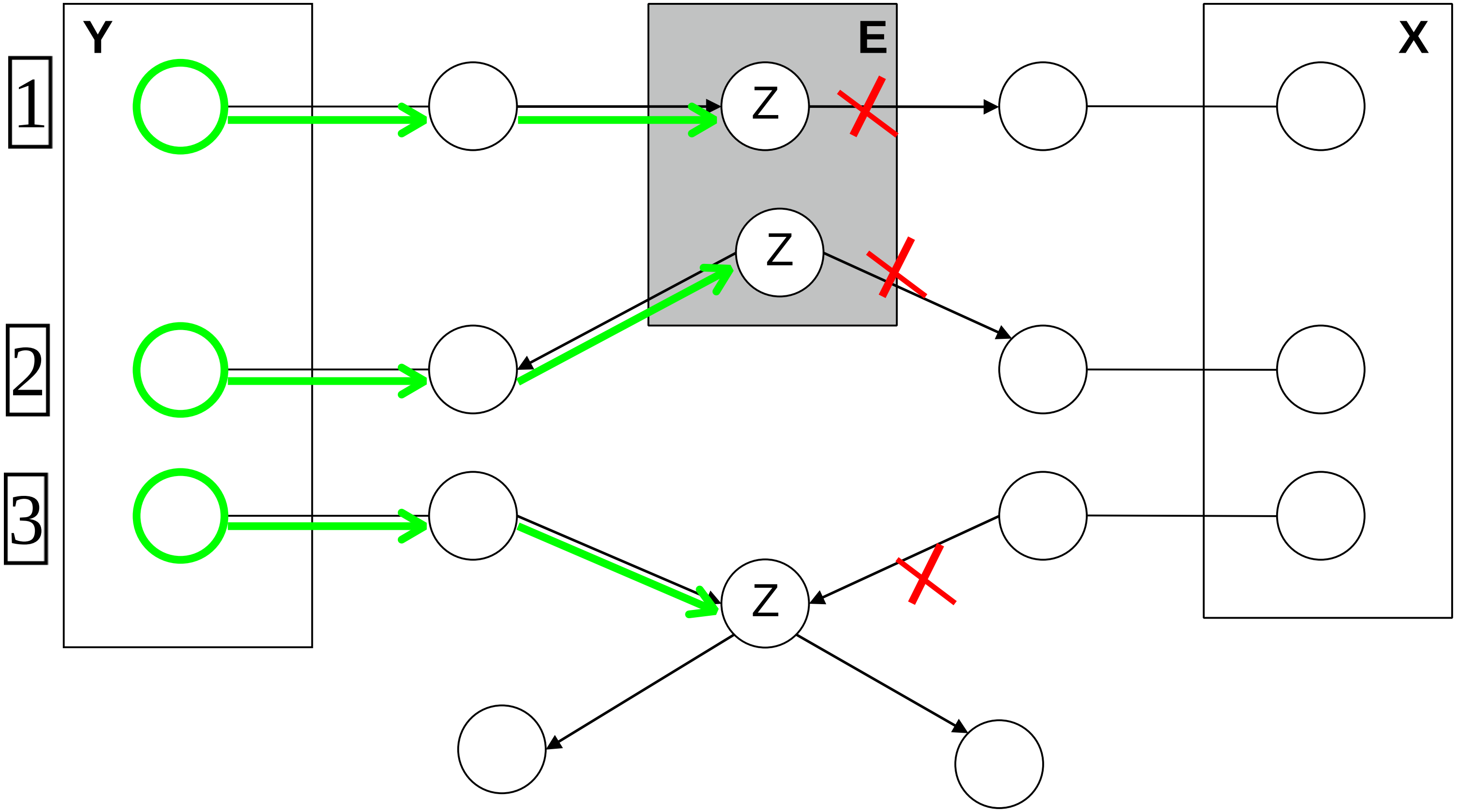


- In 3, X and Y become dependent as soon as there is evidence on Z or on any of its descendants.

- This is because knowledge of one possible cause given evidence of the effect explains away the other cause

Or Conditional Independencies

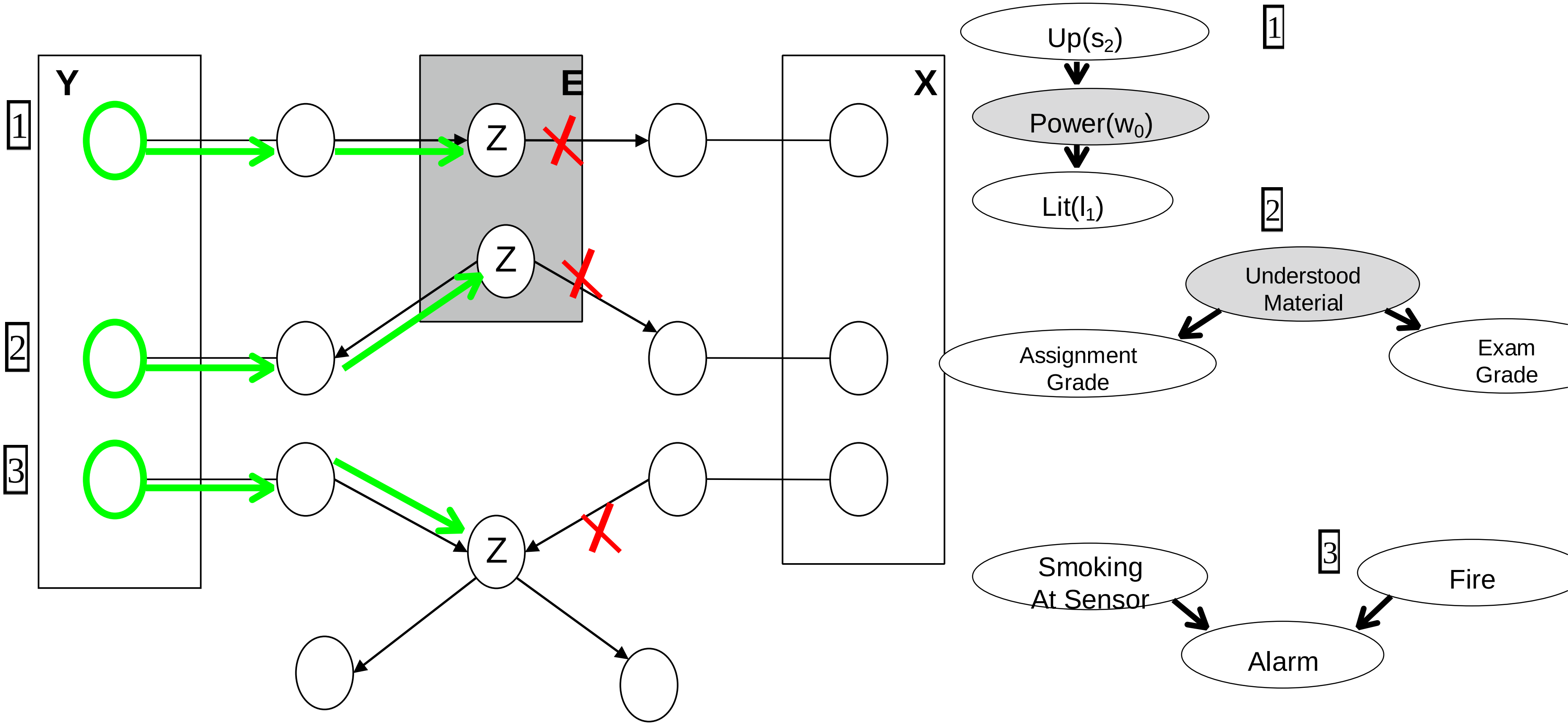
Or, blocking paths for probability propagation. Three ways in which a path between Y to X (or viceversa) can be blocked, given evidence E



- In 3, X and Y are independent if there is no evidence on their common effect (recall fire and tampering in the alarm example

Or Conditional Independencies

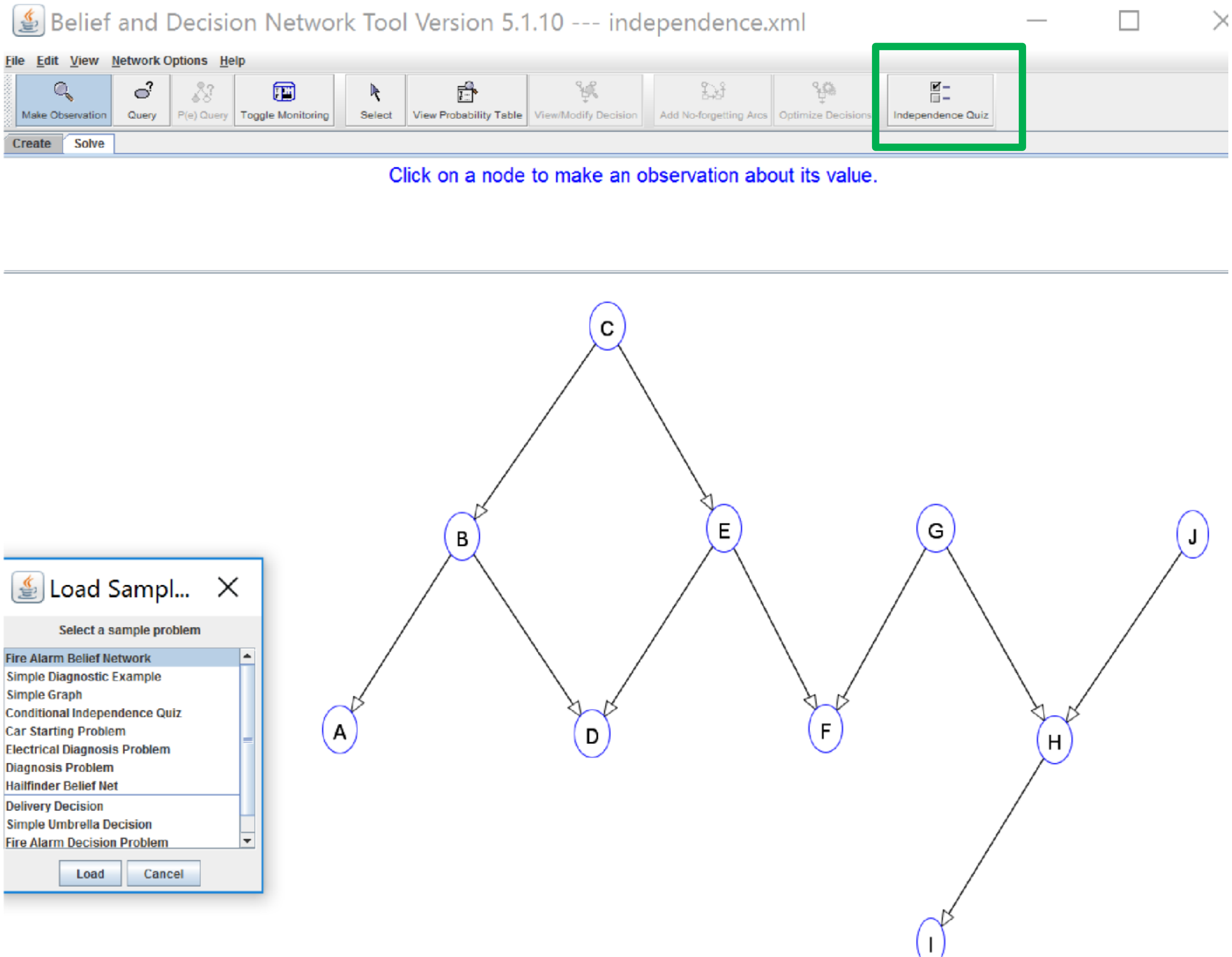
Or, blocking paths for probability propagation. Three ways in which a path between Y to X (or viceversa) can be blocked, given evidence E



- In 3, X and Y are independent if there is no evidence on their common effect (recall fire and tampering in the alarm example

Practice in the AISpace Applet

- Open the Belief and Decision Networks applet



Belief and Decision Network Tool Version 5.1.10 --- independence.xml

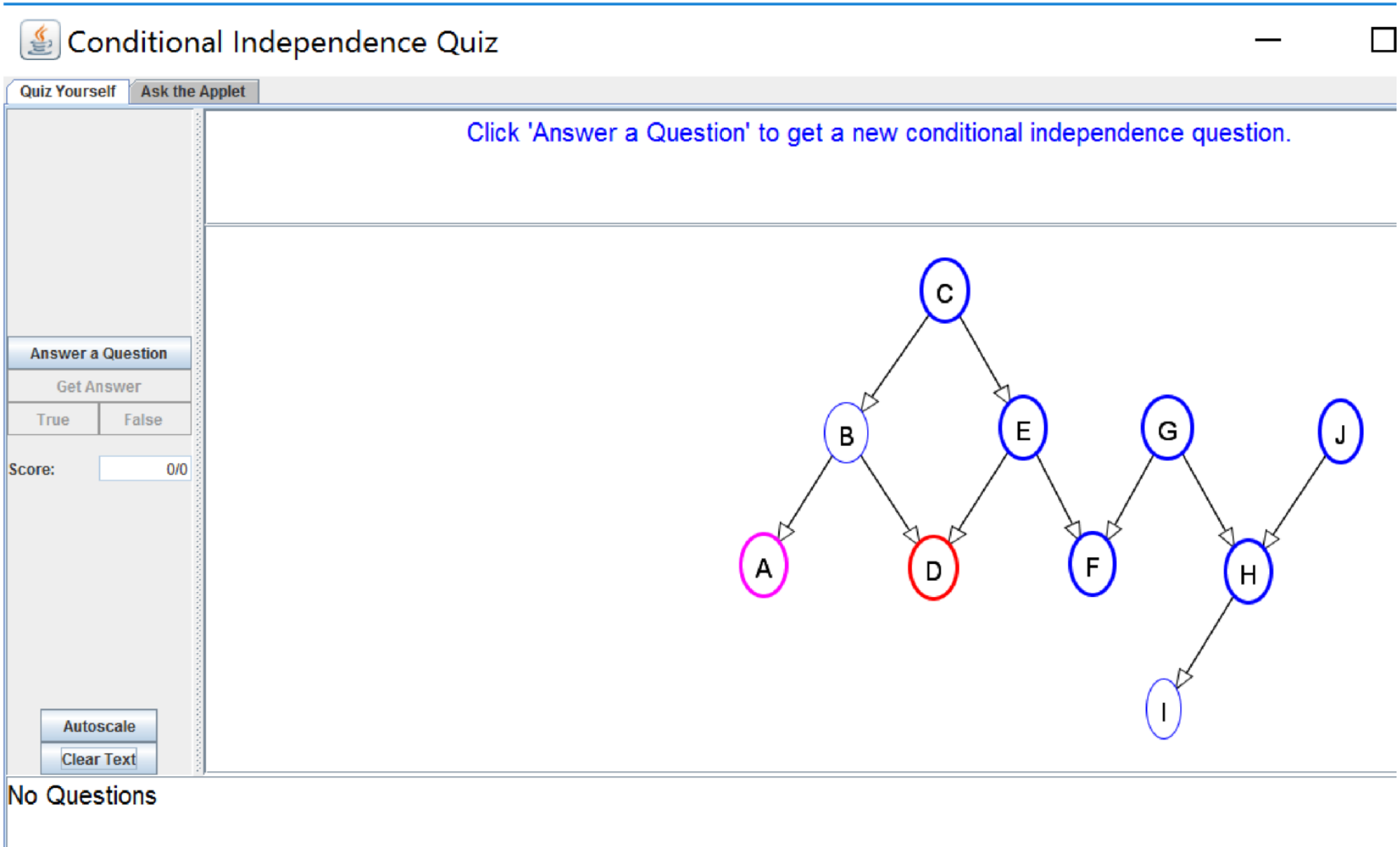
Click on a node to make an observation about its value.

Select a sample problem

- Load the problem: Conditional Independence Quiz
- Click on Independence Quiz

Practice in the AISpace Applet

- Answer Quizzes in the Conditional Independence Quiz Panel



Conditional Independence Quiz

Click 'Answer a Question' to get a new conditional independence question.

Score: 0/0

No Questions

Learning Goals so Far

- Given a JPD
- Marginalize over specific variables
- Compute distributions over any subset of the variables
- Use inference by enumeration
- to compute joint posterior probability distributions over any subset of variables given evidence
- Define and use marginal and conditional independence
- Build a Bayesian Network for a given domain (structure)
- Specify the necessary conditional probabilities
- Compute the representational savings in terms of number of probabilities required
- Identify dependencies/independencies between nodes in a Bayesian

Network

Now we will see how to do inference in BNETS

Inference Under Uncertainty

- Y: subset of variables that is queried (e.g. Temperaturein example next)
- E: subset of variables that are observed . E = e (W = yesin example)
- Z_1, …,Z_k remaining variables in the JPD (Cloudyin example)

Remember our example of Inference by Enumeration

- Given P(W,C,T) as JPD below, and evidence e : “Wind=yes”
- What is the probability that it is cold? I.e., P(T=cold | W=yes)
- Step 1: condition to get distribution P(C, T| W=yes)

Windy *W*	*Cloudy* *C*	*Temperature* *T*	*P(W, C, T)*
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
~~no~~	~~no~~	~~hot~~	~~0.06~~
~~no~~	~~no~~	~~mild~~	~~0.11~~
~~no~~	~~no~~	~~cold~~	~~0.03~~
~~no~~	~~yes~~	~~hot~~	~~0.04~~
~~no~~	~~yes~~	~~mild~~	~~0.25~~
~~no~~	~~yes~~	~~cold~~	~~0.08~~

Remember our example of Inference by Enumeration

Cloudy *C*	*Temperature* *T*	*P(C, T| W=yes)*
no	hot	
no	mild	
no	cold	
yes	hot	
yes	mild	
yes	cold	

$$PP(CC\wedge TT|WW = yyyyyy) =$$

$$= \frac{PP(CC\wedge TT\wedge WW{=}yyyyyy)}{PP(WW{=}yyyyyy)}$$

Remember our example of Inference by Enumeration

As per definition of conditional probability

- Given P(W,C,T) as JPD below, and evidence e : “Wind=yes”

Windy *W*	*Cloudy* *C*	*Temperature* *T*	*P(W, C, T)*
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
~~no~~	~~no~~	~~hot~~	~~0.06~~
~~no~~	~~no~~	~~mild~~	~~0.11~~
~~no~~	~~no~~	~~cold~~	~~0.03~~
~~no~~	~~yes~~	~~hot~~	~~0.04~~
~~no~~	~~yes~~	~~mild~~	~~0.25~~
~~no~~	~~yes~~	~~cold~~	~~0.08~~

- What is the probability that it is cold? I.e., P(T=cold | W=yes)
- Step 1: condition to get distribution P(C, T| W=yes)

Remember our example of Inference by Enumeration

Cloudy C	Temperature T		P(C W=y
	hot	0.04/0.43 ≅ 0.10	
	mild	0.09/0.43 ≅ 0.21	
	cold	0.07/0.43 ≅ 0.16	
	hot	0.01/0.43 ≅ 0.02	
	mild	0.10/0.43 ≅ 0.23	
	cold	0.12/0.43 ≅ 0.28	

$$PP(CC\wedge TT|WW = yyyyyy)$$

$$=$$

$$= \frac{PP(CC\wedge TT\wedge WW=yyyyyy)}{PP(WW=yyyyyy)}$$

Remember our example of Inference by Enumeration

Windy W	*Cloudy C*	*Temperature T*	*P(W, C, T)*
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
~~no~~	~~no~~	~~hot~~	~~0.06~~
~~no~~	~~no~~	~~mild~~	~~0.11~~
~~no~~	~~no~~	~~cold~~	~~0.03~~
~~no~~	~~yes~~	~~hot~~	~~0.04~~
~~no~~	~~yes~~	~~mild~~	~~0.25~~
~~no~~	~~yes~~	~~cold~~	~~0.08~~

Cloudy C	*Temperature T*	*P(C, T| W=yes)*
no	hot	0.04/0.43 ≅ 0.10
no	mild	0.09/0.43 ≅ 0.21
no	cold	0.07/0.43 ≅ 0.16
yes	hot	0.01/0.43 ≅ 0.02
yes	mild	0.10/0.43 ≅ 0.23
yes	cold	0.12/0.43 ≅ 0.28

$$\frac{P(C \wedge T \wedge W = yyy)}{P(W = yyy)}$$

P(W = yes) is the sum of all these probabilities

Obtained by marginalizing over Cloudy and

As per definition of conditional

Remember our example of Inference by Enumeration

probability Temperature

P(W = yes) is essentially a normalization factor that makes the new conditional probabilities sum to 1

- Given P(W,C,T) as JPD below, and evidence e : “Wind=yes”
- What is the probability that it is cold? I.e., P(T=cold | W=yes)
- Step 2: marginalize over Cloudy to get distribution P(T | W=yes)

Remember our example of Inference by Enumeration

Cloudy *C*	*Temperature* *T*	*P(C, T| W=yes)*
sunny	hot	0.10
sunny	mild	0.21
sunny	cold	0.16
cloudy	hot	0.02
cloudy	mild	0.23
cloudy	cold	0.28

Temperature *T*	*P(T| W=yes)*
hot	0.10+0.02 = 0.12
mild	0.21+0.23 = 0.44
cold	0.16+0.28 = 0.44

We get the same result if we

- first marginalize over Cloudy in the original P(W,C,T), for the entries consistent with the evidence Wind = yes
- and then normalized

We get the same result if we

- first marginalize over Cloudy in the original P(W,C,T), for the entries consistent with Wind = yes

Windy *W*	Cloudy *C*	Temperature *T*	*P(W,C,T)*
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
~~no~~	~~no~~	~~hot~~	~~0.06~~
~~no~~	~~no~~	~~mild~~	~~0.11~~
~~no~~	~~no~~	~~cold~~	~~0.03~~
~~no~~	~~yes~~	~~hot~~	~~0.04~~
~~no~~	~~yes~~	~~mild~~	~~0.25~~
~~no~~	~~yes~~	~~cold~~	~~0.08~~

$$P(T, W = yes, C = no) + P(T, W = yes, C = yes)$$

Wind *W*	Temperature *T*	*P(T, W = yes)*
yes	hot	0.05
yes	mild	0.19
yes	cold	0.19

- and then normalized

$$\frac{PP(TT \wedge WW=yyyyyy)}{PP(WW=yyyyyy \wedge TT=hoooo) + PP(WW=yyyyyy \wedge TT=mmmmmmmmm) + PP\ WW=yyyyyy \wedge TT=ccoommmm}$$

Temperature T	P(T| W=yes)
hot	0.05/0.43 = ~0.12
mild	0.19/0.43= ~0.44
cold	0.19/0.43 = ~0.44

Inference in General

- Y: subset of variables that is queried (e.g. Temperaturein previous example)
- E: subset of variables that are observed . E = e (W = yesin previous example)
- Z_1, …,Z_k remaining variables in the JPD (Cloudyin previous example)

We need to compute this numerator for each value of Y, y_i

We need to marginalize over all the variables Z_1,…Z_k not involved in the query $P(Y = y_i, E = e) = \sum_{Z_1} \ldots \sum_{Z_k} P(Z_1, \ldots, Z_k, Y = y_i, E = e)$

$P(Y,$

$P(Y \mid E=e) =$ $E=e)$ Def of conditional probability

$P(E=e)$

$P(Y,E=e)$

To compute the denominator, marginalize over Y

$\sum_Y P(Y,E=e)$

constant- Same value for every ensuring that P(Y=$\sum_Y$ y$P(Y$i).

Normalization = $y_i \mid E) = 1$

- All we need to compute is the numerator: joint probability of the query variable(s) and the evidence!
- Variable Elimination is an algorithm that efficiently performs this operation by casting it as operations between factors - introduced next

Lecture Overview

- Recap
- Final Considerations on Network Structure
- Variable Elimination
  - ➡ Factors
- Algorithm (time permitting)

Factors

- A factor is a function from a tuple of random variables to the real numbers R
- We write a factor on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$

- $P(X_1, X_2)$ is a factor $f(X_1, X_2)$

Distribution

- $P(Z \mid X,Y)$ is a factor $f(Z,X,Y)$

Set of Distributions

One for each combination of values for X and Y

- $P(Z=f \mid X,Y)$ is a factor $f(X,Y)$

Set of partial Distributions

$f(X, Y)_{Z = f}$

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

- Note: Factors do not have to sum to one

- A factor denotes one or more (possibly partial) distributions over the given tuple of variables, e.g.,

Operation 1: assigning a variable

- We can make new factors out of an existing factor

- Our first operation: we can assign some or all of the variables of a factor.

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4

More examples of assignment

f(X,Y,Z):

f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

What is the result of assigning X= t ?

$f(X=t,Y,Z) = f(X, Y, Z)_{X = t}$

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

Factor of Y,Z

Y	Z	val

f(X,Y,Z):

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f(X=t,Y,Z)

Factor of Y,Z

f(X=t,Y,Z=f):

Y	val
t	0.9
f	0.8

f(X=t,Y=f,Z=f): 0.8

Number

Factor of Y

Recap

- If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor?

Recap

If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor? • f(B).

When we assign variable A we remove it from the factor’s domain

Operation 2: Summing out a variable

- Our second operation on factors: we can marginalize out (or sum out) a variable
- Exactly as before. Only difference: factors don’t have to sum to 1
- Marginalizing out a variable X from a factor $f(X_1,\dots,X_n)$ yields a new factor defined on $\{X_1,\dots,X_n\} \setminus \{X\}$

□ □

B	A	C	val
t	t	t	0.03
t	t	f	0.07
f	t	t	0.54
f	t	f	0.36
t	f	t	0.06
t	f	f	0.14

□□$\sum f$

$X_n)$

□ X_1 □

f	f	t	0.48
f	f	f	0.32

□□$(X_2,\ldots,X_n) = \sum_{x \in dom(X_1)} f(X_1 = x, X_2, \ldots,$

$(\sum_B f_3)(A,C)$

f_3=

A	C	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

Operation 2: Summing out a variable

- Our second operation on factors: we can marginalize out (or sum out) a variable

- Exactly as before. Only difference: factors don’t sum to 1
- Marginalizing out a variable X from a factor f(X_1,… ,X_n) yields a new factor defined on {X_1,… ,X_n} \ {X}

$$\left(\sum_{X_1} f\right)(X_2,\ldots,X_n) = \sum_{x\in dom(X_1)} f\,(X_1 = x, X_2,\ldots,X_n)$$

f_3=

B	A	C	val
t	t	t	0.03
t	t	f	0.07
f	t	t	0.54
f	t	f	0.36
t	f	t	0.06
t	f	f	0.14
f	f	t	0.48
f	f	f	0.32

($\sum_B f_3$)(A,C)

A	C	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

Recap

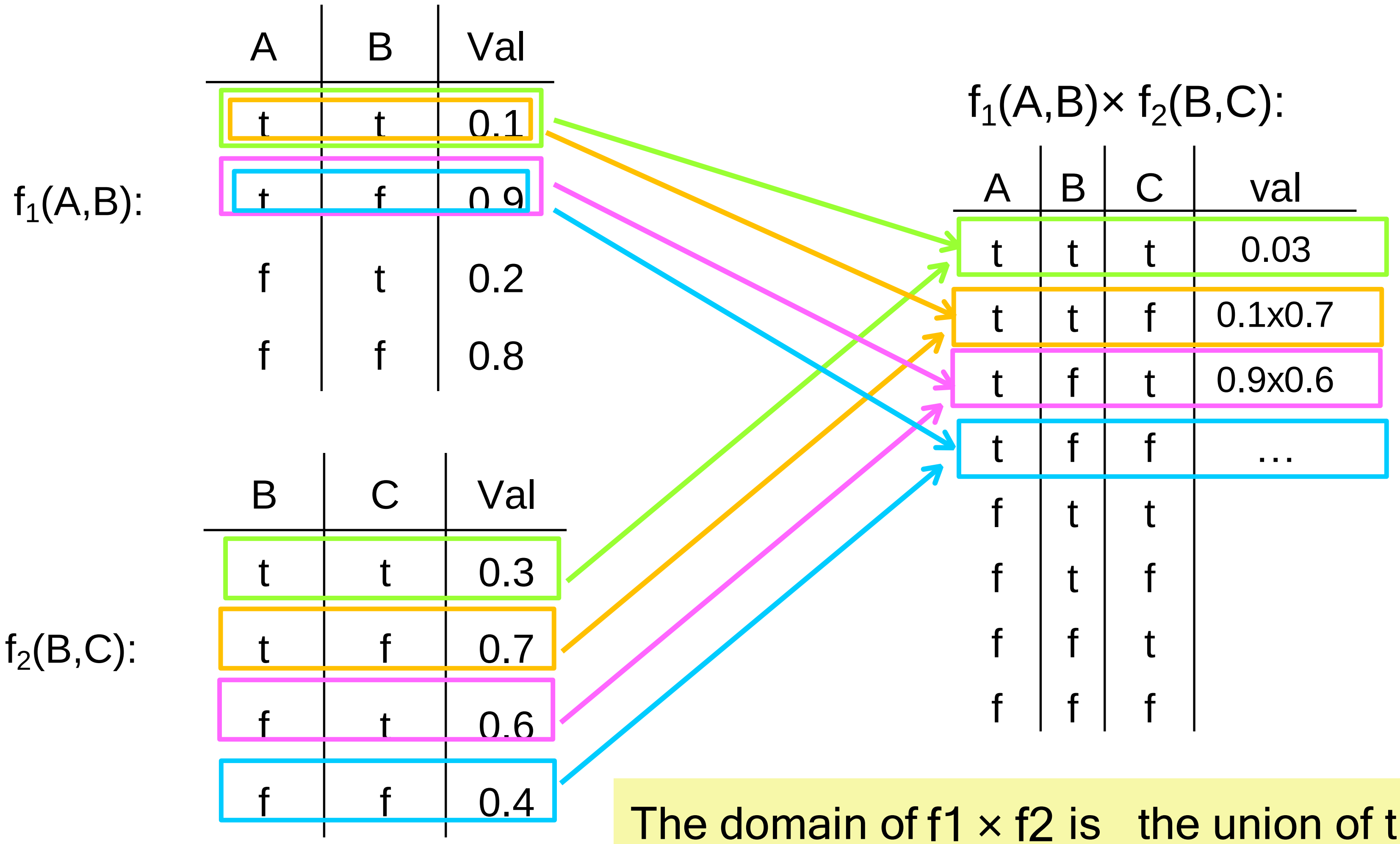
If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we assign variable A we remove it from the factor's domain

If we marginalize variable A out from factor f(A,B), what is the correct form for the resulting factor?

Operation 3: multiplying factors

The product of factors $f_1(A, B)$ and $f_2(B, C)$, where B is the variable (or set of variables) in common, is the factor $(f_1 \times f_2)(A, B, C)$defined by:



$f_1(A,B)$:

A	B	Val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$f_2(B,C)$:

B	C	Val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1(A,B)\times f_2(B,C)$:

A	B	C	val
t	t	t	0.03
t	t	f	0.1x0.7
t	f	t	0.9x0.6
t	f	f	…
f	t	t	
f	t	f	
f	f	t	
f	f	f	

The domain of f1 × f2 is the union of the domains of f1 and f2

$$(f_1 \times f_2)(A,B,C) = f_1(A,B) \times f_2(B,C)$$

Recap

If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we assign variable A we remove it from the factor's domain

If we marginalize variable A out from factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we marginalize out variable A we remove it from the factor's domain

If we multiply factors $f_4(X,Y)$ and $f_6(Z,Y)$, what is the correct form for the resulting factor?

Recap

If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we assign variable A we remove it from the factor's domain

If we marginalize variable A out from factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we marginalize out variable A we remove it from the factor's domain

If we multiply factors $f_4(X,Y)$ and $f_6(Z,Y)$, what is the correct form for the resulting factor?

- f(X,Y,Z)
- When multiplying factors, the resulting factor’s domain is the union of the multiplicands’ domains

Lecture Overview

- Recap
- Final Considerations on Network Structure
- Variable Elimination
- Factors
    - Algorithm (time permitting)

Inference in General

- Y: subset of variables that is queried
- E: subset of variables that are observed . E = e
- $Z_1, \dots, Z_k$ remaining variables in the JPD

We need to compute this numerator for each value of Y, y_i

We need to marginalize over all the variables $Z_1,\dots Z_k$ not involved in the query

$$P(Y = y_i, E = e) = \sum_{Z_1} \dots \sum_{Z_k} P(Z_1, \dots, Z_k, Y = y_i, E = e)$$

$$P(Y \mid E=e) = \frac{P(Y, E=e)}{P(E=e)}$$ Def of conditional probability

$$\frac{P(Y,E=e)}{\sum_Y P(Y,E=e)}$$

To compute the denominator, marginalize over Y
- Same value for every P(Y=y_i). Normalization constant ensuring that $\sum_Y P(Y=y_i \mid E)=1$

- All we need to compute is the numerator: joint probability of the query variable(s) and the evidence!
- Variable Elimination is an algorithm that efficiently performs this operation by casting it as operations between factors

Variable Elimination: Intro (1)

- We can express the joint probability as a factor

observed Other variables not involved in the query

- $f(Y, E_1 ..., E_j, Z_1 ...,Z_k)$

- We can compute $P(Y, E_1=e_1, ..., E_j=e_j)$ by
- Assigning $E_1=e_1, ..., E_j=e_j$
- Marginalizing out variables $Z_1, ..., Z_k$, one at a time
  - ✓the order in which we do this is called our elimination ordering

$$P(Y, E_1 = e_1, \ldots, E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y, E_1, .., E_j, Z_1, .., Z_k)_{E_1 = e_1, \ldots, E_j = e_j}$$

- Are we done?

No, this still represents the whole JPD (as a single factor)!
Need to exploit the compactness of Bayesian networks

Variable Elimination Intro (2)

$$P(Y,E_1 = e_1,\ldots,E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y,E_1,..,E_j,Z_1,..,Z_k)_{E_1=e_1,\ldots,E_j=e_j}$$

Recall the JPD of a Bayesian network

$$\mathrm{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}) = \prod_{i=1}^{n} P(X_i \mid pa(X_i))$$

We can express the joint factor as a product of factors, one for each conditional probability

$$P(X_i \mid pa(X_i)) = f(X_i, pa(X_i)) = f_i$$

$$P(Y, E_1 = e_1, \ldots, E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y, E_1, .., E_j, Z_1, .., Z_k)_{E_1=e_1,\ldots,E_j=e_j}$$

$$= \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} (f_i)_{E_1=e_1,\ldots,E_j=e_j}$$

Computing sums of products

Inference in Bayesian networks thus reduces to computing the sums of products ^{n}

$$\sum \cdots \sum \prod (f)_{E_1=e_1,\ldots,E_j} \quad {}_1 \quad i=1 \quad {}^{i} \quad =e_j$$

Z_k Z

To compute efficiently n

$$\sum \prod f_i$$

Z_k $i=1$

- Factor out those terms that don't involve Z_k, e.g.:

$$\sum_{A} \boxed{f(C,D)} \times f$$

$$(A,B,D) \times f(E,A) \times \boxed{f(D)}$$

$$\boxed{f(C,D) \times f(D)} \sum f(A,B,D) \times f(E,A)$$

$$_{A} f(C,D) \times$$

$$f(D) \times f'(B,D,E)$$

Summing out a variable efficiently

To sum out a variable Z from a product $f_1 \times \ldots \times f_k$ of factors

- Partition the factors into
  - ✓Those that do not contain Z, say $f_1 ,.., f_i$
  - ✓Those that contain Z, say $f_{i+1} ,..., f_k$

- Rewrite

$$\sum_Z f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times \left(\sum_Z f_{i+1} \times \cdots \times f_k \right)$$

- We thus have

New factor *f'* obtained by

$$\sum f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f'$$

then summing out

Z multiplying f_{i+1} ,..,f_k and Z

- Now we have summed out Z

Z_k

$$= \sum \cdots \sum$$

Simplify Sum of Product: General Case

Factors that do not contain Z_1

Factors that contain Z_1

$$\sum_{Z_k}\cdots\sum_{Z_1} f_1\times\cdots\times f_h = \sum_{Z_k}\cdots\sum_{Z_2}(f_1\times\cdots\times f_i)\left[\sum_{Z_1} f_{Z_1 1}\times\cdots\times f_{Z_1 k}\right]$$

$$=\sum\cdots\sum_{Z_2} f_1\times\cdots\times f_i\times f'$$

Factors that contain Z_2

$$(f_m\times\cdots\times f_j)\sum_{Z_2}(f_{Z_2 1}\times\cdots\times f_{Z_2 k})$$

Factors that do not contain Z_2

$$=\sum_{Z_k}\cdots\sum_{Z_3} f_m\times\cdots\times f_j\times f''$$

Z_k Z_3

Etc., continue given a predefined simplification ordering of the variables: variable elimination ordering

Analogy with “Computing sums of products”

This simplification is similar to what you can do in basic algebra with multiplication and addition

Example: it takes 14 multiplications or additions to evaluate the expression ab + ac + ad + aeh + afh + agh.

How can this expression be evaluated efficiently?

- Factor out the a and then the h giving $a(b + c + d + h(e + f + g))$
- This takes only 7 operations