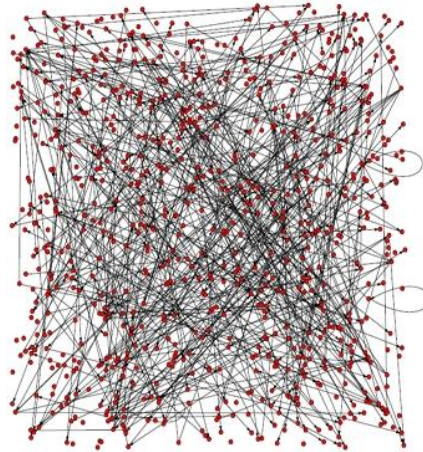


L10. Bayesian Network



Probability + **Statistics** + **Graph Theory**

L7. Bayesian Network (Modeling)

- (Static) Bayesian Network
- Dynamic Bayesian Network

(Static) Bayesian Network

Degree of Belief and Probability

How to compare the plausibility of different statements?



G : “we can be a billionaire if we go to graduate school”

vs

S : “we can be a billionaire if we go to Samsung”



- If you believe G more than S , you can write $G \succ S$
- If you believe S more than G , you can write $G \prec S$
- If you have the same belief, you can write $G \sim S$

Assumptions about relationships of \succ and \sim

- *Universal comparability* : either $G \succ S$, $G \prec S$ or $G \sim S$
- *Transitivity* : if $G \succ S$ and $S \succ V$, then $G \succ V$

Due to the two assumptions, the **degree of belief** can be represented by a real-valued function:

- $P(G) > P(S)$ if and only if $G \succ S$
- $P(G) = P(S)$ if and only if $G \sim S$

We are going to use very simple probability theories to construct Probabilistic Graphical Model

- conditional probability :

$$P(A|B) = \frac{P(B|A)}{P(B)}$$

- Law of total probability :

$$P(A) = \sum_{B \in \mathcal{B}} P(A|B) P(B) \quad \text{or} \quad P(A) = \int_{\mathcal{B}} P(A|B) P(B) dB$$

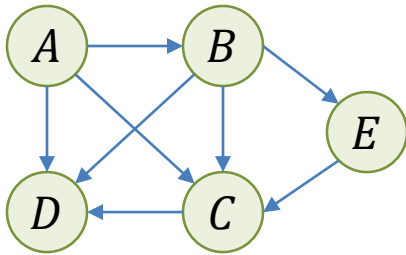
- Bayes' rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

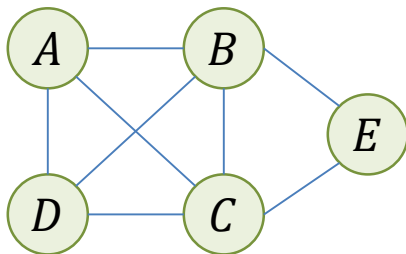
Introduction to Graph Theory

Graph

- A graph G consists of nodes (also called vertices) and edges (also called links) between the nodes.



A directed graph G consists of directed edges between nodes

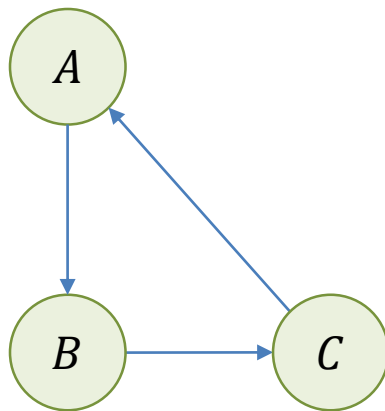


An undirected graph G consists of undirected edges between nodes

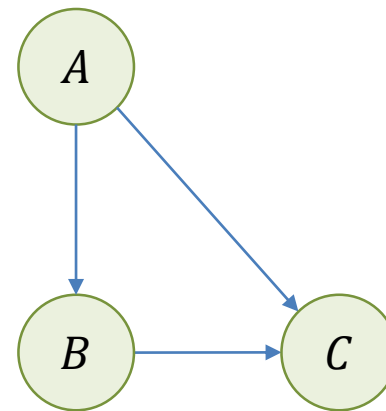
Directed Acyclic Graph (DAG)

- A DAG is a graph G with directed edges (arrows on each link) between the nodes such that by following a path of nodes from one node to another along the direction of each edge no path will revisit a node.

Cyclic Graph



Acyclic Graph



- DAG will play a central role in modeling environments with many variables
 - will be used for the belief networks
 - can encode the direction dependence between the parent nodes and child nodes.

Introduction to Graph Theory

Path

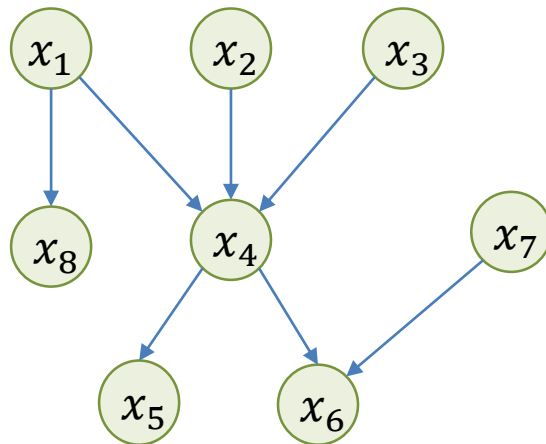
- A path $A \rightarrow B$ from node A to node B is a sequence of nodes that connects A to B

Ancestors

- In directed graph, the nodes A such that $A \rightarrow B$ and $B \nrightarrow A$ are the ancestors of B

Descendants

- In directed graph, the nodes B such that $A \rightarrow B$ and $B \nrightarrow A$ are the descendants of A



Representations

- Edge list

$$L = \{(x_1, x_4), (x_2, x_4), (x_3, x_4), (x_1, x_8), (x_4, x_5), (x_4, x_6), (x_7, x_6)\}$$

- Adjacency matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- ✓ A path $x_1 \rightarrow x_6$ is $x_1 \rightarrow x_4 \rightarrow x_6$
- ✓ The ancestors of x_6 are $ac(x_6) = \{x_1, x_2, x_3, x_4\}$
- ✓ The descendants of x_2 are $dc(x_2) = \{x_4, x_5, x_6\}$
- ✓ The parents of x_4 are $pa(x_4) = \{x_1, x_2, x_3\}$
- ✓ The children of x_4 are $ch(x_4) = \{x_5, x_6\}$

Motivation of Bayesian Network

Full Joint Distribution

Example distribution

A	B	C	$P(A, B, C)$
0	0	0	0.08
0	0	1	0.15
0	1	0	0.05
0	1	1	0.10
1	0	0	0.14
1	0	1	0.18
1	1	0	0.19
1	1	1	0.11

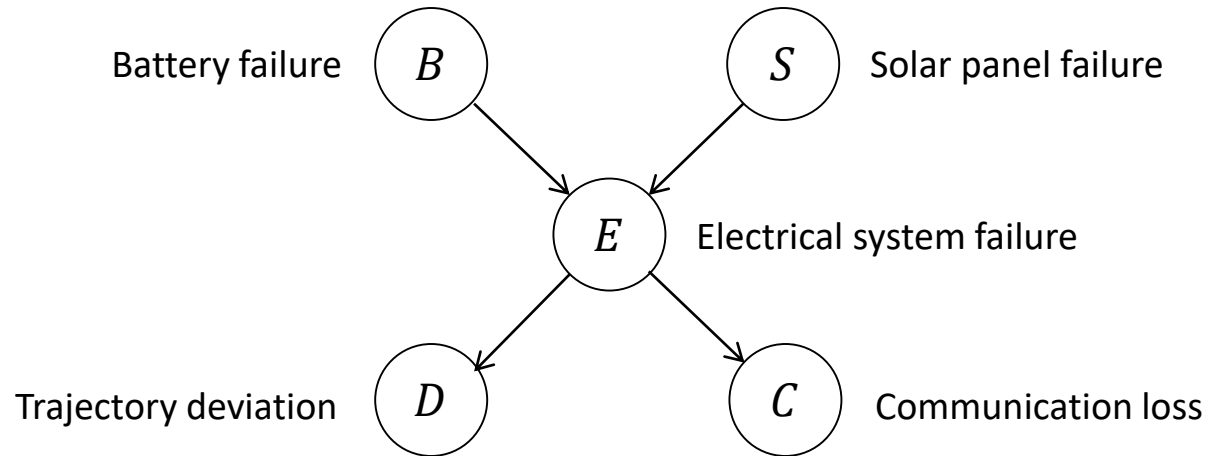
- Binary variables: A, B, C (e.g., $A = 1$ or 0)
- 2^3 entities are required to construct the table
- $2^3 - 1$ independent parameters are required to fully specify the joint probability distribution
- $2^N - 1$ parameters are required for N binary variables
- If each variable has M different choices, $M^N - 1$ parameters are required

The number of parameters grows exponentially

→ **Difficult to represent Probability distribution and learn the parameters from data**

Motivation of Bayesian Network

Full Joint Probability Distribution



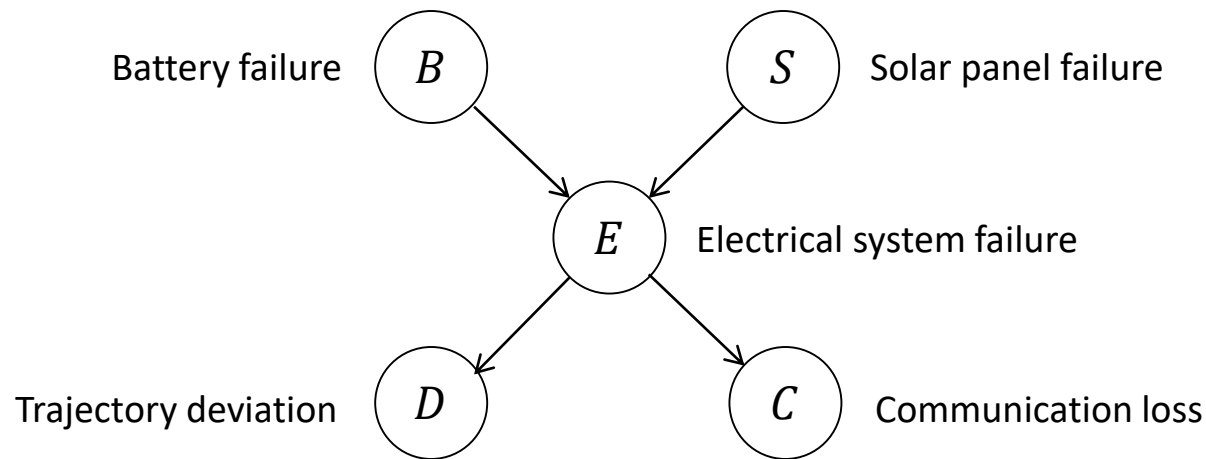
- Binary variables: B, S, E, D, C (e.g., $B = 1$ or 0)
- 2^5 entities are required to construct the table
- $2^5 - 1$ independent parameters are required to fully specify the joint probability distribution
- $2^N - 1$ parameters are required for N binary variables
- If each variable has M different choices,
 $M^N - (M - 1)$ parameters are required

The number of parameters grows exponentially

→ **Difficult to represent Probability distribution and learn the parameters from data**

Motivation of Bayesian Network

A Bayesian network is a compact representation of a joint distribution

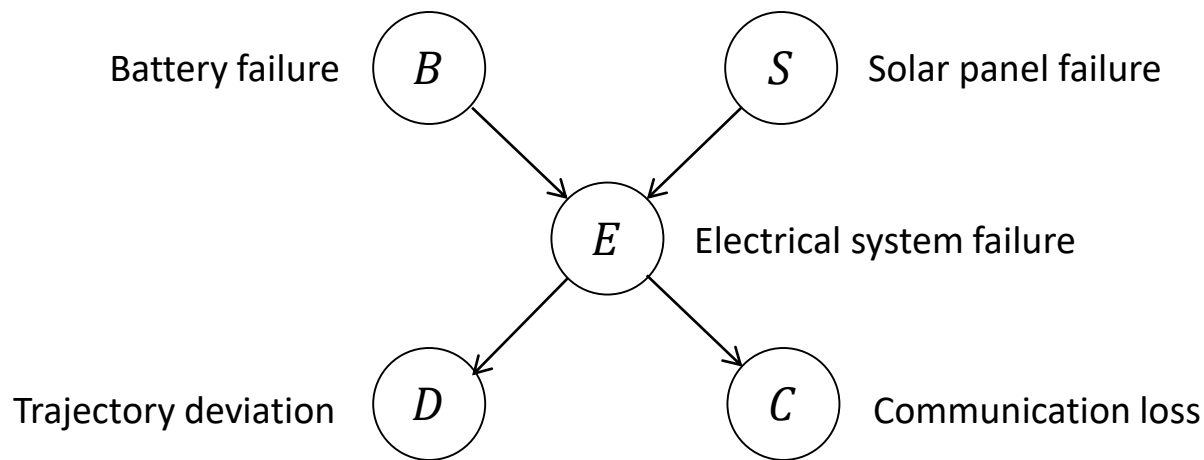


Probability + **Statistics** + **Graph Theory**

- A Bayesian Network introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables. For inferencing, it uses statistics
- Provide a good representation to model the probabilistic structures between random variables.
 - Nodes represent random variables
 - Edges represent probabilistic dependency, namely conditional probability among variables
- Conditional independence described by the graph, greatly reduces the computational effort to learn the model and inferencing random variables.

Motivation of Bayesian Network

A Bayesian network is a compact representation of a joint distribution



- Each node corresponds to a random variable
- Directed edges connect pairs of nodes, indicating direct probabilistic relationships
- $P(x_i | \text{pa}_{x_i})$ represents the probability distribution of x_i conditional on the parent nodes pa_{x_i} of X_i e.g., $P(E|B,S)$: *B* and *S* are the parent nodes of *E*

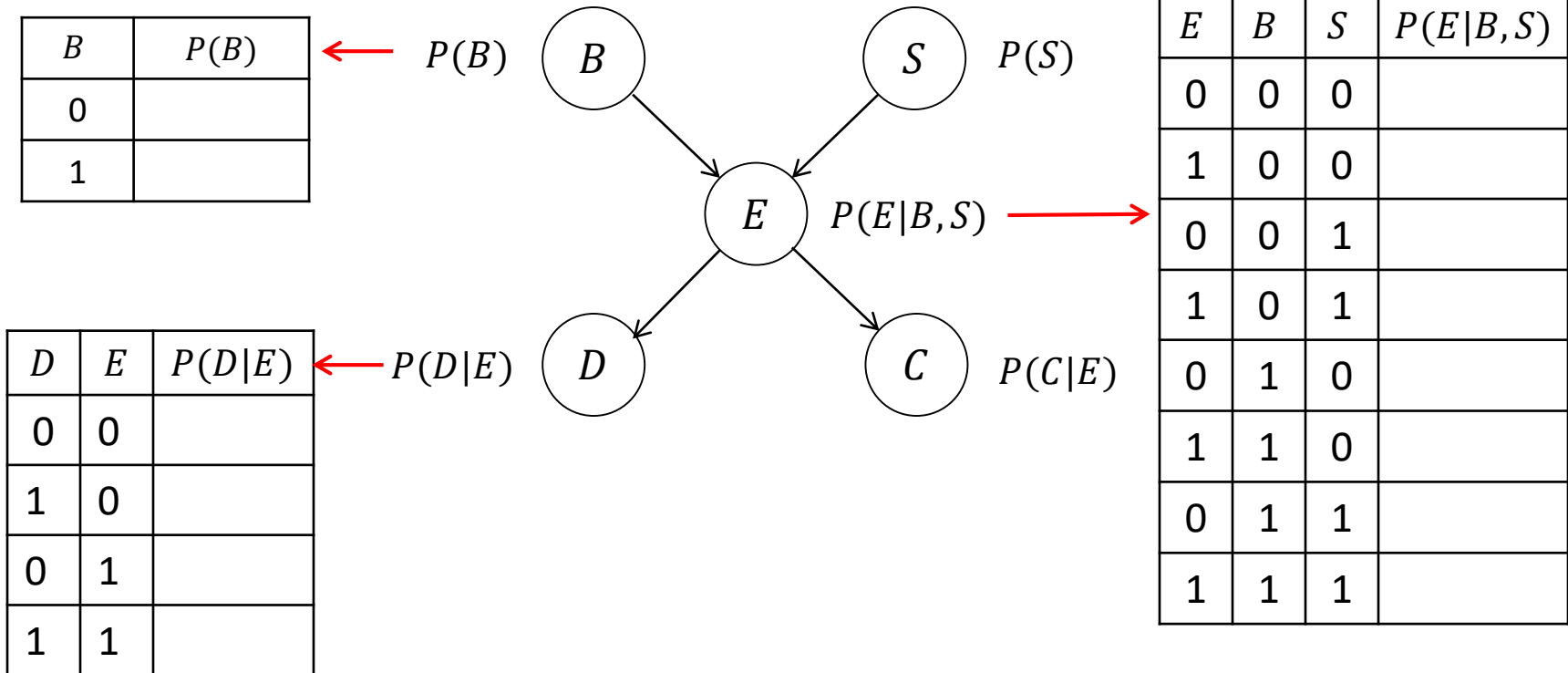
The chain rule for Bayesian networks specifies how to construct a joint distribution from the local conditional probability distribution

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{pa}_{x_i})$$

local conditional probability distribution

Motivation of Bayesian Network

A Bayesian network is a compact representation of a joint distribution



- Chain rule: $P(B, S, E, D, C) = P(B)P(S)P(E|B, S)P(D|E)P(C|E)$
- Required independent parameters to fully specify the joint PDF

$P(B) : 1, P(S) : 1, P(E|B, S) : 4, P(D|E) : 2, P(C|E) : 2$ (total 10 compared to $2^5 - 1 = 31$)

Bayesian network can greatly reduce the number of parameters

Formal Definition Bayesian Network

- A Bayesian network (BN) is a distribution of the form

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | \text{pa}_{x_i})$$

- ✓ pa_{x_i} represents the parental variables of variable x_i
 - ✓ BN is represented as a directed acyclic graph with an arrow pointing from a parent variable to child variable
- Every probability distribution can be written as a BN:

$$\begin{aligned} p(x_1, \dots, x_n) &= p(x_n | x_1, \dots, x_{n-1}) p(x_1, \dots, x_{n-1}) \\ &= p(x_n | x_1, \dots, x_{n-1}) p(x_{n-1} | x_1, \dots, x_{n-2}) p(x_1, \dots, x_{n-2}) \\ &= p(x_1) \prod_{i=2}^n p(x_i | \text{pa}_{x_i}) \end{aligned}$$

- The particular role of BN is that the structure of the DAG corresponds to a set of **conditional independence assumptions**, namely which ancestral parental variables are sufficient to specify each conditional probability table

Definition : Independence

$$X \perp Y$$

$$p(X, Y) = p(X)p(Y) \text{ for all states of } X, Y$$

or equivalently $P(X|Y) = P(X)$

Definition : Conditional Independence

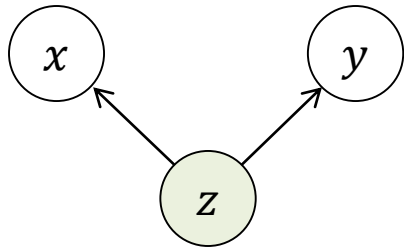
$$X \perp Y|Z$$

$$p(X, Y|Z) = p(X|Z)p(Y|Z) \text{ for all states of } X, Y, Z$$

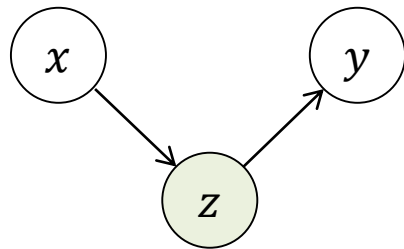
or equivalently $P(X|Y, Z) = P(X|Z)$

- ✓ The two sets of variables X and Y are independent of each other provided we know the state of the set of variables Z
- ✓ The information of Y does not give further information on X

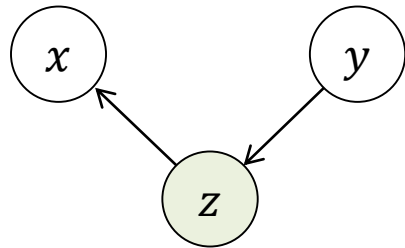
V-structure (or collider)



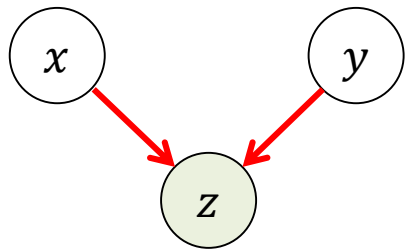
$$p(x, y|z) = \frac{p(x, y, z)}{p(z)} = \frac{p(z)p(x|z)p(y|z)}{p(z)} = p(x|z)p(y|z)$$



$$\begin{aligned} p(x, y|z) &= \frac{p(x, y, z)}{p(z)} = \frac{p(x)p(z|x)p(y|z)}{p(z)} \\ &= \frac{p(x, z)p(y|z)}{p(z)} = p(x|z)p(y|z) \end{aligned}$$



$$\begin{aligned} p(x, y|z) &= \frac{p(x, y, z)}{p(z)} = \frac{p(y)p(z|y)p(x|z)}{p(z)} \\ &= \frac{p(y, z)p(x|z)}{p(z)} = p(y|z)p(x|z) \end{aligned}$$

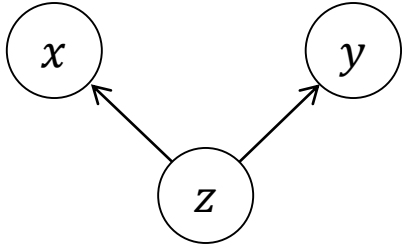


$$p(x, y|z) = \frac{p(x, y, z)}{p(z)} = \frac{p(x)p(y)p(z|x, y)}{p(z)} \neq p(y|z)p(x|z)$$

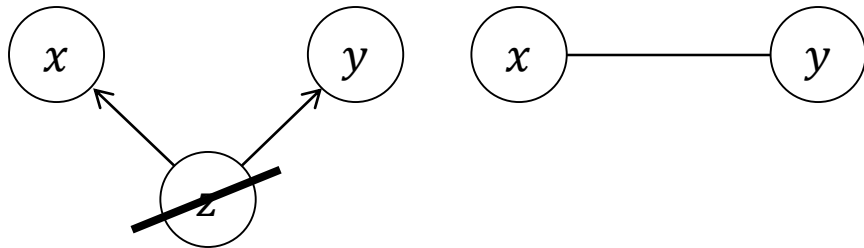
- BN with $x \rightarrow z \leftarrow y$
- ✓ x and y are unconditionally independent
 - ✓ x and y are dependent conditional on z

V-structure (or collider)

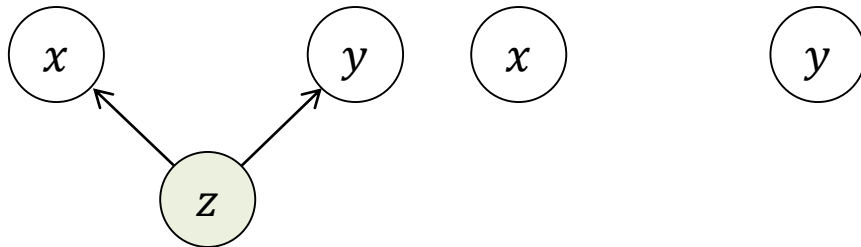
$$p(x, y, z) = p(x|z)p(y|z)$$



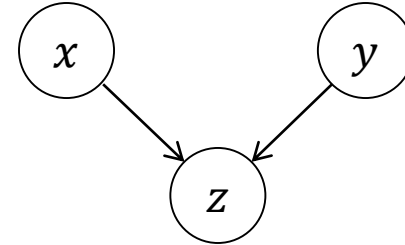
Marginalization over z



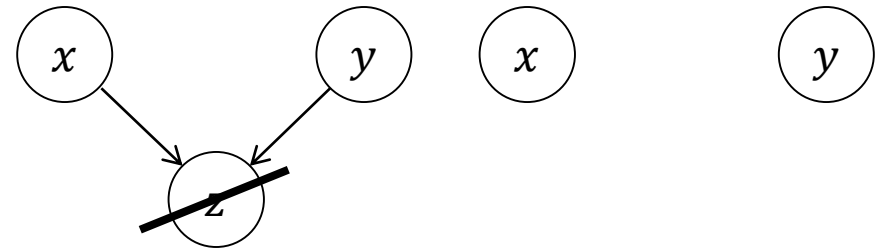
Conditionalization on z



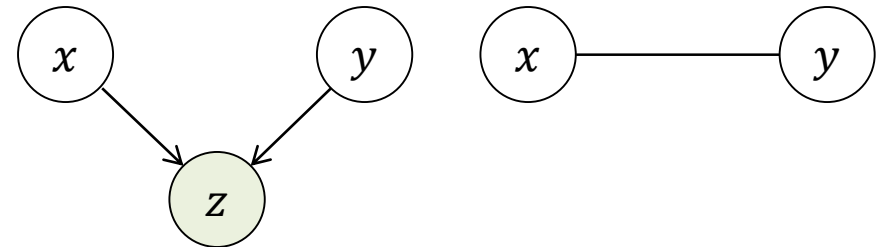
$$p(x, y, z) = p(z|x, y)p(x)p(y)$$



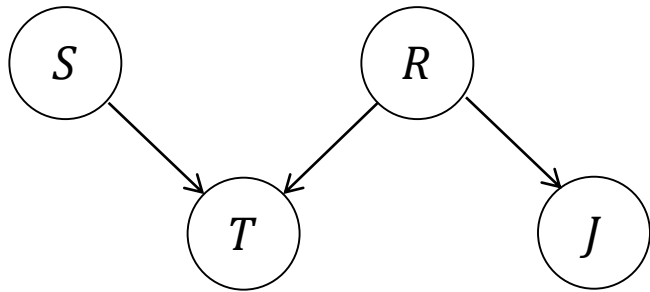
Marginalization over z



Conditionalization on z



Example : Wet Grass



$R \in \{0,1\} : R = 1$ means that it has been raining

$S \in \{0,1\} : S = 1$ Sprinkler is turned on

$J \in \{0,1\} : J = 1$ Jack's grass is wet

$T \in \{0,1\} : T = 1$ Tracey's grass is wet

Joint distribution based on chain rule

$$p(T, J, R, S) = p(T|J, R, S)p(J, R, S)$$

$$= p(T|J, R, S)p(J|R, S)p(R, S)$$

$$= p(T|J, R, S)p(J|R, S)p(R|S)p(S)$$

$$8 + 4 + 2 + 1 = 2^4 - 1 = 15 \text{ parameters are required}$$

Joint distribution conditional independence

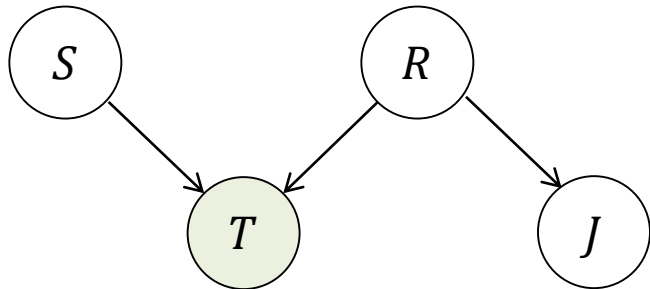
$$p(T, J, R, S) = p(T|J, R, S)p(J|R, S)p(R|S)p(S)$$

$$= p(T|R, S) \times p(J|R) \times p(R) \times p(S)$$

$$= p(T|R, S)p(J|R)p(R)p(S)$$

Example : Wet Grass

Modeling



$R \in \{0,1\} : R = 1$ means that it has been raining

$S \in \{0,1\} : S = 1$ Sprinkler is turned on

$J \in \{0,1\} : J = 1$ Jack's grass is wet

$T \in \{0,1\} : T = 1$ Tracey's grass is wet

$$p(T, J, R, S) = p(T|R, S)p(J|R)p(R)p(S)$$

$p(T|S, R)$

Tracey's Grass wet=1	Rain	Sprinkler
1	1	1
1	1	0
0.9	0	1
0	0	0

$p(J|R)$

Jack's Grass wet=1	Rain
1	1
0.2	0

$$p(S = 1) = 0.1$$

$$p(R = 1) = 0.2$$

The tables and graphical structure fully specify the distribution

Example : Wet Grass

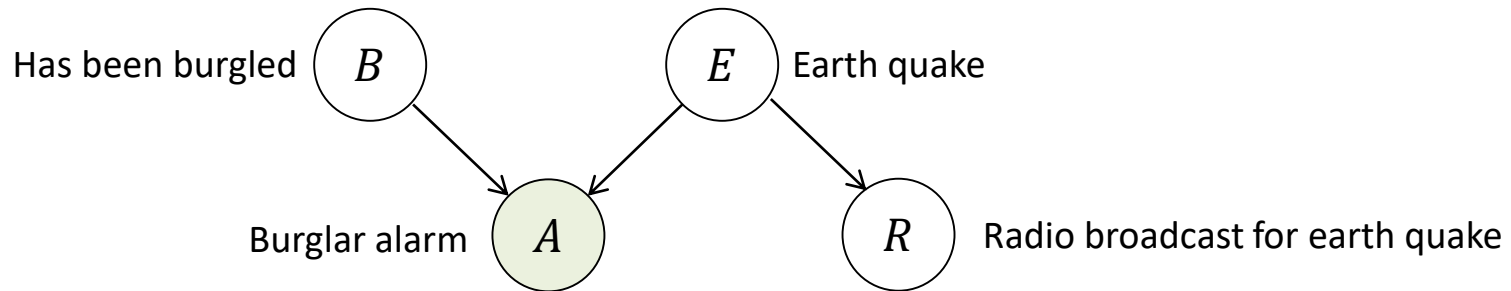
Inference

$$\begin{aligned} p(S = 1|T = 1) &= \frac{p(S = 1, T = 1)}{p(T = 1)} = \frac{\sum_{J,R} p(T = 1, J, R, S = 1)}{\sum_{J,R,S} p(T = 1, J, R, S)} \\ &= \frac{\sum_{J,R} p(J|R)p(T = 1|R, S = 1)p(R)p(S = 1)}{\sum_{J,R,S} p(J|R)p(T = 1|R, S)p(R)p(S)} \\ &= \frac{\sum_R p(T = 1|R, S = 1)p(R)p(S = 1)}{\sum_{R,S} p(T = 1|R, S)p(R)p(S)} \quad \because \sum_J p(J|R) = 1 \\ &= \frac{0.9 \times 0.8 \times 0.1 + 1 \times 0.2 \times 0.1}{0.9 \times 0.8 \times 0.1 + 1 \times 0.2 \times 0.1 + 0 \times 0.8 \times 0.9 + 1 \times 0.2 \times 0.9} = 0.3382 \end{aligned}$$

$$\begin{aligned} p(S = 1|T = 1, J = 1) &= \frac{p(S = 1, T = 1, J = 1)}{p(T = 1, J = 1)} \\ &= \frac{\sum_R p(T = 1, J = 1, R, S = 1)}{\sum_{R,S} p(T = 1, J = 1, R, S)} \\ &= \frac{\sum_R p(J = 1|R)p(T = 1|R, S = 1)p(R)p(S = 1)}{\sum_{R,S} p(J = 1|R)p(T = 1|R, S)p(R)p(S)} \\ &= \frac{0.0344}{0.2144} = 0.1604 \end{aligned}$$

The fact that Jack's grass is also wet increases the chance that the rain has played a role in making Tracey's grass wet

Example : Burglar Alarm



$$p(B, E, A, R) = p(A|B, E)p(R|E)p(E)p(B)$$

$$p(A|B, E)$$

Alarm=1	Burglar	Earthquake
0.9999	1	1
0.99	1	0
0.99	0	1
0.0001	0	0

$$p(R|E)$$

Radio=1	Earthquake
1	1
0	0

$$p(E = 1) = 0.01$$

$$p(E = 0) = 0.99$$

$$\begin{aligned}
 p(B = 1|A = 1) &= \frac{p(B, A = 1)}{p(A = 1)} = \frac{\sum_{E,R} p(B = 1, E, A = 1, R)}{\sum_{B,E,R} p(B, E, A = 1, R)} \\
 &= \frac{\sum_{E,R} p(A = 1|B = 1, E)p(R|E)p(E)p(B = 1)}{\sum_{B,E,R} p(A = 1|B, E)p(R|E)p(E)p(B)} \approx 0.99
 \end{aligned}$$

$$p(B = 1|A = 1, R = 1) \approx 0.01$$

Conditional Independence

What causes the number of parameters to be reduced?

→ **The conditional independence assumptions** encoded by the structure of a Bayesian network

- X and Y are independent if and only if

$$P(X, Y) = P(X)P(Y)$$

or equivalently $P(X|Y) = P(X)$

$$\because P(X|Y) = \frac{P(X, Y)}{P(Y)} = \frac{P(X)P(Y)}{P(Y)} = P(X)$$

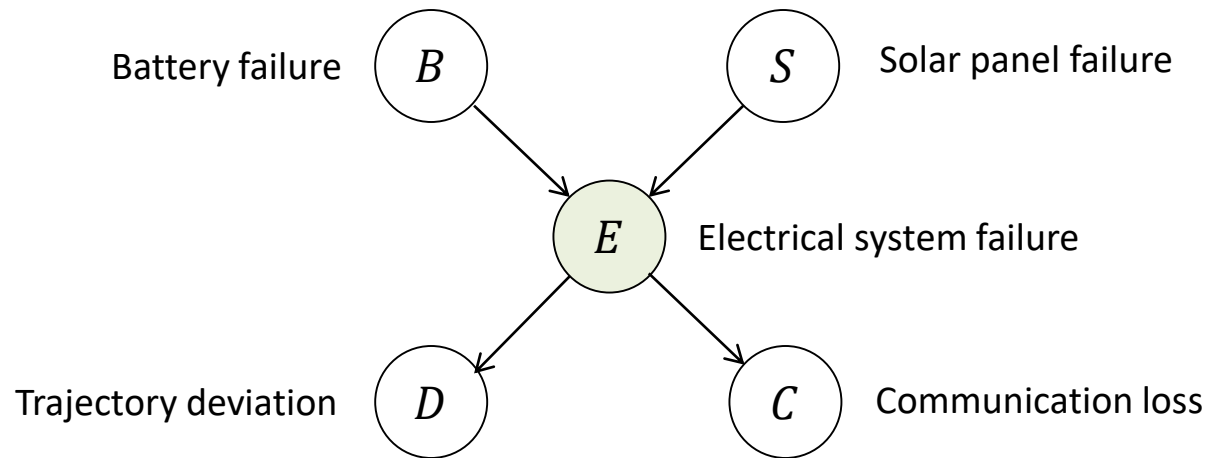
- X and Y are conditionally independent given Z if and only if

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$

or equivalently $P(X|Z) = P(X|Y, Z)$

Independence assumptions reduce the number of parameters used to represent a joint pdf

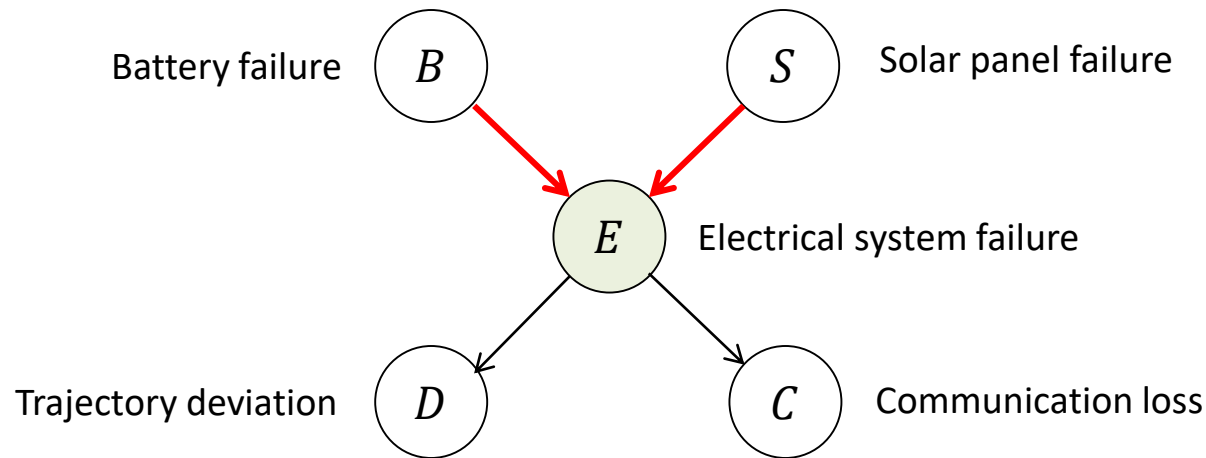
Conditional Independence examples



- C is independent of B given E : $(C \perp B|E)$
 - Information about Battery failure does not affect my belief on communication loss if I already know (observed) the status of electrical system failure
- D is independent of S given E : $(D \perp S|E)$
 - Information about Solar failure does not affect my belief on a trajectory deviation if I already know (observed) the status of electrical system failure

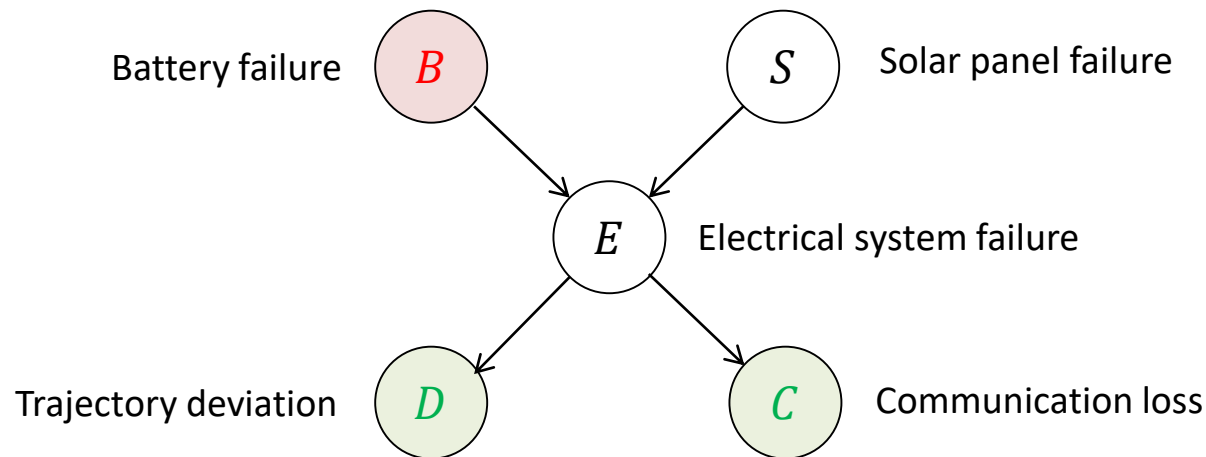
Conditional Independence examples

V-structure



- B is independent S (E is not observed)
→ Knowing there is a battery failure does not affect my belief regarding solar panel failure
- B is dependent S **given E**
→ If there was an electrical system failure (observed) and there was no battery failure, there it is likely that a solar panel fails
- Influence flows only through $B \rightarrow E \leftarrow S$ when E is known

Inference



- Once a joint probability distribution is constructed, inference can be performed to determine the distribution over one or more unobserved variables given the values associated with a set of observed variables

$P(B|d^1, c^1)$ Probability distribution of Battery failure

Query variable

given the trajectory deviation and the communication loss

Evidence variable

E *S* : Hidden variables

Inference

How to compute $P(B|d^1, c^1)$?

Exact inference

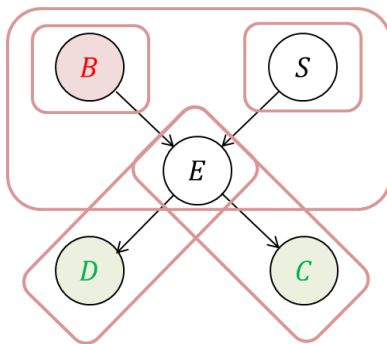
$$\begin{aligned} P(b^1|d^1, c^1) &\propto \sum_s \sum_e P(b^1, s, e, d^1, c^1) \\ &= \sum_s \sum_e P(b^1)P(s)P(e|b^1, s)P(d^1|e)p(c^1|e) \quad \text{By conditional independence} \\ &= P(b^1) \sum_e P(d^1|e)p(c^1|e) \sum_s P(s)P(e|b^1, s) \end{aligned}$$

The number of terms to be added together can grow exponentially with the number of hidden variables

Inference

How to compute $P(B|d^1, c^1)$?

Variable Elimination



Conditional distributions are represented by the following tables

$$T_1(B)T_2(S)T_3(E, B, S)T_4(d^1, E)T_5(c^1, E)$$

$$T_1(B)T_2(S)T_3(E, B, S)T_6(E)T_7(E)$$

Observe evidence (d^1 and c^1)

$$T_1(B)T_2(S)T_8(B, S)$$

$$T_8(B, S) = \sum_e T_3(e, B, S)T_6(e)T_7(e)$$

$$T_1(B)T_9(B)$$

$$T_9(B) = \sum_s T_2(s)T_8(B, s)$$

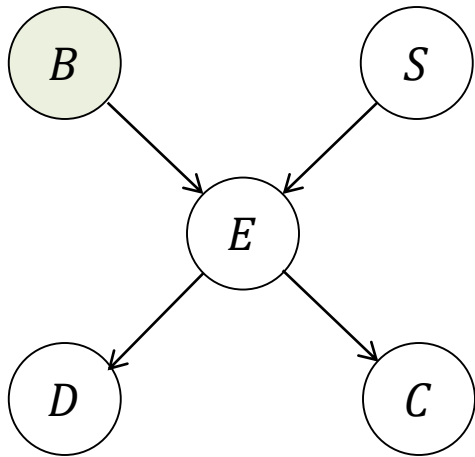
Normalizing the product of the two factors ($T_1(B)$ and $T_9(B)$) results in $P(B|d^1, c^1)$

Variable elimination algorithm relies on **heuristic ordering** of variables to eliminate in sequence
→ Often linear but sometimes exponential

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



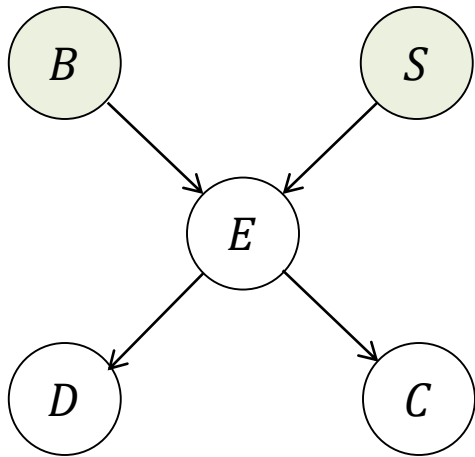
Sample from $P(B)$

B	S	E	D	C
1				

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



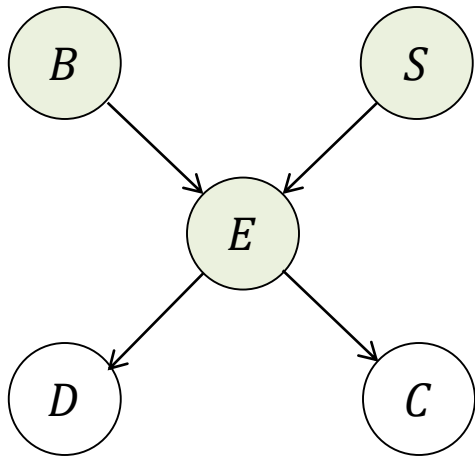
Sample from $P(S)$

B	S	E	D	C
1	1			

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



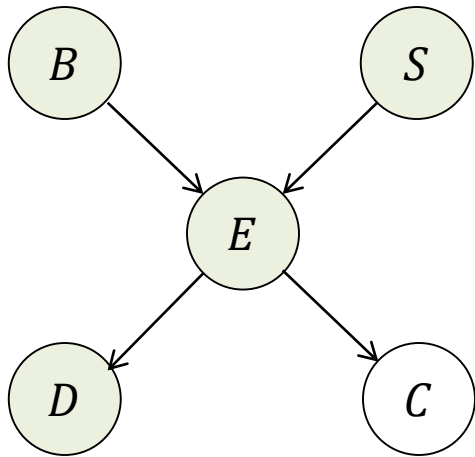
B	S	E	D	C
1	1	1		

Sample from $P(E|B = 1, S = 1)$

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



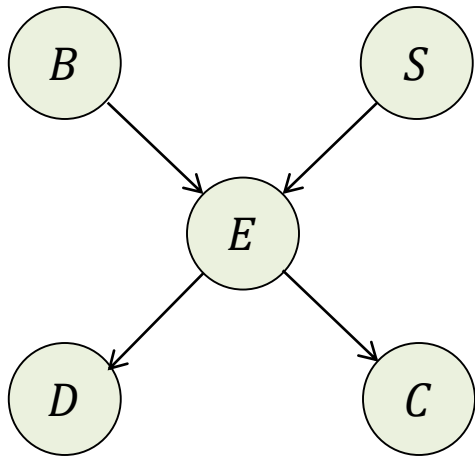
B	S	E	D	C
1	1	1	0	

Sample from $P(D|E = 1)$

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



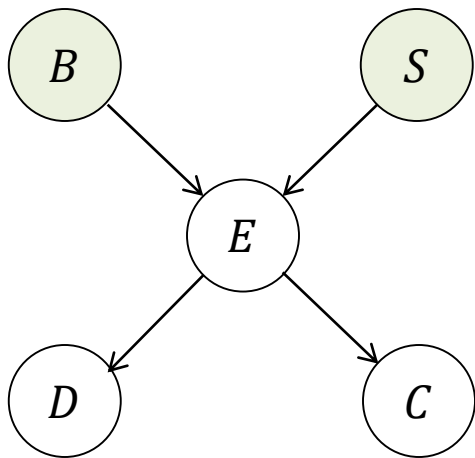
B	S	E	D	C
1	1	1	0	0

Sample from $P(C|E = 1)$

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



$$P(b^1|d^1, c^1) = 1/3$$

$$P(b^0|d^1, c^1) = 2/3$$

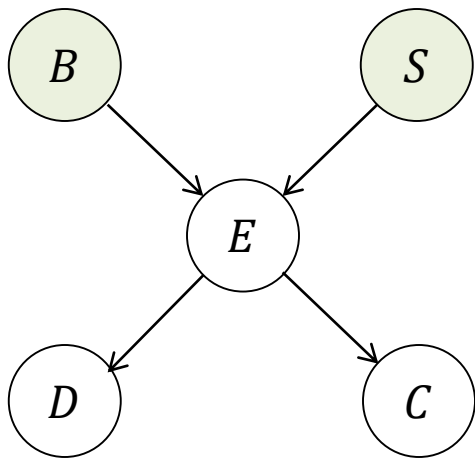
B	S	E	D	C
1	1	1	0	0
0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
0	1	1	1	1
0	1	0	0	1
0	0	0	1	0
0	1	1	1	0
0	1	0	1	1

Three cases coincide observations d^1, c^1

Inference

How to compute $P(B|d^1, c^1)$?

Approximate inference (Sampling based methods)



$$P(b^1|d^1, c^1) = 1/3$$

$$P(b^0|d^1, c^1) = 2/3$$

B	S	E	D	C
1	1	1	0	0
0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
0	1	1	1	1
0	1	0	0	1
0	0	0	1	0
0	1	1	1	0
0	1	0	1	1

Three cases coincide observations d^1, c^1

If likelihood of evidence is small, then many samples are required!!

Inference

How to compute $P(B|d^1, c^1)$?

Likelihood sampling

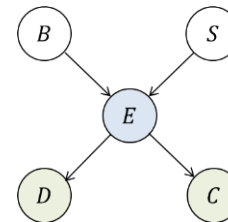
B	S	E	D	C
1	1	1	0	0
0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
0	1	1	1	1
0	1	0	0	1
0	0	0	1	0
0	1	1	1	0
0	1	0	1	1

Algorithm 2.5 Likelihood-weighted sampling from a Bayesian network

```

1: function LIKELIHOODWEIGHTEDSAMPLE( $B, o_{1:n}$ )
2:    $X_{1:n} \leftarrow$  a topological sort of nodes in  $B$ 
3:    $w \leftarrow 1$ 
4:   for  $i \leftarrow 1$  to  $n$ 
5:     if  $o_i = \text{NIL}$ 
6:        $x_i \leftarrow$  a random sample from  $P(X_i \mid \text{pa}_{x_i})$ 
7:     else
8:        $x_i \leftarrow o_i$ 
9:        $w \leftarrow w \times P(x_i \mid \text{pa}_{x_i})$ 
10:  return  $(x_{1:n}, w)$ 

```



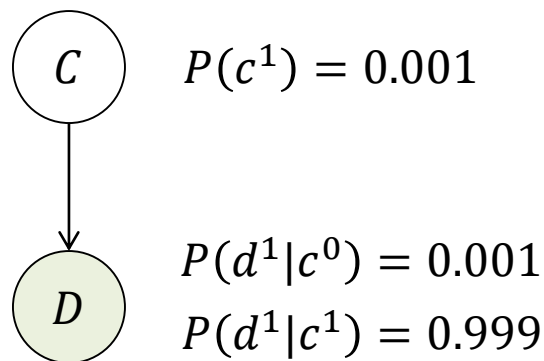
B	S	E	D	C	weight
1	0	1	1	1	$P(d^1 e^1) P(c^1 e^1)$
0	1	1	1	1	$P(d^1 e^1) P(c^1 e^1)$
0	1	0	1	1	$P(d^1 e^0) P(c^1 e^0)$

$$P(b^1|d^1, c^1) = \frac{P(d^1|e^1) P(c^1|e^1)}{P(d^1|e^1) P(c^1|e^1) + P(d^1|e^1) P(c^1|e^1) + P(d^1|e^0) P(c^1|e^0)}$$

Inference

How to compute $P(B|d^1, c^1)$?

Likelihood sampling has a still problem!



Bayesian approach :

$$\begin{aligned} P(c^1|d^1) &= \frac{P(d^1|c^1)P(c^1)}{P(d^1|c^1)P(c^1) + P(d^1|c^0)P(c^0)} \\ &= \frac{0.999 \times 0.001}{0.999 \times 0.001 + 0.001 \times 0.999} \\ &= 0.5 \end{aligned}$$

To use likelihood weighting sampling approach:

$c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, c^0, \dots, c^1$

$P(d^1|c^1) = 0$ because c^1 is not sampled due to the low prior

How to compute $P(B|d^1, c^1)$?

Gibbs sampling, a kind of Markov chain Monte Carlo technique

- The sequence of samples forms a Markov chain
- In the limit, samples are drawn exactly from the joint distribution over the unobserved variables given the observations
- Simulate samples by sweeping through all the posterior conditionals, one random variables at a time

Algorithm : Gibbs sampler

Initialize $X^{(0)} \sim q(x)$

for iteration $i = 1, \dots$ *do*

$$x_1^{(i)} \sim P\left(X_1 = x_1 \mid X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \dots, X_D = x_D^{(i-1)}\right)$$

$$x_2^{(i)} \sim P\left(X_2 = x_2 \mid X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \dots, x_D = x_D^{(i-1)}\right)$$

$$x_3^{(i)} \sim P\left(X_3 = x_3 \mid X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \dots, x_D = x_D^{(i-1)}\right)$$

$$\vdots$$

$$x_D^{(i)} \sim P\left(X_D = x_D \mid X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \dots, X_{D-1} = x_{D-1}^{(i)}\right)$$

end for

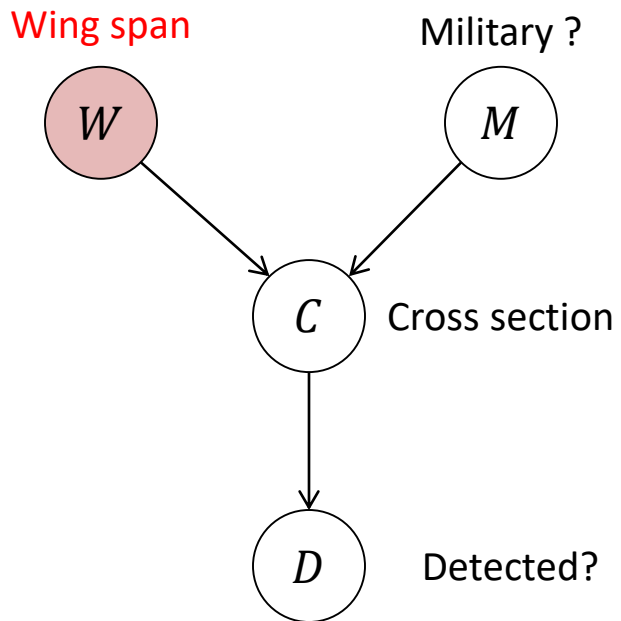
Because samples from the early iterations are not from the target posterior, it is common to discard these samples “**burn-in**” period”

Sampling method comparisons

Jupyter Demo Simulation
Wet grass (PyMC)

Hybrid Bayesian Networks

Bayesian networks can contain a mixture of both discrete and continuous variables

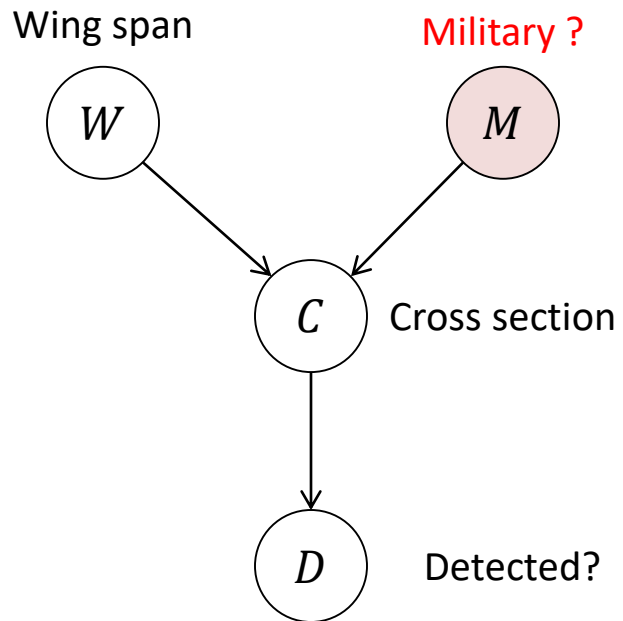


Wing span is a continuous variable and modeled as a Gaussian distribution

$$P(w) = N(w|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{w-\mu}{\sigma}\right)^2}$$

Hybrid Bayesian Networks

Bayesian networks can contain a mixture of both discrete and continuous variables



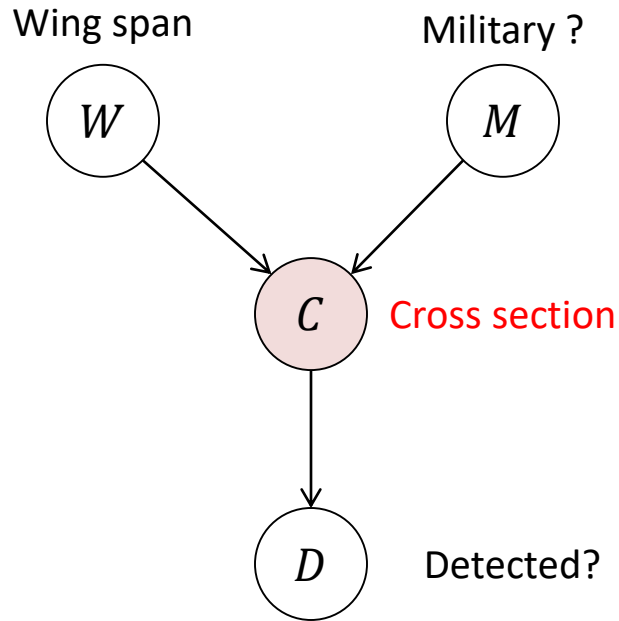
Whether a target is a military vehicle can be modeled with a single parameter θ

$$P(m^1) = \theta$$

$$P(m^0) = 1 - \theta$$

Hybrid Bayesian Networks

Bayesian networks can contain a mixture of both discrete and continuous variables



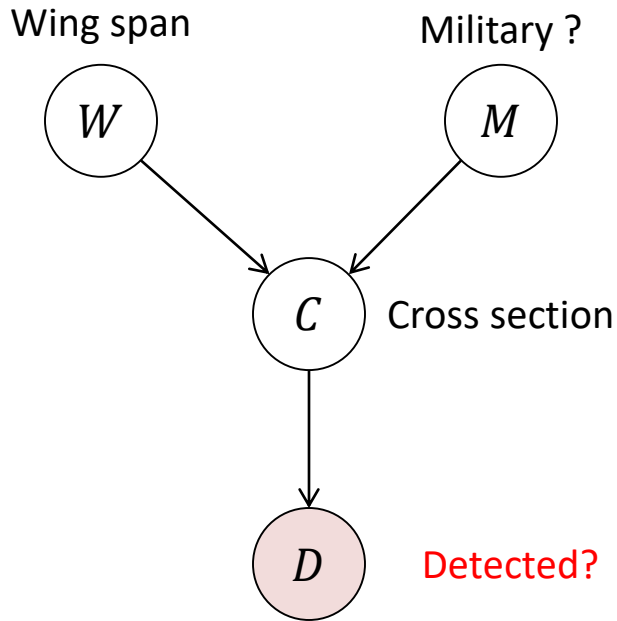
Radar cross section can be modeled as a conditional Gaussian

$$P(c|w, m) = \begin{cases} N(c|a_0w + b_0, \sigma_0^2) & \text{if } m = m^0 \\ N(c|a_1w + b_1, \sigma_1^2) & \text{if } m = m^1 \end{cases}$$

(Conditional linear Gaussian)

Hybrid Bayesian Networks

Bayesian networks can contain a mixture of both discrete and continuous variables

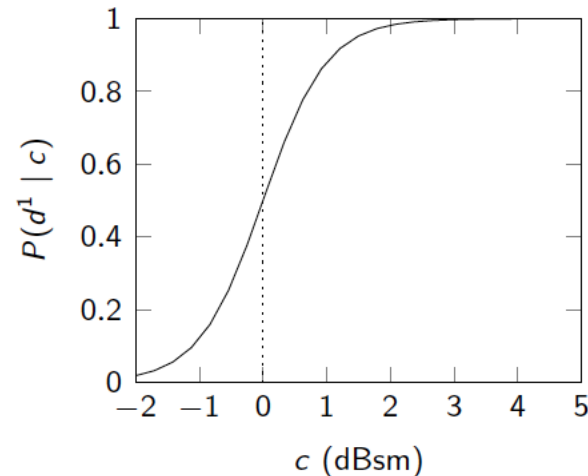


- Logit model:

$$P(d^1|c) = \frac{1}{1 + \exp\left(-2\frac{c - \alpha}{\beta}\right)}$$

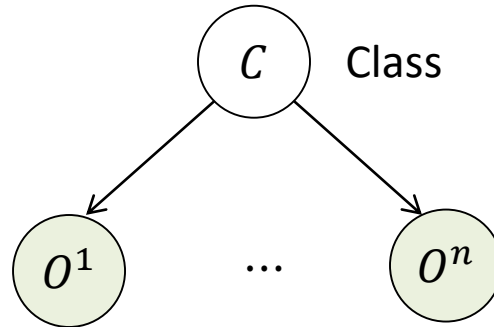
- Probit model:

$$P(d^1|c) = \Phi\left(\frac{c - \alpha}{\beta}\right)$$



Bayesian Network for Classification

Naïve Bayes Model



Prior: $P(C)$

Class conditional distribution $P(O^i|C)$

$$P(C|O^{1:n}) = \frac{P(C, O^{1:n})}{P(O^{1:n})} = \frac{P(C) \prod_{i=1}^n P(O^i|C)}{P(O^{1:n})}$$

$$P(O^{1:n}) = \sum_c P(C, O^{1:n})$$

$$P(C|O^{1:n}) \propto P(C) \prod_{i=1}^n P(O^i|C)$$

We already know how to estimate the parameters for probability distributions

MLE or Bayesian approach

- Bayesian Score $P(G|D)$ for a certain graph G given data D is defined as

$$\begin{aligned} P(G|D) &= \frac{P(G)P(D|G)}{P(D)} \\ &= \frac{P(G) \int_{\theta} P(D|\theta, G)P(\theta|G)d\theta}{P(D)} \end{aligned}$$

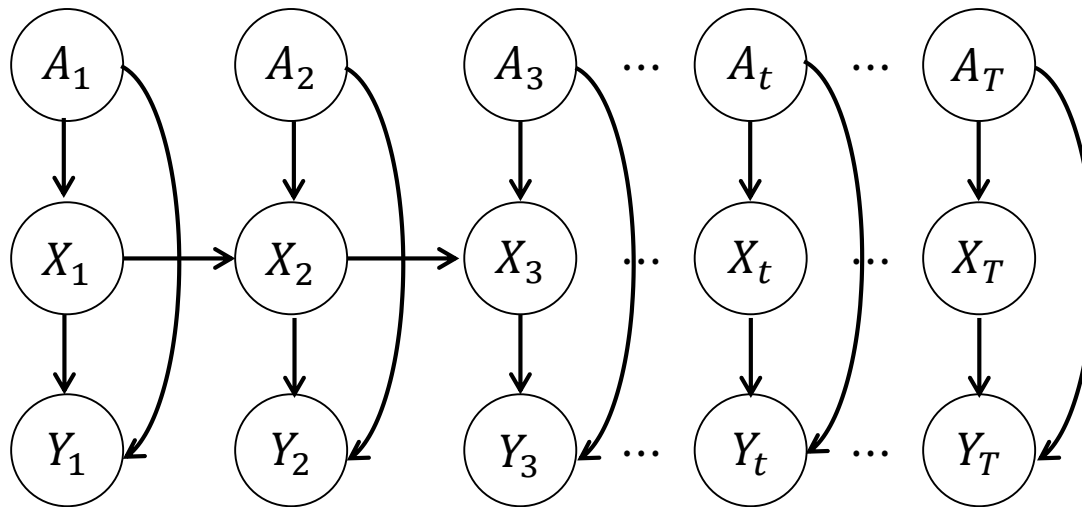
- A Bayesian approach to structure learning involves finding the graph G that maximizes the Bayesian Score $P(G|D)$ as

$$G^* = \operatorname{argmax}_G P(G|D)$$

- Not feasible to enumerate every possible structure, so use local search for graph with largest Bayesian score

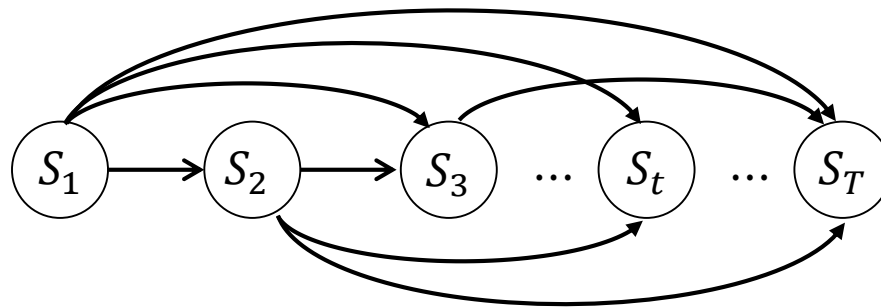
Dynamic Bayesian Network

Dynamic Bayesian Network relates variables to each other **over adjacent time steps**.



Dynamic Bayesian Networks (temporal model)

- We are interested in reasoning about the state of the world as it evolves over time
- **System state** S_t is a snapshot of the relevant attributes of the system at time t
- Trajectory of states S_1, \dots, S_t represents the evolution of the target system
- $P(S_1, \dots, S_t)$ is very complex probability space
→ we need a series of simplifying assumptions

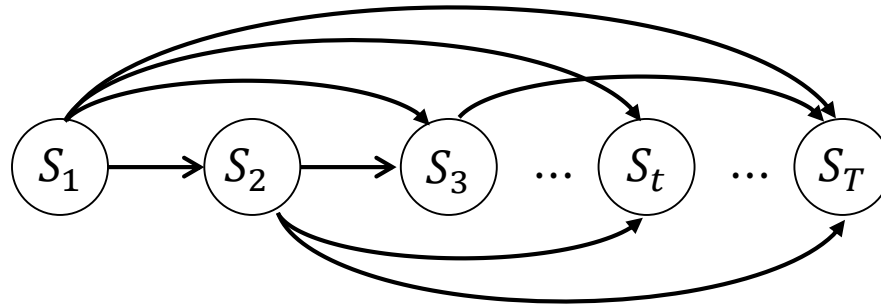


Discrete-State Markov models

Consider a distribution over trajectories sampled over a prefix of time $t = 1, \dots, T$

$$P(S_1, S_2, \dots, S_T) = P(S_1) \prod_{t=2}^T P(S_t | S_{1:t-1})$$

Cascade decomposition



Discrete-State Markov models

Consider a distribution over trajectories sampled over a prefix of time $t = 1, \dots, T$

$$P(S_1, S_2, \dots, S_T) = P(S_1) \prod_{t=2}^T P(S_t | S_{1:t-1})$$

Cascade decomposition

1. Markov Chain:

A Markov chain is defined on either discrete or continuous variables $S_{1:t}$ and the following **conditional independence** assumption holds:

$$P(S_t | S_1, \dots, S_{t-1}) = P(S_t | S_{t-L}, \dots, S_{t-1})$$

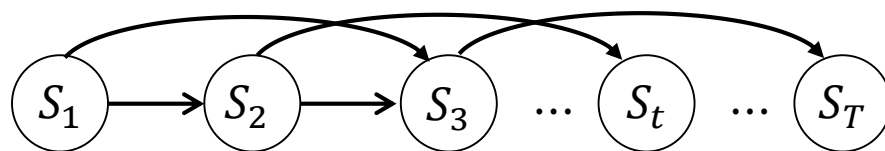
where $L \geq$ is the order of the Markov chain. First order Markov chain can be represented

$$P(S_1, S_2, \dots, S_T) = P(S_1) \prod_{t=2}^T P(S_t | S_{t-1})$$

the future is conditionally independent of the past given the present: $(S^{t+1} \perp S^{(1:t-1)} | S^t)$



First Order Markov Chain



Second Order Markov Chain

Discrete-State Markov models

Consider a distribution over trajectories sampled over a prefix of time $t = 1, \dots, T$

$$P(S_1, S_2, \dots, S_T) = P(S_1) \prod_{t=2}^T P(S_t | S_{1:t-1})$$

Cascade decomposition

1. Markov Chain:

A Markov chain is defined on either discrete or continuous variables $S_{1:t}$ and the following **conditional independence** assumption holds:

$$P(S_t | S_1, \dots, S_{t-1}) = P(S_t | S_{t-L}, \dots, S_{t-1})$$

where $L \geq 1$ is the order of the Markov chain. First order Markov chain can be represented

$$P(S_1, S_2, \dots, S_T) = P(S_0) \prod_{t=1}^T P(S_t | S_{t-1})$$

2. Stationary assumption

The state transition probability $P(S^{t+1} | S^t)$ is the same for all t

$$P(S^{t+1} = s' | S^t = s) = P(S' = s' | S = s) \text{ for any } t$$

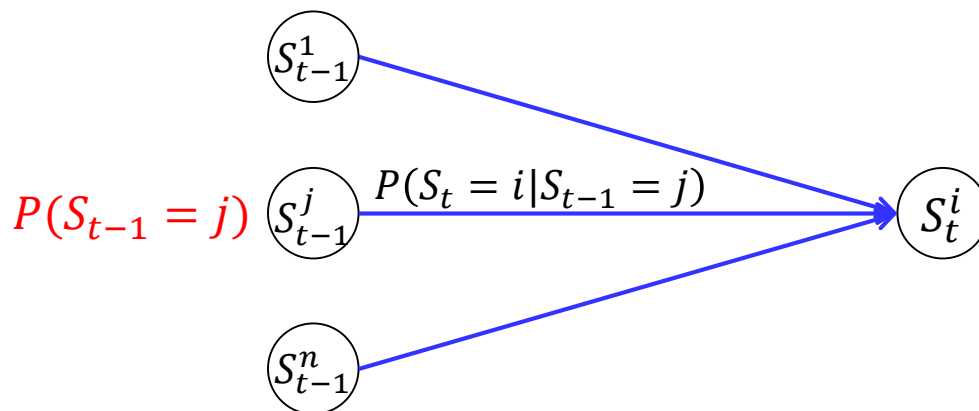
→ The number of parameters are reduced substantially

Equilibrium and stationary distribution of a Markov chain

- The marginal $P(S_t)$ evolves through time. For discrete time,

$$P(S_t = i) = \sum_j P(S_t = i | S_{t-1} = j) P(S_{t-1} = j)$$

- ✓ $P(S_t = i)$: the frequency that we visit state i at time t , given we started with a sample from $P(S_1)$ and subsequently repeatedly drew samples from the transition $P(S_\tau | S_{\tau-1})$



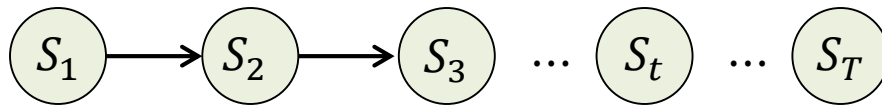
- Denoting $(\mathbf{p}_t)_i = P(S_t = i)$,

$$\mathbf{p}_t = \mathbf{M}\mathbf{p}_{t-1} = \mathbf{M}^{t-1}\mathbf{p}_1$$

- If, for $t \rightarrow \infty$, \mathbf{p}_t is independent of the initial distribution \mathbf{p}_1 , then \mathbf{p}_∞ is called the equilibrium distribution of the chain, that is

$$\mathbf{p}_\infty = \mathbf{M}\mathbf{p}_\infty$$

Fitting Markov Models



Given a sequence $(S_1 = s_1, S_2 = s_2, \dots, S_T = s_T)$, how to construct the transition matrix?

$$\theta_{i|j} = P(S_\tau = i | S_{\tau-1} = j) \propto \sum_{t=2}^T \mathbb{I}[S_t = i, S_{t-1} = j]$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{1|1} & \theta_{1|2} & \theta_{1|3} & \theta_{1|4} & \theta_{1|5} \\ \theta_{2|1} & \theta_{2|2} & \theta_{2|3} & \theta_{2|4} & \theta_{2|5} \\ \theta_{3|1} & \theta_{3|2} & \theta_{3|3} & \theta_{3|4} & \theta_{3|5} \\ \theta_{4|1} & \theta_{4|2} & \theta_{4|3} & \theta_{4|4} & \theta_{4|5} \\ \theta_{5|1} & \theta_{5|2} & \theta_{5|3} & \theta_{5|4} & \theta_{5|5} \end{bmatrix}$$

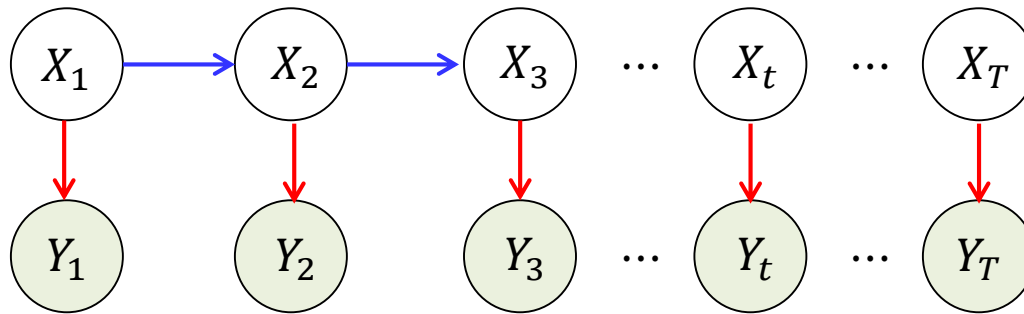
State transition matrix

$$\sum_i \theta_{i|j} = 1$$

If we have $s_{1:t} = \underset{\downarrow}{1}, \underset{\downarrow}{3}, 2, 4, 1, 4, 3, 5, \underset{\downarrow}{1}, \underset{\downarrow}{3}, 4, 2, 1, 4, 4, 2, 4, 5, \underset{\downarrow}{1}, \underset{\downarrow}{3}, 3, 4, \dots \longrightarrow \theta_{3|1} = \frac{3}{5}$

Definition of Hidden Markov Models

- The Hidden Markov Model (HMM) defines a Markov chain on hidden variables $X_{1:t}$
- The observed variables are dependent on the hidden variables through an emission $P(Y_t|X_t)$



- The joint distribution on the hidden variables and observations are

$$P(X_{1:t}, Y_{1:t}) = P(X_1)P(Y_1|X_1) \prod_{t=2}^T P(X_t|X_{t-1})P(Y_t|X_t)$$

- **Transition distribution:** For a stationary HMM the transition distribution $P(X_t|X_{t-1})$ is defined as the $H \times H$ matrix

$$M_{i,j} = P(X_t = i | X_{t-1} = j)$$

- **Emission distribution:** For a stationary HMM and emission distribution with discrete states $Y_t \in \{1, \dots, V\}$, we define $V \times H$ matrix

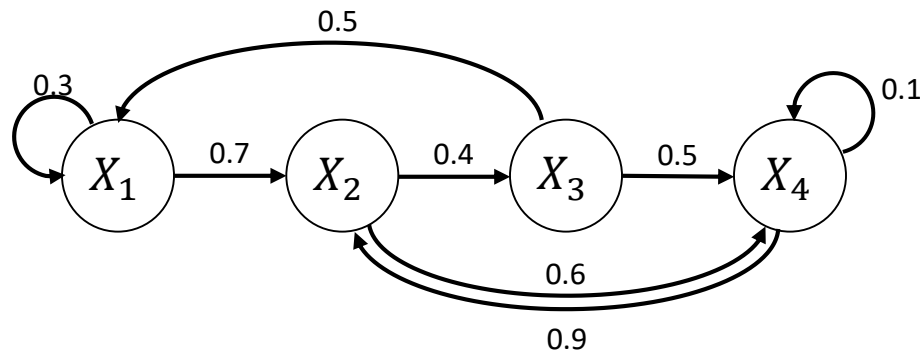
$$O_{i,j} = P(Y_t = i | X_t = j)$$

Hidden Markov Model

The state variable X_t is discrete

- The state transition model $P(X'|X)$ is usually sparse,
→ can be represented as a **directed graph**

$X = (X_1, X_2, X_3, X_4) : 4 \text{ discretized states}$

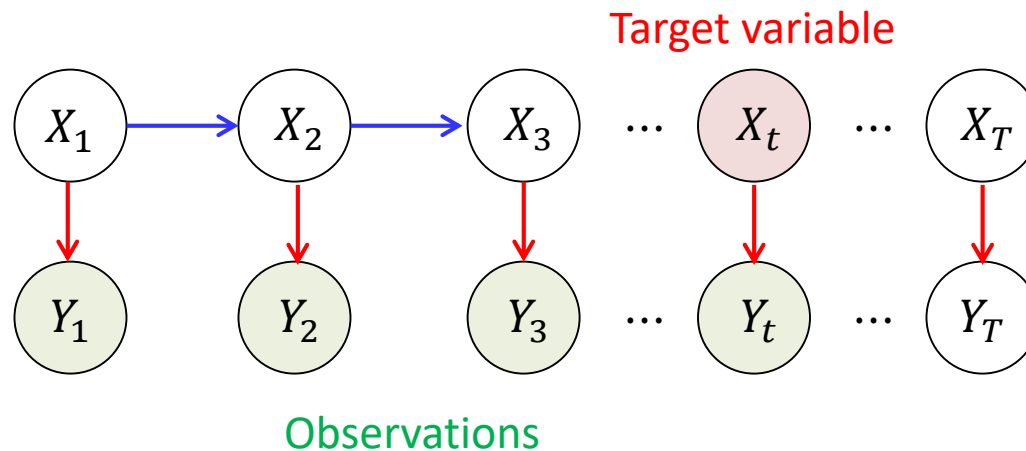


$$\sum_i P(X'_i|X) = 1$$

- The observation model : $P(Y |X)$ can be deterministic or random

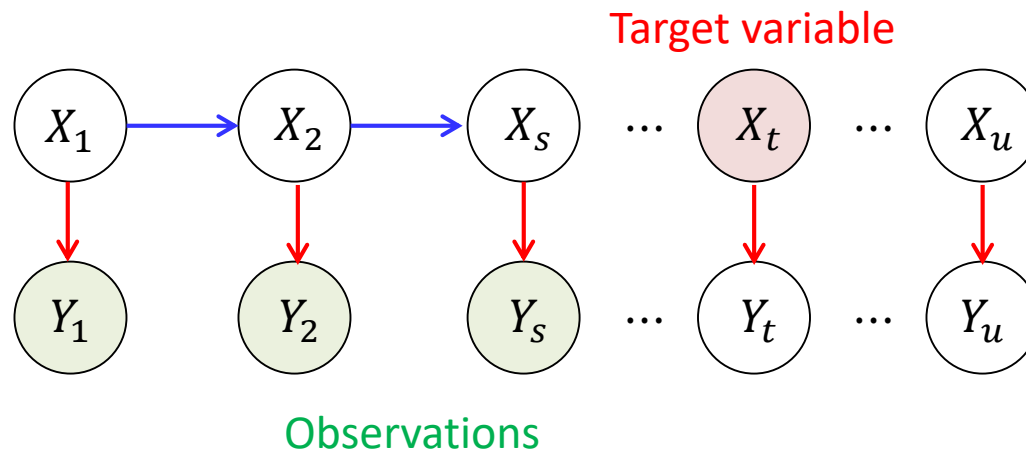
Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$
- Prediction (inferencing the future) $P(x_t|y_{1:s}) \quad t > s$
- Smoothing (inferencing the past) $P(x_t|y_{1:u}) \quad t < u$
- Likelihood (inferencing the past) $P(x_{1:t})$
- Most likely hidden path (Viterbi alignment) $\operatorname{argmax}_{x_{1:t}} P(x_{1:t}|y_{1:t})$



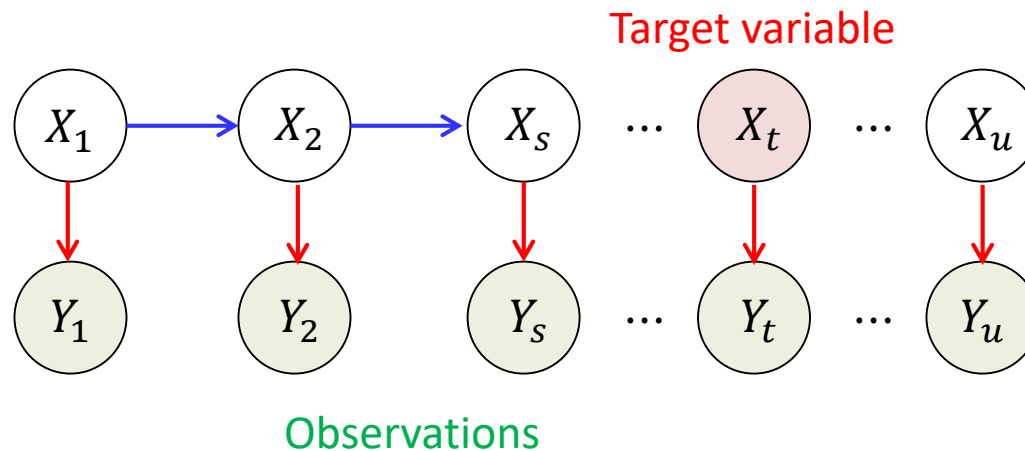
Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$
- Prediction (inferencing the future) $P(x_t|y_{1:s}) \quad t > s$
- Smoothing (inferencing the past) $P(x_t|y_{1:u}) \quad t < u$
- Likelihood (inferencing the past) $P(x_{1:t})$
- Most likely hidden path (Viterbi alignment) $\operatorname{argmax}_{x_{1:t}} P(x_{1:t}|y_{1:t})$



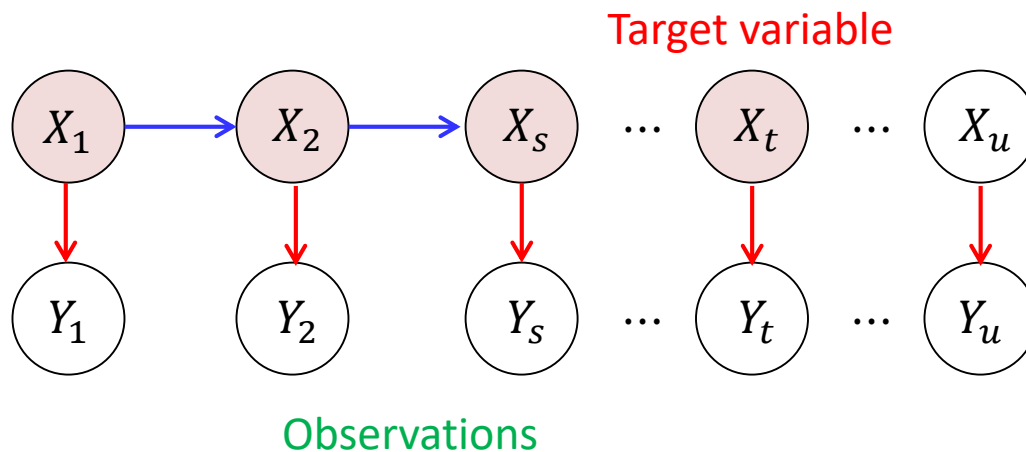
Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$
- Prediction (inferencing the future) $P(x_t|y_{1:s}) \quad t > s$
- Smoothing (inferencing the past) $P(x_t|y_{1:u}) \quad t < u$
- Likelihood (inferencing the past) $P(x_{1:t})$
- Most likely hidden path (Viterbi alignment) $\operatorname{argmax}_{x_{1:t}} P(x_{1:t}|y_{1:t})$



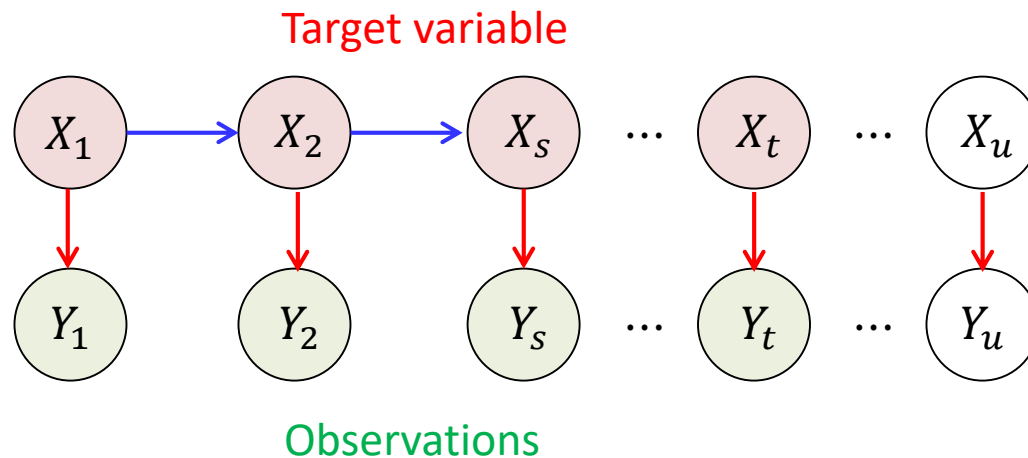
Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$
- Prediction (inferencing the future) $P(x_t|y_{1:s}) \quad t > s$
- Smoothing (inferencing the past) $P(x_t|y_{1:u}) \quad t < u$
- Likelihood (inferencing the past) $P(x_{1:t})$
- Most likely hidden path (Viterbi alignment) $\operatorname{argmax}_{x_{1:t}} P(x_{1:t}|y_{1:t})$



Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$
- Prediction (inferencing the future) $P(x_t|y_{1:s}) \quad t > s$
- Smoothing (inferencing the past) $P(x_t|y_{1:u}) \quad t < u$
- Likelihood (inferencing the past) $P(x_{1:t})$
- Most likely hidden path (Viterbi alignment) $\operatorname{argmax}_{x_{1:t}} P(x_{1:t}|y_{1:t})$



Inferencing of Hidden Markov Models

- Filtering (inferencing the present) $P(x_t|y_{1:t})$

$$P(x_t|y_{1:t}) = \frac{P(x_t, y_{1:t})}{P(y_{1:t})} \propto P(x_t, y_{1:t})$$

$$\begin{aligned}
 P(x_t, y_{1:t}) &= \sum_{x_{t-1}} P(x_t, \cancel{x_{t-1}}, y_{1:t-1}, y_t) \\
 &\quad \sum_{x_{t-1}} P(y_t | \cancel{y_{1:t-1}}, x_t, \cancel{x_{t-1}}) P(x_t | \cancel{y_{1:t-1}}, x_{t-1}) P(x_{t-1}, y_{1:t-1}) \\
 &\quad \sum_{x_{t-1}} P(y_t | x_t) P(x_t | x_{t-1}) P(x_{t-1}, y_{1:t-1}) \quad \because \text{Conditional independence} \\
 &\quad \underbrace{P(y_t | x_t)}_{\text{corrector}} \underbrace{\sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}, y_{1:t-1})}_{\text{predictor}}
 \end{aligned}$$

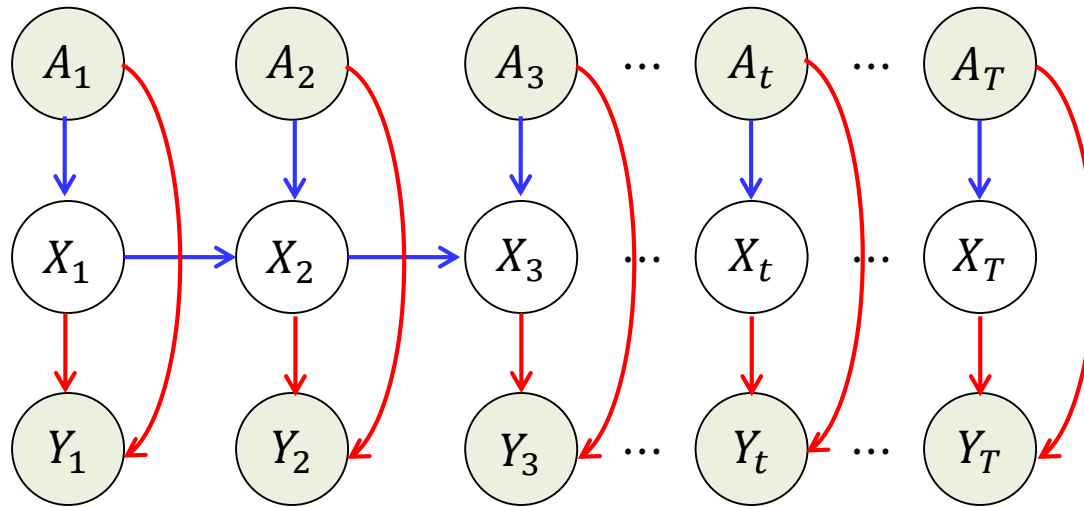
Bayesian view

$$P(x_t|y_{1:t}) \propto \underbrace{\sum_{x_{t-1}} P(y_t | x_t) P(x_t | x_{t-1})}_{\text{Regarded as likelihood}} \underbrace{P(x_{t-1} | y_{1:t-1})}_{\text{Modified prior distribution}}$$

Regarded as likelihood

Modified prior distribution has an effect of removing all nodes in the graph before time $t - 1$

Input and output hidden Markov Model (IOHMM)



The state transition model : $P(X_t|X_{t-1}, A_t)$

The observation model : $P(Y_t|X_t, A_t)$

Continuous-state Markov models

- In many practical time series applications, the data is naturally continuous (i.e., variables are not discretized), particularly for models of the physical environment
- Restrict the form of the continuous transition $p(X_t|X_{t-1})$
- A simple yet powerful class of such transitions are the **linear dynamical systems**
- A *deterministic linear dynamical system* defines the temporal evolution of a vector x_t according to the discrete-time update equation

$$x_t = A_t x_{t-1}$$

where A_t is the transition matrix at time t

- If A_t is invariant with t , the process is called stationary or time-invariant

Observed linear dynamic system

- A *stochastic linear dynamical system* defines the temporal evolution of a vector x_t according to the discrete-time update equation

$$x_t = A_t x_{t-1} + \eta_t$$

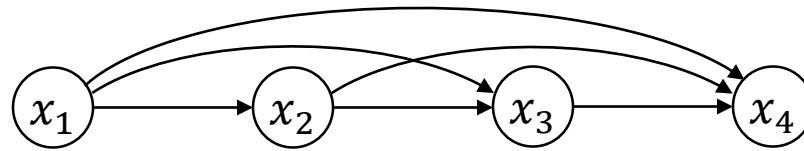
where η_t is a noise vector sampled from a Gaussian distribution

$$\eta_t \sim N(\mu_t, \Sigma_t)$$

- This is equivalent to a first-order Markov model with transition

$$p(x_t | x_{t-1}) = N(x_t | A_t x_{t-1} + \eta_t, \Sigma_t)$$

Auto-regressive models



- A scalar time-invariant Auto-Regressive (AR) model is defined by

$$x_t = \sum_{l=1}^L a_l x_{t-l} + \eta_t, \quad \eta_t \sim N(\mu, \sigma^2)$$

where $a = (a_1, a_2, \dots, a_L)^T$ are AR coefficients and σ^2 is innovation noise.

- As a belief network, the AR model can be written as an L th-order Markov model:

$$p(x_{1:T}) = \prod_{t=1}^T p(x_t | x_{t-1}, \dots, x_{t-L}), \quad \text{with } x_i = 0 \text{ for } i \leq 0$$

$\hat{x}_{t-1} = (x_{t-1}, \dots, x_{t-L})$

$$\text{with } p(x_t | x_{t-1}, \dots, x_{t-L}) = N(x_t | \sum_{l=1}^L a_l x_{t-l}, \sigma^2) = N(x_t | a^T \hat{x}_{t-1}, \sigma^2)$$

Similar to Bayesian Regression

- Heavily used in financial time series prediction, being able to capture simple trends in the data
- The AR coefficients form a compressed representation of the signal

Training Auto-regressive model

- Maximum likelihood training of the AR coefficients is straightforward based on

$$\begin{aligned}\log p(x_{1:T}) &= \log \prod_{t=1}^T p(x_t | x_{t-1}, \dots, x_{t-L}) \\ &= \sum_{t=1}^T \log(x_t | \hat{x}_{t-1}) \\ &= -\frac{1}{2\sigma^2} \sum_{t=1}^T (x_t - a^T \hat{x}_{t-1})^2 - \frac{T}{2} \log(2\pi\sigma^2)\end{aligned}$$

- Differentiating w.r.t. a and equating to zero we arrive at

$$\begin{aligned}\sum_{t=1}^T (\mathbf{x}_t - a^T \hat{x}_{t-1}) \hat{x}_{t-1} &= 0 \\ \rightarrow a &= [\sum_t \hat{x}_{t-1} \hat{x}_{t-1}^T]^{-1} \sum_t \mathbf{x}_t \hat{x}_{t-1} \quad \mathbf{x}_t : \text{target output (scalar)}\end{aligned}$$

- Similarly,

$$\sigma^2 = \frac{1}{T} \sum_{t=1}^T (x_t - a^T \hat{x}_{t-1})^2$$

Time-varying Auto-regressive model

- Learning the AR coefficients as a problem in inference in a latent linear dynamical system (LDS):

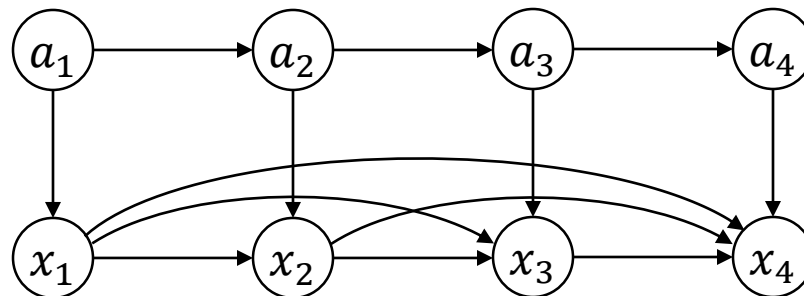
$$x_t = \hat{x}_{t-1}^T a_t + \eta_t, \quad \eta_t \sim N(0, \sigma^2)$$

which can be viewed as the emission distribution of a latent LDS in which the hidden variable is a_t and the time dependent emission matrix is given by \hat{x}_{t-1}^T

- By placing a simple latent transition

$$a_t = a_{t-1} + \eta_t^a, \quad \eta_t^a \sim N(0, \sigma_a^2 \mathbf{I})$$

which encourages the AR coefficients to change slowly with time



Time-varying Auto-regressive model

- Learning the AR coefficients as a problem in inference in a latent linear dynamical system (LDS):

$$x_t = \hat{x}_{t-1}^T a_t + \eta_t, \quad \eta_t \sim N(0, \sigma^2)$$

which can be viewed as the emission distribution of a latent LDS in which the hidden variable is a_t and the time dependent emission matrix is given by \hat{x}_{t-1}^T

- By placing a simple latent transition

$$a_t = a_{t-1} + \eta_t^a, \quad \eta_t^a \sim N(0, \sigma_a^2 \mathbf{I})$$

which encourages the AR coefficients to change slowly with time

- The joint distribution between the observation $x_{1:T}$ and the coefficients $\mathbf{a}_{1:t}$

$$p(\mathbf{a}_{1:T} | x_{1:T}) \propto p(x_{1:T}, \mathbf{a}_{1:T}) = \prod_{t=2}^T p(x_t | a_t, \hat{x}_{t-1}) p(a_t | a_{t-1})$$

then we can compute

$$\mathbf{a}_{1:T}^* = \operatorname{argmax}_{\mathbf{a}_{1:T}} p(\mathbf{a}_{1:T} | x_{1:T})$$

from which the MAP estimates for the AR coefficients can be determined

Time-varying variance Auto-regressive model

- For some applications, particularly in finance, the variance can change with time due to volatility

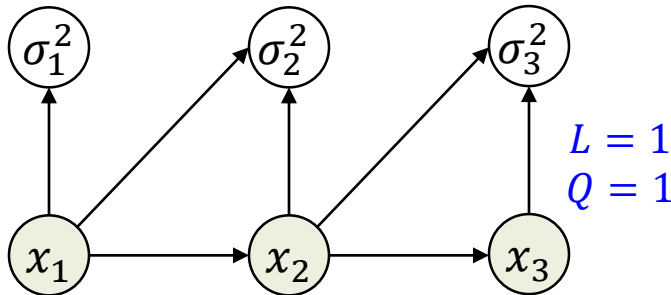
$$x_t = \sum_{l=1}^L a_l x_{t-l} + \eta_t, \quad \eta_t \sim N(0, \sigma_t^2)$$

Time varying variance

$$\bar{x}_t = \sum_{l=1}^L a_l x_{t-l}$$

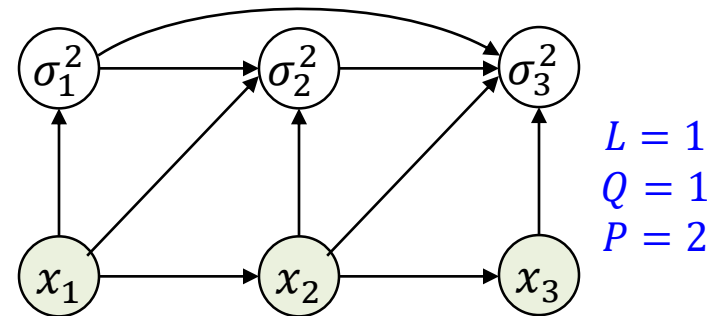
- The estimated time varying variance of noise can be computed

Auto Regressive Conditional Heteroscedasticity (ARCH)



$$\sigma_t^2 = \sigma_0 + \sum_{i=1}^Q \alpha_i (x_{t-i} - \bar{x}_{t-i})^2$$

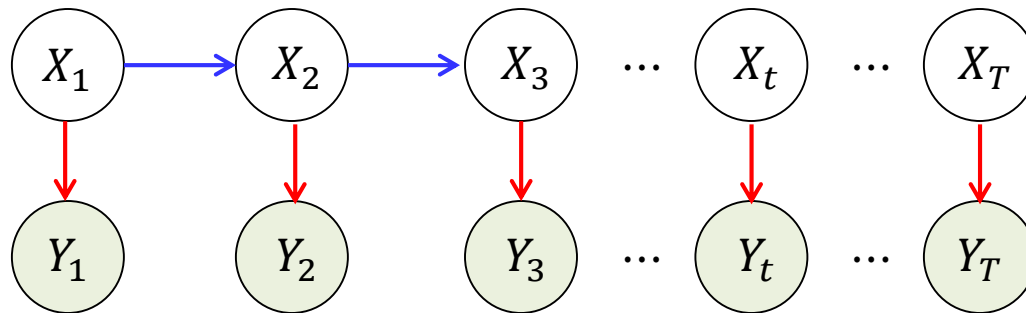
Generalized ARCH model (GARCH)



$$\sigma_t^2 = \sigma_0 + \sum_{i=1}^Q \alpha_i (x_{t-i} - \bar{x}_{t-i})^2 + \sum_{i=1}^P \beta_i \sigma_{t-i}^2$$

Linear Gaussian State Space Model

- The latent LDS defines a stochastic linear dynamical system in a latent space on a sequence of states $x_{1:T}$
- Observations $y_{1:T}$ are used to infer the hidden states that tracks or explains the system evolution



Transition model : $x_t = A_t x_{t-1} + \eta_t^x$,

Emission model : $y_t = B_t x_t + \eta_t^y$,

$\eta_t^x \sim N(\bar{x}_t, \Sigma_t^x)$

$\eta_t^y \sim N(\bar{y}_t, \Sigma_t^y)$

A_t : transition matrix

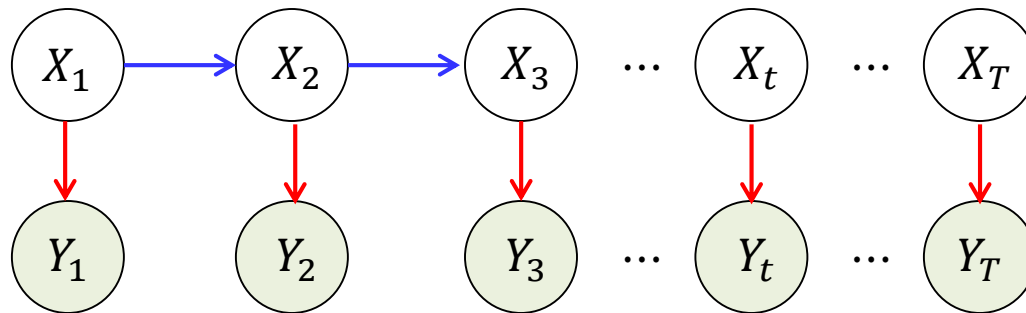
B_t : emission matrix

η_t^x transition noise vector with a hidden bias \bar{x}_t

η_t^y emission noise vector with a hidden bias \bar{y}_t

Linear Gaussian State Space Model

- The latent LDS defines a stochastic linear dynamical system in a latent space on a sequence of states $x_{1:T}$
- Observations $y_{1:T}$ are used to infer the hidden states that tracks or explains the system evolution



Transition model : $p(x_t|x_{t-1}) = N(x_t|A_tx_{t-1} + \bar{x}_t, \Sigma_t^x)$, $p(x_1) = N(x_1|\mu_\pi, \Sigma_\pi)$

Emission model : $p(y_t|x_t) = N(y_t|B_tx_t + \bar{y}_t, \Sigma_t^y)$

- The first order Markov model is then defined as

$$p(x_{1:T}, y_{1:T}) = p(x_1)p(y_1|x_1) \prod_{t=2}^T p(x_t|x_{t-1})p(y_t|x_t)$$

Kalman Filter

- Recall the filtering recursion for HMM:

$$P(x_t|y_{1:t}) \propto \sum_{x_{t-1}} P(y_t|x_t)P(x_t|x_{t-1})P(x_{t-1}|y_{1:t-1})$$

- For linear Gaussian State-space model, the recursion becomes

$$P(x_t|y_{1:t}) \propto \int_{x_{t-1}} P(y_t|x_t)P(x_t|x_{t-1})P(x_{t-1}|y_{1:t-1}) \quad \text{for } t > 1$$

- Since the product of two Gaussians is another Gaussian, and the integral of a Gaussian is another Gaussian, $P(x_t|y_{1:t})$ is Gaussian:

$$P(x_t|y_{1:t}) = N(x_t|f_t, F_t)$$

- Thus the recursion is for computing the mean f_t and the variance F_t for $P(x_t|y_{1:t})$ using f_{t-1} and the variance F_{t-1} for $P(x_{t-1}|y_{1:t-1})$



$$P(x_{t-1}|y_{1:t-1}) = N(x_{t-1}|f_{t-1}, F_{t-1})$$

$$P(x_t|y_{1:t}) = N(x_t|f_t, F_t)$$