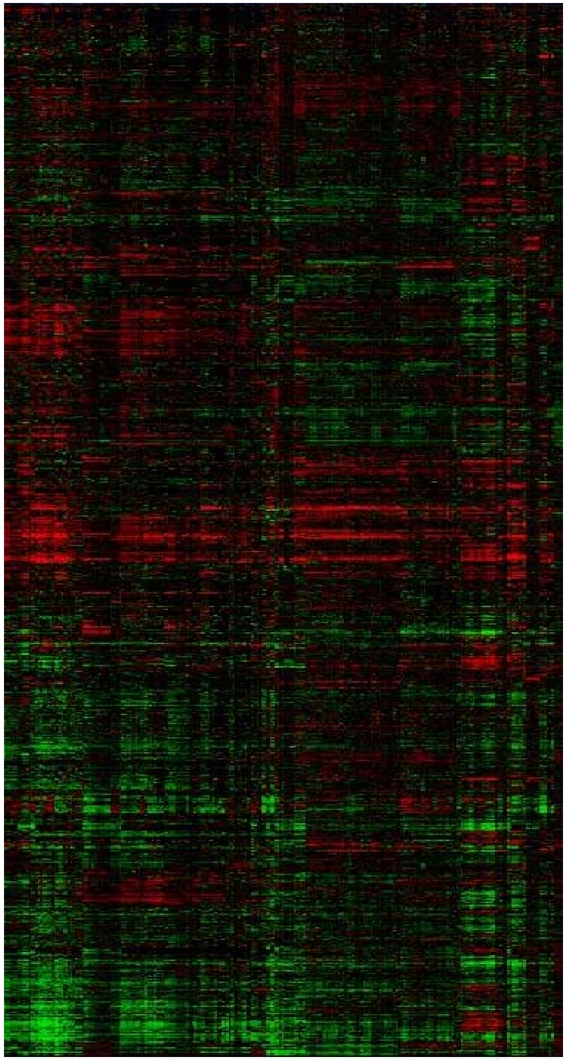
6.047 / 6.878 Computational Biology: Genomes, Networks, Evolution
Fall 2008

Introduction to Bayesian Networks

Overview

- We have looked at a number of graphical representations of probability distributions
  - DAG example: HMM
  - Undirected graph example: CRF
- Today we will look at a very general graphical model representation – Bayesian Networks
- One application – modeling gene expression
- Aviv Regev guest lecture – an extension of this basic idea

Probabilistic Reconstruction

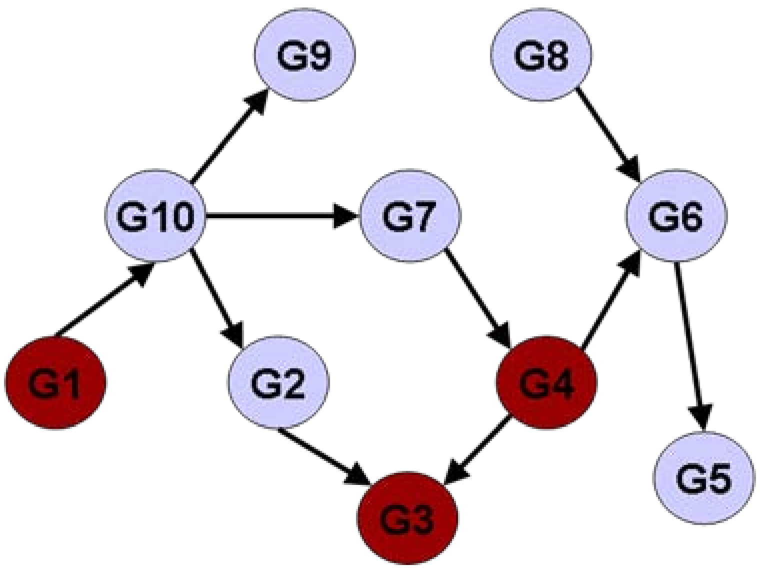


- Expression data gives us information about what genes tend to be expressed with others
- In probability terms, information about the joint distribution over gene states X:

$$\mathbf{P(X)=P(X_1,X_2,X_3,X_4,\ldots,X_m)}$$

Can we model this joint distribution?

Bayesian Networks



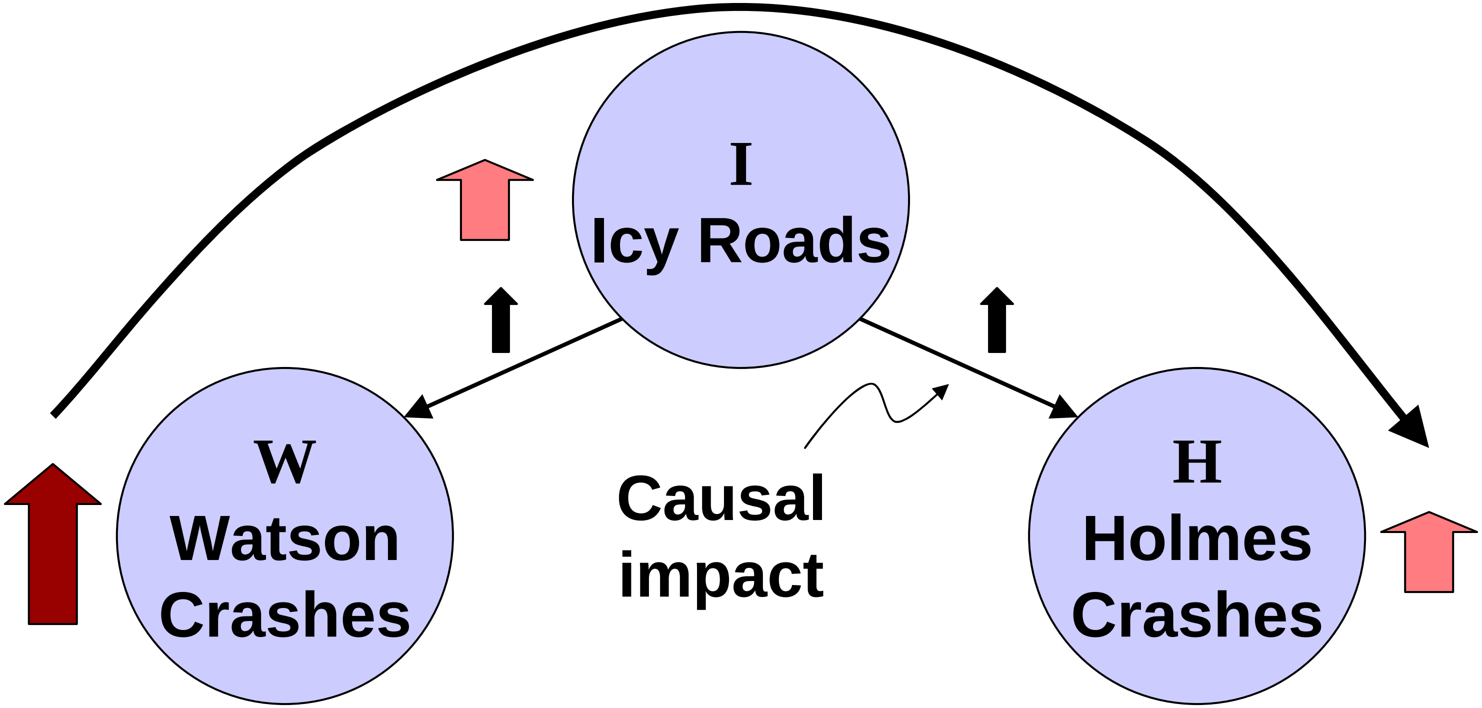
- Directed graph encoding joint distribution variables X

$$P(X) = P(X1,X2,X3,\ldots,XN)$$

- Learning approaches
- Inference algorithms

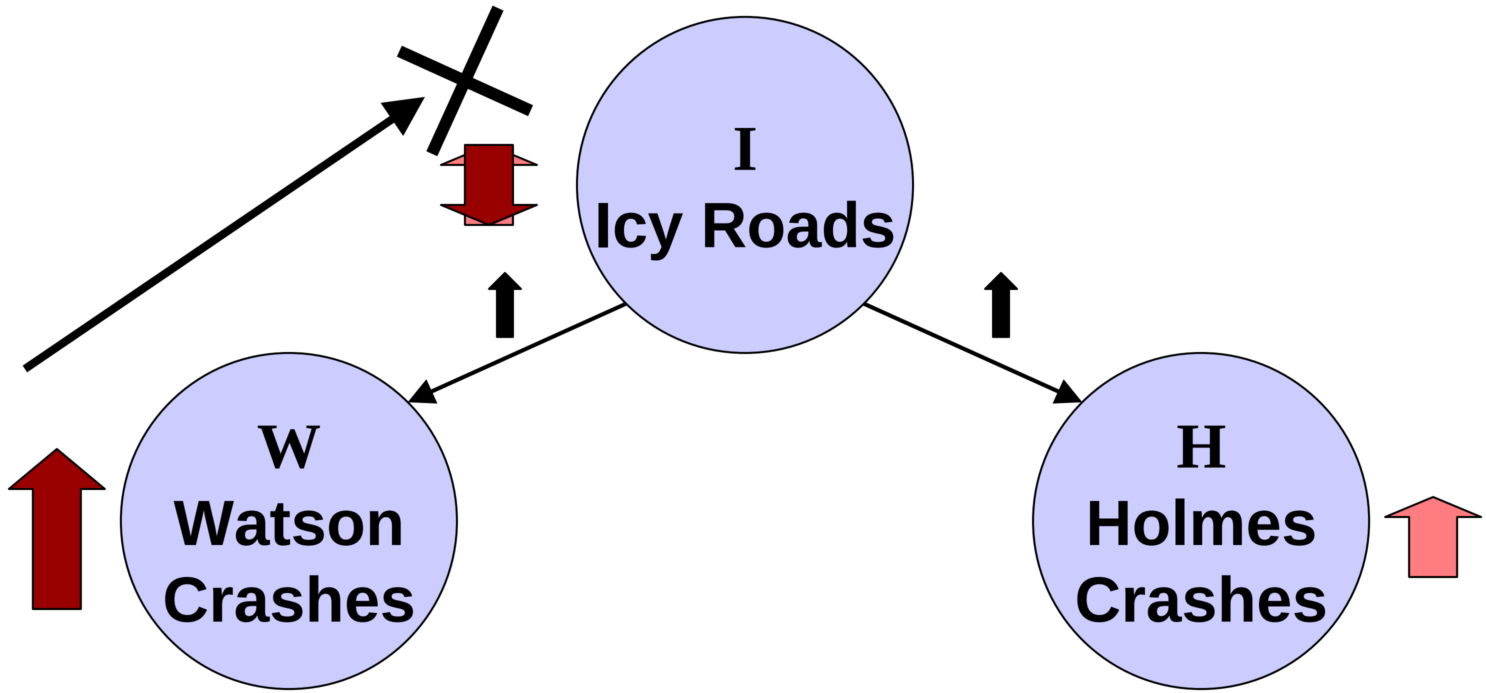
- Captures information about *dependency structure* of P(X)

Example 1 – Icy Roads



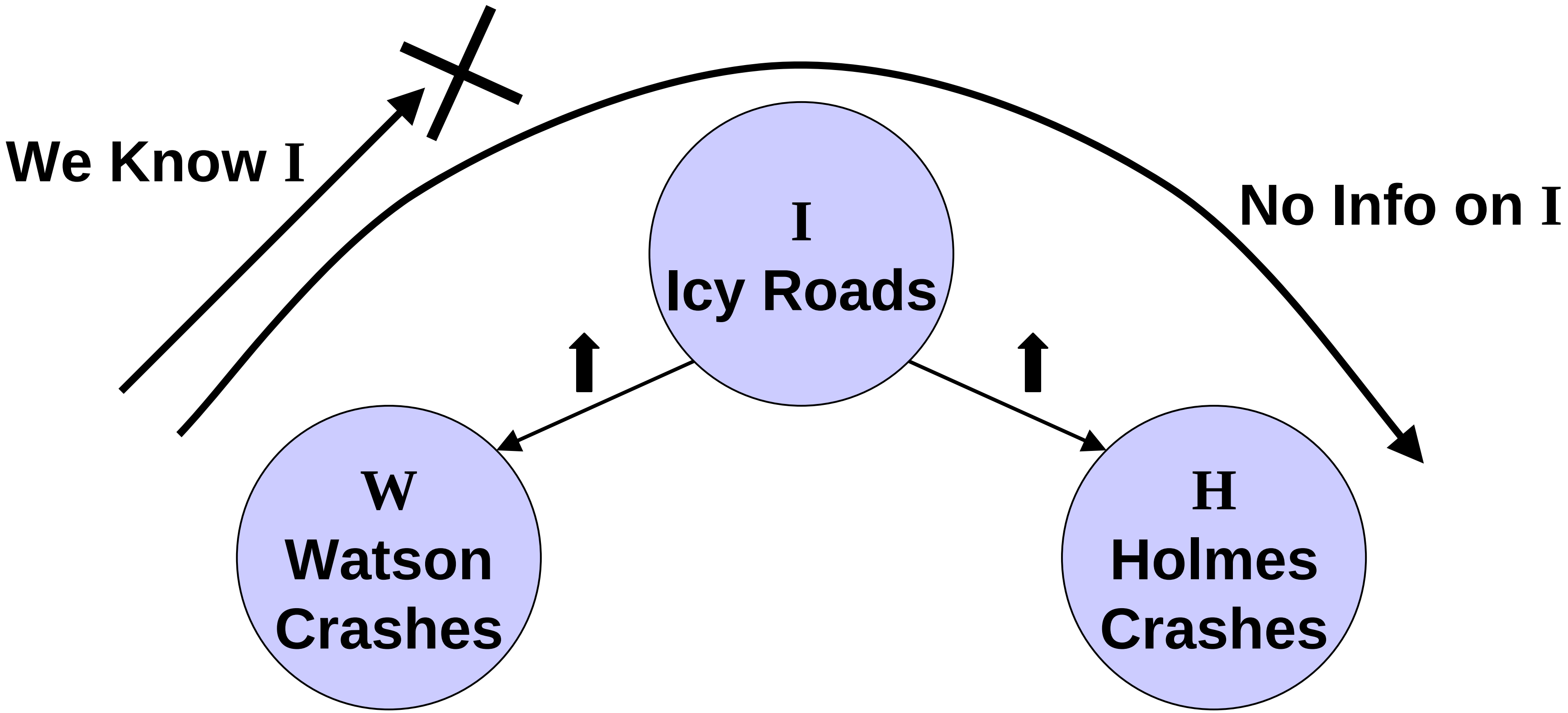
Assume we learn that Watson has crashed
Given this causal network, one might fear Holmes has crashed too. Why?

Example 1 – Icy Roads



Now imagine we have learned that roads are not icy
We would no longer have an increased fear that Holmes has crashed

Conditional Independence



If we know nothing about I, W and H are dependent
If we know I, W and H are conditionally independent

Conditional Independency

- Independence of 2 random variables

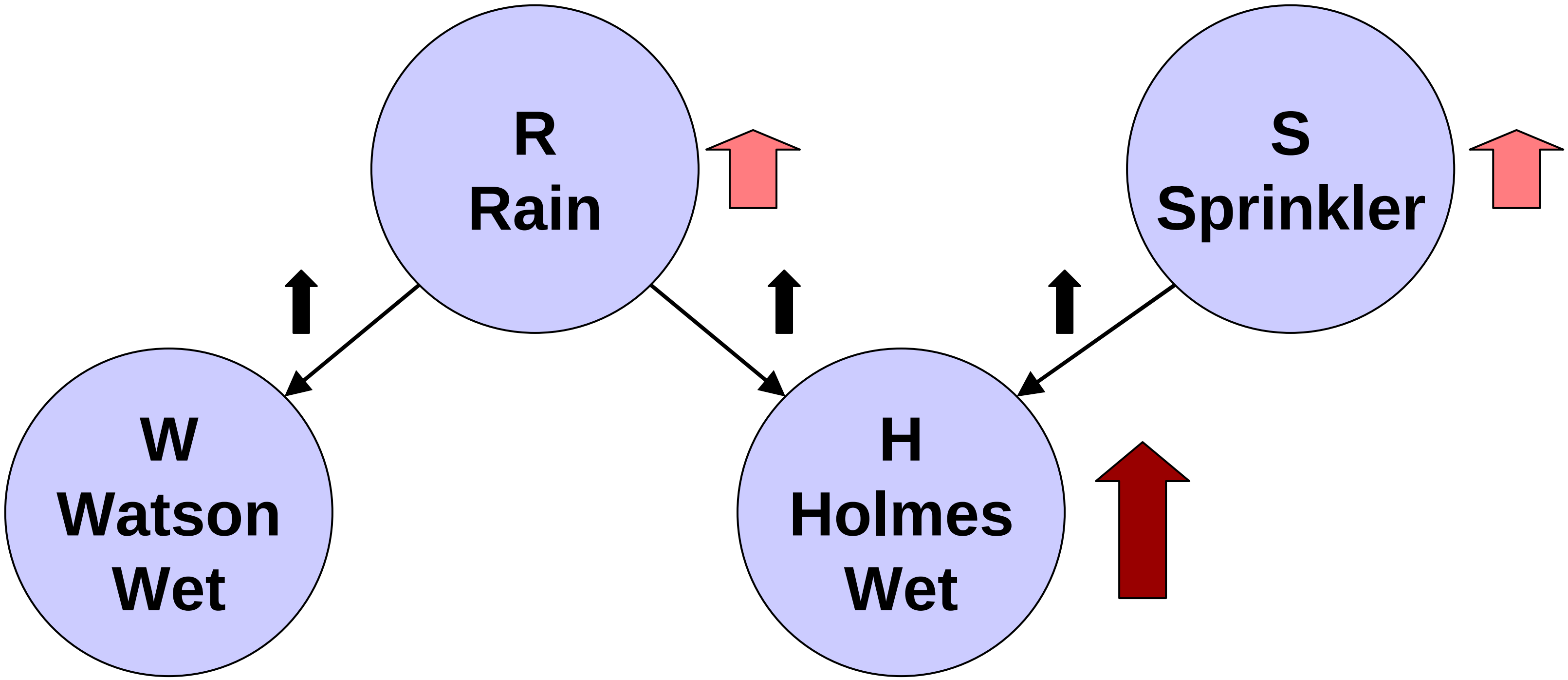
$$X \perp Y \Leftrightarrow P(X, Y) = P(X)P(Y)$$

- *Conditional* independence given a third

$$X \perp Y \mid Z \Leftrightarrow P(X, Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$

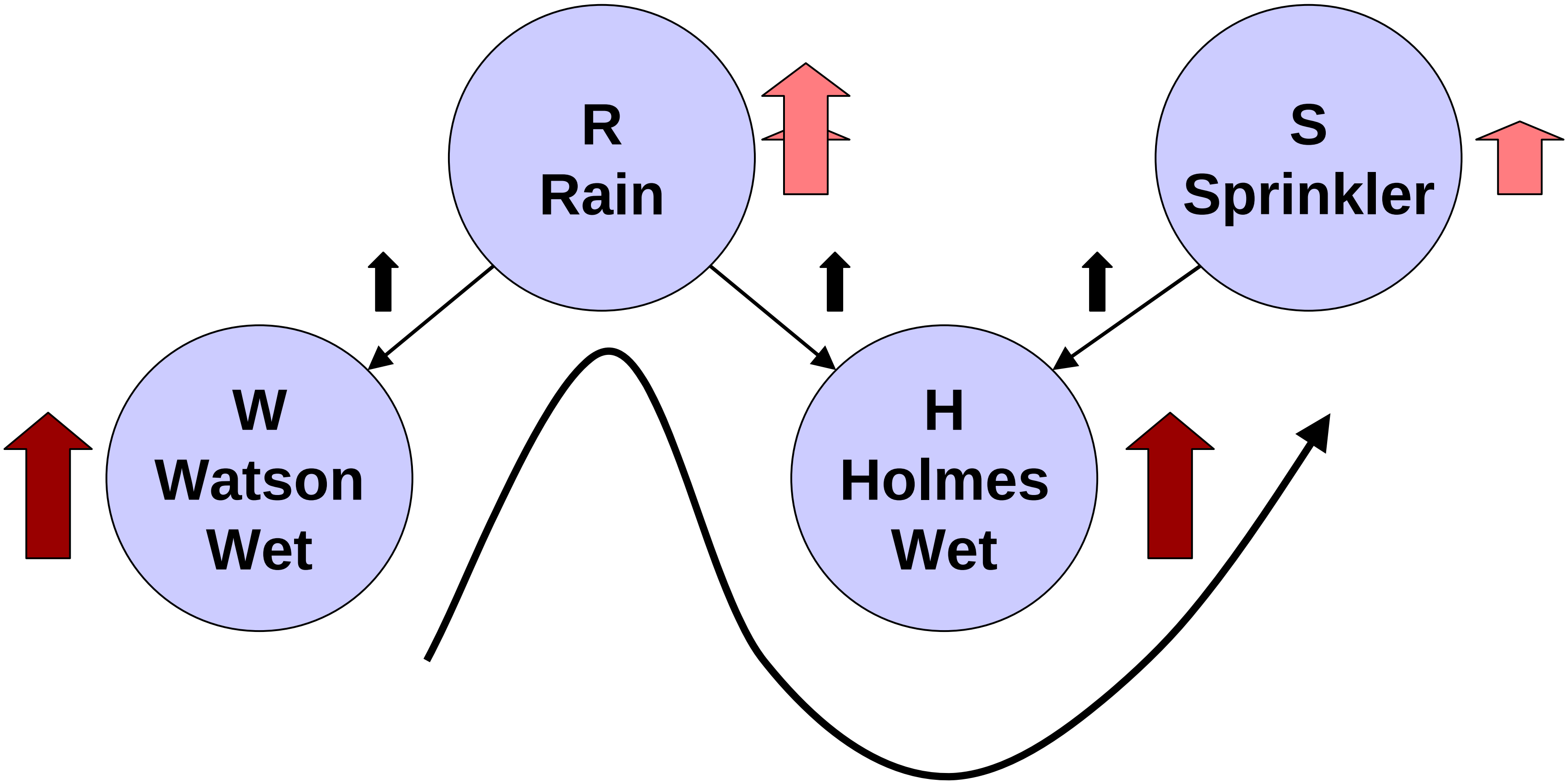
$$\text{but} \quad P(X, Y) \neq P(X)P(Y) \ \text{ necessarily}$$

Example 2 – Rain/Sprinkler



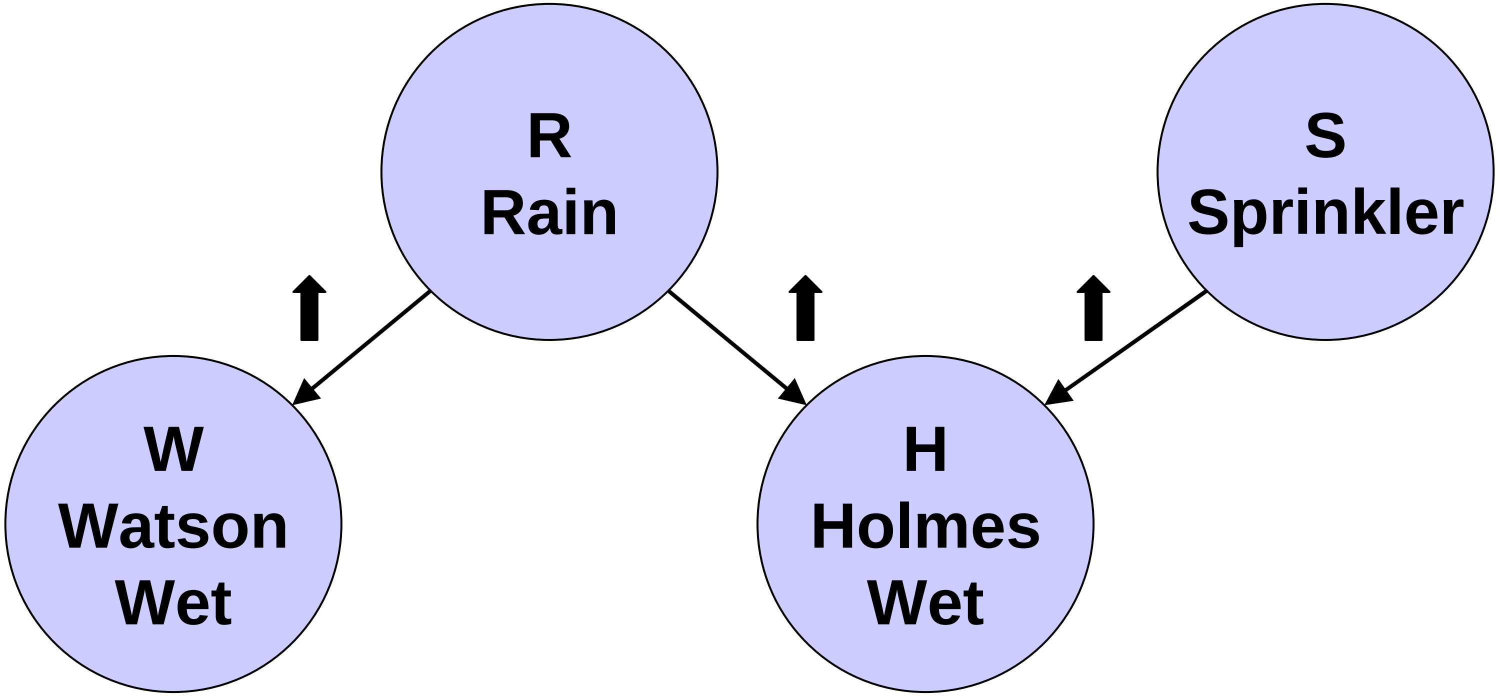
Holmes discovers his house is wet.
Could be rain or his sprinkler.

Example 2 – Rain/Sprinkler



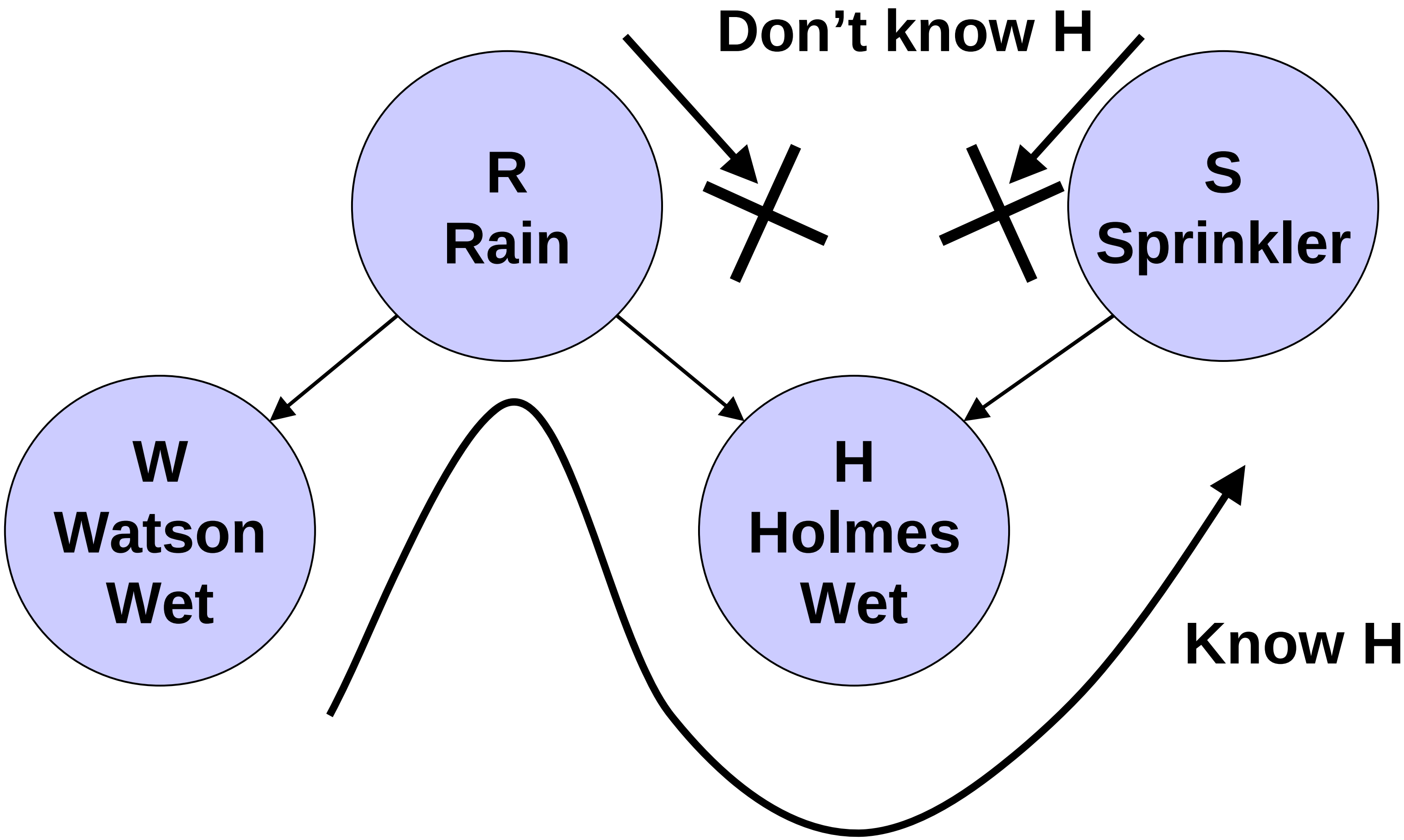
Now imagine Holmes sees Watsons grass is wet
Now we conclude it was probably rain
And probably *not* his sprinkler

Explaining Away



Initially we had two explanations for Holmes' wet grass. But once we had more evidence for R, this *explained away* H and thus no reason for increase in S

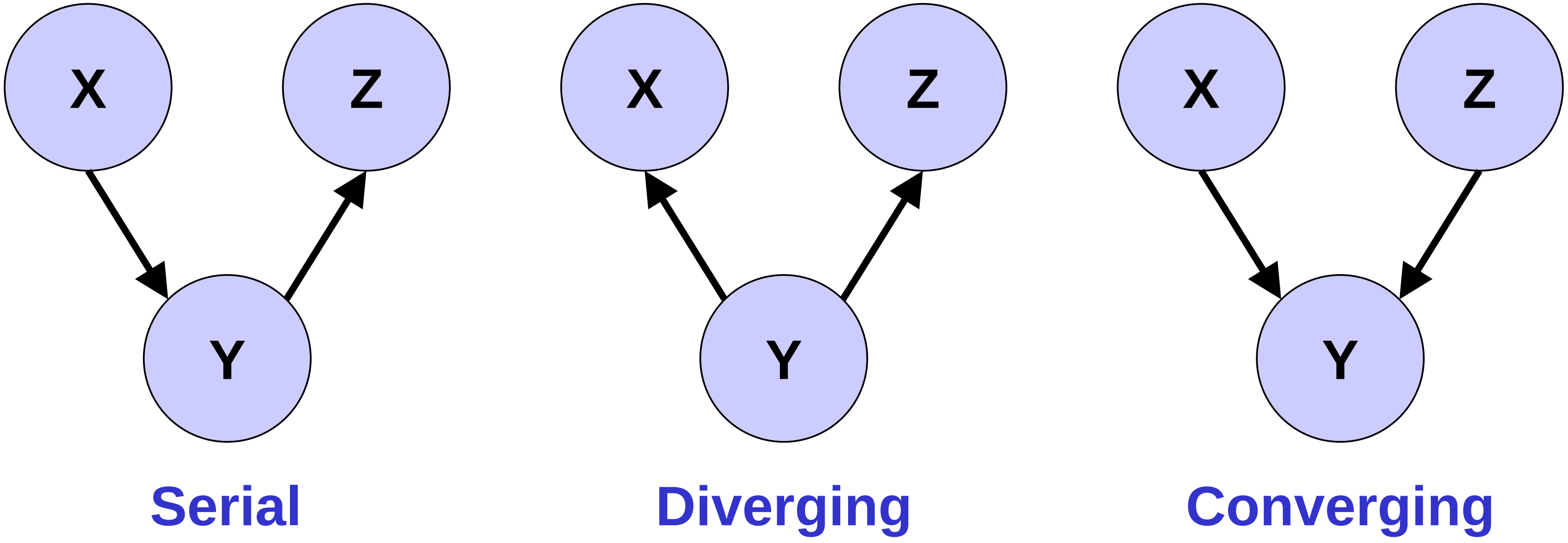
Conditional Dependence



If we don't know H, R and S are …
independent
But if we know H, R and S are *conditionally dependent*

Graph Semantics

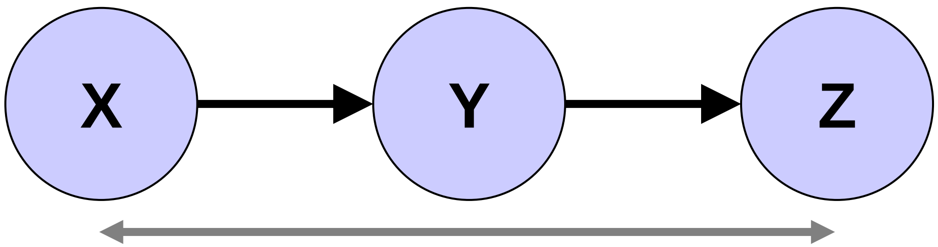
Three basic building blocks



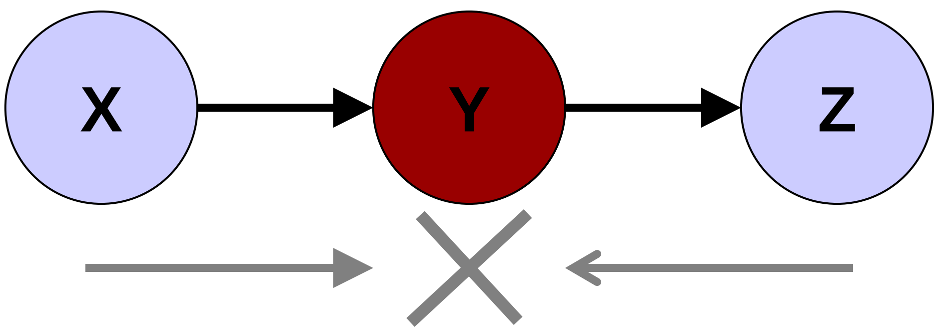
Each implies a particular independence relationship

Chain/Linear

Conditional Independence



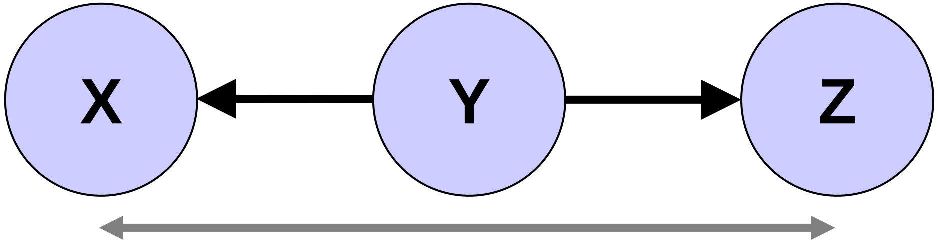
$$X \not\perp Z \mid \emptyset$$



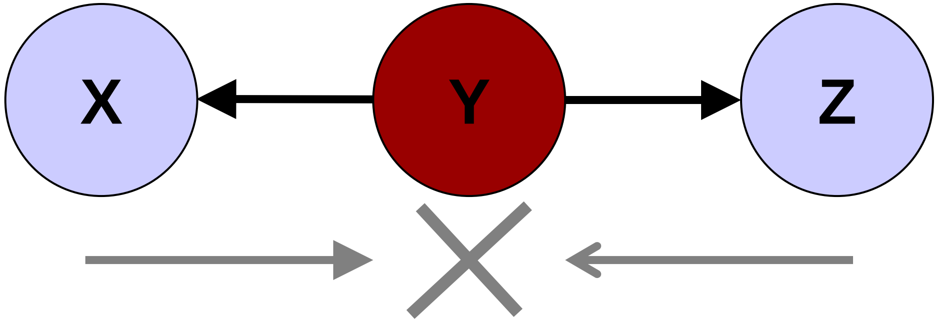
$$\boxed{X \perp Z \mid Y}$$

Diverging

Conditional Independence



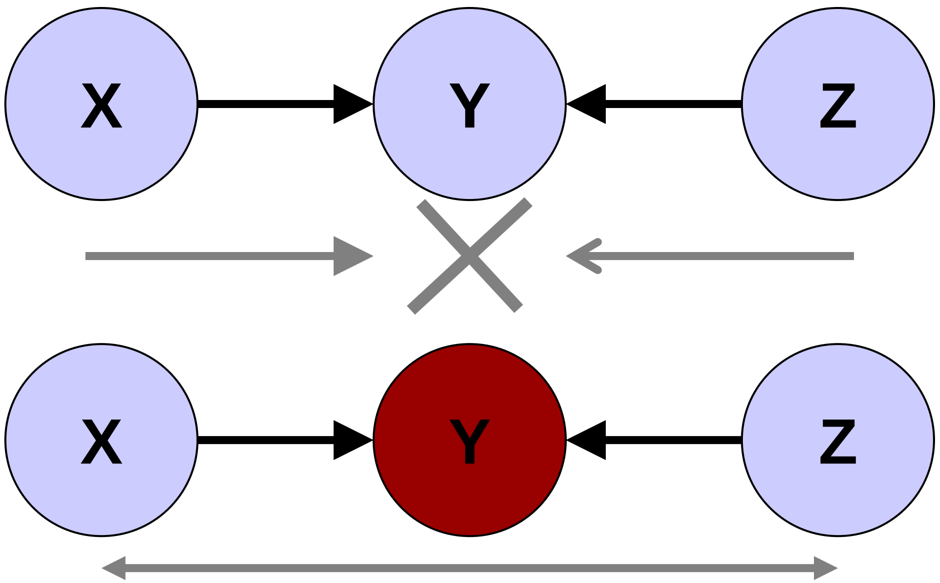
$$X \not\perp Z \mid \emptyset$$



$$\boxed{X \perp Z \mid Y}$$

Converging

Conditional Dependence - Explaining Away

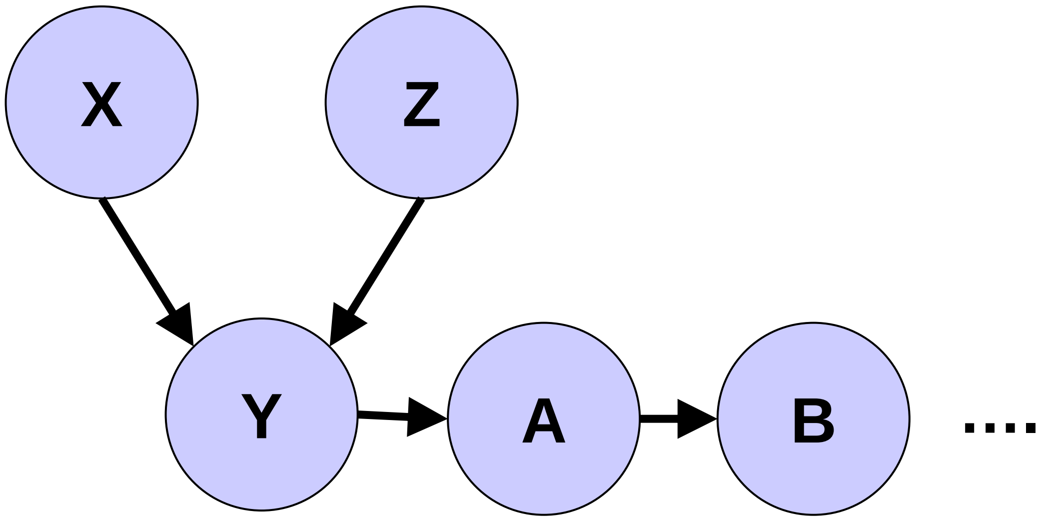


$$X \perp Z \mid \emptyset$$

$$\boxed{X \not\perp Z \mid Y}$$

Graph Semantics

Three basic building blocks



Converging

D-Separation

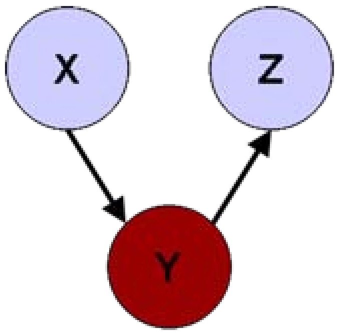
Three semantics combine in concept of d-separation

Definition : Two nodes A and B are **d-separated** (or **blocked**) if for every path *p* between A and B there is a node V such that either

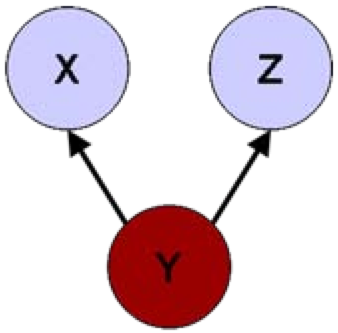
1. The connection is serial or diverging and V is known

2. The connection is converging and V *and all of its descendants* are unknown

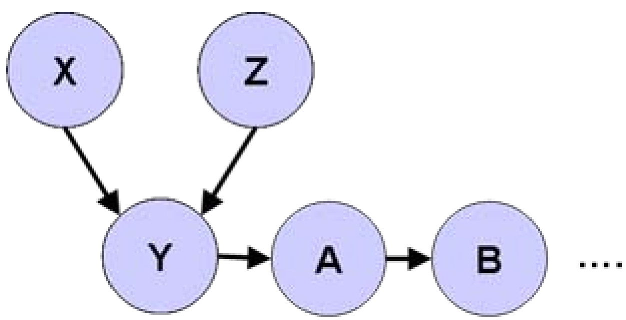
If A and B are not d-separated, they are d-connected



Serial



Diverging

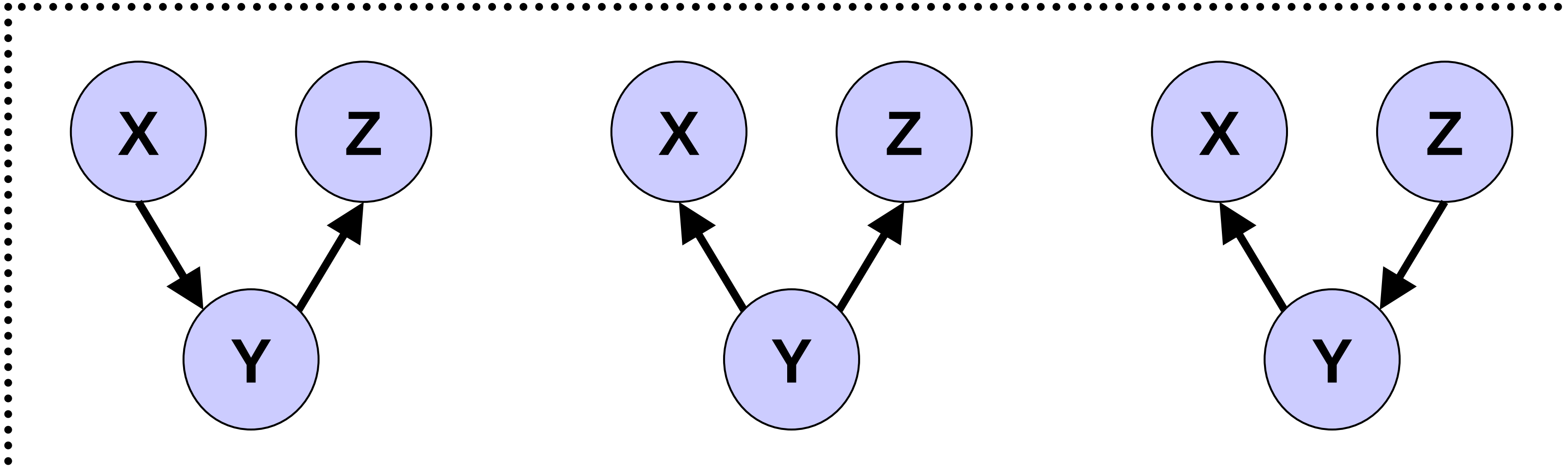


Converging

Equivalence of Networks

Two structures are equivalent if they represent same independence relationship - they encode the same space of probability distributions

Example



Will return to this when we consider causal vs probabilistic networks

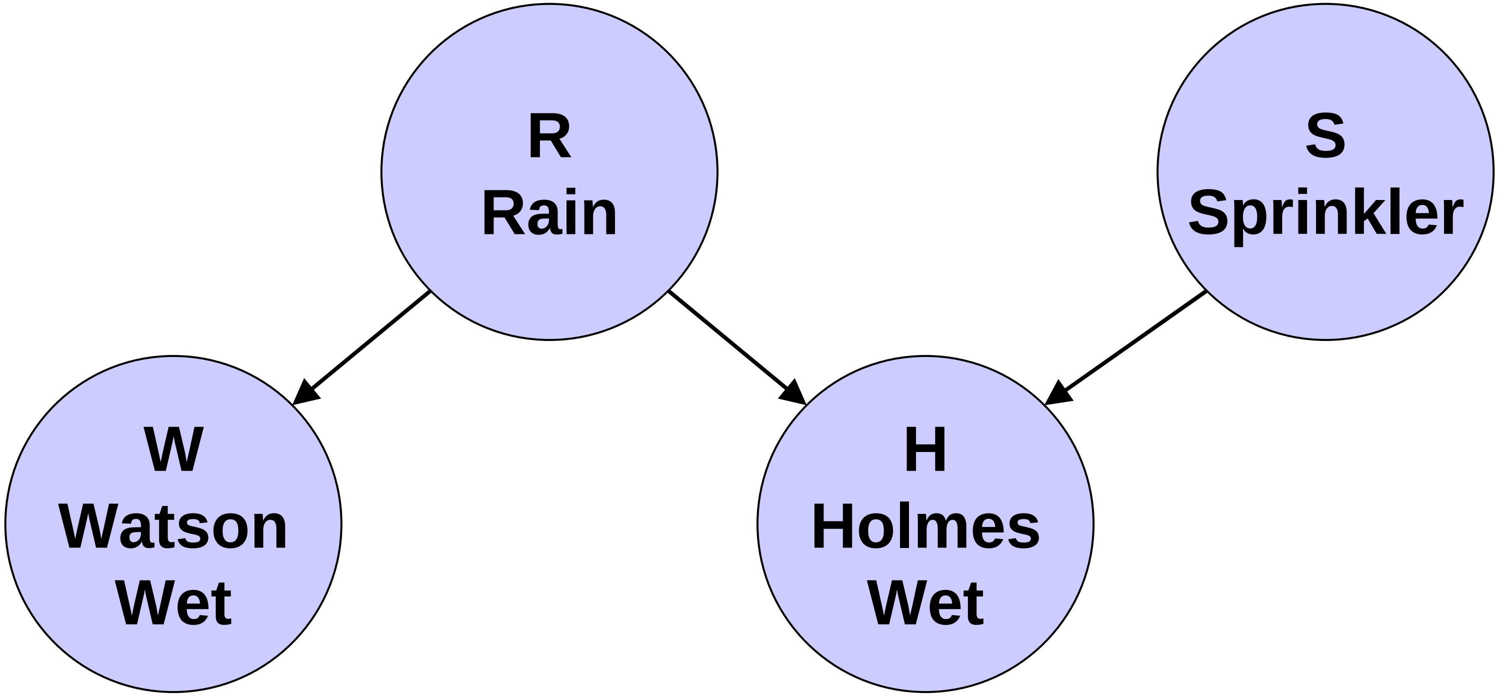
Bayesian Networks

A Bayesian network (BN) for $\mathcal{X} = \{X_1, X_2, X_3, \ldots, X_n\}$ consists of:

- A network structure S
  - Directed acyclic graph (DAG)
  - Nodes => random variables $\mathcal{X}$
  - *Encodes graph independence sematics*

- Set of probability distributions $\mathcal{P}$
  - Conditional Probability Distribution (CPD)
  - Local distributions for X

Example Bayesian Network



$$P(X) = P(R)P(S \mid R)P(W \mid R,S)P(H \mid R,S,W)$$

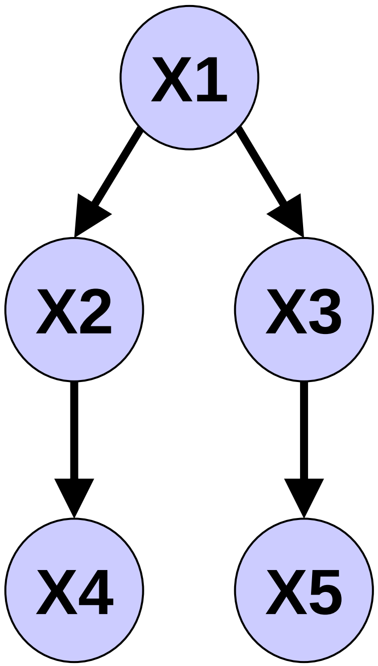
BN Probability Distribution

Only need distributions over nodes and their parents

$$P(X) = \prod_{i=1}^{n} P(X_i \,|\, X_1, \ldots, X_{i-1})$$

$$= \prod_{i=1}^{n} P(X_i \,|\, pa(X_i))$$

BNs are Compact Descriptions of P(X)



Independencies allow us to *factorize* distribution

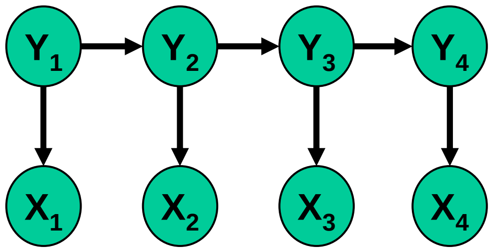
Example

- Assume 2 states per node

$$P(X) = \mathrm{P}(\mathrm{X}_1)\mathrm{P}(\mathrm{X}_2|\mathrm{X}_1)\mathrm{P}(\mathrm{X}_3|\mathrm{X}_2,\mathrm{X}_1)$$
$$\mathrm{P}(\mathrm{X}_4|\mathrm{X}_3,\mathrm{X}_2,\mathrm{X}_1)\mathrm{P}(\mathrm{X5}|\mathrm{X}_4,\mathrm{X}_3,\mathrm{X}_2,\mathrm{X}_1)$$
$$\Rightarrow 2+4+8+16+32=62 \text{ entries}$$

$$P(X) = \prod_{i=1}^{n} P(X_i \,|\, pa(X_i))$$
$$= \mathrm{P(X1)P(X2|X1)P(X3|X1)}$$
$$\mathrm{P(X4|X2)P(X5|X3)}$$
$$\Rightarrow 2+4+4+4+4=18 \text{ entries}$$

Recall from HMM/CRF Lecture

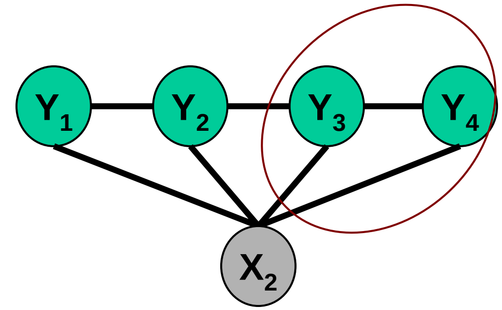


Directed Graph Semantics

Factorization

$$P(X,Y) = \prod_{\text{all nodes v}} P(v|\text{parents}(v))$$

$$= \prod P\left(Y_i \mid Y_{i-1}\right) P\left(X_i | Y_i\right)$$



Potential Functions over Cliques
(conditioned on X)

Markov Random Field

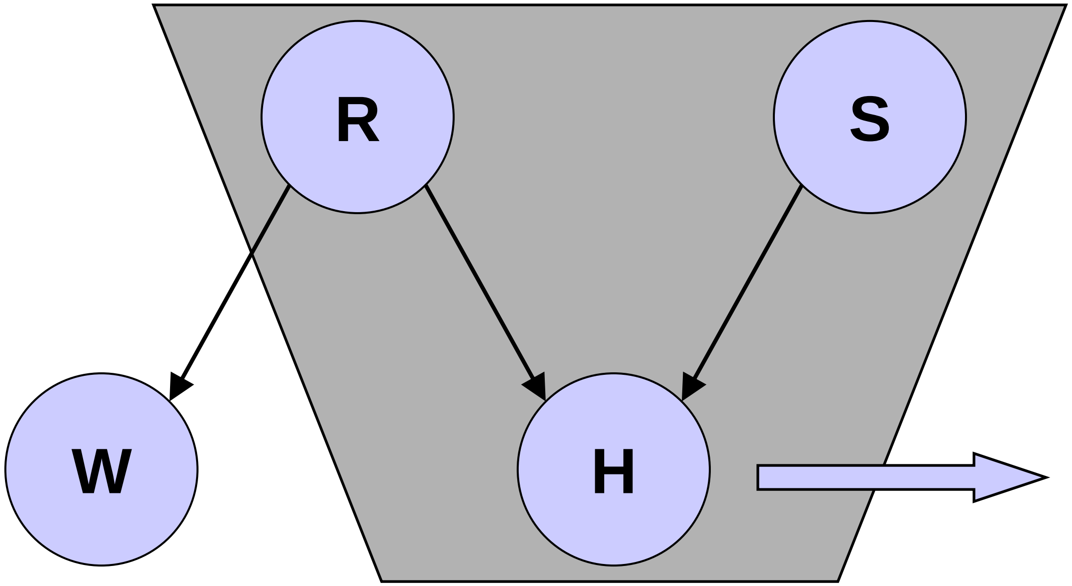
Factorization

$$P(Y|X) = \prod_{\text{all nodes v}} P(v|\text{clique}(v),X)$$

$$= \prod P\left(Y_i \mid Y_{i-1},X\right)$$

CPDs

Discrete



R	S	H	P(H|R,S)
0	0	1	0.1
0	0	0	0.9
0	1	1	0.8
0	1	0	0.2
1	0	1	0.9
1	0	0	0.1
1	1	1	0.1
1	1	0	0.9

Continuous

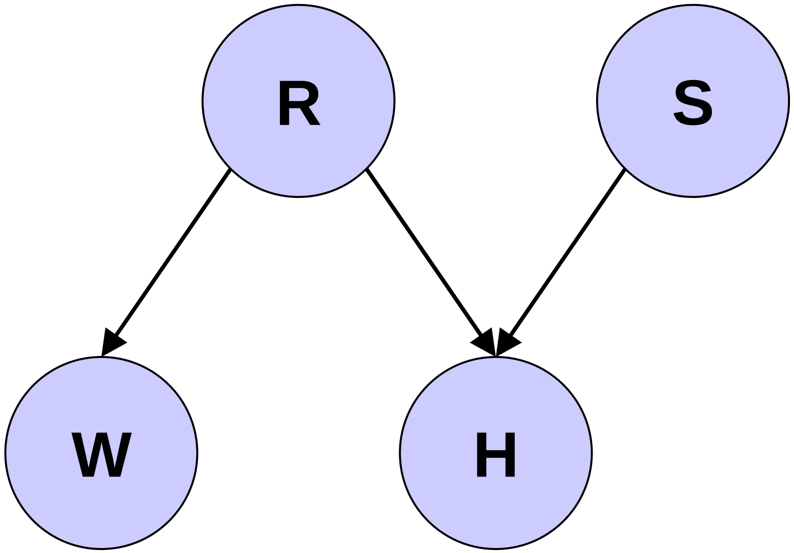
$$P(X \mid Y_1, \ldots, Y_N) = N\left(a_o + \sum_{i=1}^{N} a_i Y_i, \sigma^2 \right)$$

Bayesian Networks for Inference

Observational inference

- Observe values (evidence on) of a set of nodes, want to predict state of other nodes
- Exact Inference
  - Junction Tree Algorithm
- Approximate Inference
  - Variational approaches, Monte Carlo sampling

Walking Through an Example



P_0(R)

R=y	R=n
0.2	0.8

P_0(S)

S=y	S=n
0.1	0.9

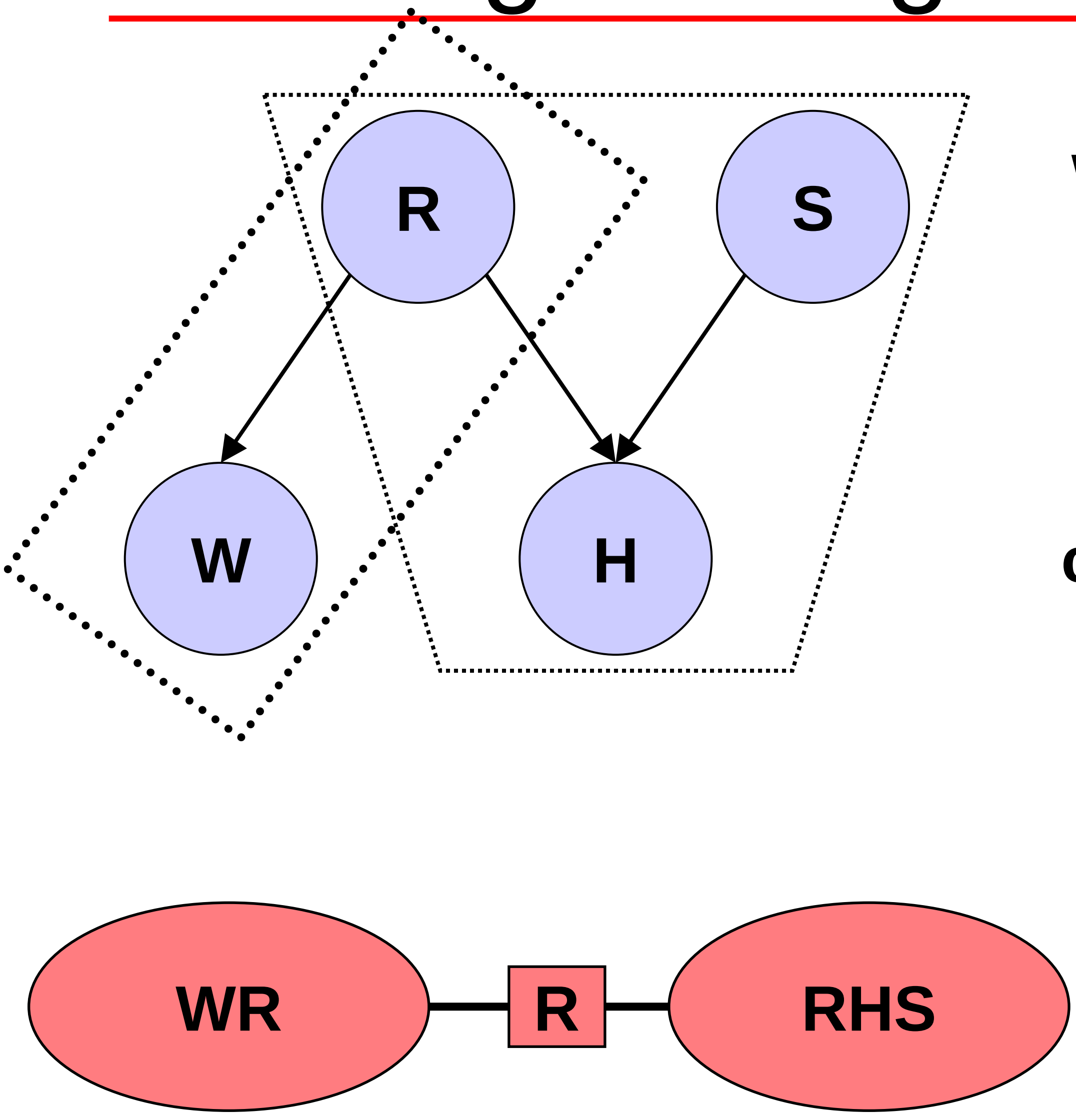
P_0(W|R)

	R=y	R=n
W=y	1	0.2
W=n	0	0.8

P_0(H|R,S)

	R=y	R=n
S=y	1,0	0.9,0.1
S=n	0,0	0,1

Walking Through an Example



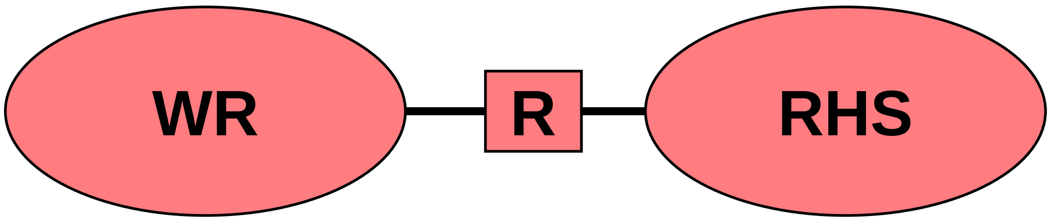
We define two clusters:
-WR, RHS

The **key idea**: the clusters only communicate through R

If they agree on R, all is good

Walking Through an Example

We will find it easier to work on this representation:



We then need P(WR) and P(RHS):

P(WR) =P(R)P(W|R)
P(RHS) =P(R)P(S)P(H|R,S)

P_0(R)

R=y	R=n
0.2	0.8

P_0(S)

S=y	S=n
0.1	0.9

P_0(W|R)

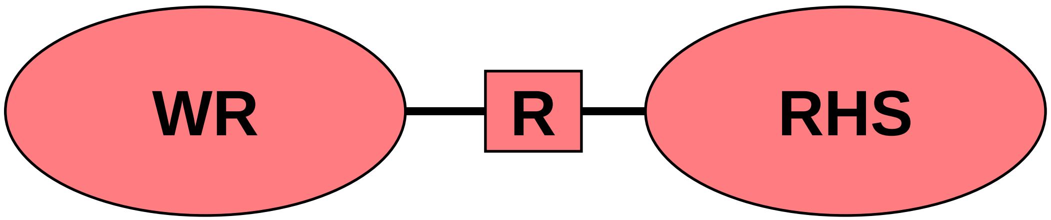
	R=y	R=n
W=y	1	0.2
W=n	0	0.8

P_0(H|R,S)

	R=y	R=n
S=y	1,0	0.9,0.1
S=n	0,0	0,1

Walking Through an Example

We will find it easier to work on this representation:



$P_0(R)$

R=y	R=n
0.2	0.8

We then need P(WR) and P(RHS):

$$P(WR) = P(R)P(W|R)$$
$$P(RHS) = P(R)P(S)P(H|R,S)$$

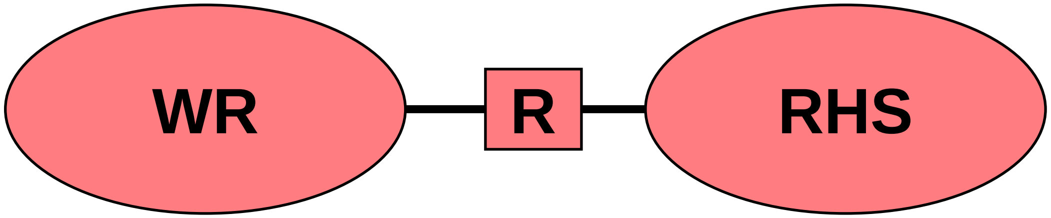
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_0(R,H,S)$

	R=y	R=n
S=y	0.02,0	0.072,0.008
S=n	0.18,0	0,0.72

Walking Through an Example



$P_0(R)$

R=y	R=n
0.2	0.8

Note that by marginalizing out W from $P_0(W,R)$ we get

$P_0(W)=(0.36,0.64)$

This is our initial belief in Watsons grass being (wet,not wet)

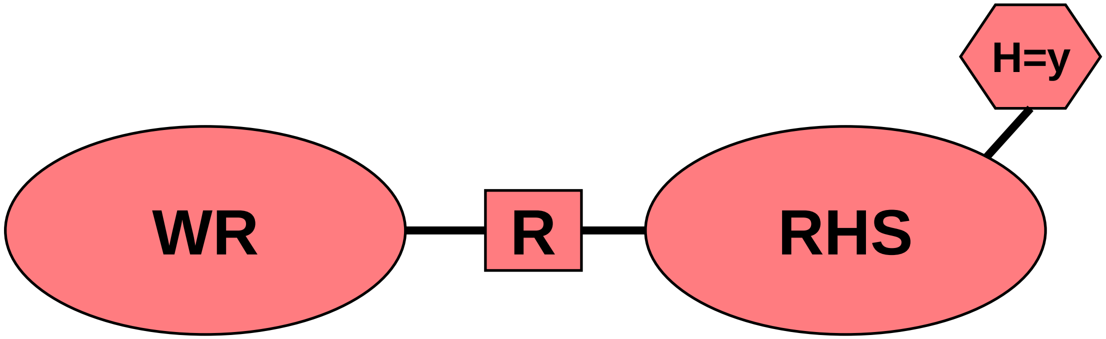
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_0(R,H,S)$

	R=y	R=n
S=y	0.02,0	0.072,0.008
S=n	0.18,0	0,0.72

Walking Through an Example



$P_0(R)$

R=y	R=n
0.2	0.8

Now we observe H=y

We need to do three things:

1. **Update RHS with this info**
2. **Calculate a new $P_1(R)$**
3. **Transmit $P_1(R)$ to update WR**

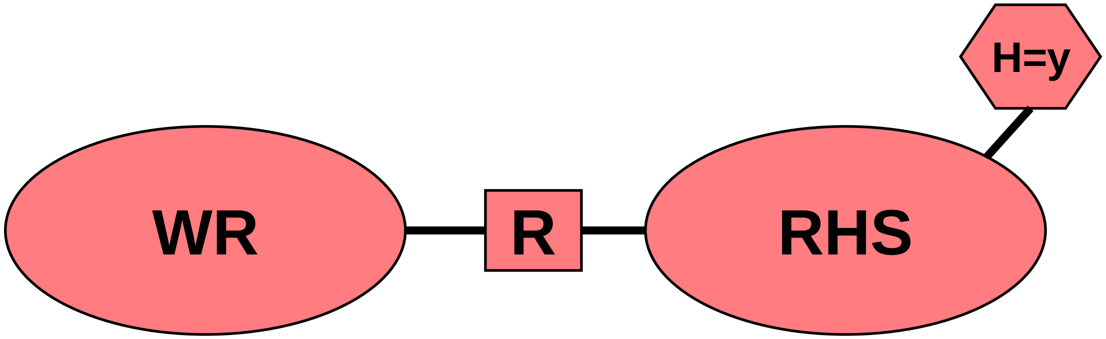
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_0(R,H,S)$

	R=y	R=n
S=y	0.02,0	0.072,0.008
S=n	0.18,0	0,0.72

Walking Through an Example



P_0(R)

R=y	R=n
0.2	0.8

Updating RHS with H=y

We can simply

- **Zero out all entries in RHS where H=n**

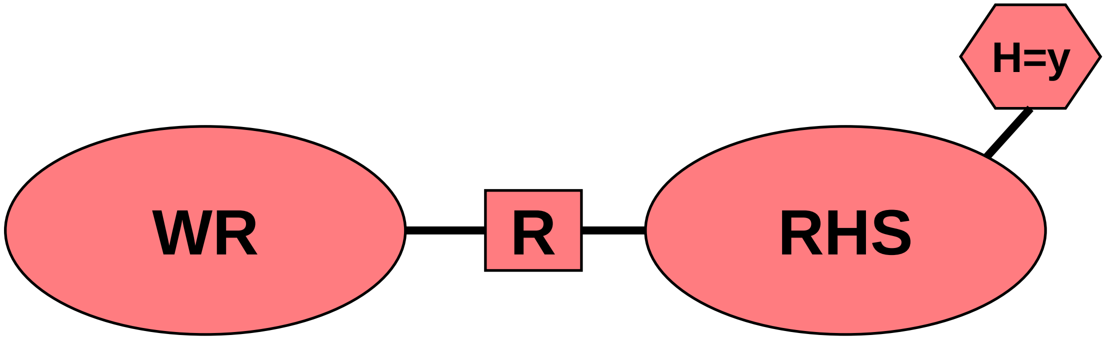
P_0(W,R)

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

P_0(R,H,S)

	R=y	R=n
S=y	0.02,0	0.072,**0**
S=n	0.18,0	0,**0**

Walking Through an Example



$P_0(R)$

R=y	R=n
0.2	0.8

Updating RHS with H=y

We can simply

- **Zero out all entries in RHS where H=n**
-

But you can see that this changes P(R) from the perspective of RHS

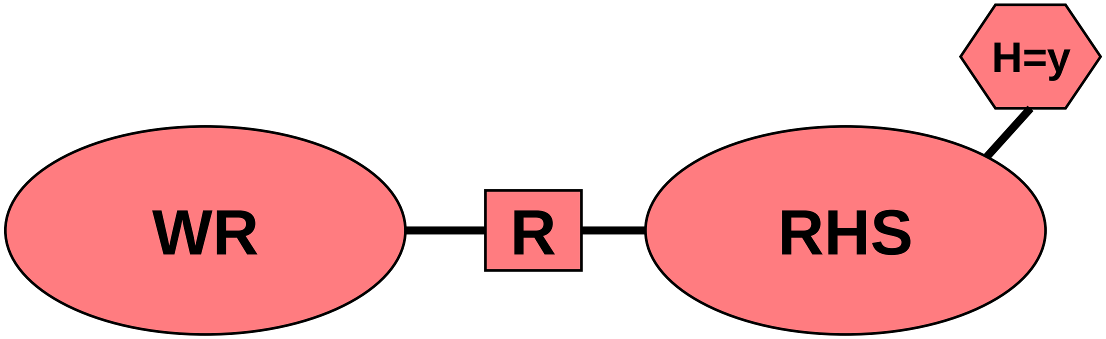
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_1(R,H,S)$

	R=y	R=n
S=y	**0.074**,0	**0.264**,0
S=n	**0.662**,0	0,0

Walking Through an Example



$P_0(R)$

R=y	R=n
0.2	0.8

2. Calculate new $P_1(R)$

Marginalize out H,S from RHS for:

$P_1(R)$ = (0.736,0.264)

Note also

$P_1(S)$ =(0.339,0.661)
$P_0(S)$ =(0.1,0.9)

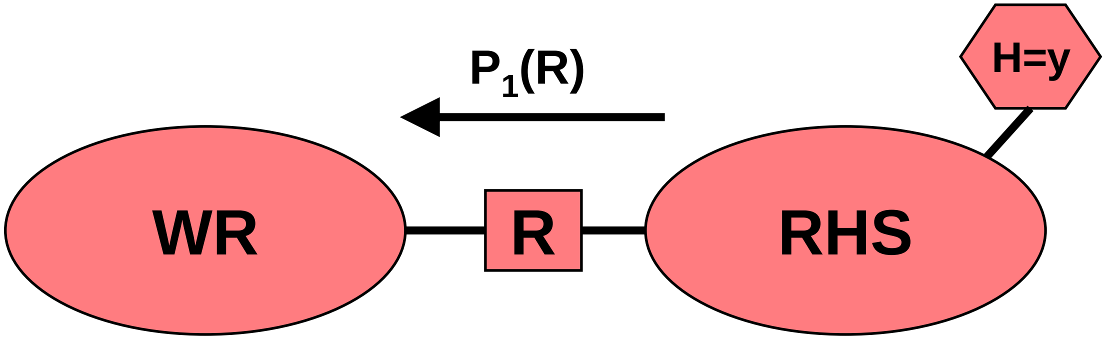
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_1(R,H,S)$

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



$P_1(R)$

R=y	R=n
0.736	**0.264**

2. Transmit $P_1(R)$ to update WR

$$P_1(W,R) = P(W|R)P_1(R)$$

$$= P_0(W,R)\frac{P_1(R)}{P_0(R)}$$

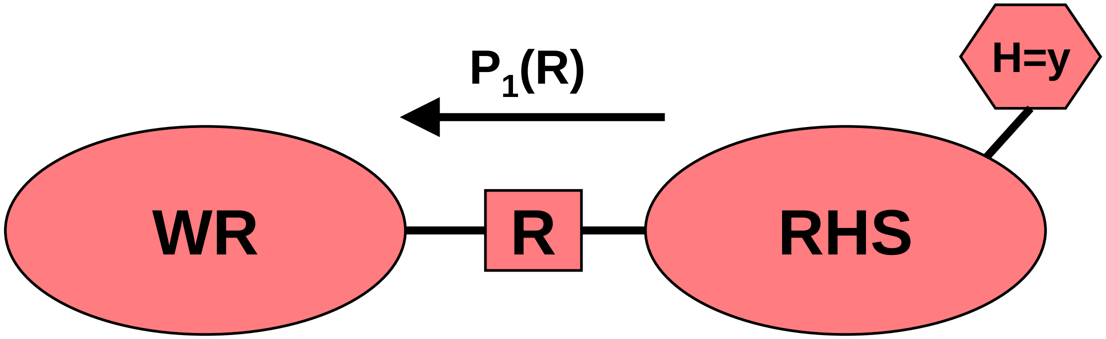
$P_0(W,R)$

	R=y	R=n
W=y	0.2	0.16
W=n	0	0.64

$P_1(R,H,S)$

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



P_1(R)

R=y	R=n
0.736	0.264

2. Transmit P_1(R) to update WR

$$P_1(W,R)=P(W|R)P_1(R)$$

$$=P_0(W,R)\frac{P_1(R)}{P_0(R)}$$

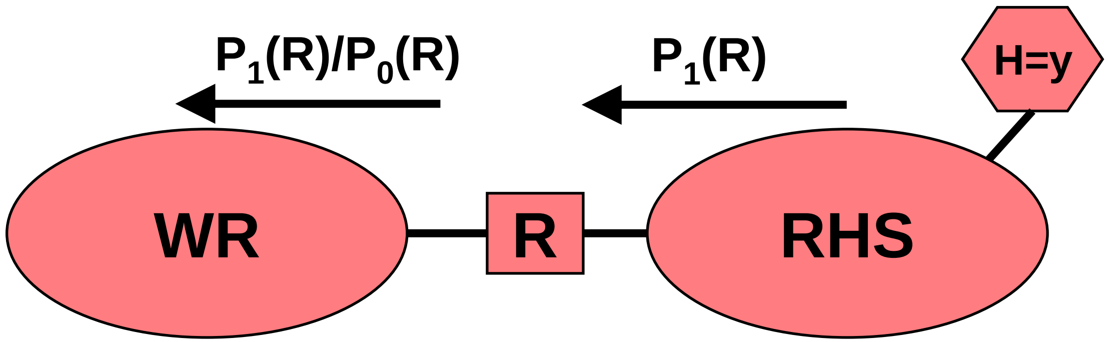
P_1(W,R)

	R=y	R=n
W=y	**0.736**	**0.052**
W=n	**0**	**0.211**

P_1(R,H,S)

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



$P_1(R)$

R=y	R=n
0.736	0.264

2. Transmit $P_1(R)$ to update WR

$$P_1(W,R)=P(W|R)P_1(R)$$

$$=P_0(W,R)\boxed{\frac{P_1(R)}{P_0(R)}}$$

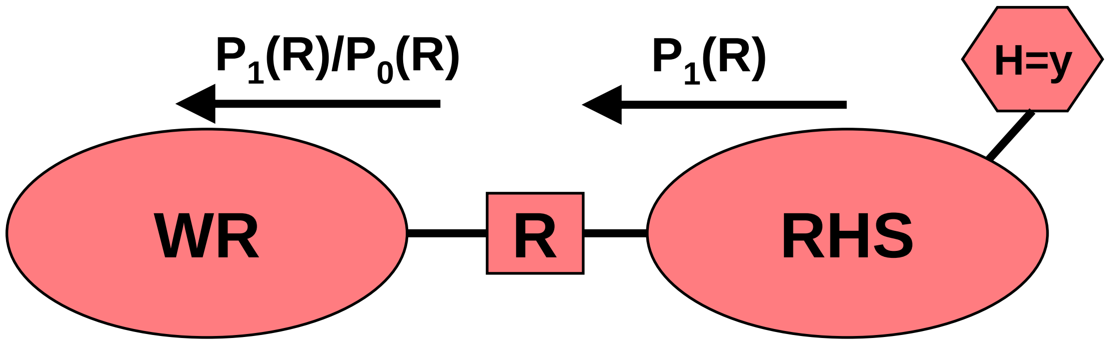
$P_1(W,R)$

	R=y	R=n
W=y	**0.736**	**0.052**
W=n	**0**	**0.211**

$P_1(R,H,S)$

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



$P_1(R)$

R=y	R=n
0.736	0.264

2. Transmit $P_1(R)$ to update WR

$$P_1(W,R)=P(W|R)P_1(R)$$

$$=P_0(W,R)\boxed{\frac{P_1(R)}{P_0(R)}}$$

$$P_1(W{=}y)=0.788$$

$$P_0(W{=}y)=0.360$$

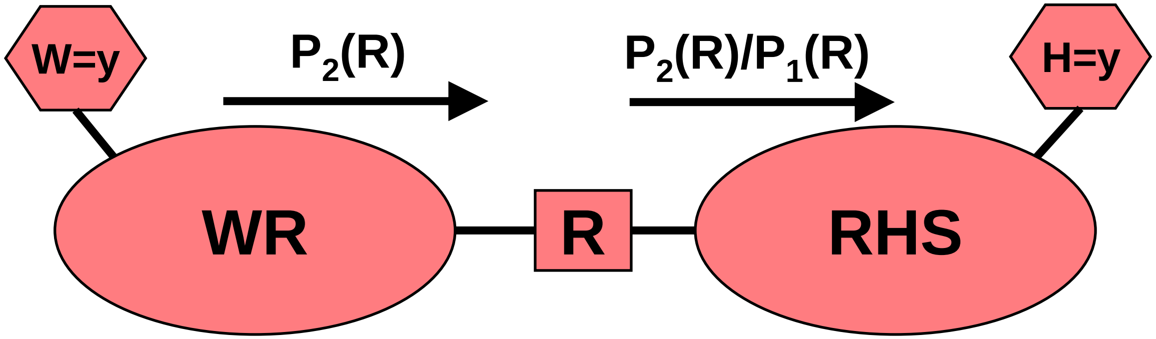
$P_1(W,R)$

	R=y	R=n
W=y	**0.736**	**0.052**
W=n	**0**	**0.211**

$P_1(R,H,S)$

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



$P_1(R)$

R=y	R=n
0.736	0.264

Now we observe W=y

1. **Update WR with this info**
2. **Calculate a new $P_2(R)$**
3. **Transmit $P_2(R)$ to update WR**

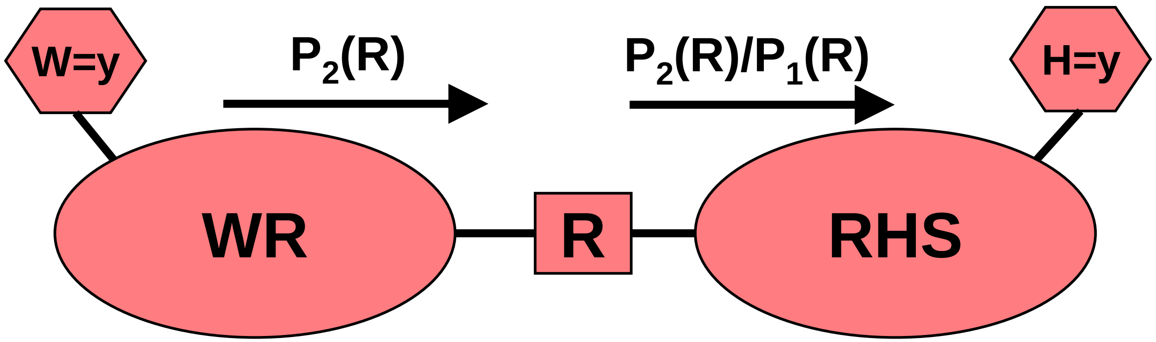
$P_1(W,R)$

	R=y	R=n
W=y	0.736	0.052
W=n	0	0.211

$P_1(R,H,S)$

	R=y	R=n
S=y	0.074,0	0.264,0
S=n	0.662,0	0,0

Walking Through an Example



$P_2(R)$

R=y	R=n
0.93	**0.07**

$P_2(S=y)$=0.161
$P_1(S=y)$=0.339
$P_0(S=y)$=0.1

$P_2(W,R)$

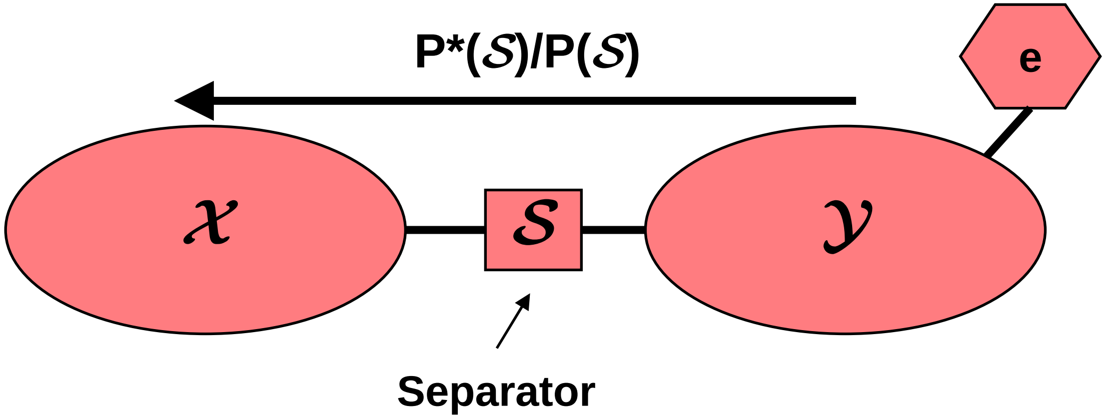
	R=y	R=n
W=y	**0.93**	**0.07**
W=n	0	**0**

- **R is almost certain**
- **We have *explained away* H=y**
- **S goes low again**

$P_2(R,H,S)$

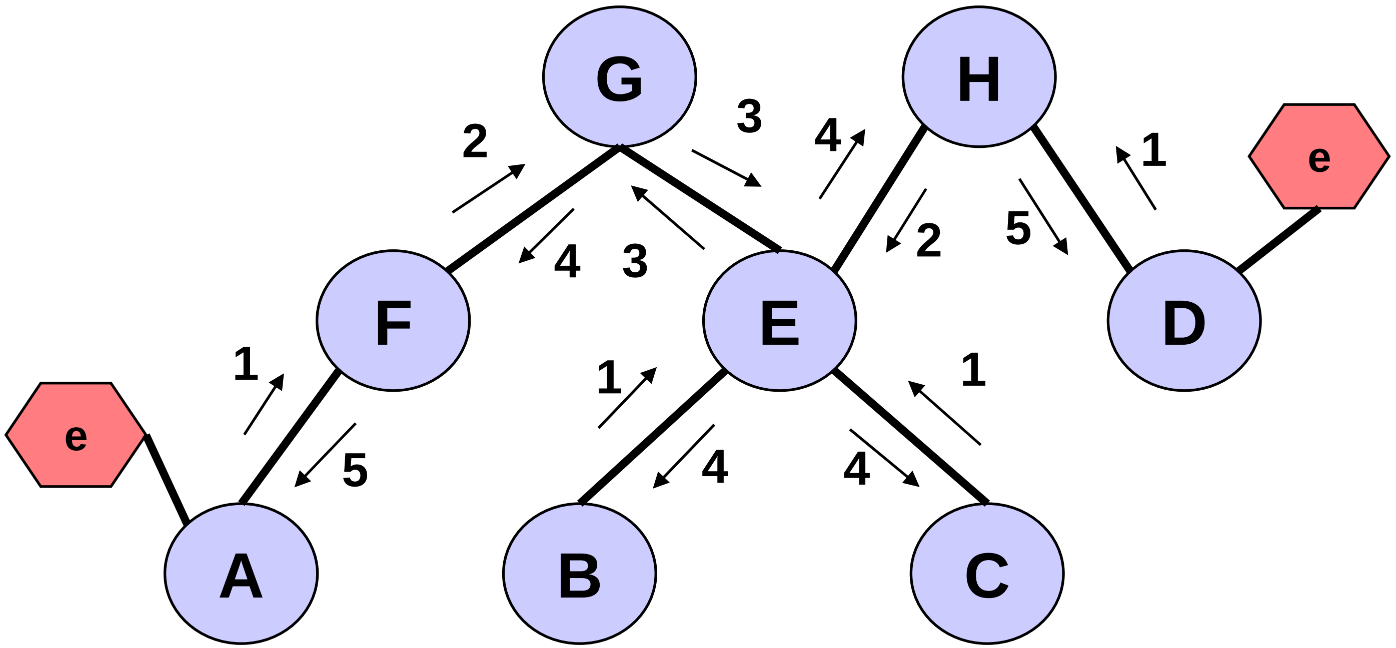
	R=y	R=n
S=y	**0.094**,0	**0.067**,0
S=n	**0.839**,0	0,0

Message Passing in Junction Trees



- State of separator $\mathcal{S}$ is information shared between $\mathcal{X}$ and $\mathcal{Y}$
- When $\mathcal{Y}$ is updated, it sends a message to $\mathcal{X}$
- Message has information to update $\mathcal{X}$ to agree with $\mathcal{Y}$ on state of $\mathcal{S}$

Message Passing in Junction Trees



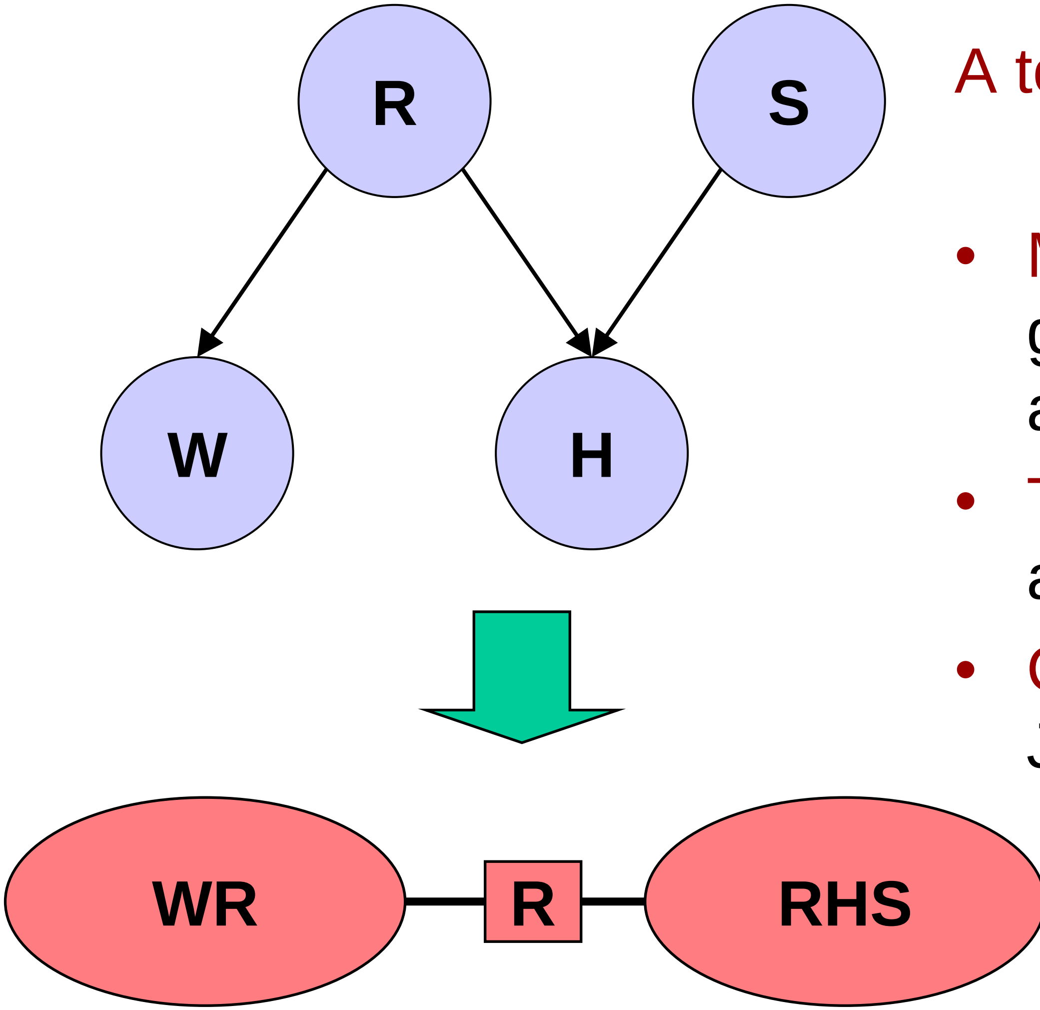
- A node can send one message to a neighbor, only after receiving all messages from each other neighbor
- When messages have been passed both ways along a link, it is consistent
- Passing continues until all links are consistent

HUGIN Algorithm

A simple algorithm for coordinated message passing in junction trees

- Select one node, V, as root
- Call CollectEvidence(V):
  - Ask all neighbors of V to send message to V.
  - If not possible, recursively pass message to all neighbors but V
- Call DistributeEvidence(V):
  - Send message to all neighbors of V
  - Recursively send message to all neighbors but V

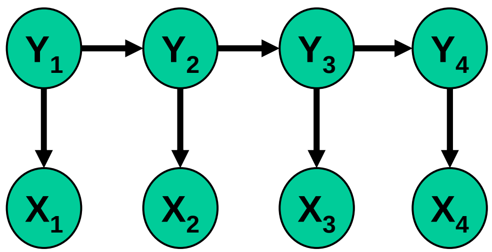
BN to Junction Tree



A topic by itself. In summary:

- Moral Graph – undirected graph with links between all parents of all nodes
- Triangulate – add links so all cycles>3 have cord
- Cliques become nodes of Junction Tree

Recall from HMM/CRF Lecture

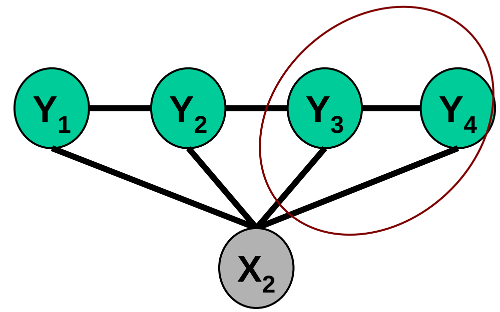


Directed Graph Semantics

Factorization

$$P(X,Y) = \prod_{\text{all nodes v}} P(v|\text{parents}(v))$$

$$= \prod P\left(Y_i \mid Y_{i-1}\right) P\left(X_i | Y_i\right)$$



Potential Functions over Cliques (conditioned on X)

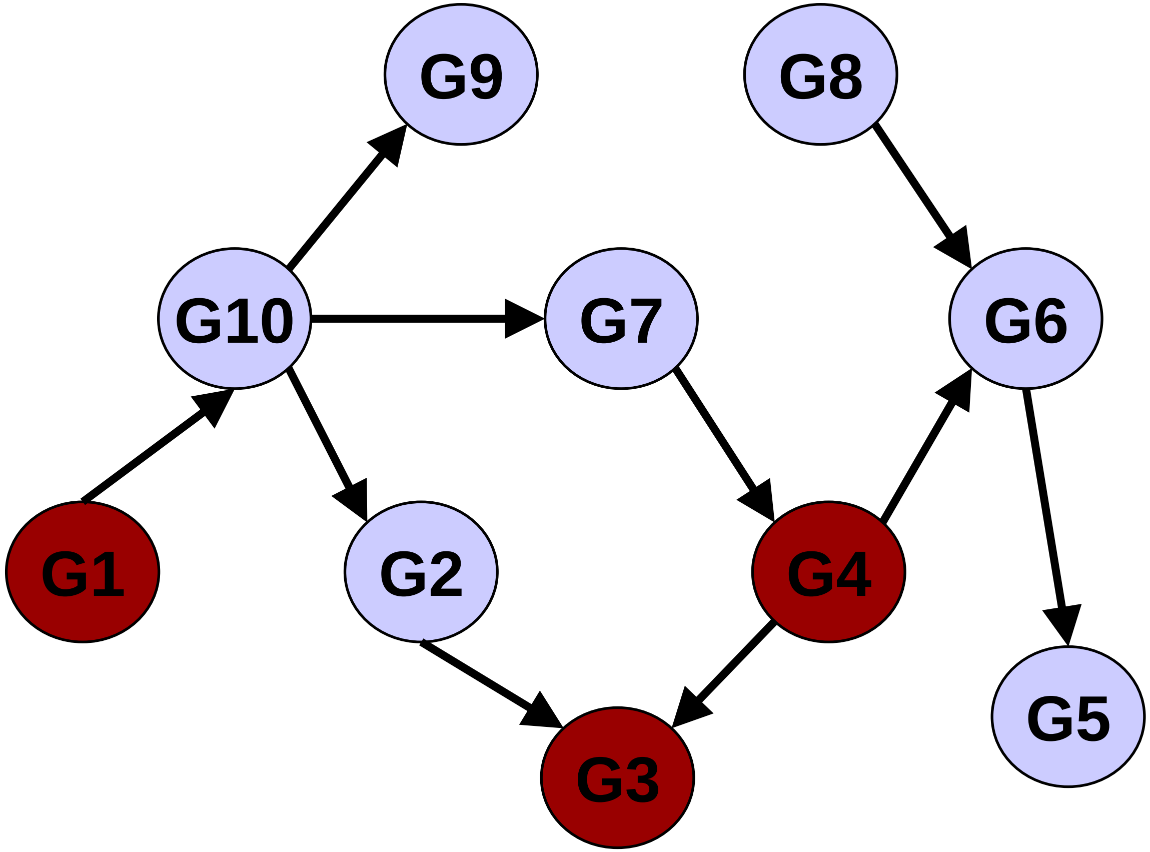
Markov Random Field

Factorization

$$P(Y|X) = \prod_{\text{all nodes v}} P(v|\text{clique}(v),X)$$

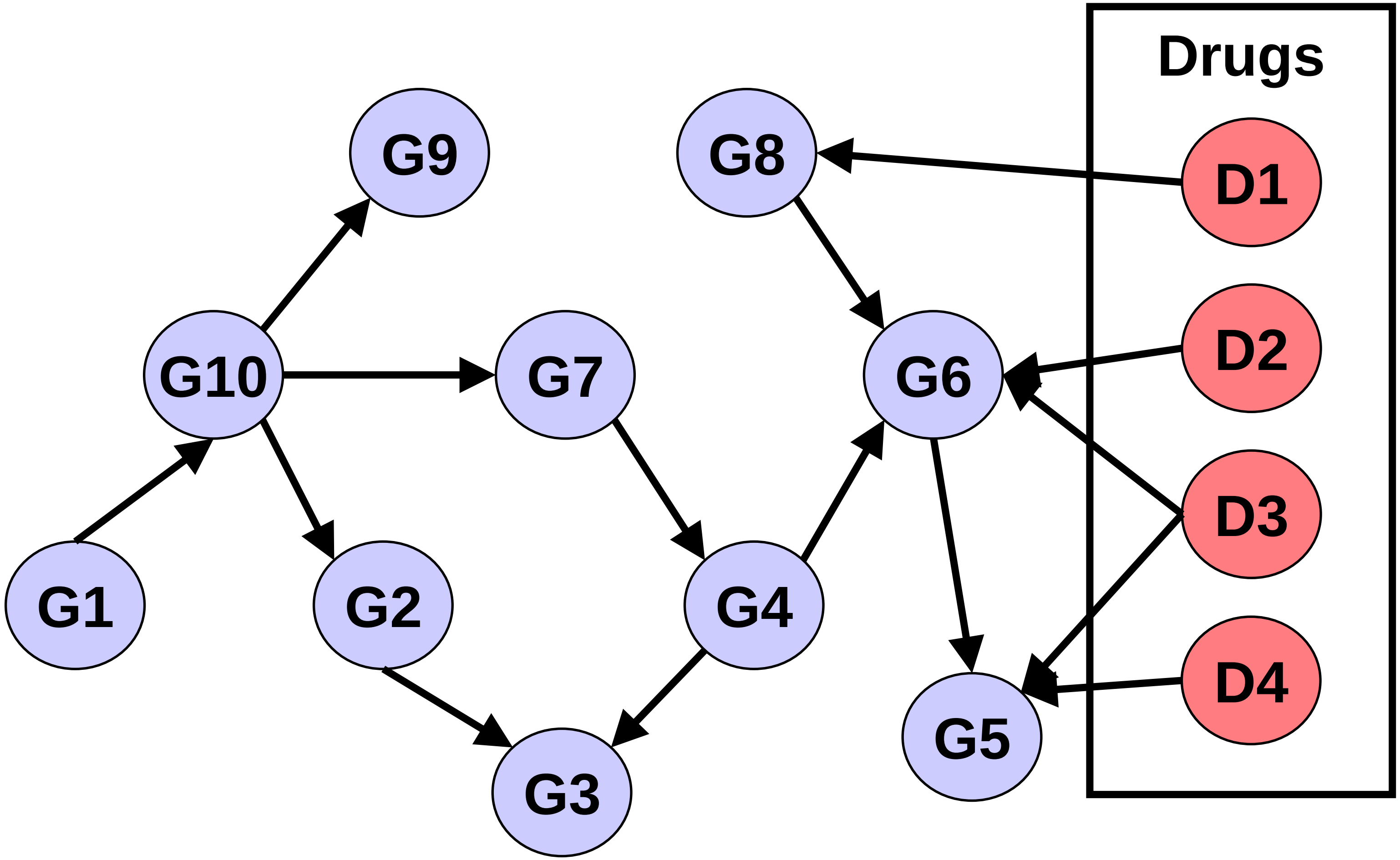
$$= \prod P\left(Y_i \mid Y_{i-1},X\right)$$

Applications

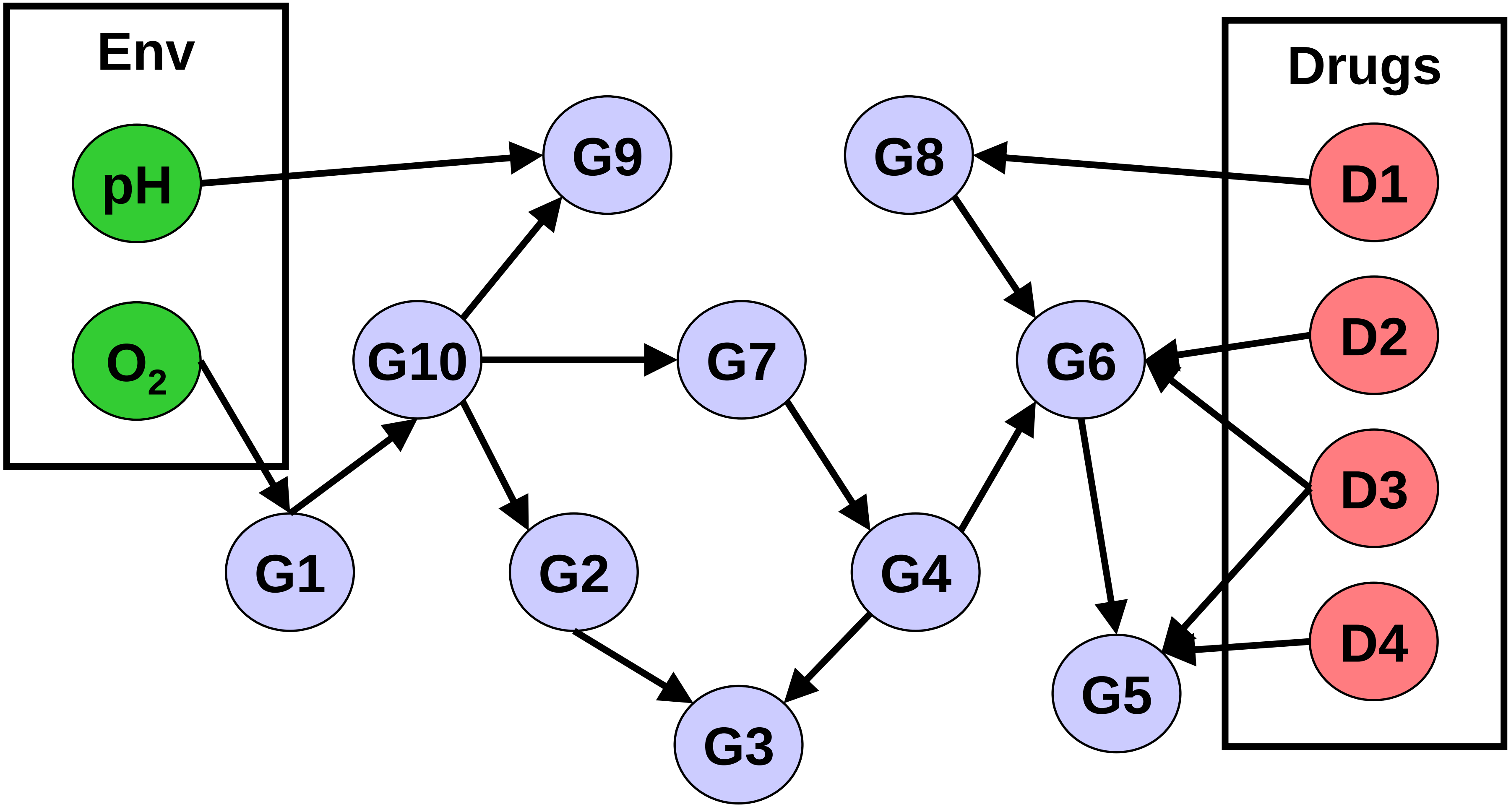


Measure some gene expression – predict rest

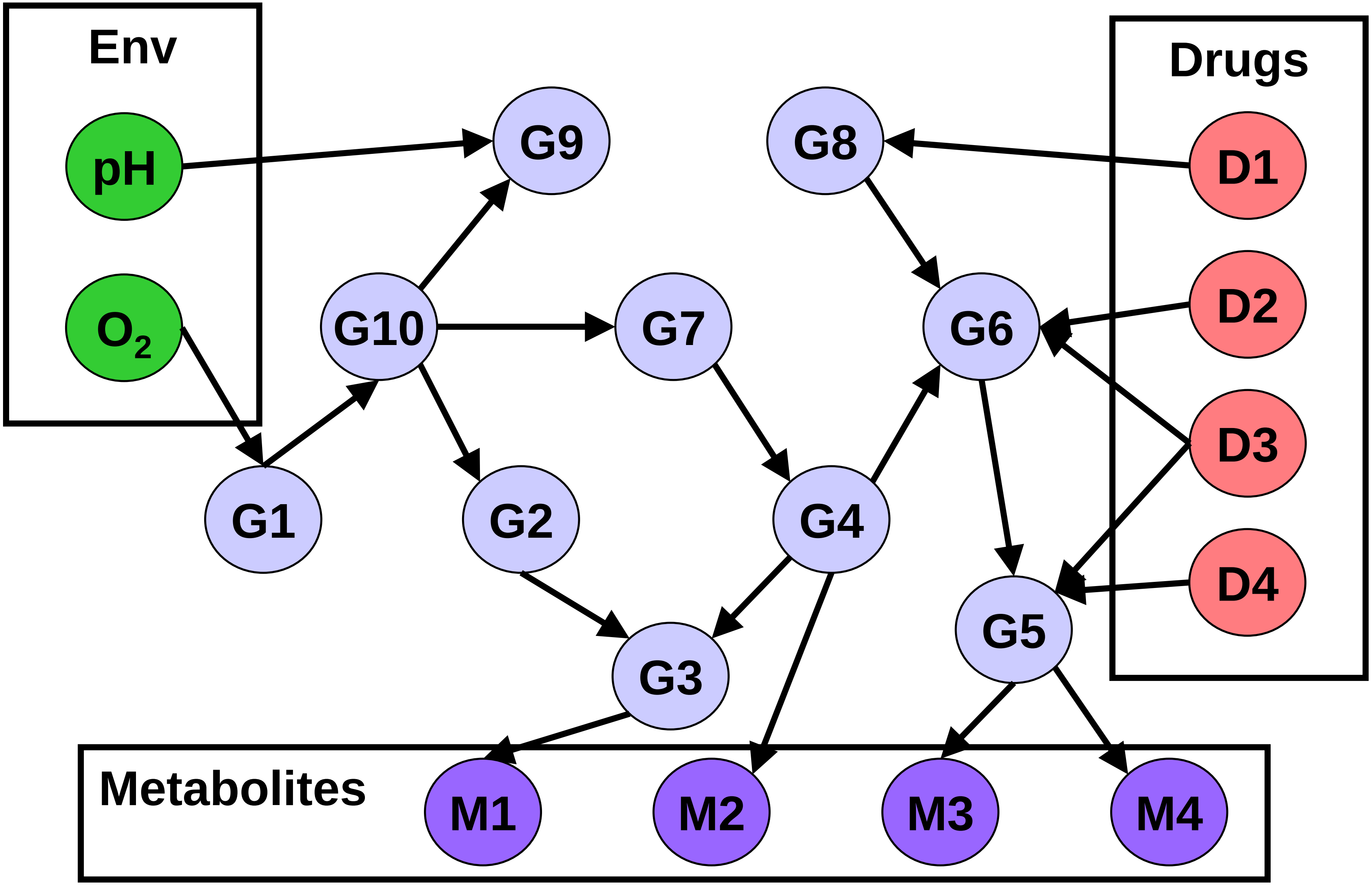
Latent Variables



Latent Variables



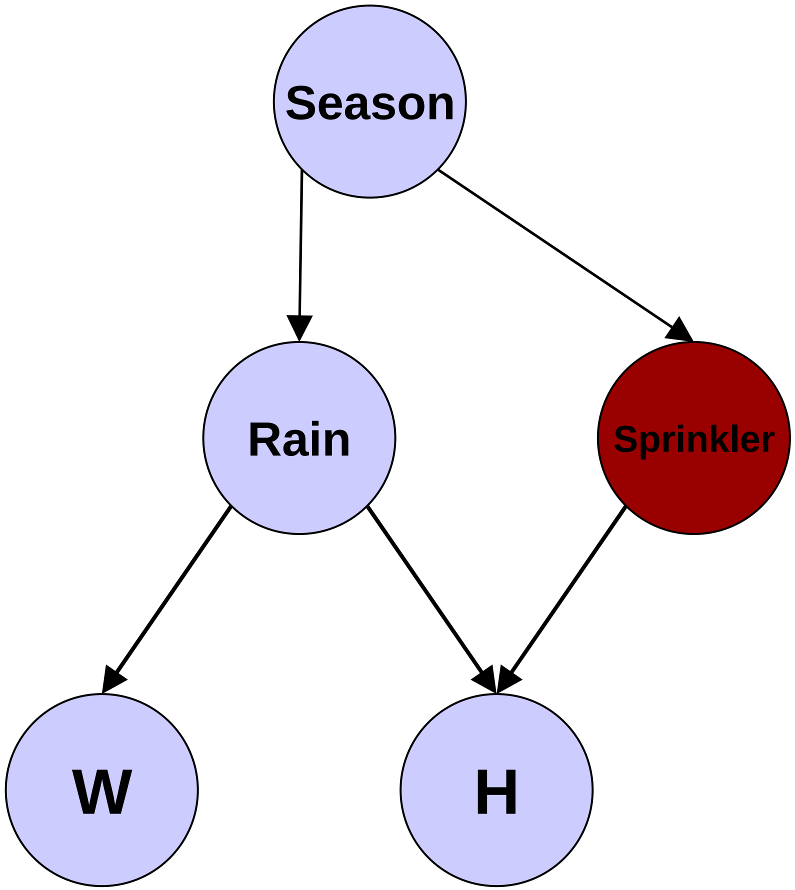
Latent Variables



Observation vs Intervention

- Arrows not necessarily causal
  - BN models probability and correlation between variables
- For applications so far, we *observe* evidence and want to know states of other nodes most likely to go with observation
- What about *interventions*?

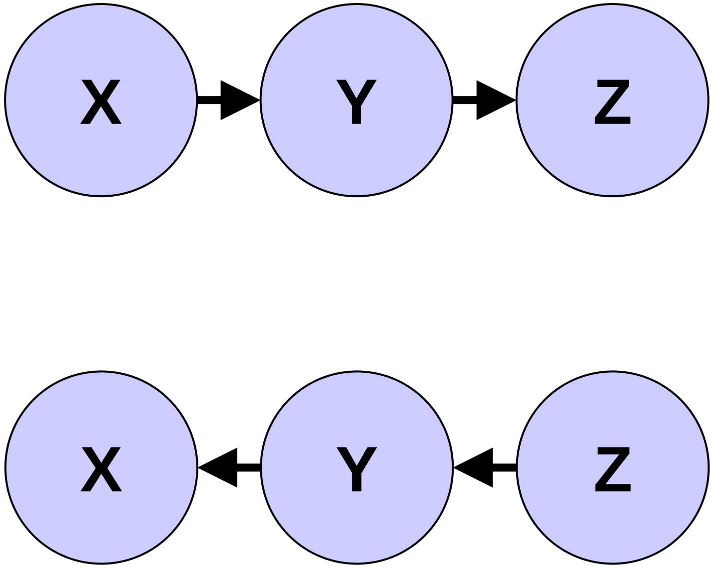
Example – Sprinkler



- If we **observe** the Sprinkler on
- Holmes grass more likely wet
- And more likely summer
- But what if we **force** sprinkler on

Intervention – cut arrows from parents

Causal vs Probabilistic



- This depends on getting the arrows correct!
- Flipping all arrows does not change independence relationships
- But changes causality for interventions

Learning Bayesian Networks

Given a set of observations D (i.e. expression data set) on X, we want to find:

1. A network structure $\mathcal{S}$

2. Parameters, Θ, for probability distributions on each node, given $\mathcal{S}$

Relatively Easy

Learning Θ

- Given S, we can choose maximum likelihood parameter Θ

$$\widehat{\theta} = \arg\max_{\theta} P(D \mid \theta, S) = \prod_{i=1}^{n} P(X_i \left| pa(X_i), \theta \right)$$

- We can also choose to include prior information $P(\Theta)$ in a bayesian approach

$$P(\theta \mid S, \mathrm{D}) = P(S, D \mid \theta) P(\theta)$$

$$\theta_{bayes} = \int \theta P(\theta \mid S, D) d\theta$$

Learning Bayesian Networks

Given a set of observations D (i.e. expression data set) on X, we want to find:

1. A network structure $\mathcal{S}$

   NP-Hard

2. Parameters, Θ, for probability distributions on each node, given $\mathcal{S}$

Learning $\mathcal{S}$

Find optimal structure $\mathcal{S}$ given D

$$P(S \mid D) \propto P(D \mid S)P(S)$$

$$P(D \mid S) = \int P(D \mid \theta, S)P(\theta \mid S)d\theta$$

In special circumstances, integral analytically tractable

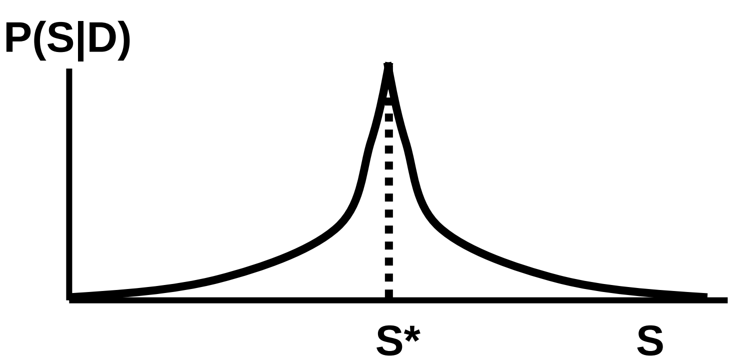
(e.g. no missing data, multinomial, dirichlet priors)

Learning $\mathcal{S}$ – Heuristic Search

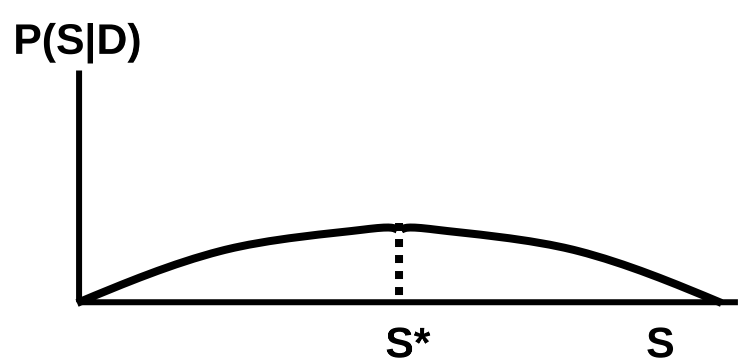
- Compute P(S|D) for all networks S
- Select S* that maximizes P(S|D)

Problem 1: number of S grows super-exponentially with number of nodes – no exhaustive search, use hill climbing, etc..

Problem 2: Sparse data and overfitting



Lots of data



Sparse data

Model Averaging

- Rather than select only one model as result of search, we can draw many samples from P(M|D) and model average

$$P(E_{xy} \mid D) = \sum_{samples} P(Exy \mid D, S) P(S \mid D)$$

$$= \sum_{samples} 1_{xy}(S) P(S \mid D)$$

How do we sample....?

Sampling Models - MCMC

Markov Chain Monte Carlo Method

Sample from $$P(S \mid D) = \frac{P(D \mid S_s) P(S_k)}{\sum_k P(D \mid S_s) P(S_k)}$$

Direct approach intractable due to partition function

MCMC

- **Propose – Given S_{old}, propose new S_{new} with probability $Q(S_{new}|S_{old})$**

- **Accept/Reject – Accept S_{new} as sample with**

$$p = \min\left\{1, \frac{P(D \mid S_{new}) P(S_{new})}{P(D \mid S_{old}) P(S_{old})} \times \frac{Q(S_{old} \mid S_{new})}{Q(S_{new} \mid S_{old})}\right\}$$