This lecture is partially based on an excellent set of lecture slides by Andrew Moore, available at: http://www.cs.cmu.edu/~awm/tutorials

Machine Learning for Cities
CUSP-GX 5006.001, Spring 2019

Lectures 8-9: Bayesian Networks and Causality

Some motivation for Bayes Nets

Urban systems consist of many interconnected sub-systems. We wish to understand the complex dependencies between these systems, both predictive and causal. We use this understanding to **model** "typical" system behavior, to **detect** anomalies, trends and patterns, and to inform possible **interventions**.

In this course, we will use **Bayesian networks** both to model probabilistic dependencies between many variables and to infer the causal relationships between them.

Why Bayesian networks?

- An easily interpretable graphical representation of the relationships between a set of variables.
- Bayes Nets can be specified manually or learned automatically from data, and enable computationally efficient probabilistic inferences.
- Many practical and successful applications in medicine, manufactuing, failure diagnosis:
  - Diagnosis: infer Pr(problem type | symptoms)
  - Prediction: infer probability distributions for values that are expensive or impossible to measure.
  - Anomaly Detection: detect observations that are very unlikely (i.e. have low probabilities given the model).
  - Active Learning: choose the most informative diagnostic test to perform given these observations.

Introduction to Bayesian networks

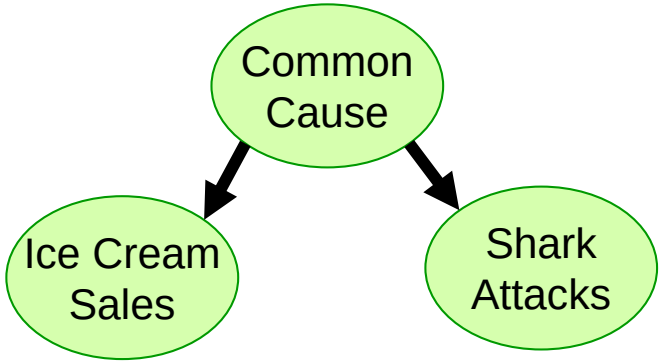
Recent Dow-Jones Change	Number of Ice Creams Sold Today	Number of Shark Attacks Today
UP	3500	4
STEADY	41	0
UP	2300	5
DOWN	3400	4
UP	18	0
STEADY	105	0
STEADY	4	0
STEADY	6310	3
UP	70	0

More ice creams = More shark attacks!

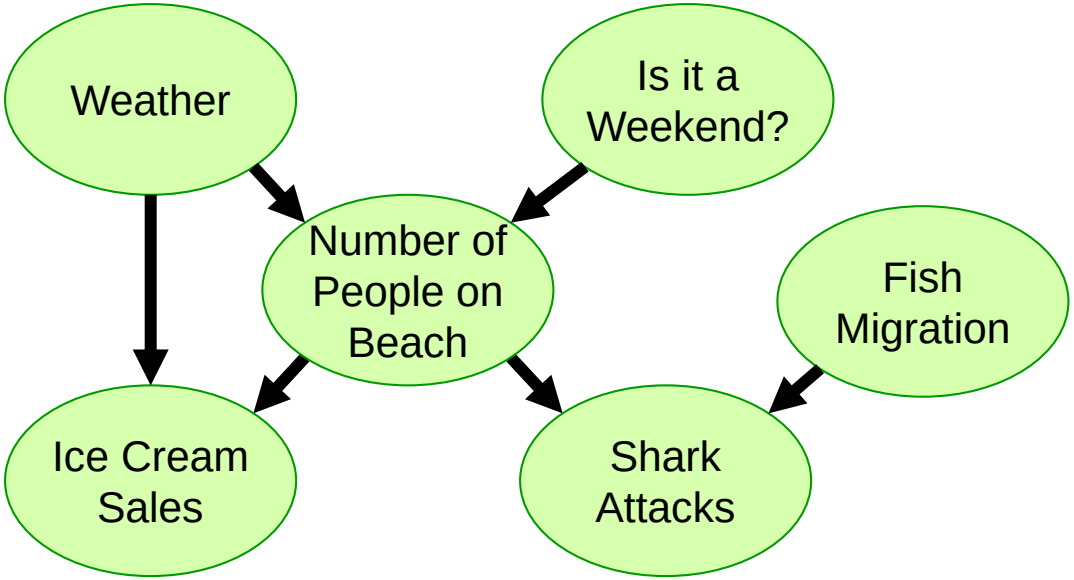
Does eating ice cream cause shark attacks?

Do shark attacks affect ice cream consumption?

Perhaps the two events have a common cause!



Introduction to Bayesian networks



Many other factors may influence some or all of these variables…

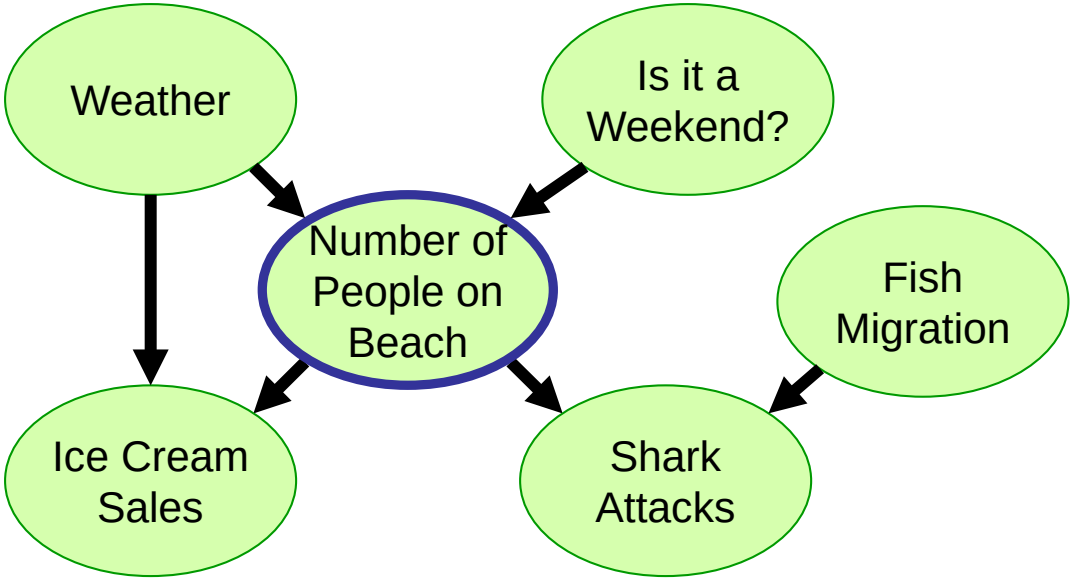
What we have here is a way of representing probabilistic relationships between variables. We call this a **Bayesian network**.

Causal interpretation: Link X → Y means that variable X directly **causes** variable Y.

Probabilistic interpretation: The links encode **conditional dependencies** between variables.

"X and Y are conditionally independent given Z":
Pr(X | Z) = Pr (X | Y, Z) and Pr(Y | Z) = Pr(Y | X, Z)

Introduction to Bayesian networks



Many other factors may influence some or all of these variables…

What we have here is a way of representing probabilistic relationships between variables. We call this a **Bayesian network**.

Each node has a probability distribution that is conditioned on its parents' values.

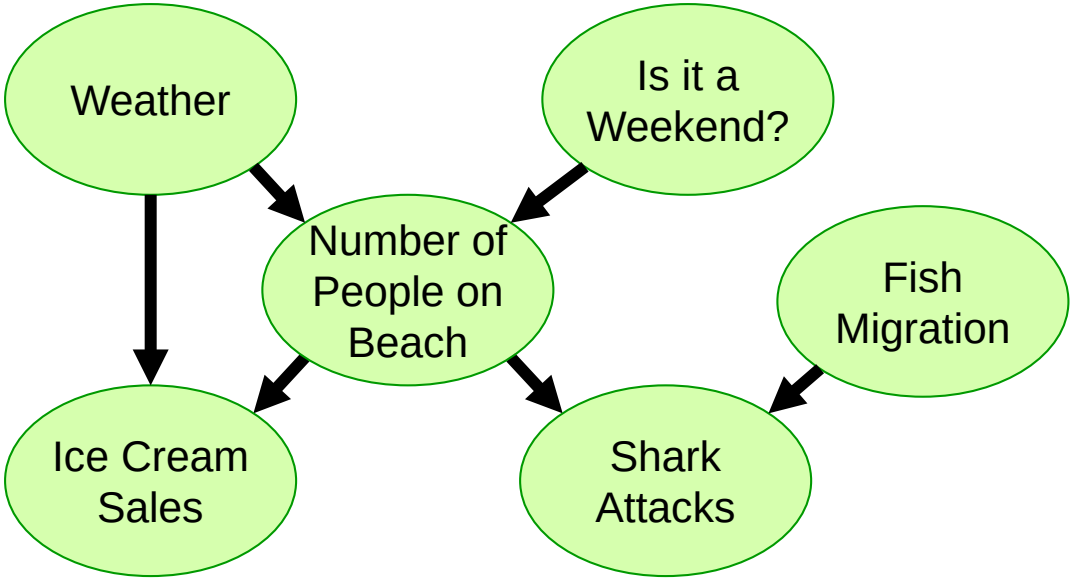
Causal interpretation: Link X → Y means that variable X directly **causes** variable Y.

Probabilistic interpretation: The links encode **conditional dependencies** between variables.

"X and Y are conditionally independent given Z":
Pr(X | Z) = Pr (X | Y, Z) and Pr(Y | Z) = Pr(Y | X, Z)

Sunny, Weekend:
Pr (Beach Crowded) = 90%
Sunny, Not Weekend:
Pr (Beach Crowded) = 40%
Not Sunny, Weekend:
Pr (Beach Crowded) = 20%
Not Sunny, Not Weekend:
Pr (Beach Crowded) = 1%

Introduction to Bayesian networks



Many other factors may influence some or all of these variables…

What we have here is a way of representing probabilistic relationships between variables. We call this a **Bayesian network**.

In a Bayes Net, each node is **conditionally independent** of its non-descendents given its parents.

Causal interpretation: Link X → Y means that variable X directly **causes** variable Y.

Probabilistic interpretation: The links encode **conditional dependencies** between variables.

"X and Y are conditionally independent given Z": Pr(X | Z) = Pr (X | Y, Z) and Pr(Y | Z) = Pr(Y | X, Z)

For example, if you already know the number of people on the beach and have fish migration data, knowing today's weather doesn't give you any more information about shark attacks.

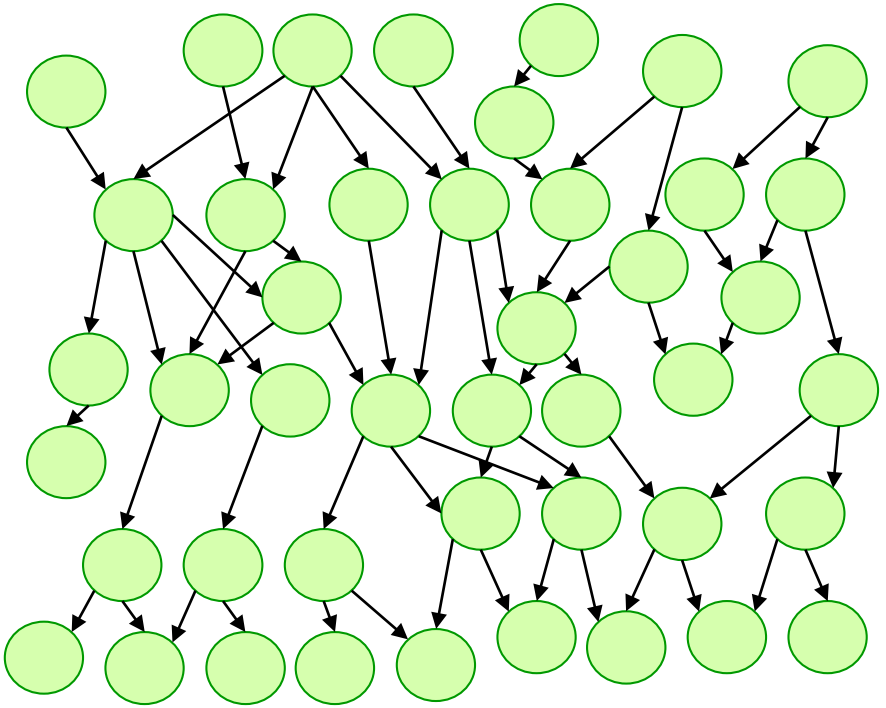
CI(Shark Attacks, Weather | Number of People on Beach, Fish Migration)

Real-world Bayes Nets

Bayesian networks for real-world application domains may have hundreds or thousands of nodes.

They can be built manually, consulting domain experts for the structure and probabilities.

More often, the probabilities (and in some cases, the structure) are **learned** from training data.



Pathfinder (a diagnostic system for lymph-node diseases) uses a Bayes Net with 60 diseases, 100 symptoms, and 14,000 probabilities.

The Bayes Net was constructed manually by human experts: 8 hours to choose variables, 35 hours for structure, 40 hours for probabilities.

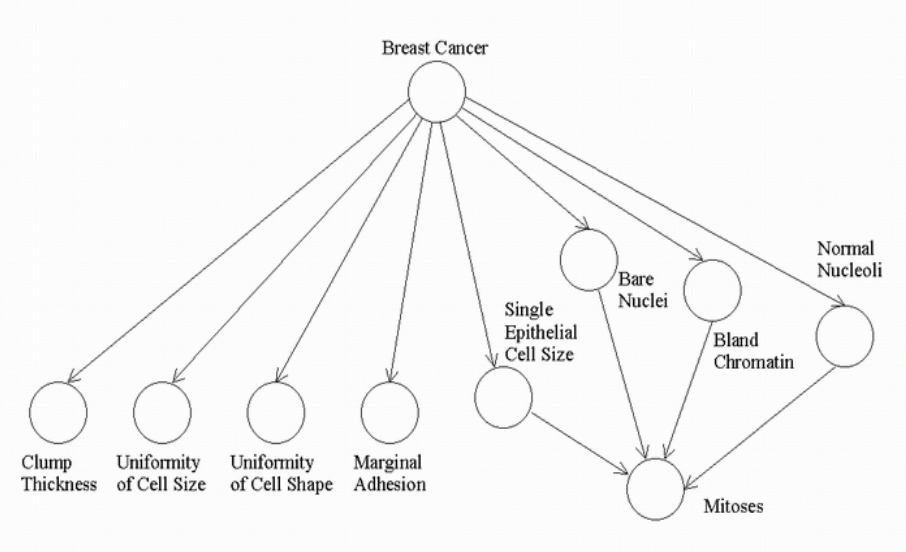
Pathfinder was shown to outperform human experts in diagnosis accuracy, and has been extended to other medical domains.

Real-world Bayes Nets

Bayesian networks for real-world application domains may have hundreds or thousands of nodes.

They can be built manually, consulting domain experts for the structure and probabilities.

More often, the probabilities (and in some cases, the structure) are **learned** from training data.



Medical diagnosis is a common application of Bayes Nets: here's a simple network which can be used to infer Pr(breast cancer | symptoms).

Pathfinder (a diagnostic system for lymph-node diseases) uses a Bayes Net with 60 diseases, 100 symptoms, and 14,000 probabilities.

The Bayes Net was constructed manually by human experts: 8 hours to choose variables, 35 hours for structure, 40 hours for probabilities.

Pathfinder was shown to outperform human experts in diagnosis accuracy, and has been extended to other medical domains.

Joint probability distributions

a.k.a. “Why are Bayes Nets so useful for representing probabilities?”

To create the joint distribution of M discrete-valued variables:

Make a truth table listing all combinations of values of your variables. If there are M binary variables, the table has 2^M rows.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Joint probability distributions

a.k.a. “Why are Bayes Nets so useful for representing probabilities?”

To create the joint distribution of M discrete-valued variables:

Make a truth table listing all combinations of values of your variables. If there are M binary variables, the table has 2^M rows.

For each combination of values, say how probable it is. These probabilities must sum to 1.

What if we have M = 100 variables: how can we use less than 2^{100} rows?

Use information about conditional independencies between variables to infer a Bayesian network structure!

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Joint probability distributions

a.k.a. “Why are Bayes Nets so useful for representing probabilities?”

To create the joint distribution of M discrete-valued variables:

Make a truth table listing all combinations of values of your variables. If there are M binary variables, the table has 2^M rows.

For each combination of values, say how probable it is. These probabilities must sum to 1.

What if we have M = 100 variables: how can we use less than 2^{100} rows?

Use information about conditional independencies between variables to infer a Bayesian network structure!

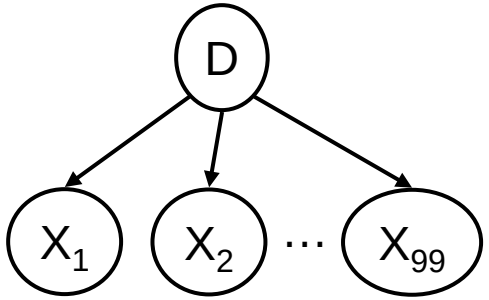
Here’s a simple example:

D = “there is a disease outbreak”

$X_1..X_{99}$ = “person i goes to the hospital”

If D is known, also knowing X_i does not change the probability of X_j:

$$Pr(X_j \mid D, X_i) = Pr(X_j \mid D)$$



$$Pr(D, X_1..X_{99}) = Pr(D) \prod_{i=1..99} Pr(X_i \mid D)$$

Bayes Nets often allow a much more compact representation of the joint distribution!

(In general, how many probabilities do we need?)

Building a Bayes Net

Small Bayes Nets are easy to build by hand, assuming that we understand the relationships between variables and are able to estimate their conditional probabilities.

Large Bayes Nets may require many person-hours to build, but they can also be **learned** automatically from data.

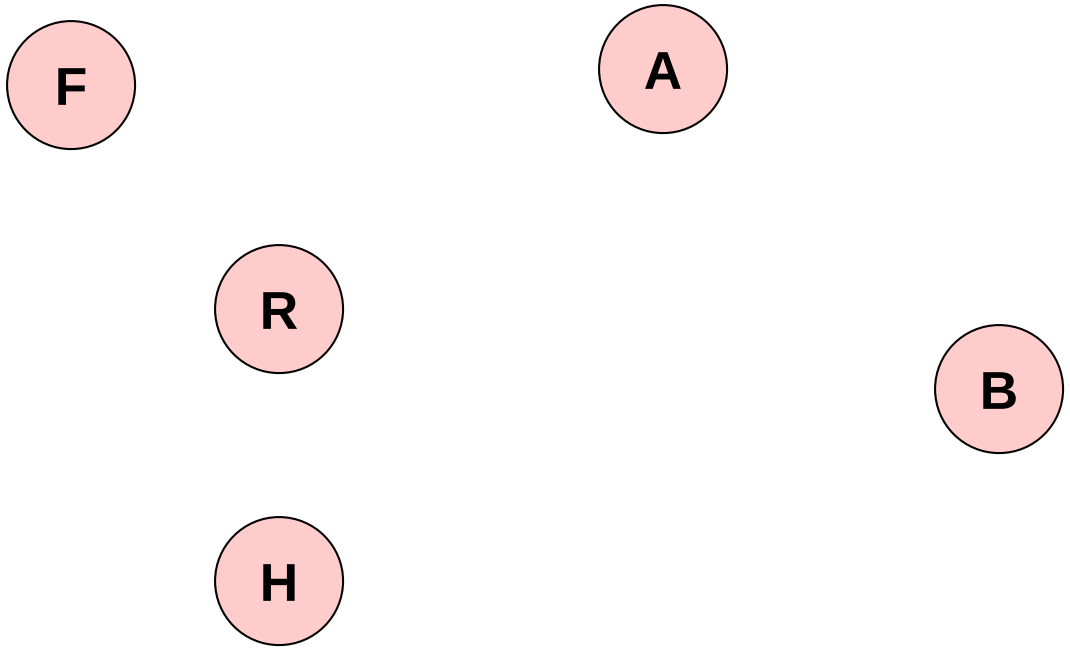
For example, let's assume that we want to build a Bayes Net to determine whether a terrorist anthrax attack has occurred.

1. An anthrax attack is likely to increase the level of respiratory illness.
2. Seasonal influenza is also likely to cause an increase in respiratory illness.
3. The CDC has a **hospital surveillance system** which alerts when the number of ED visits is abnormally high, and has additionally deployed **bio-sensors** for airborne anthrax detection.
4. The hospital surveillance system and the bio-sensors are not perfect: both false alarms and missed outbreaks are possible.

Define the following variables:

F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert

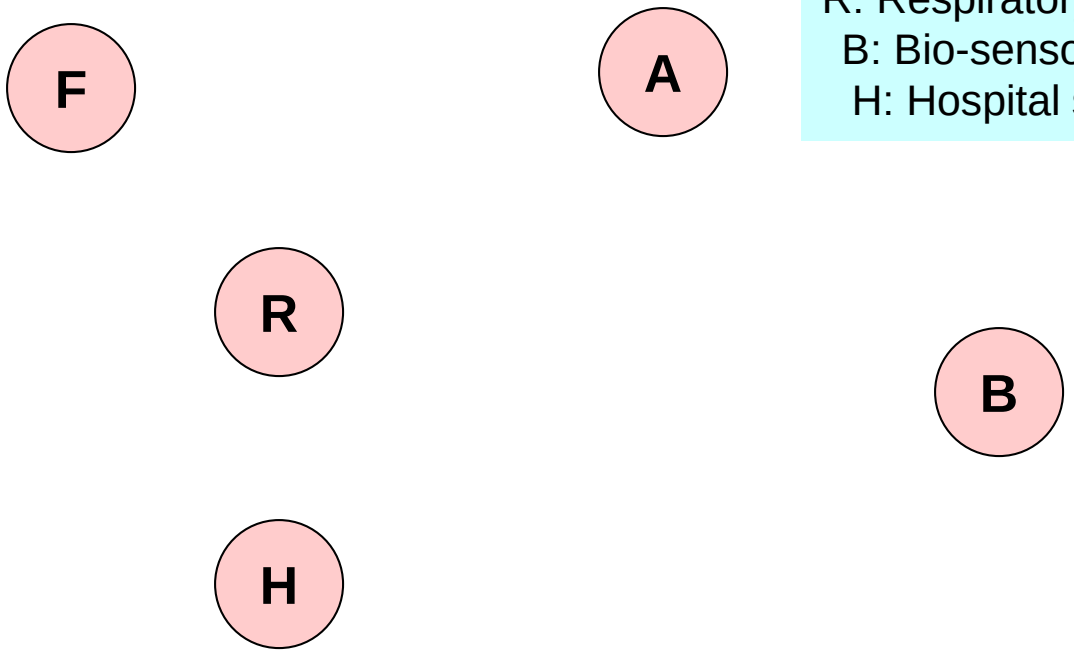
Building a Bayes Net



Step 1: Choose a set of relevant variables, and represent each variable by a node.

Building a Bayes Net

F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert



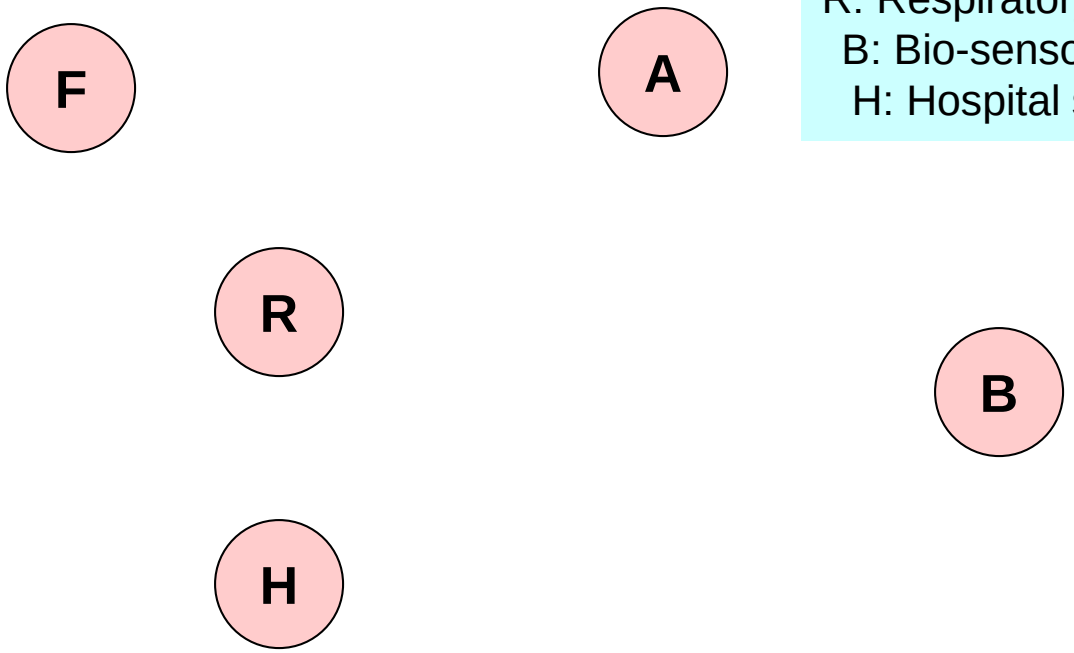
Step 2: Choose an ordering for the variables $X_1..X_M$, such that if X_i influences X_j, then $i < j$.

Hint: put environmental and event variables first, then latent variables, then observations.

Any ordering will produce a valid Bayes Net structure, but using the causal information will produce more compact (fewer links) and more interpretable structures.

Building a Bayes Net

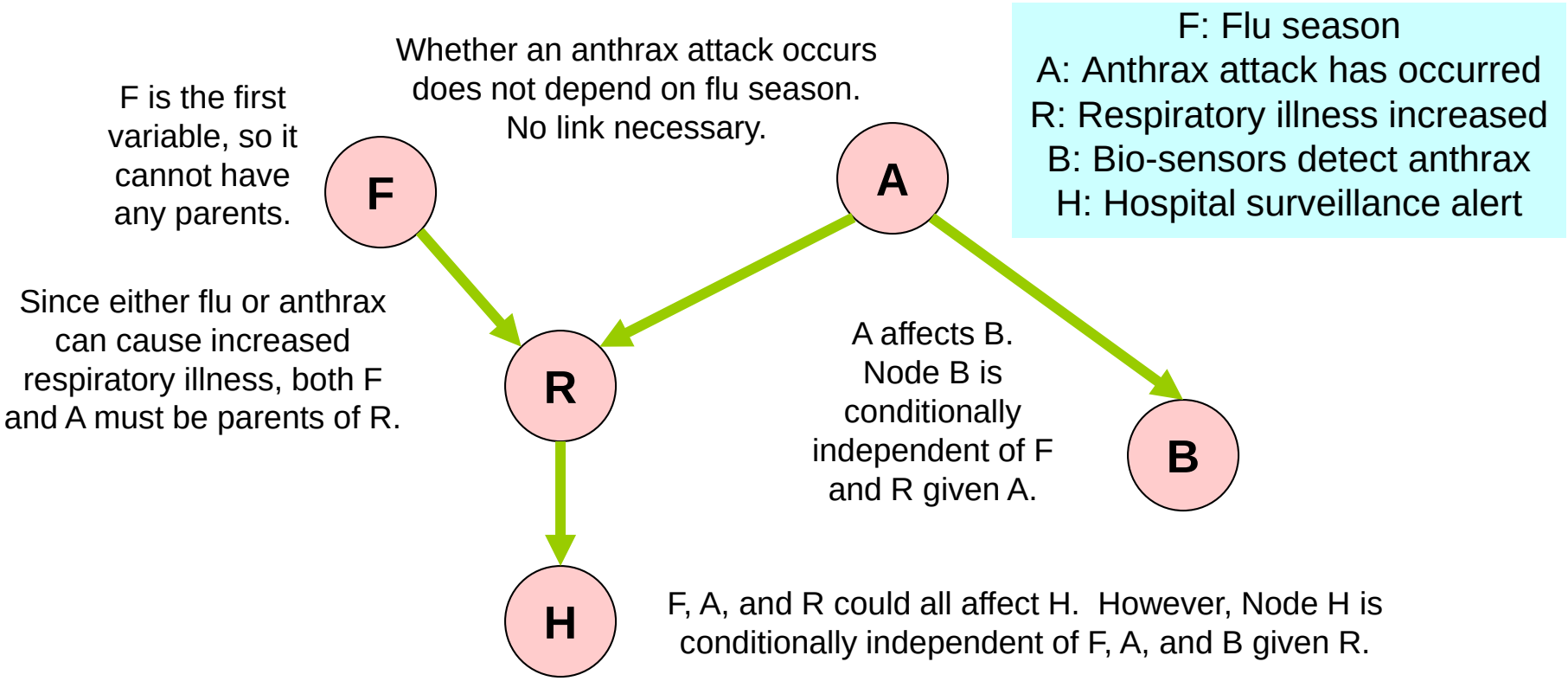
F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert



Step 3: Add links.

- The link structure must be acyclic.
- If node X is given parents $Q_1,Q_2,\ldots Q_m$, you are promising that any variable that's a non-descendent of X is conditionally independent of X given $\{Q_1,Q_2,..Q_m\}$

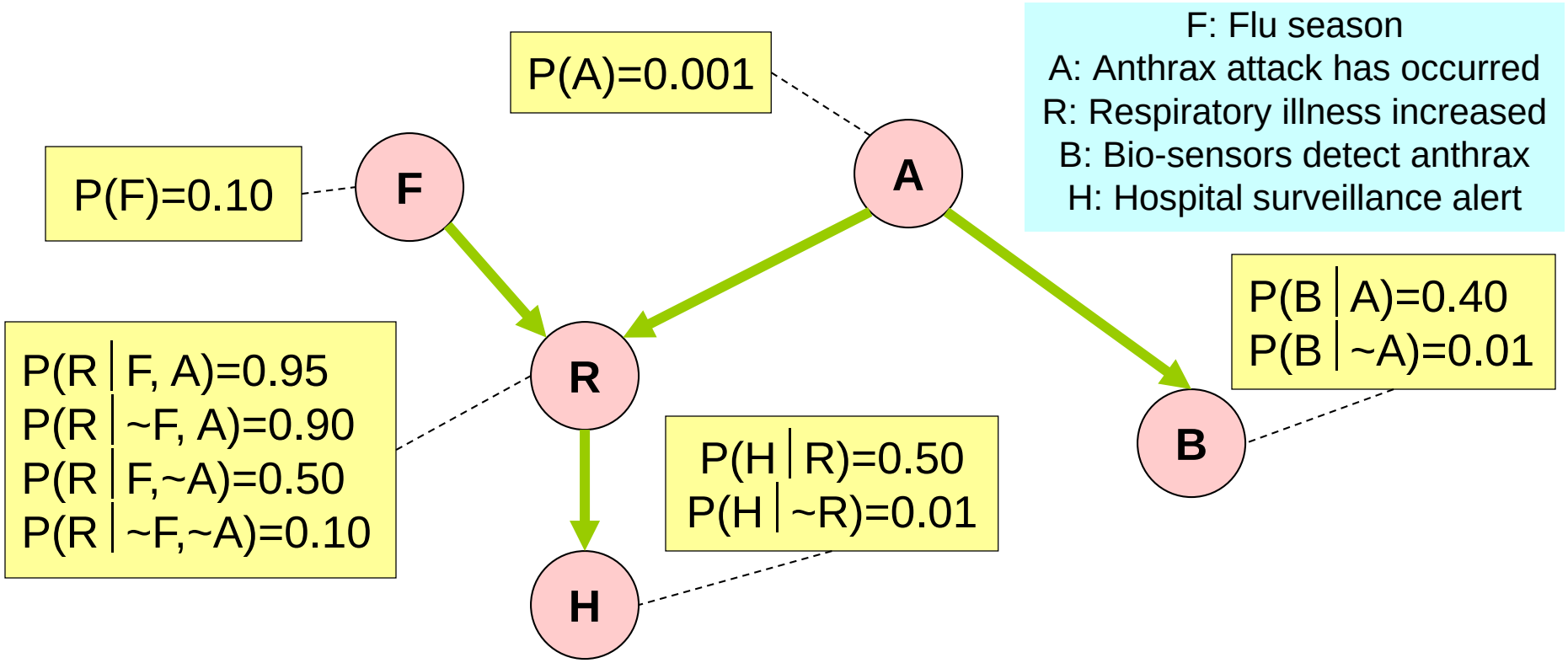
Building a Bayes Net



Step 3: Add links.

- For each variable X_i (for i = 2…M), choose a minimal subset of parents from $X_1..X_{i-1}$, such that X_i is conditionally independent of the rest of $X_1..X_{i-1}$ given its parents.

Building a Bayes Net



Step 4: Add a conditional probability table for each node.

- The table for node X must list Pr(X | Parents(X)) for each value of X and each combination of parent values.
- If X is binary, we know that Pr(~X | Parents) = 1 – Pr(X | Parents), and do not need to write this explicitly.

Building a Bayes Net

Practice problem: suppose that we're building a nuclear power station, and want to report when the core temperature is low.

The gauge is meant to read the temperature of the core, and the alarm is meant to sound if the gauge reads a low temperature.

However, the gauge or the alarm (or both) could be faulty.

Define the following variables:

G = Gauge reads low.

C = Core temperature is low.

FG = Gauge is faulty.

FA = Alarm is faulty.

A = Alarm sounds.

Which Bayesian network structure makes the most sense for these five variables?

Building a Bayes Net

Practice problem: suppose that we're building a nuclear power station, and want to report when the core temperature is low.

The gauge is meant to read the temperature of the core, and the alarm is meant to sound if the gauge reads a low temperature.

However, the gauge or the alarm (or both) could be faulty.

Define the following variables:

G = Gauge reads low.

C = Core temperature is low.

FG = Gauge is faulty.

FA = Alarm is faulty.

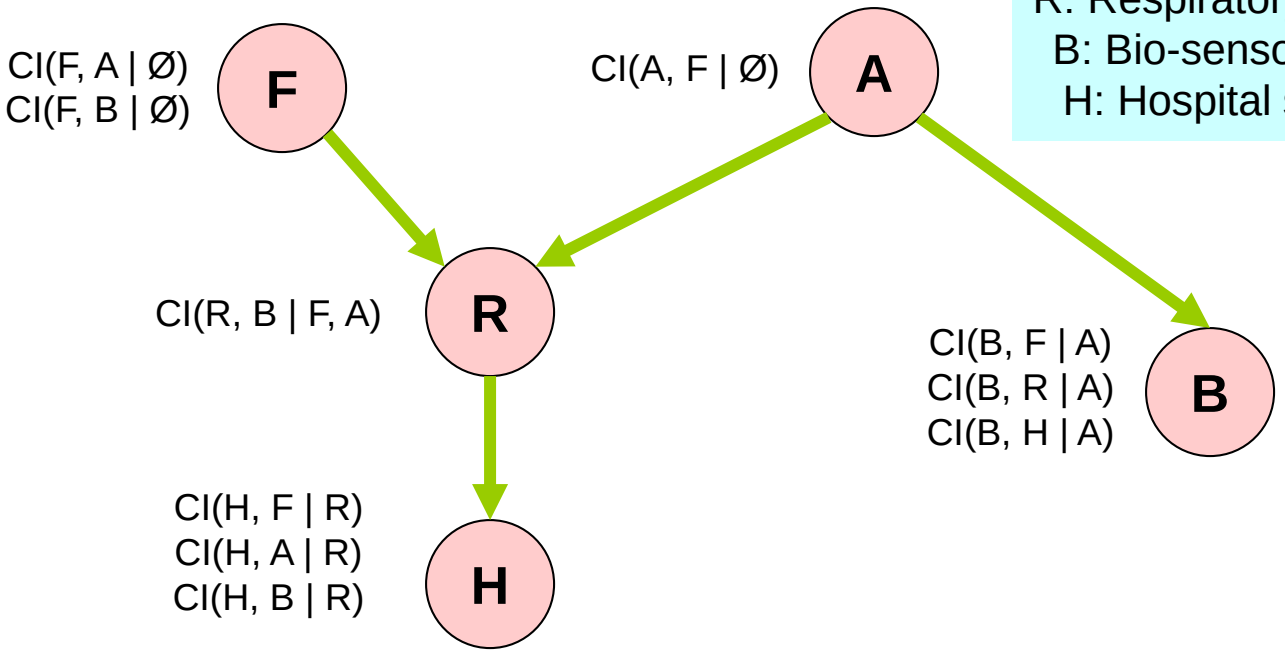
A = Alarm sounds.

Which Bayesian network structure makes the most sense for these five variables?

Delete me in PPT version to reveal the answer

Interpreting a Bayes Net structure

F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert

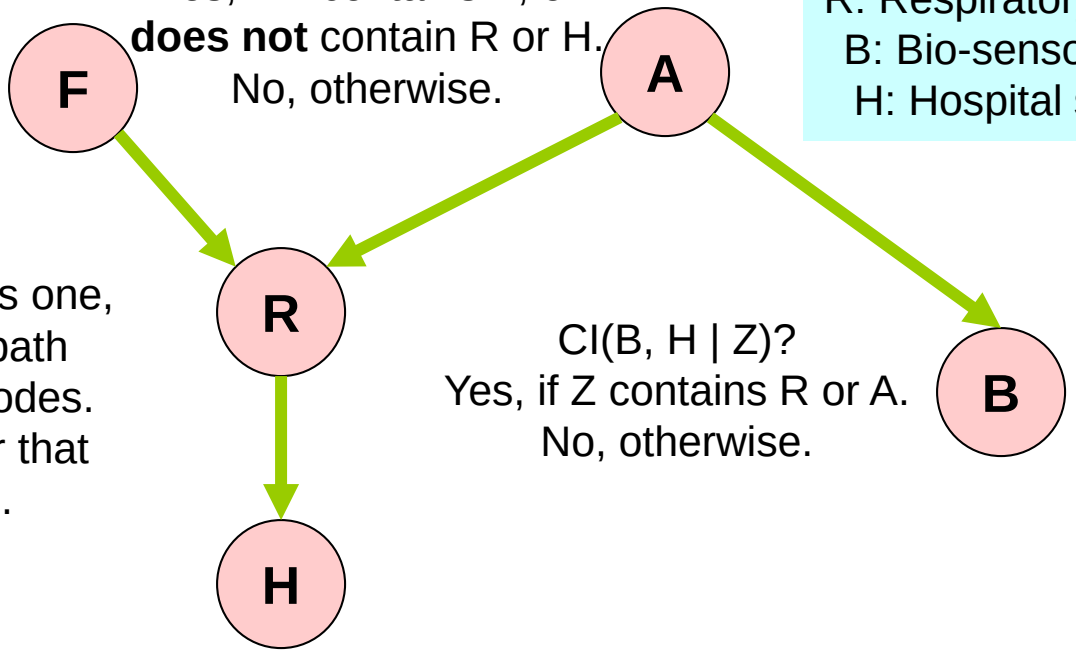


- Key property: each node is conditionally independent of all its non-descendants in the tree, given its parents.
- Two unconnected variables may still be correlated.
- Whether any two variables are conditionally independent can be deduced from a Bayes Net using “d-separation”.

d-separation

F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert

CI(F, B | Z)?
Yes, if Z contains A, or **does not** contain R or H.
No, otherwise.

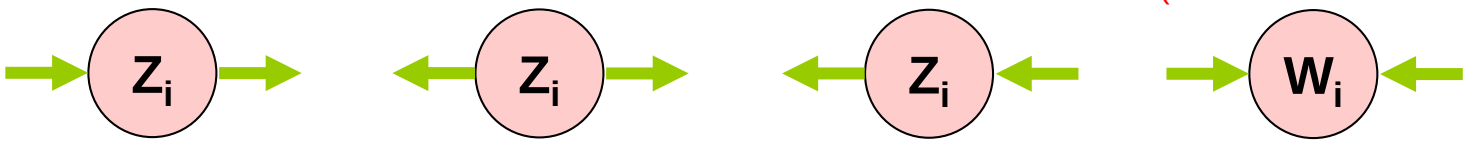


In a **polytree** like this one, there is only one path between any two nodes. Just check whether that path is blocked.

CI(B, H | Z)?
Yes, if Z contains R or A.
No, otherwise.

- Nodes X_i and X_j are conditionally independent given a set of nodes Z iff every undirected path between X_i and X_j is "blocked" by at least one of the following: $(Z_i \in Z, W_i \notin Z)$

(and no descendent of W_i is in Z)



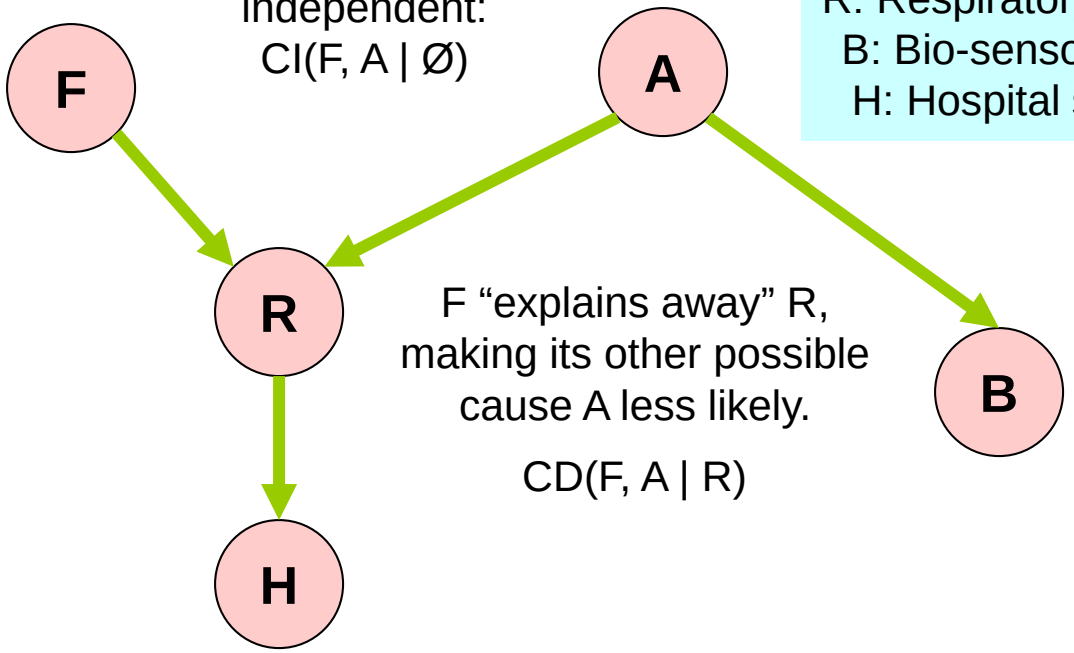
“Explaining away”

What if we know that respiratory illness has increased, and want to know whether an anthrax attack has occurred?

If it is flu season, the increase in respiratory illness is probably due to flu, not anthrax.

F: Flu season
A: Anthrax attack has occurred
R: Respiratory illness increased
B: Bio-sensors detect anthrax
H: Hospital surveillance alert

F and A are independent:
CI(F, A | Ø)

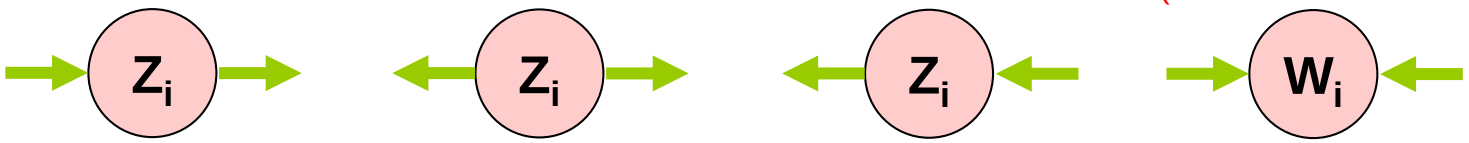


F “explains away” R, making its other possible cause A less likely.

CD(F, A | R)

- Nodes X_i and X_j are conditionally independent given a set of nodes Z iff every undirected path between X_i and X_j is “blocked” by at least one of the following: $(Z_i \in Z, W_i \notin Z)$

(and no descendent of W_i is in Z)

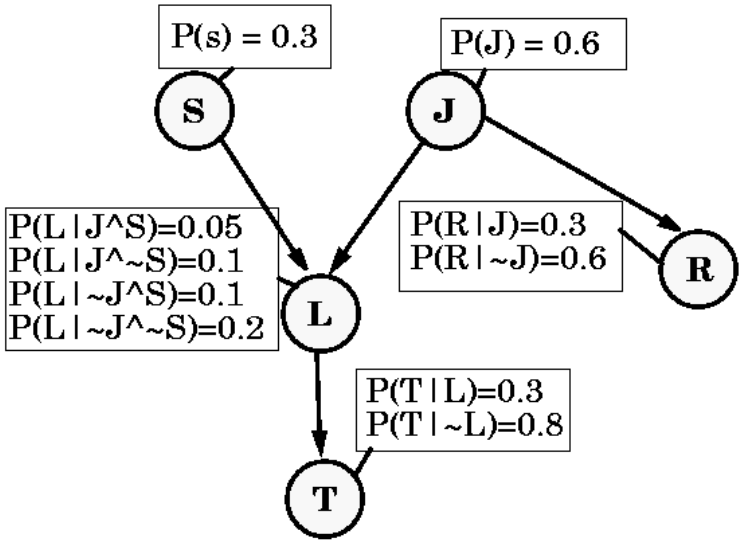


Where are we now?

- We can build a Bayesian network by hand, specifying the structure and the conditional probability tables.
- The network structure represents the conditional dependencies and independencies between variables, and can also have a causal interpretation.
- Bayes Nets are a more compact representation of the joint probability distribution: we only need to store a number of probabilities exponential in the number of parents per node, not the total number of nodes.
- We will now answer two main questions:
  - How can we use Bayes Nets for probabilistic **inference**? ("What is the probability of an anthrax attack, given that the hospital surveillance system and bio-sensors both alerted?")
  - How can we **learn** the Bayes Net structure and parameters automatically, using a large training dataset?

Bayes Net inference

Question 1: How to compute joint probabilities using a Bayes Net?



Compute Pr(S, ~J, L, ~R, T)

Answer: use conditional independence.

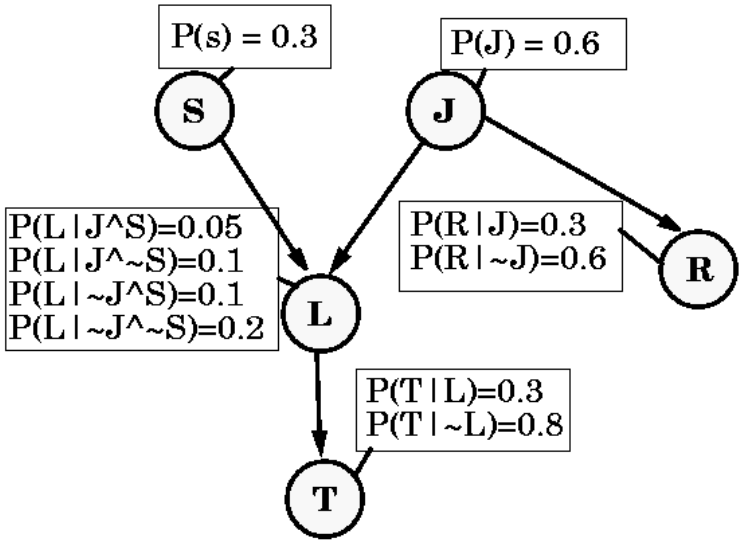
$Pr(X_1..X_M) = \prod_{i=1..M} Pr(X_i \mid Parents(X_i))$

Pr(S, ~J, L, ~R, T) =
Pr(S) Pr(~J) Pr(L | ~J, S) Pr(~R | ~J) Pr(T | L) =
(0.3) (0.4) (0.1) (0.4) (0.3) = 0.00144.

We can efficiently compute the joint probability of any given assignment of values to variables.

Bayes Net inference

Question 2: How to compute arbitrary conditional probabilities?



Compute Pr(S, T | ~J, L)

Express the conditional probability as a ratio:

$$Pr(S, T \mid \sim J, L) = Pr(S, \sim J, L, T) / Pr(\sim J, L)$$

Express numerator and denominator as sums of joint probabilities:

$$Pr(S, \sim J, L, T) = Pr(S, \sim J, L, R, T) + Pr(S, \sim J, L, \sim R, T)$$

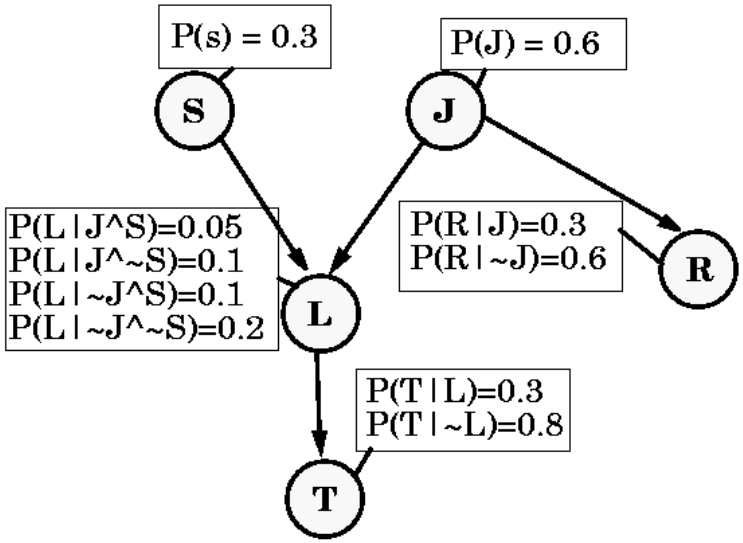
$$Pr(\sim J, L) = Pr(S, \sim J, L, R, T) + Pr(S, \sim J, L, R, \sim T) + Pr(\sim S, \sim J, L, R, T) + Pr(\sim S, \sim J, L, R, \sim T) + Pr(S, \sim J, L, \sim R, T) + Pr(S, \sim J, L, \sim R, \sim T) + Pr(\sim S, \sim J, L, \sim R, T) + Pr(\sim S, \sim J, L, \sim R, \sim T)$$

$$Pr(X \mid Y) = \frac{\sum_{\text{joint entries matching X and Y}} Pr(\text{joint entry})}{\sum_{\text{joint entries matching Y}} Pr(\text{joint entry})}$$

You have m binary variables in your Bayes Net, and expression Y uses k variables. How many rows of the joint do you have to calculate?

Bayes Net inference

Question 2: How to compute arbitrary conditional probabilities?



Compute Pr(S, T | ~J, L)

Express the conditional probability as a ratio:

$$Pr(S, T \mid \sim J, L) = Pr(S, \sim J, L, T) / Pr(\sim J, L)$$

Good news: we can sometimes simplify the probability calculations.

$$Pr(S, \sim J, L, T) = Pr(S)\, Pr(\sim J)\, Pr(L \mid S, \sim J)\, Pr(T \mid L)$$

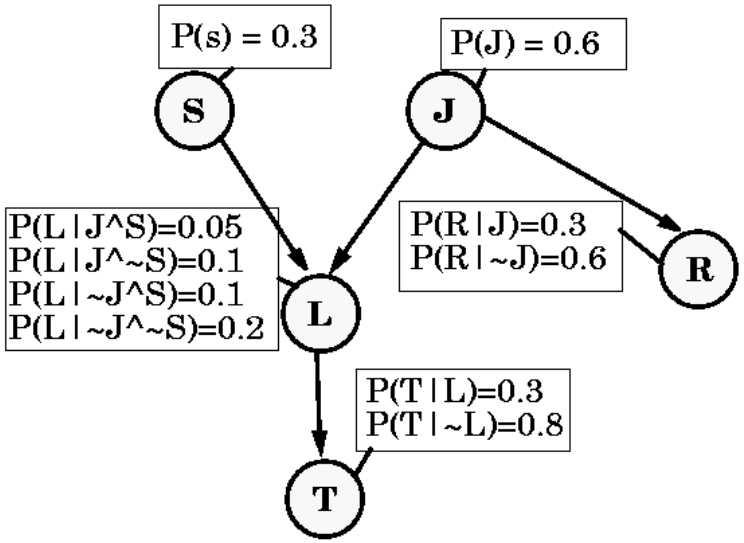
$$Pr(\sim J, L) = Pr(\sim J)\, Pr(L \mid \sim J)$$
$$Pr(L \mid \sim J) = Pr(L \mid \sim J, S)\, Pr(S) + Pr(L \mid \sim J, \sim S)\, Pr(\sim S)$$

$$Pr(X \mid Y) = \frac{\sum_{\text{joint entries matching X and Y}} Pr(\text{joint entry})}{\sum_{\text{joint entries matching Y}} Pr(\text{joint entry})}$$

You have m binary variables in your Bayes Net, and expression Y uses k variables. How many rows of the joint do you have to calculate?

Bayes Net inference

Question 2: How to compute arbitrary conditional probabilities?



Express the conditional probability as a ratio:

$$\Pr(S, T \mid \sim J, L) = \Pr(S, \sim J, L, T) / \Pr(\sim J, L)$$

Good news: we can sometimes simplify the probability calculations.

$$\Pr(S, \sim J, L, T) = \Pr(S)\Pr(\sim J)\Pr(L \mid S, \sim J)\Pr(T \mid L)$$

$$\Pr(\sim J, L) = \Pr(\sim J)\Pr(L \mid \sim J)$$

$$\Pr(L \mid \sim J) = \Pr(L \mid \sim J, S)\Pr(S) + \Pr(L \mid \sim J, \sim S)\Pr(\sim S)$$

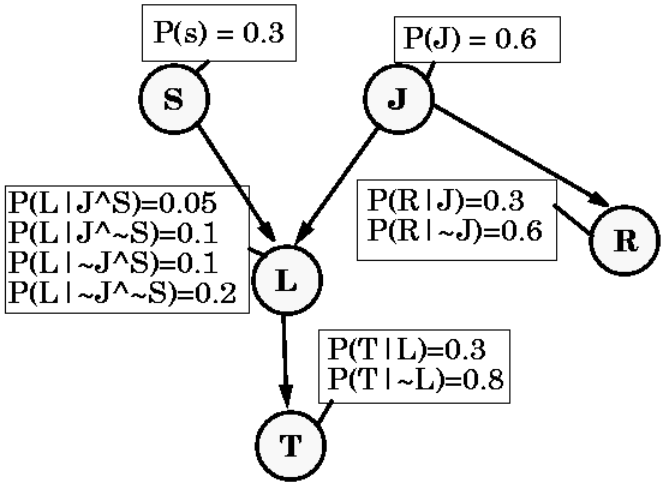
Bad news: doing exact inference for Bayes Nets is computationally hard.

$$\Pr(X \mid Y) = \frac{\sum_{\text{joint entries matching X and Y}} \Pr(\text{joint entry})}{\sum_{\text{joint entries matching Y}} \Pr(\text{joint entry})}$$

But it's tractable in some special cases (e.g. trees). We can also do efficient **approximate** inference.

You have m binary variables in your Bayes Net, and expression Y uses k variables. How many rows of the joint do you have to calculate?

Approximate inference



Compute Pr(S, T | ~J, L)

Problem: many of these N samples are wasted because Y is false.

Solution: only generate samples where Y is true, but weight them so that this property still holds.

To sample from the joint distribution of S, J, L, R, T

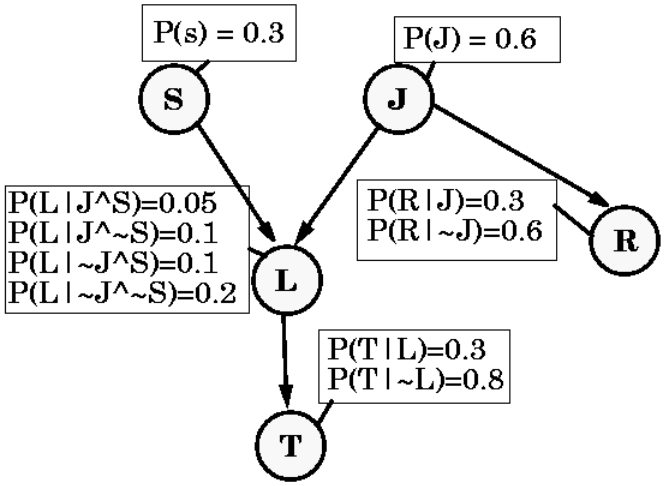
1. Randomly choose S (True with probability 0.3)
2. Randomly choose J (True with probability 0.6)
3. Randomly choose L. The probability that L is true depends on the assignments of S and J. If steps 1 and 2 produced S = True, J = False, then probability that L is true is 0.1.
4. Randomly choose R (Probability depends on J)
5. Randomly choose T (Probability depends on L)

To estimate any conditional probability Pr(X | Y)

1. Draw N samples from the joint distribution
2. Count N_Y = number of samples where Y is true
3. Count N_{XY} = number of samples where both X and Y are true.
4. Calculate Pr(X | Y) = N_{XY} / N_Y

→ For large N, the ratio of N_{XY} to N_Y converges to the true probability Pr(X | Y).

Likelihood weighted sampling



Compute Pr(S, T | ~J, L)

Problem: many of these N samples are wasted because Y is false.

Solution: only generate samples where Y is true, but weight them so that this property still holds.

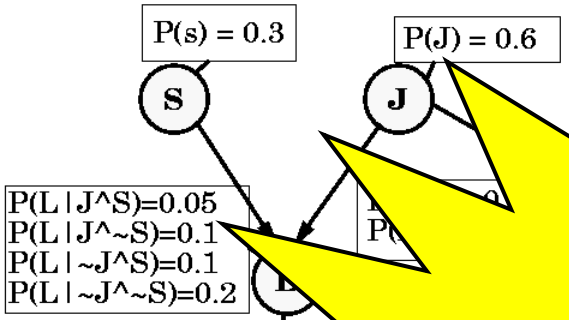
Choosing a sample from the joint, subject to ~J, L:

0. Set initial weight w = 1.
1. Randomly choose S (True with probability 0.3)
2. Multiply w by Pr(J = False) = 0.4. Set J = False.
3. Multiply w by Pr(L = True) given the current assignments of S and J. For example, if steps 1-2 produced S = True and J = False then multiply w by 0.1. Set L = True.
4. Randomly choose R (True with probability 0.6)
5. Randomly choose T (True with probability 0.3)

To estimate any conditional probability Pr(X | Y)

1. Draw N samples from the joint distribution, subject to the constraint Y.
2. For each sample and its weight w ≤ 1:
   Increment N_Y by w.
   If X is true, increment N_{XY} by w.
3. Calculate $Pr(X \mid Y) = N_{XY} / N_Y$

Likelihood weighted sampling



Now we know how to perform inference with a Bayes Net. This is great if we already have the network structure and parameters specified by an expert… but what if we want to **learn** the Bayes Net from data?

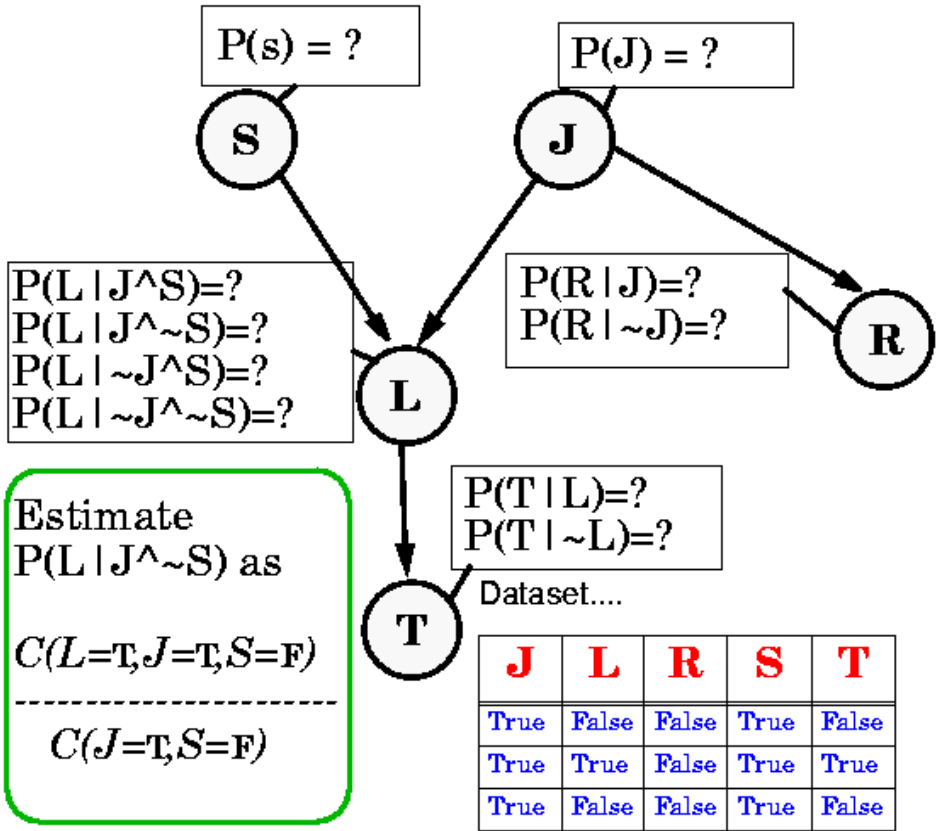
Solution: only gene… samples where Y is true, but weight them so that this property still holds.

Bayes Net parameter learning

Given a Bayesian network structure and a training dataset, we can learn the parameters of each node by maximum likelihood.

Given node X with parent nodes $Q_1..Q_m$, we learn the conditional distribution of X for each distinct combination of parent values.

For example, if $Q_1..Q_m$ are all binary, there are 2^m distributions to learn.

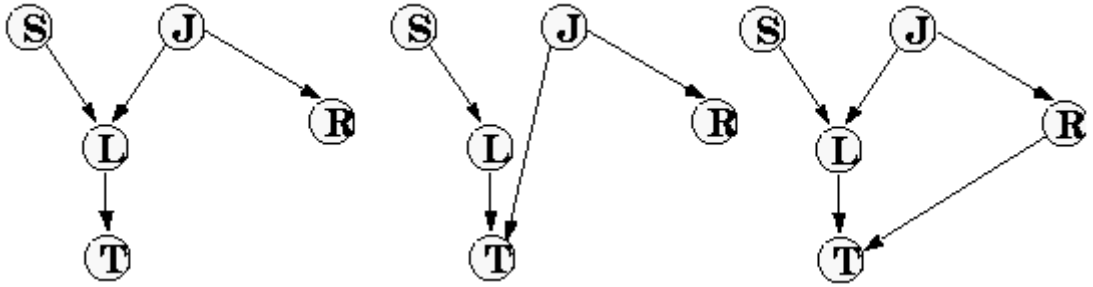


J	L	R	S	T
True	False	False	True	False
True	True	False	True	True
True	False	False	True	False
False	True	True	False	False
True	True	False	True	True
:	:	:	:	:

We learn the distribution Pr(X | $Q_1 = v_1$, $Q_2 = v_2$, …, $Q_m = v_m$) by looking at the subset of training records with the given parent values and computing the proportion with each X value.

Bayes Net structure search

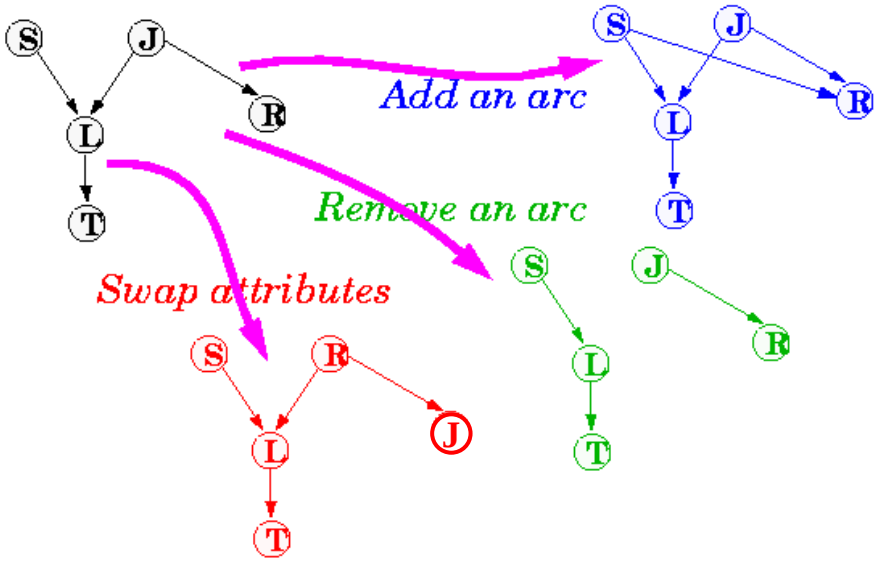
How to automatically find the Bayesian network structure that best fits the data?



This is a hard **state-space search** problem: use hill-climbing or simulated annealing with restarts.

What moveset to use?
How to score a structure?

Here's one possible moveset:



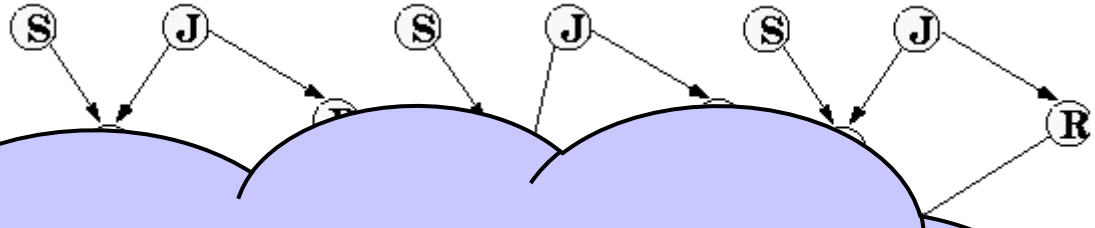
To score a structure, learn all parameters from the training data by maximum likelihood.

Then compute the log-likelihood of the training dataset given the structure and parameters.

Score = log-likelihood – λk, where λ is a constant and k is the total number of parameters.

Bayes Net structure search

How to automatically find the Bayesian network structure that best fits the data?



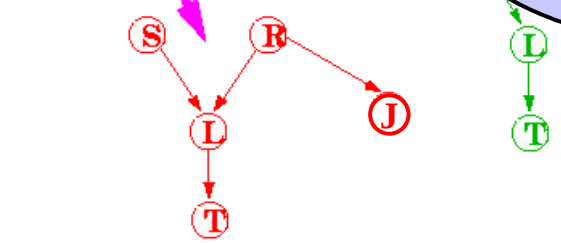
This "score-based" learning approach can find Bayes Net structures that accurately capture the conditional independence relationships in the data. But what if we want to be able to interpret edges **causally**?

<u>One option</u>: incorporate prior knowledge. The "K2" structure learning algorithm is a hill-climbing approach that relies on a **causal partial ordering** of the variables and only allows edges from X_i to X_j for $i < j$.

…od.

…g-likelihood …taset given the structure and parameters.

Score = log-likelihood – λk, where λ is a constant and k is the total number of parameters.

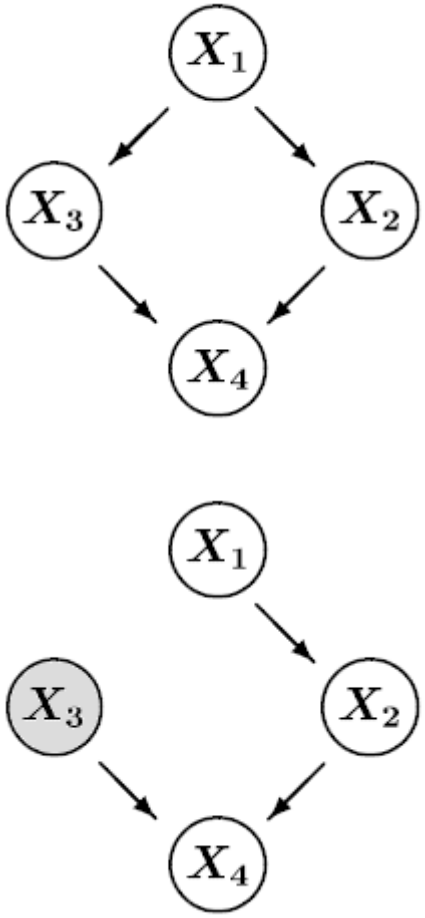


Causal Bayesian Networks

- CBN = BN, where edge X → Y is assumed to indicate that X is a direct cause of Y.
- Markov condition: given its parents (causes), each variable is conditionally independent of its non-descendents (non-effects).
  - $Pr(X_1..X_N) = \prod Pr(X_i \mid Parents(X_i))$
  - All this is just like a regular Bayes Net.
- We can also reason about interventions:

  $Pr(X_i \mid Parents(X_i), do(X = x)) = 1\{X_i = x_i\}$ for intervened variables ($X_i = x_i$ in X)

  $Pr(X_i \mid Parents(X_i), do(X = x)) = Pr(X_i \mid Parents(X_i))$ for non-intervened variables (X_i not in X).

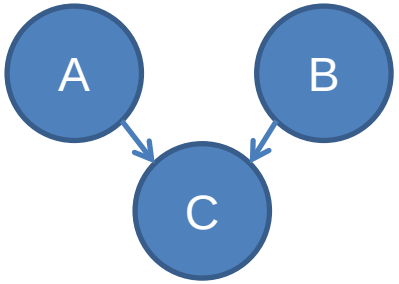


After intervention $do(X_3 = x_3)$

Causal Structure Learning from Observational Data

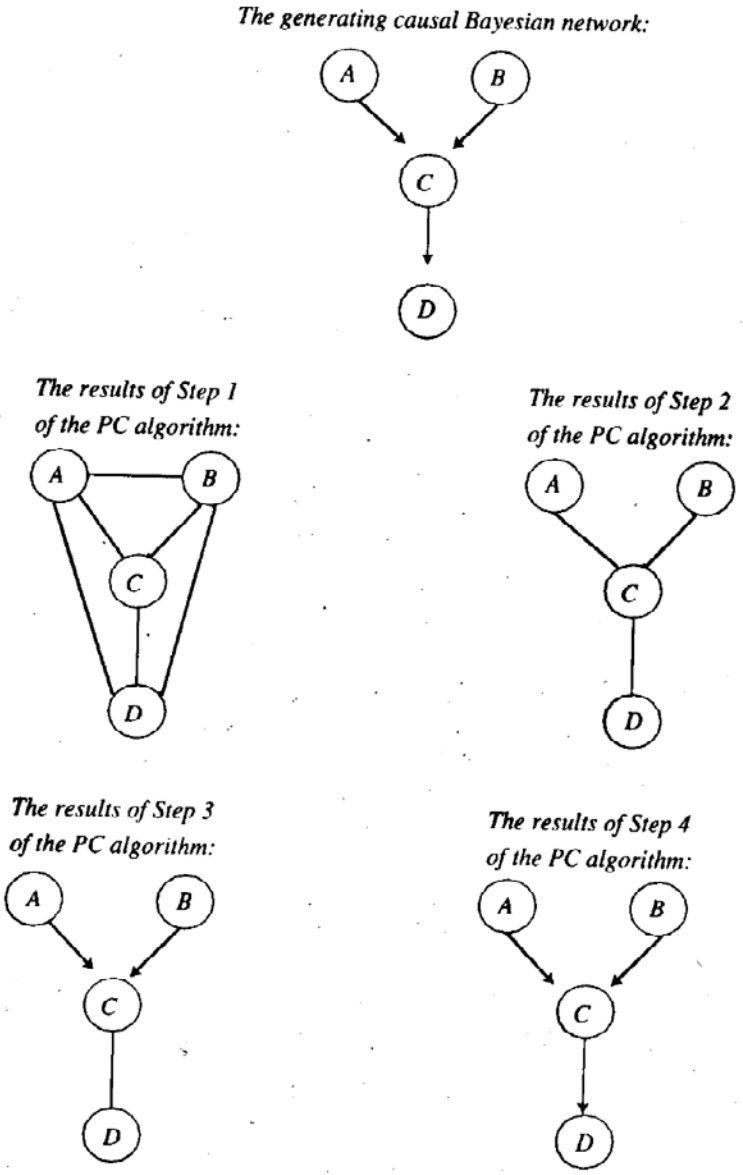
- Key thing to keep in mind: we cannot distinguish between networks which have same conditional independence relationships but different causation.

- We get an equivalence class of structures (some edges may be directed, some undirected).
- If we want to do better, need prior knowledge, additional assumptions, or different data (time series, intervention).
- How can we ever get a directed edge?
  - Answer: V-structures!
  - CI(A,B) but CD(A,B | C)



Constraint-Based Structure Learning

- Relies on results of statistical tests for conditional independence between variables; finds an equivalence class of structures satisfying these constraints.
- PC algorithm (Spirtes et al.)
  - Start with complete undirected graph.
  - For each pair of variables X and Y, delete edge if they are conditionally independent given any subset of the other vars.
  - For any triplet X – Y – Z without X – Z, if X and Z are conditionally dependent given Y (and any subset of other vars), replace with V-structure X → Y ← Z.
  - Use this information to direct other edges (avoid creating directed cycles and additional V-structures).



Assumptions made by PC

(and most other causal learning methods)

- **Causal Markov:** a variable is probabilistically independent of its non-descendants (non-effects) conditional on its direct causes.
  - Permits inference from dependence to causal connection.
- **Causal Faithfulness:** conditional independence between variables does not occur by accident (e.g., via "canceling out" settings of parameters), but only because of the lack of a (direct) causal relationship.
  - Permits inference from independence to causal separation.
- **Causal Sufficiency:** no unmeasured common causes
- **Acylicity:** no variable is an (indirect) cause of itself.
  - Would be violated, for example, if X → Y, Y → Z, and Z → X.

(Some text on this slide was borrowed from Fredrick Eberhardt's presentation, "All of Causal Discovery", and from Stephen Fancsali's doctoral dissertation.)

Assumptions made by PC

(and most other causal learning methods)

- **Causal Markov:** a
  inde
  co

Two ways to proceed from here:

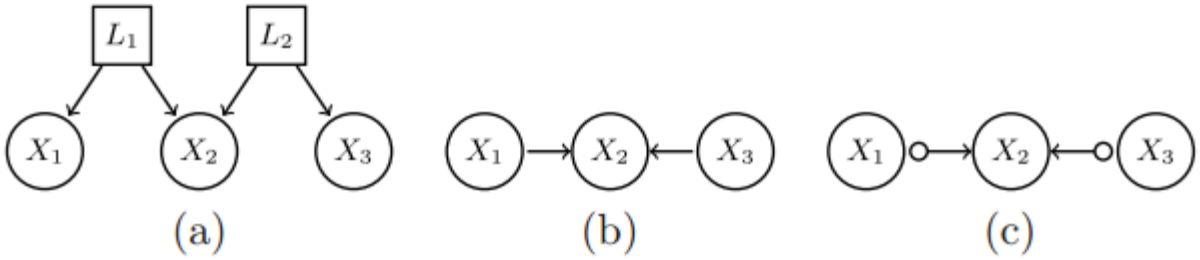
Weaker assumptions, such as allowing unmeasured common causes, lead to a larger equivalence class of structures. (Fewer edges can be oriented)

Stronger assumptions, such as parametric model assumptions, lead to a smaller equivalence class of structures . (More edges can be oriented)

- **Cau** causes
- **Ac** cause of itself.
  - be violated, for example, if X → Y, Y → Z, and Z → X.

(Some text on this slide was borrowed from Fredrick Eberhardt's presentation, "All of Causal Discovery", and from Stephen Fancsali's doctoral dissertation.)

Fast Causal Inference (FCI)



Like PC, but handles selection bias and unobserved confounders.

Fewer assumptions (does not require causal sufficiency).
But results in a larger equivalence class of structures.

Learns a **partial ancestral graph** (PAG): A → B means A is an ancestor of B.
Can sometimes distinguish this from unobserved common cause A ← X → B.
Cannot ever rule out unobserved intervening causes A → X → B.

Not very fast, doesn't scale to many variables.

(Some text and the figure on this slide were borrowed from Seth Flaxman's presentation, "Algorithmic Approaches to Causal Inference")

Causal orientation methods

With **additional parametric model assumptions**, can distinguish between X → Y and Y → X even without the presence of a third variable. These methods work by exploiting **asymmetries** in the shapes of the conditional probability densities.

Statnikov et al. (2012) does a big bake-off to compare many of these methods for a genomics application (X: transcription factor; Y: target gene).

Example 1: LiNGaM (assumes **linear** model with **non-Gaussian** errors)

Estimate models $Y = bX + \varepsilon$ and $X = b'Y + \varepsilon'$, where ε and ε' are independent.
Choose direction with smaller slope b.

Example 2: ANM (assumes **non-linear** model with **additive noise**)

If x and y are dependent:

- Estimate residuals from non-linear regression $y = f(x) + \varepsilon$
- Check whether residuals and x are independent
- Independent? Accept model x → y

Repeat with x and y switched.

(Some text on this slide was borrowed from Seth Flaxman's presentation, "Algorithmic Approaches to Causal Inference")

The many uses of Bayes Nets

Bayes Nets provide a useful graphical representation of the probabilistic (+ causal) relationships between variables.

Automatic learning of Bayes net structure can be used for exploratory analysis of datasets with many attributes.

We can often improve the performance of model-based classification by moving from Naïve Bayes to Bayes Nets.

We can also use Bayes Nets to detect **anomalies**, by finding points with low probabilities given the Bayes Net.

Bayes Nets provide a compact structure which enables us to efficiently compute probability distributions for any unobserved variables given observations of other variables.

Medical diagnosis Failure troubleshooting Drug discovery

Environmental modeling Computational biology

References

Bayes Nets:

- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- S. Russell and P. Norvig. *AI: A Modern Approach*, Ch. 15.
- Several excellent tutorials on Bayes Nets available at http://www.autonlab.org/tutorials.

Causality:

- P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2000.
- J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009.
- P. Spirtes. Introduction to causal inference. *Journal of Machine Learning Research* 11: 1643-1662, 2010.
- M. Kalish, P. Buhlmann. Causal structure learning and inference: a selective review. *Qual Technol Quant Manag.* 11:3–21, 2014.
- A. Statnikov et al. New methods for separating causes from effects in genomics data. *BMC Genomics* 2012, 13(Suppl 8):S22. http://www.biomedcentral.com/1471-2164/13/S8/S22