15-780: Grad AI
Lecture 19: Graphical models, Monte Carlo methods

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Erik Zawadzki, Abe Othman

Admin

- Reminder: midterm March 29
- Reminder: project milestone reports due March 31

Review: scenarios

- Converting QBF+ to PBI/MILP by scenarios
  - Replicate decision variables for each scenario
  - Replicate clauses: share first stage vars; set scenario vars by scenario index; replace decision vars by replicates
  - Sample random scenarios
- Example: PSTRIPS

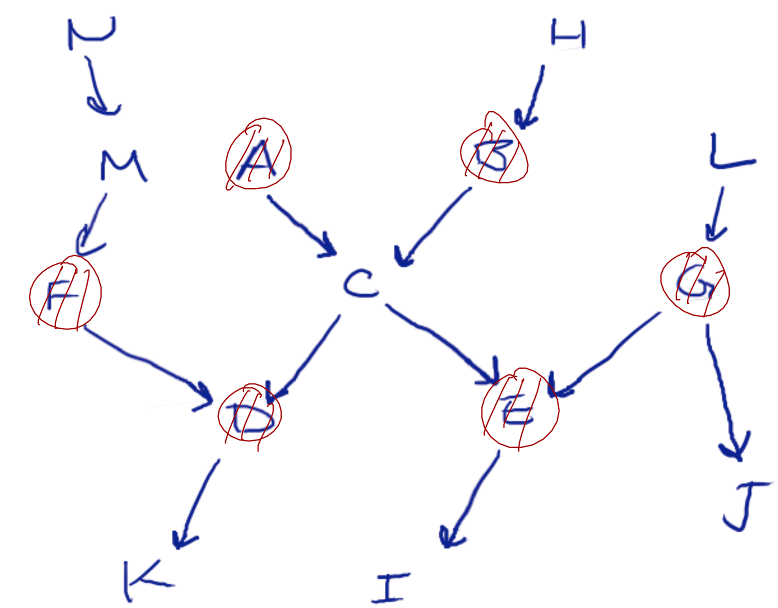
Review: dynamic programming

- Solving #SAT by dynamic programming (variable elimination)
  - repeatedly move sums inward, combine tables, sum out
  - treewidth and runtime/space

Review: graphical models

- Bayes net = DAG + CPTs
  - For each RV (say X), there is one CPT specifying P(X | pa(X))
  - Can simulate with propositional logic + random causes
- Inference: similar to #SAT DP—move sums inward
  - Can do partly analytically
  - Allows us to prove independences and conditional ind's from DAG alone

Review: graphical models

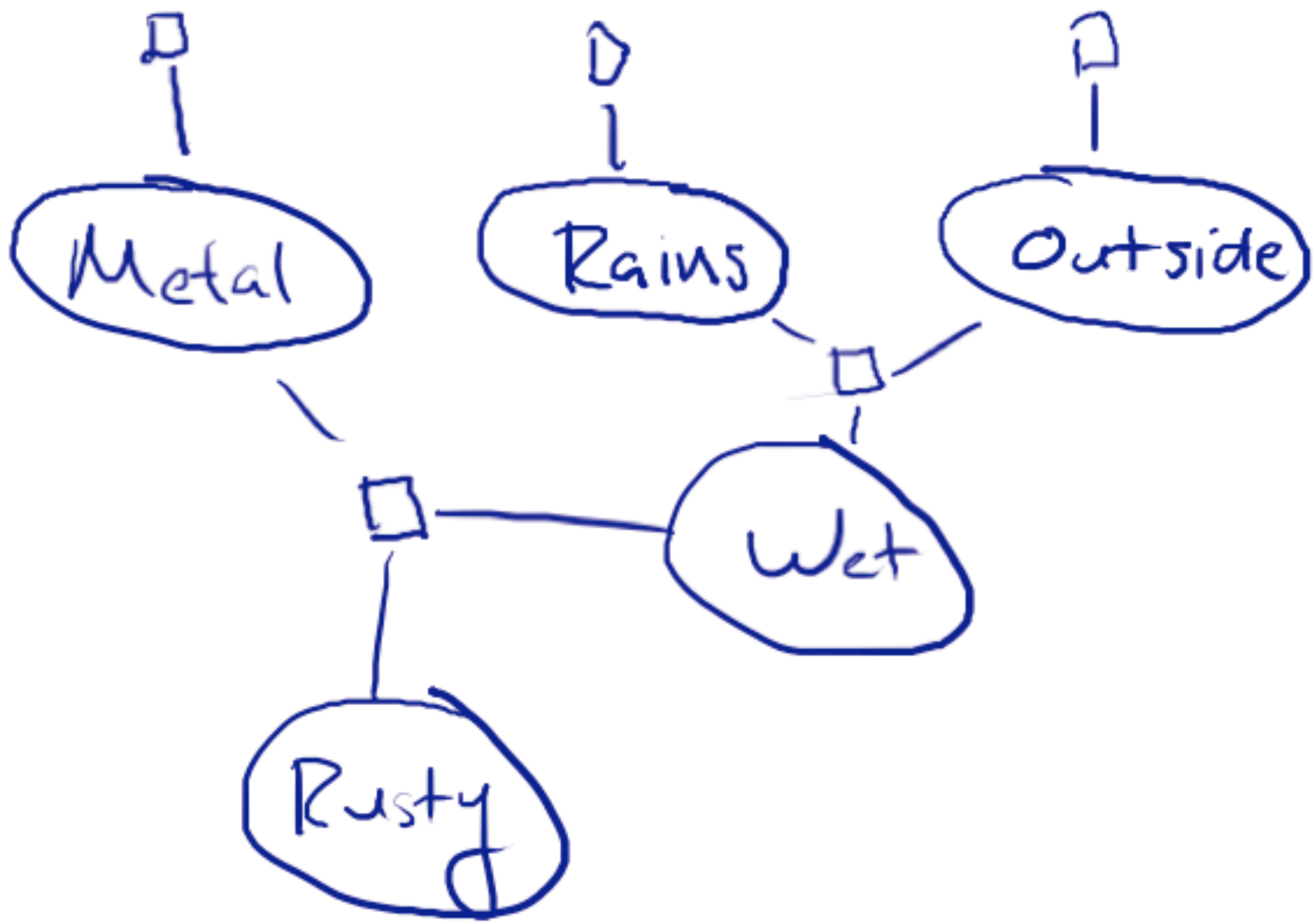


- Blocking, explaining away
- Markov blanket
- Learning: counting, Laplace smoothing
  - if hidden variables: take 10-708 or use a toolbox

Factor graphs

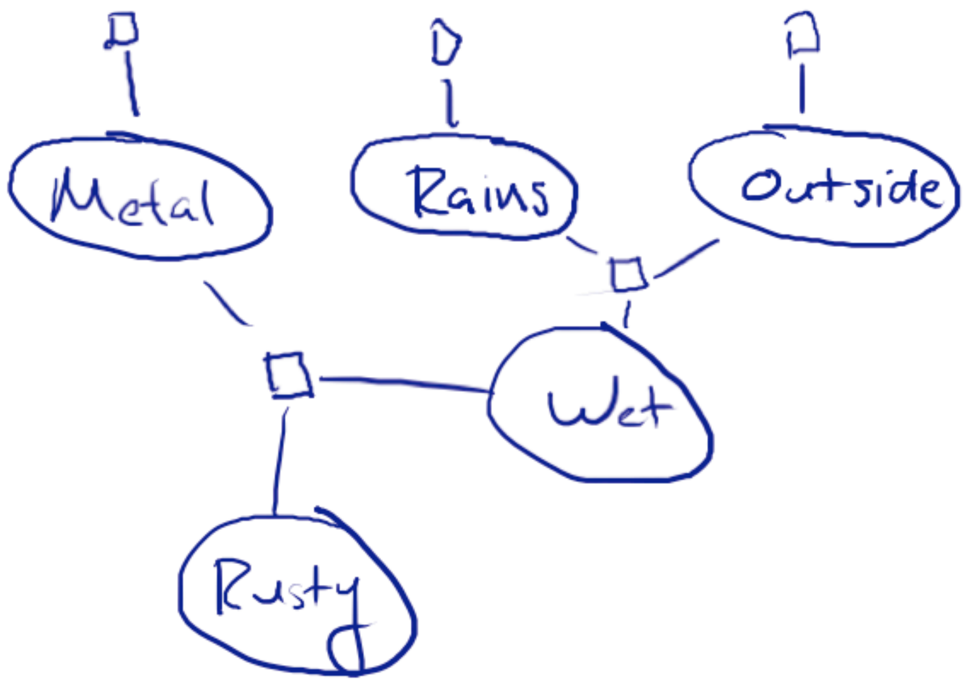
- Another common type of graphical model
- Uses ***undirected, bipartite*** graph instead of DAG

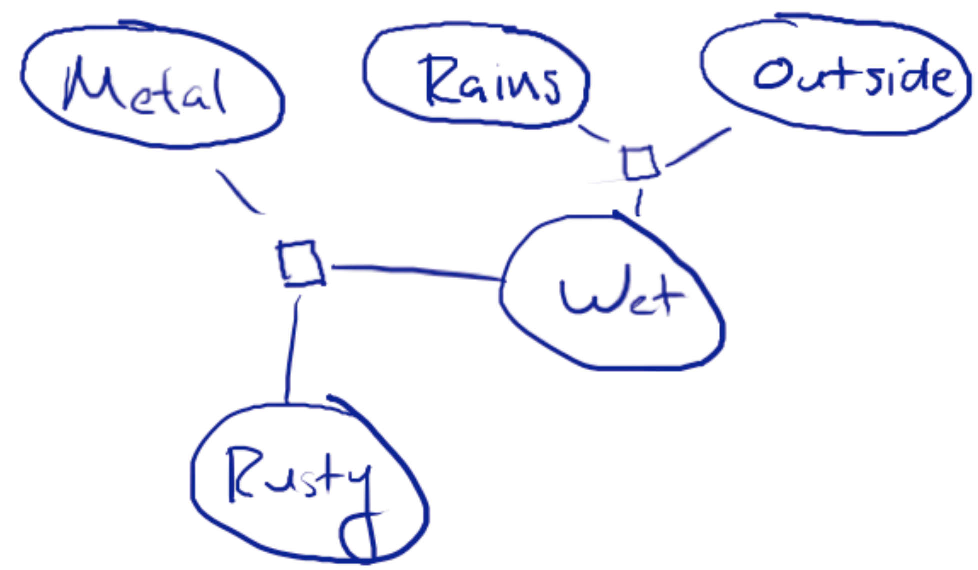
Rusty robot: factor graph



P(M) P(Ra) P(O) P(W|Ra,O) P(Ru|M,W)

Convention





- Don’t need to show unary factors
- Why? They don’t affect algorithms below.

Non-CPT factors

- Just saw: easy to convert Bayes net → factor graph
- In general, factors need not be CPTs: any nonnegative #s allowed
- In general, P(A, B, …) =
- Z =

Hard v. soft factors

Hard

X

Y		0	1	2
	0	0	0	0
Y	1	0	0	1
	2	0	1	1

Soft

X

Y		0	1	2
	0	1	1	1
Y	1	1	1	3
	2	1	3	3

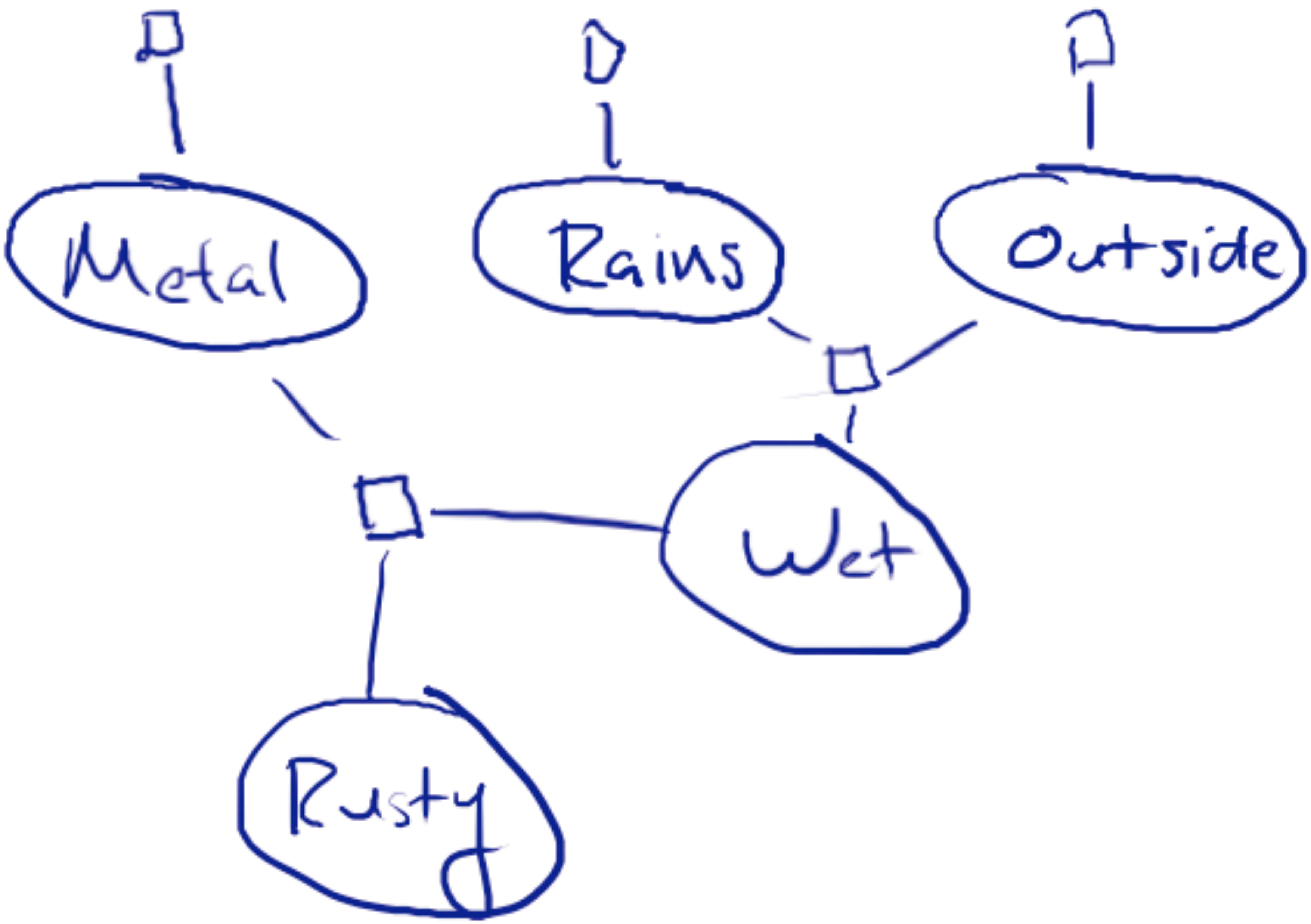
Factor graph → Bayes net

- Conversion possible, but more involved
  - Each representation can handle ***any*** distribution
  - But, size/complexity of graph may differ
- 2 cases for conversion:
  - without adding nodes:
  - adding nodes:

Independence

- Just like Bayes nets, there are graphical tests for independence and conditional independence
- Simpler, though:
  - Cover up all observed nodes
  - Look for a path

Independence example



Modeling independence

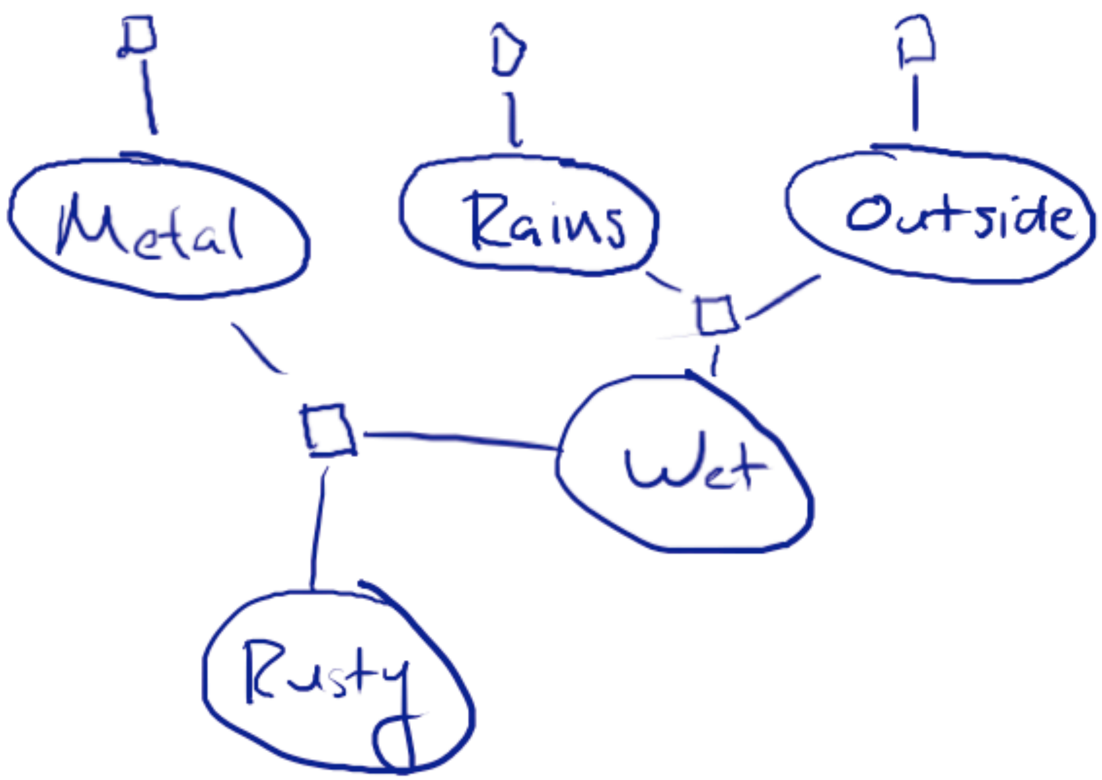
- Take a Bayes net, list the (conditional) independences
- Convert to a factor graph, list the (conditional) independences
- Are they the same list?
- What happened?

Inference

- Inference: prior + evidence → posterior
- We gave examples of inference in a Bayes net, but not a general algorithm
- Reason: general algorithm uses factor-graph representation
- Steps: instantiate evidence, eliminate nuisance nodes, normalize, answer query

Inference

$$P(M, Ra, O, W, Ru) = \phi_1(M)\,\phi_2(Ra)\,\phi_3(O)\,\phi_4(Ra, O, W)\,\phi_5(M, W, Ru) / Z$$



$\phi_1(M) =$

T	0.9
F	0.1

$\phi_2(Ra) =$

T	0.7
F	0.3

$\phi_3(O) =$

T	0.2
F	0.8

$\phi_4(Ra, O, W) =$

TTT	0.9
TTF	0.1
TFT	0.1
TFF	0.9
FTT	0.1
FTF	0.9
FFT	0.1
FFF	0.9

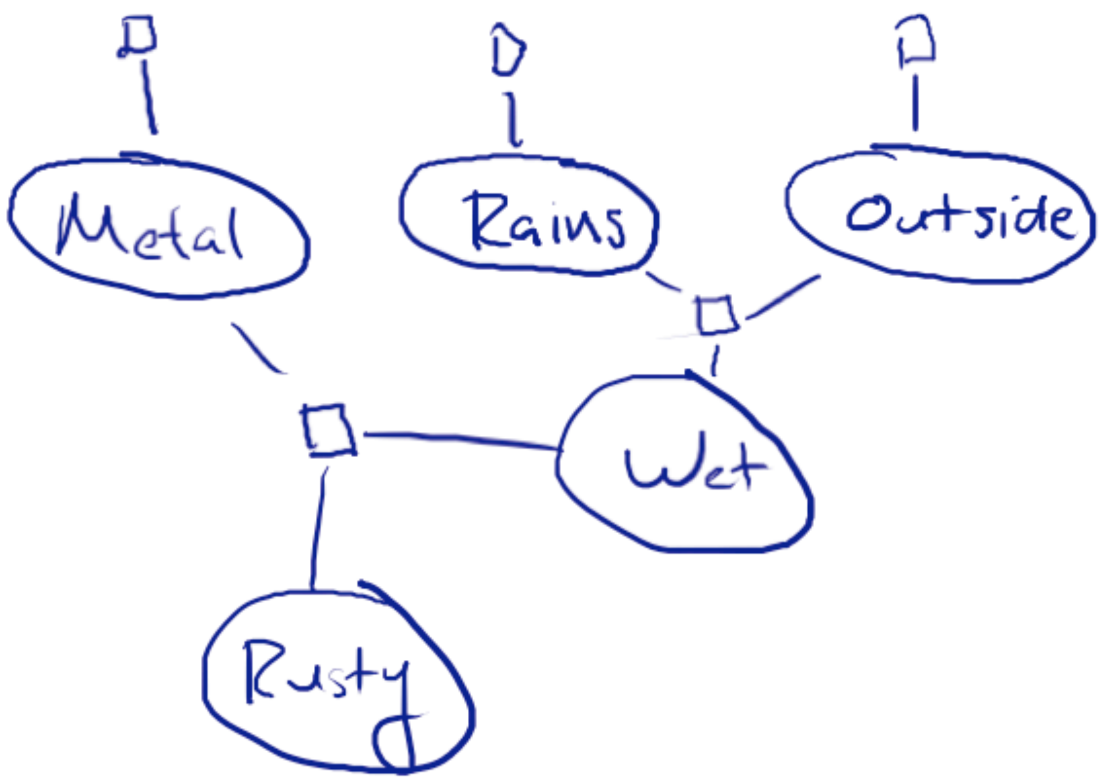
$\phi_5(M, W, Ru) =$

TTT	0.8
TTF	0.2
TFT	0.1
TFF	0.9
FTT	0
FTF	1
FFT	0
FFF	1

- Typical Q: given Ra=F, Ru=T, what is P(W)?

Incorporate evidence

$$P(M, Ra, O, W, Ru) = \phi_1(M)\,\phi_2(Ra)\,\phi_3(O)\,\phi_4(Ra, O, W)\,\phi_5(M, W, Ru)/Z$$



$\phi_1(M) =$

T	0.9
F	0.1

$\phi_2(Ra) =$

T	0.7
F	0.3

$\phi_3(O) =$

T	0.2
F	0.8

$\phi_4(Ra, O, W) =$

TTT	0.9
TTF	0.1
TFT	0.1
TFF	0.9
FTT	0.1
FTF	0.9
FFT	0.1
FFF	0.9

$\phi_5(M, W, Ru) =$

TTT	0.8
TTF	0.2
TFT	0.1
TFF	0.9
FTT	0
FTF	1
FFT	0
FFF	1

Condition on Ra=F, Ru=T

Eliminate nuisance nodes

$$P(M, \cancel{Ra}, O, W, \cancel{Ru}) = \phi_1(M)\,\cancel{\phi_2(Ra)}\,\phi_3(O)\,\phi_4(\cancel{Ra}, O, W)\,\phi_5(M, W, \cancel{Ru}) / Z$$

- Remaining nodes: M, O, W
- Query: P(W)
- So, O&M are nuisance—marginalize away
- Marginal =

Elimination order

$$\sum_M \sum_O \phi_1(M)\phi_3(O)\phi_4(O,W)\phi_5(M,W)/Z$$

- Sum out the nuisance variables in turn
- Can do it in any order, but some orders may be easier than others
- Let's do O, then M

$\phi_3(O) =$

O	
T	0.2
F	0.8

$\phi_4(O,W) =$

O W	
T T	0.1
T F	0.9
F T	0.1
F F	0.9

One last elimination

$\phi_1(M) =$

T	0.9
F	0.1

$\phi_6(W) =$

T	0.1
F	0.9

$\phi_5(M, W) =$

T	T	0.8
T	F	0.1
F	T	0
F	F	0

Checking our work

- http://www.aispace.org/bayes/version5.1.6/bayes.jnlp

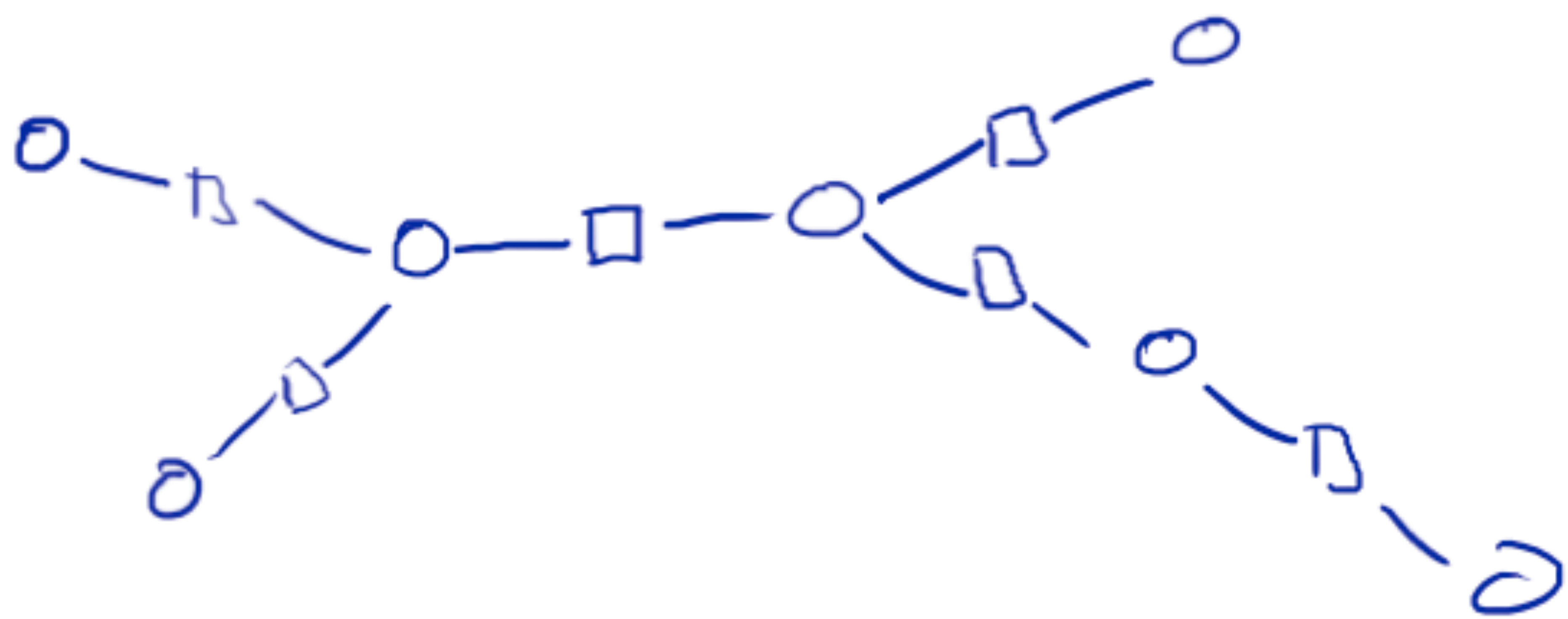
Discussion

- Steps: instantiate evidence, eliminate nuisance nodes, normalize, answer query
  - each elimination introduces a new table, makes some old tables irrelevant
- Normalization
- Each elim. order introduces different tables
  - some tables bigger than others
- FLOP count; treewidth

Treewidth examples

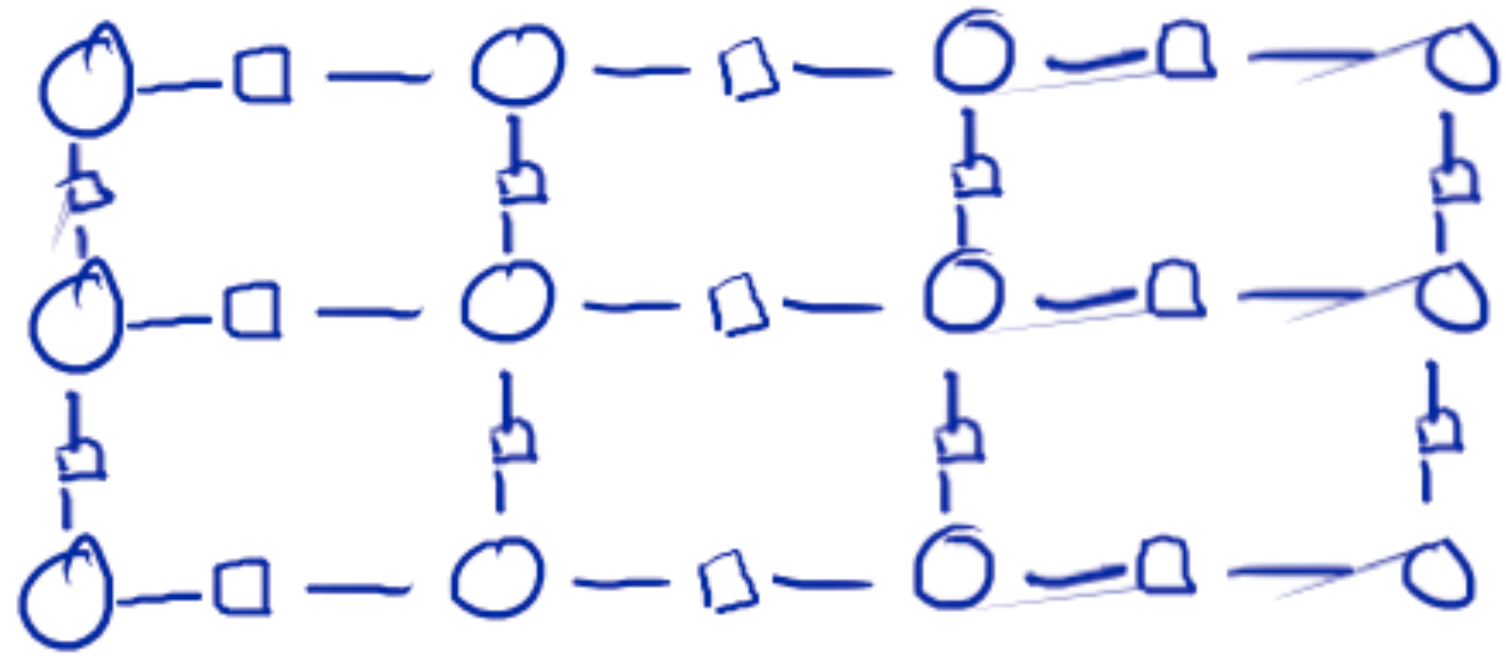
Chain

Tree

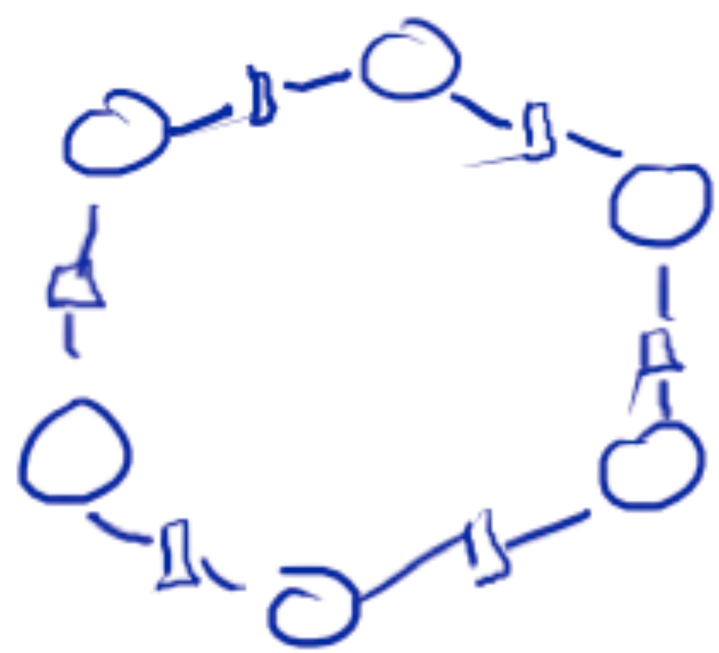


Treewidth examples

Parallel chains



Cycle



Discussion

- Several relationships between GMs and logic (similar DP algorithm, use of independent choices + logical consequences to represent a GM, factor graph with 0-1 potentials = CSP, MAP assignment = ILP)
- Directed v. undirected: advantages to both
- Lifted reasoning
  - Propositional logic + objects = FOL
  - FO GMs are a current hot topic of research (plate models, MLNs, ICL)—not solved yet!

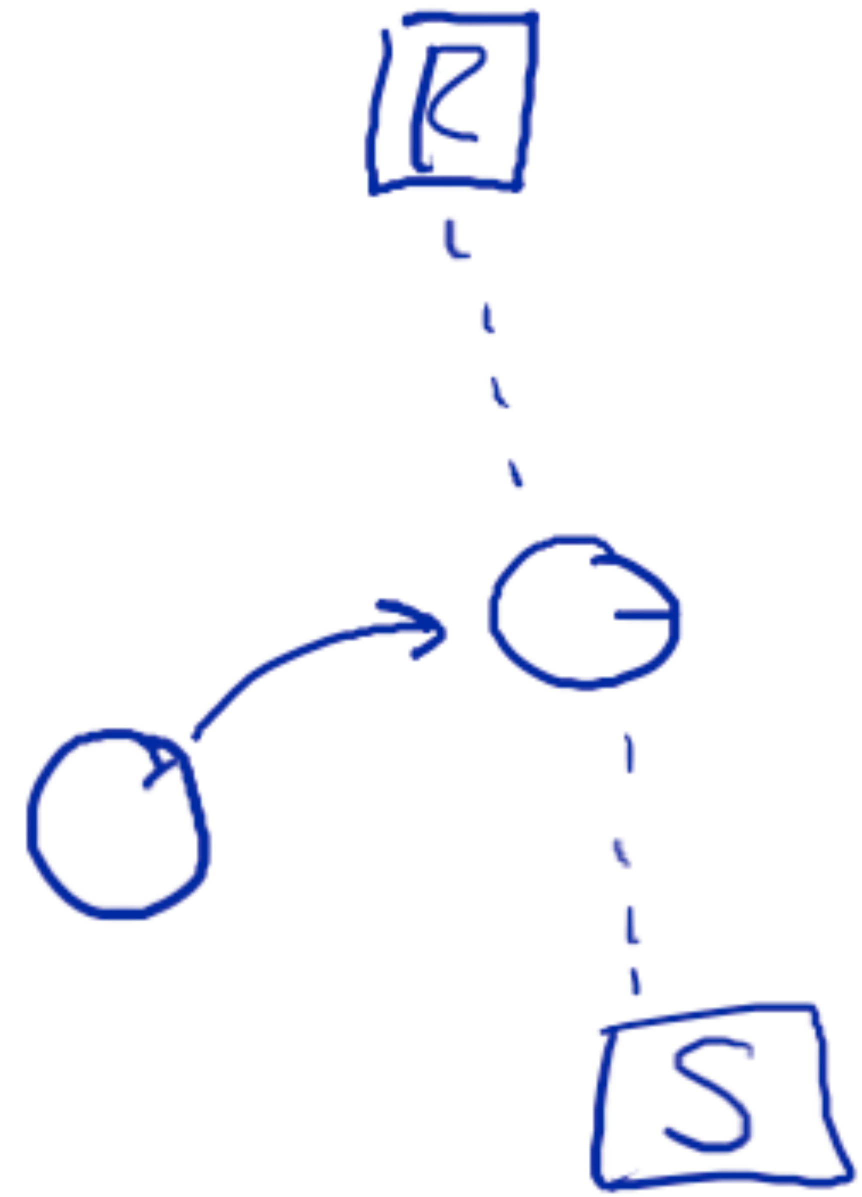
Discussion: belief propagation

- Suppose we want all 1-variable marginals
- Could do N runs of variable elimination
- Or: the BP algorithm simulates N runs for the price of 2
- For details: Kschischang et al. reading

HMMs and DBNs

Inference over time

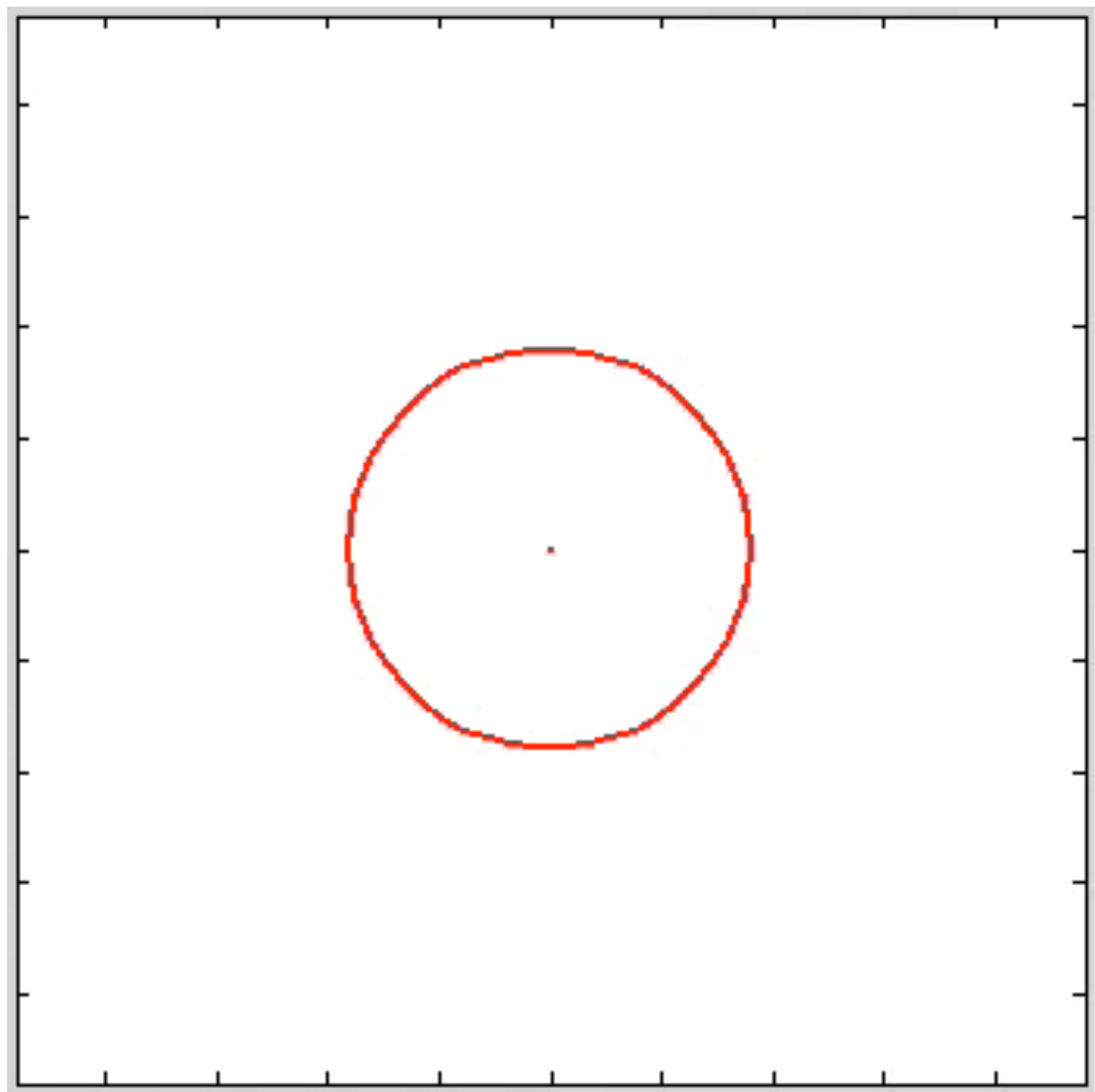
- Consider a robot:
  - true state (x, y, θ)
  - controls (v, w)
  - N range sensors (here N=2: r, s)



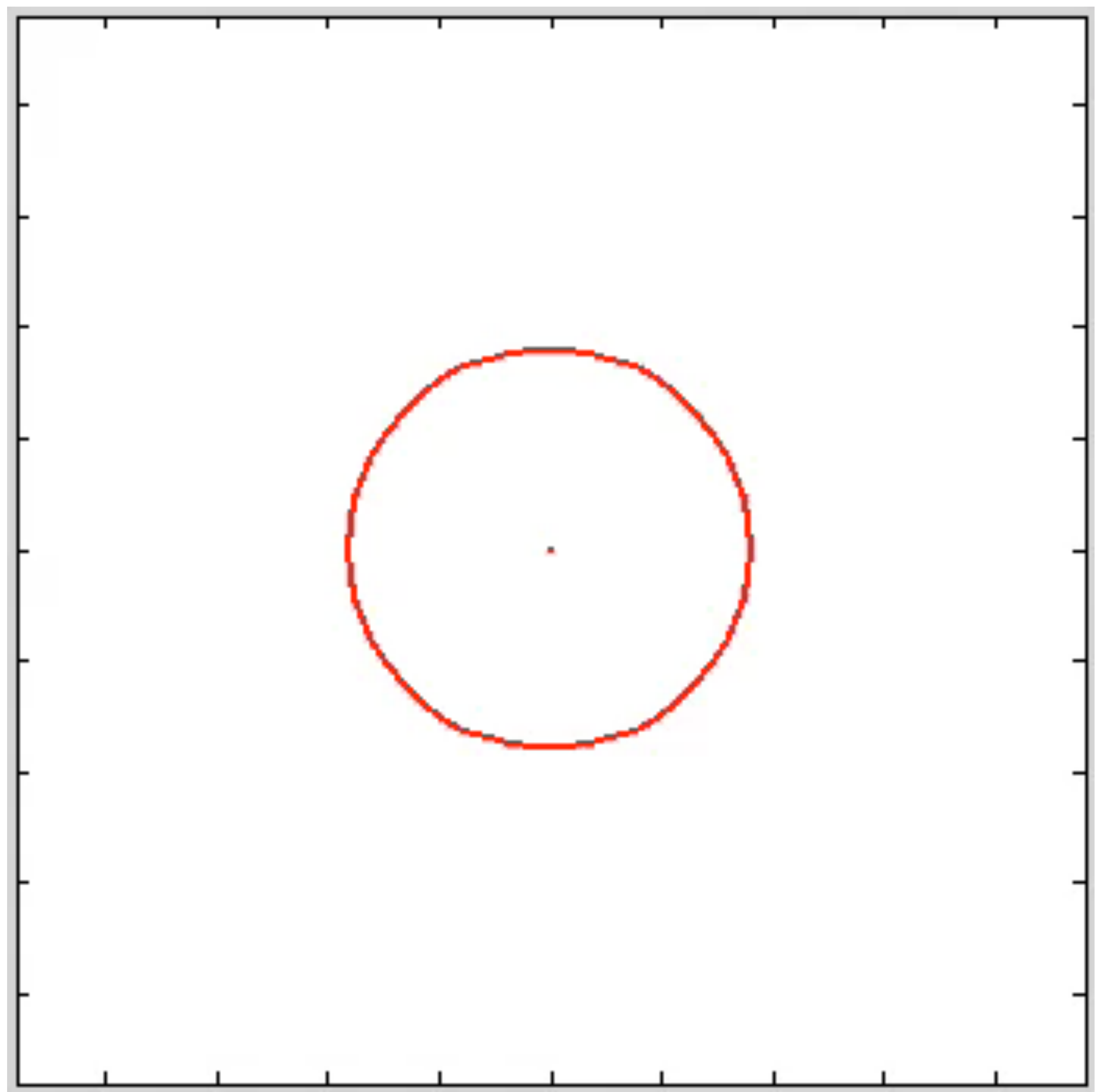
Model

$$
\begin{aligned}
x_{t+1} &= x_t + v_t \cos\theta_t + \text{noise} \\
y_{t+1} &= y_t + v_t \sin\theta_t + \text{noise} \\
\theta_{t+1} &= \theta_t + w_t + \text{noise} \\
r_t &= \sqrt{(x_t - x^R)^2 + (y_t - y^R)^2} + \text{noise} \\
s_t &= \sqrt{(x_t - x^S)^2 + (y_t - y^S)^2} + \text{noise}
\end{aligned}
$$

Model of x, y, θ (r, s unobserved)

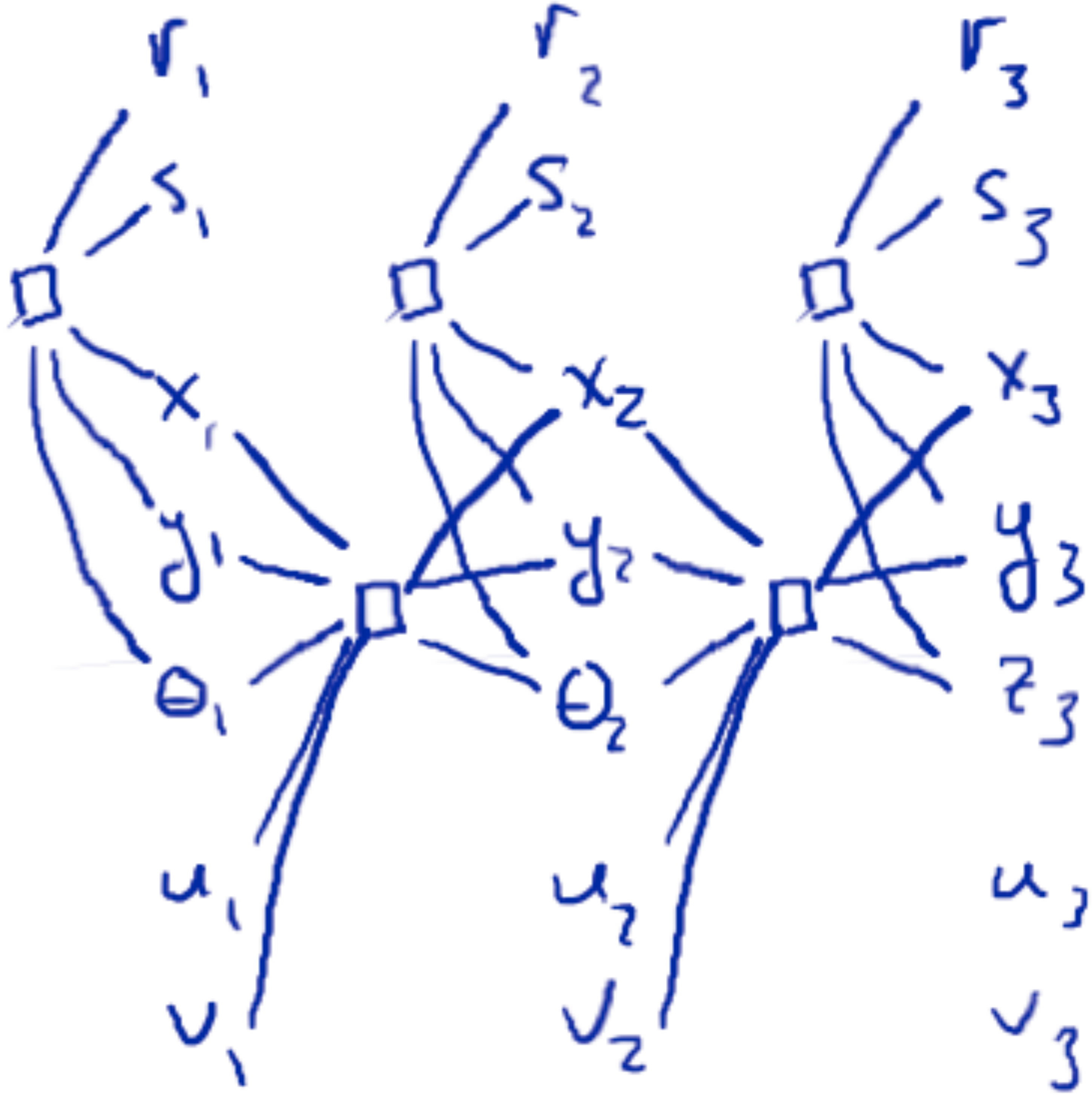


Goal: inference over time



- N=1 sensor, repeatedly observe range = 1m + noise

Factor graph

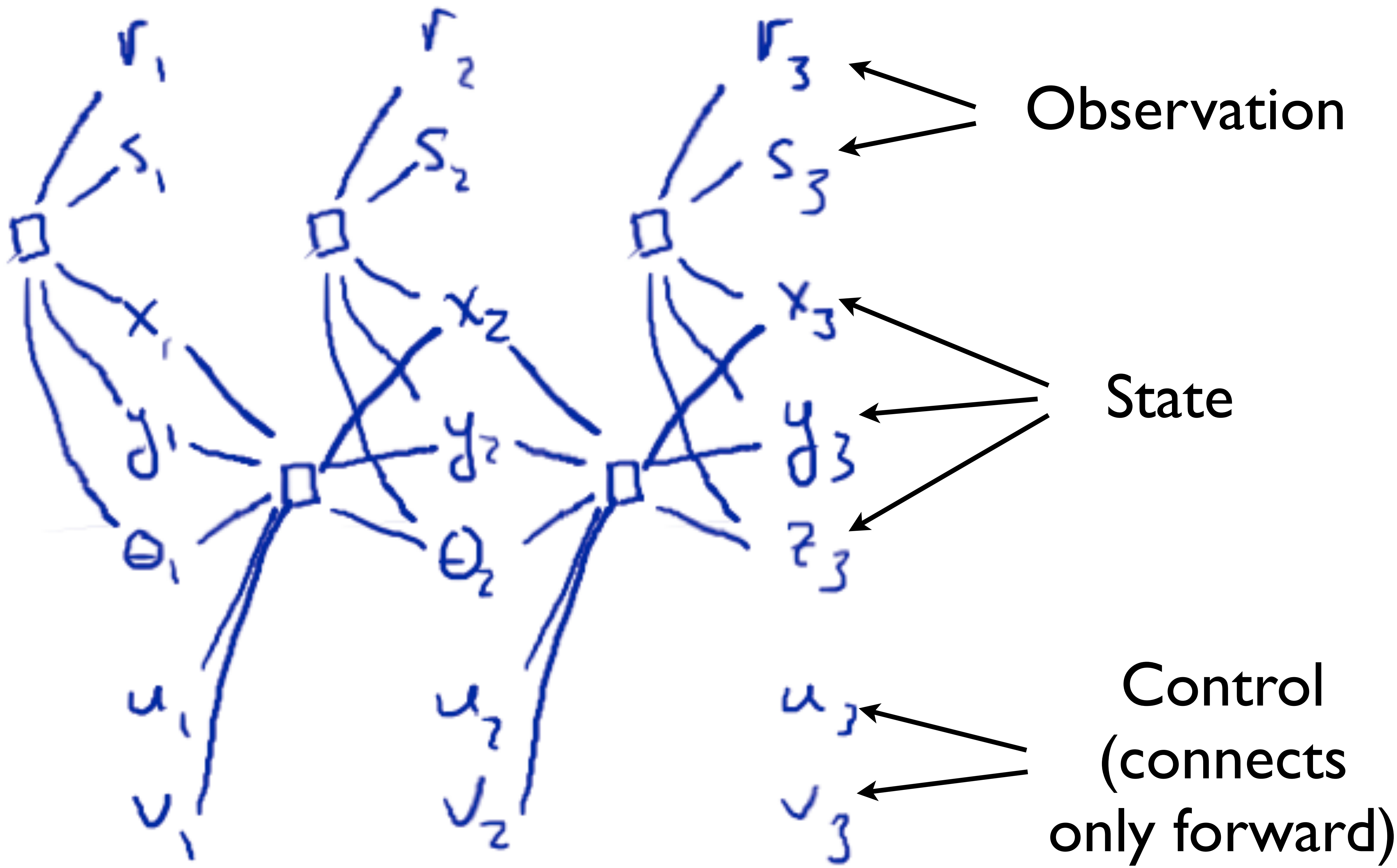


Dynamic Bayes Network

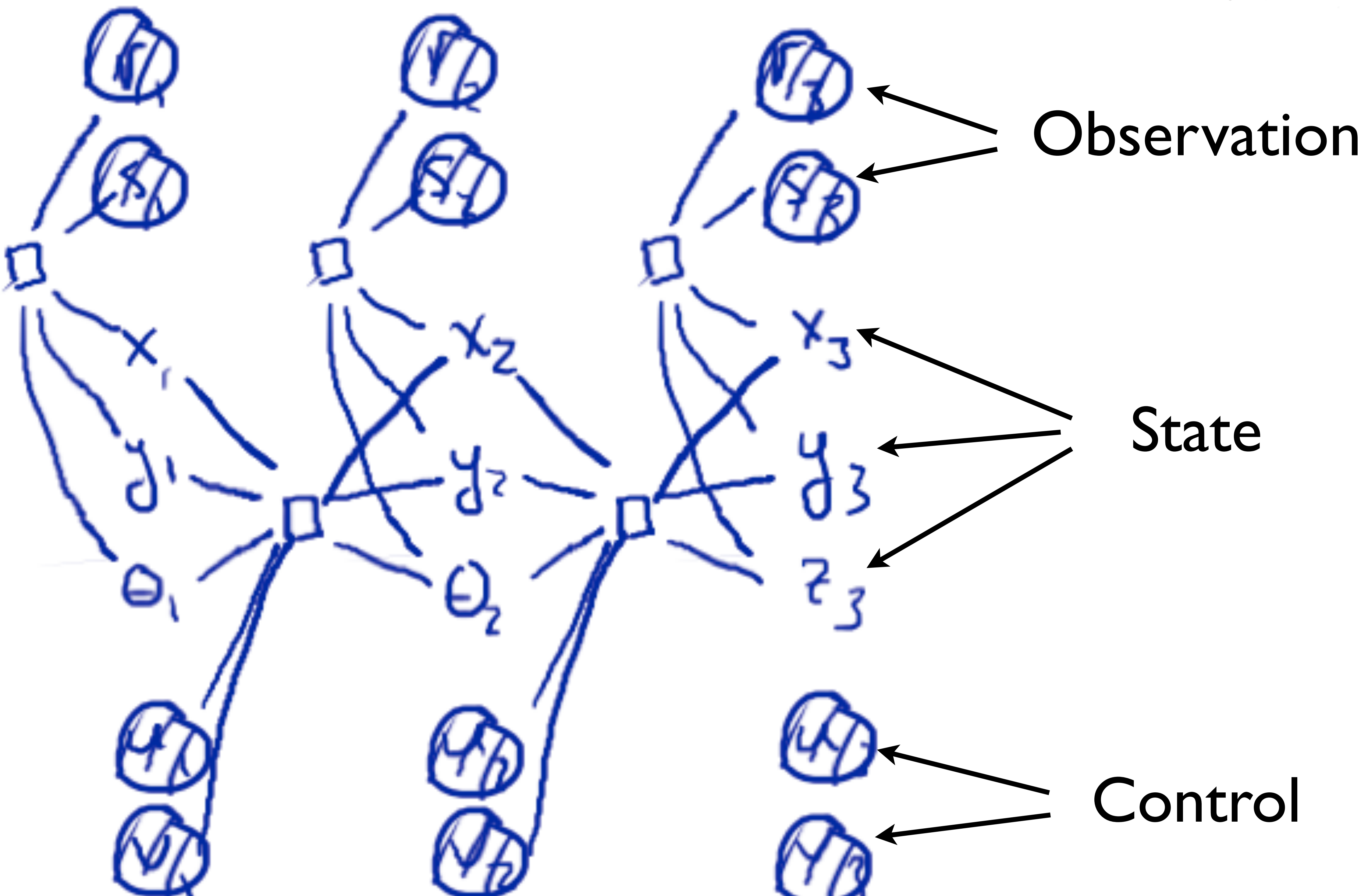
- DBN: factor graph composed of a single structural unit repeated over time
  - conceptually infinite to right, but in practice cut off at some maximum T
- Factors **must** be conditional distributions

$$\begin{aligned} \forall x_t, y_t, \theta_t, u_t, v_t \quad \textstyle\sum_{x_{t+1}, y_{t+1}, \theta_{t+1}} \phi(x_t, y_t, \theta_t, u_t, v_t, x_{t+1}, y_{t+1}, \theta_{t+1}) &= 1 \\ \forall x_t, y_t, \theta_t \quad \textstyle\sum_{r_t, s_t} \phi(x_t, y_t, \theta_t, r_t, s_t) &= 1 \end{aligned}$$

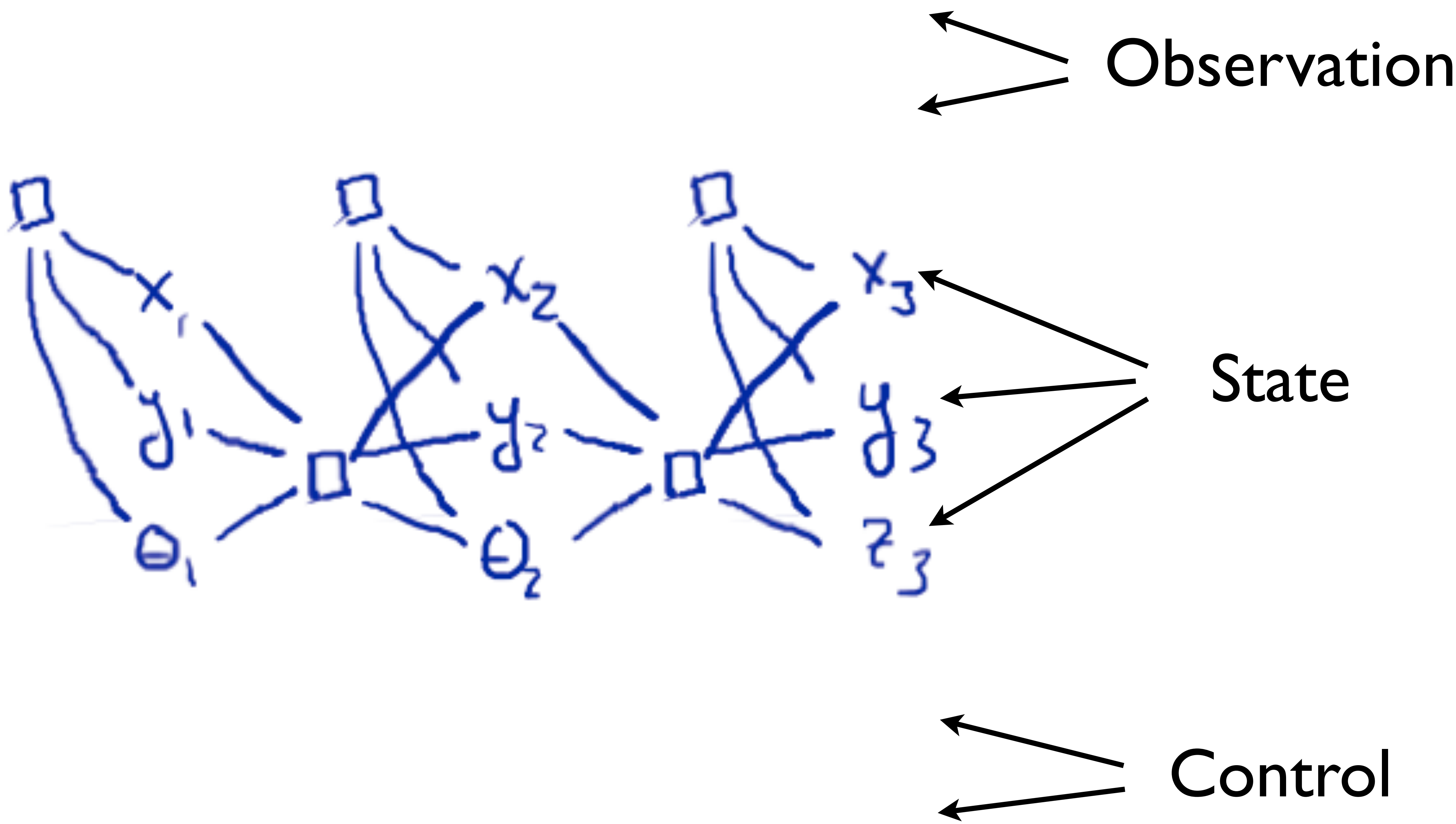
Three kinds of variable



Condition on obs, do(control)



Condition on obs, do(control)



Simplified version

- State: $x_t \in \{1, 2, 3\}$
- Observation: $y_t \in \{L, H\}$
- Control: just one (i.e., no choice)—"keep going"

Hidden Markov Models

- This is an HMM—a DBN with:
  - one state variable
  - one observation variable

Potentials

X_{t+1}

X_t	1	2	3
1	0.7	0.3	0
2	0.3	0.3	0.3
3	0	0.3	0.7

Y_t

X_t	L	H
1	0.67	0.33
2	0.5	0.5
3	0.33	0.67

HMM inference

- Condition on $y_1 = H, y_2 = H, y_3 = L$
- What is $P(X_2 \mid HHL)$?

HMM factors after conditioning

x_1	ϕ_1
1	.33
2	.5
3	.67

x_2	ϕ_2
1	.33
2	.5
3	.67

x_3	ϕ_3
1	.67
2	.5
3	.33

ϕ_4

x_1 \ x_2			
	.67	.33	0
	.33	.33	.33
	0	.33	.67

ϕ_5

x_2 \ x_3			
	.67	.33	0
	.33	.33	.33
	0	.33	.67

Eliminate x_1 and x_3

$\phi_1 \phi_4$ α_{12}

Rows x_1, columns x_2:

2/9	1/9	0
1/6	1/6	1/6
0	2/9	4/9

$\rightarrow$

x_2
7/18
1/2
11/18

$\phi_3 \phi_5$ β_{23}

Rows x_2, columns x_3:

4/9	1/6	0
2/9	1/6	1/9
0	1/6	2/9

$\rightarrow$

x_2
11/18
1/2
7/18

Multiply remaining potentials and renormalize

α_{12}

X_2
7/18
1/2
11/18

β_{23}

X_2
11/18
1/2
7/18

$\alpha_{12}\beta_{23}\phi_2$
.079
.125
.158

$/Z$

.22

.34

.44

Forward-backward

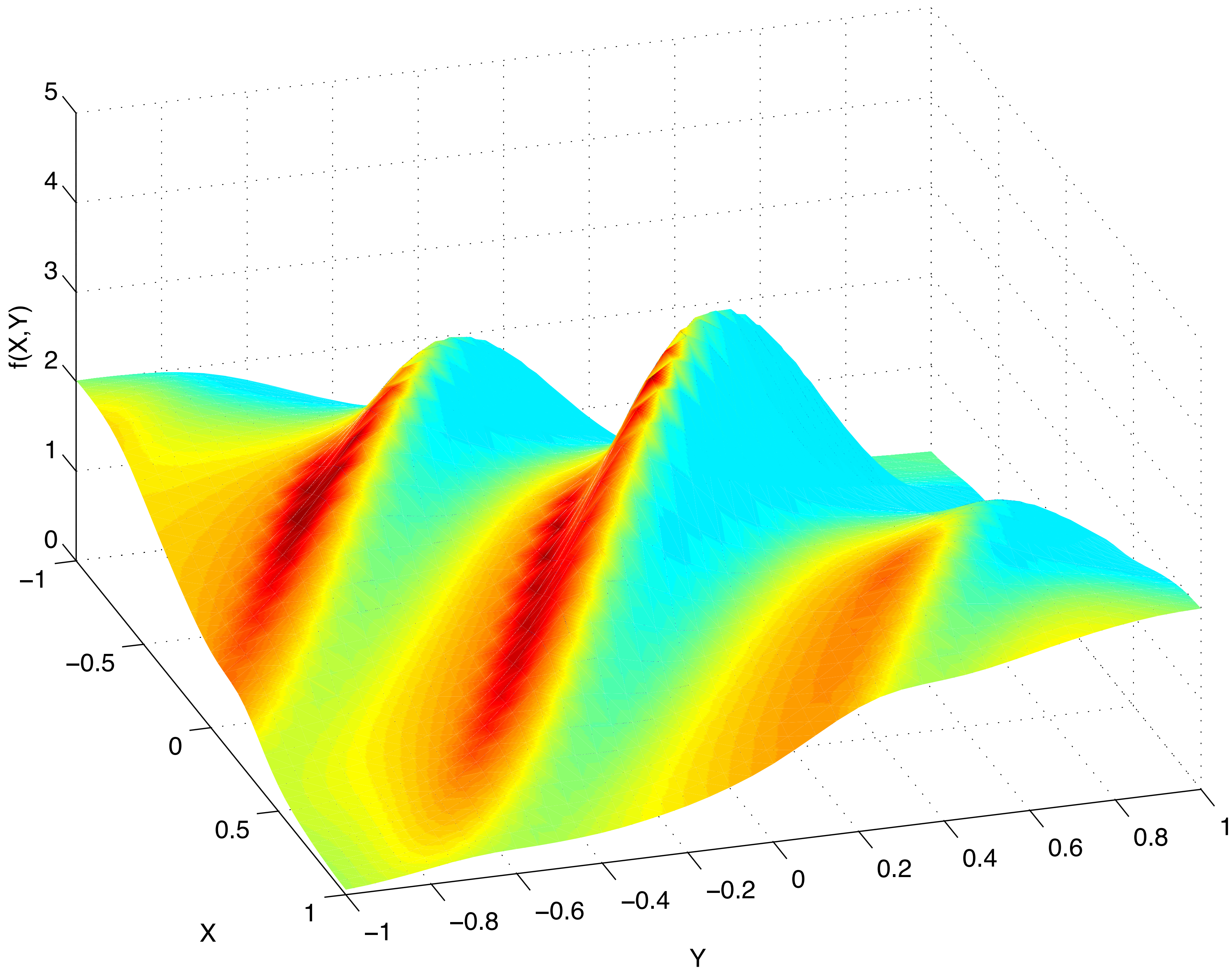
- You may recognize the above as the forward-backward algorithm
- Special case of dynamic programming / variable elimination / belief propagation

Approximate Inference

Most of the time…

- Treewidth is big
- Variables are high-arity or continuous
- Can’t afford exact inference
- Need numerical integration (and/or summation)
- We’ll look at randomized algorithms

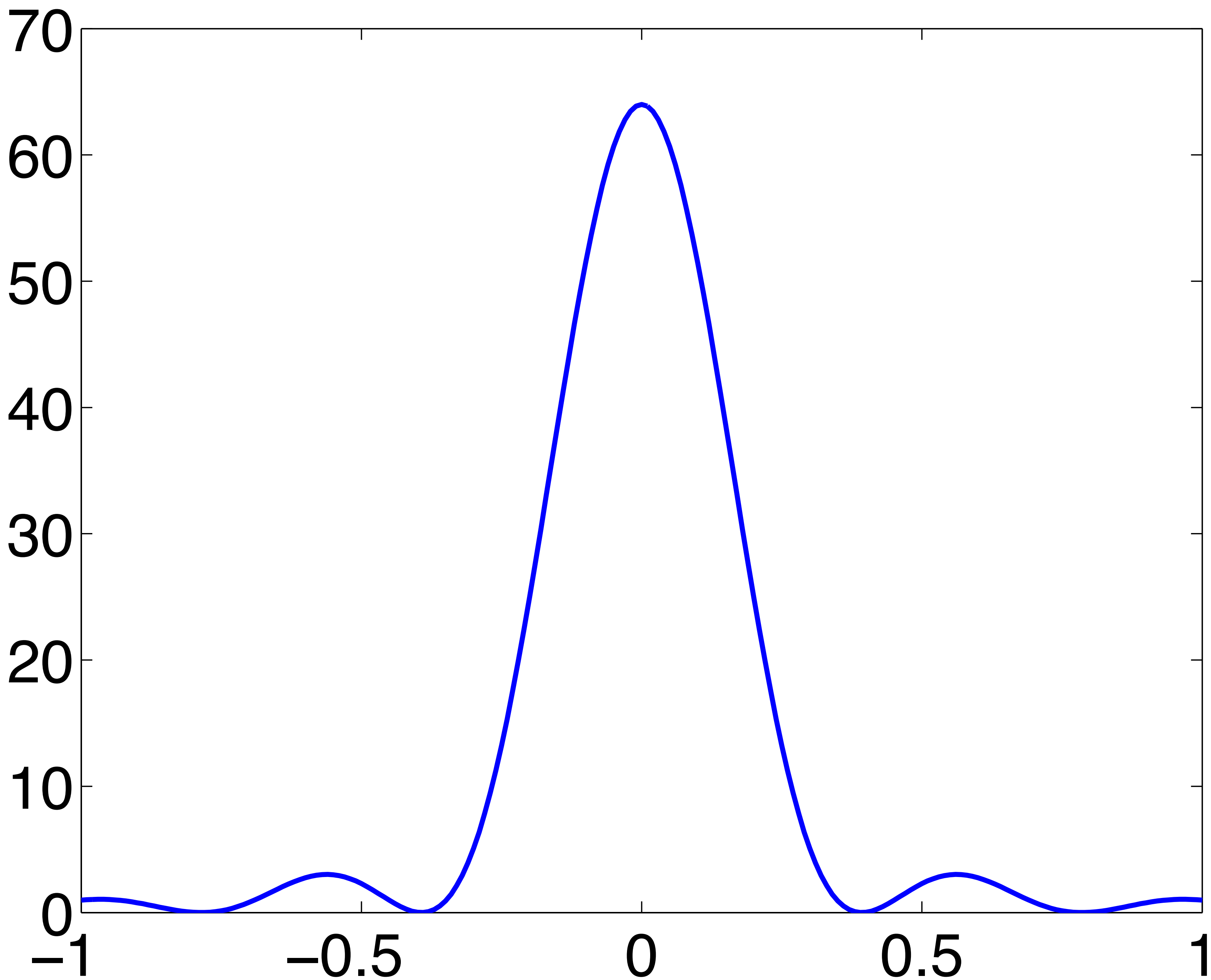
Numerical integration



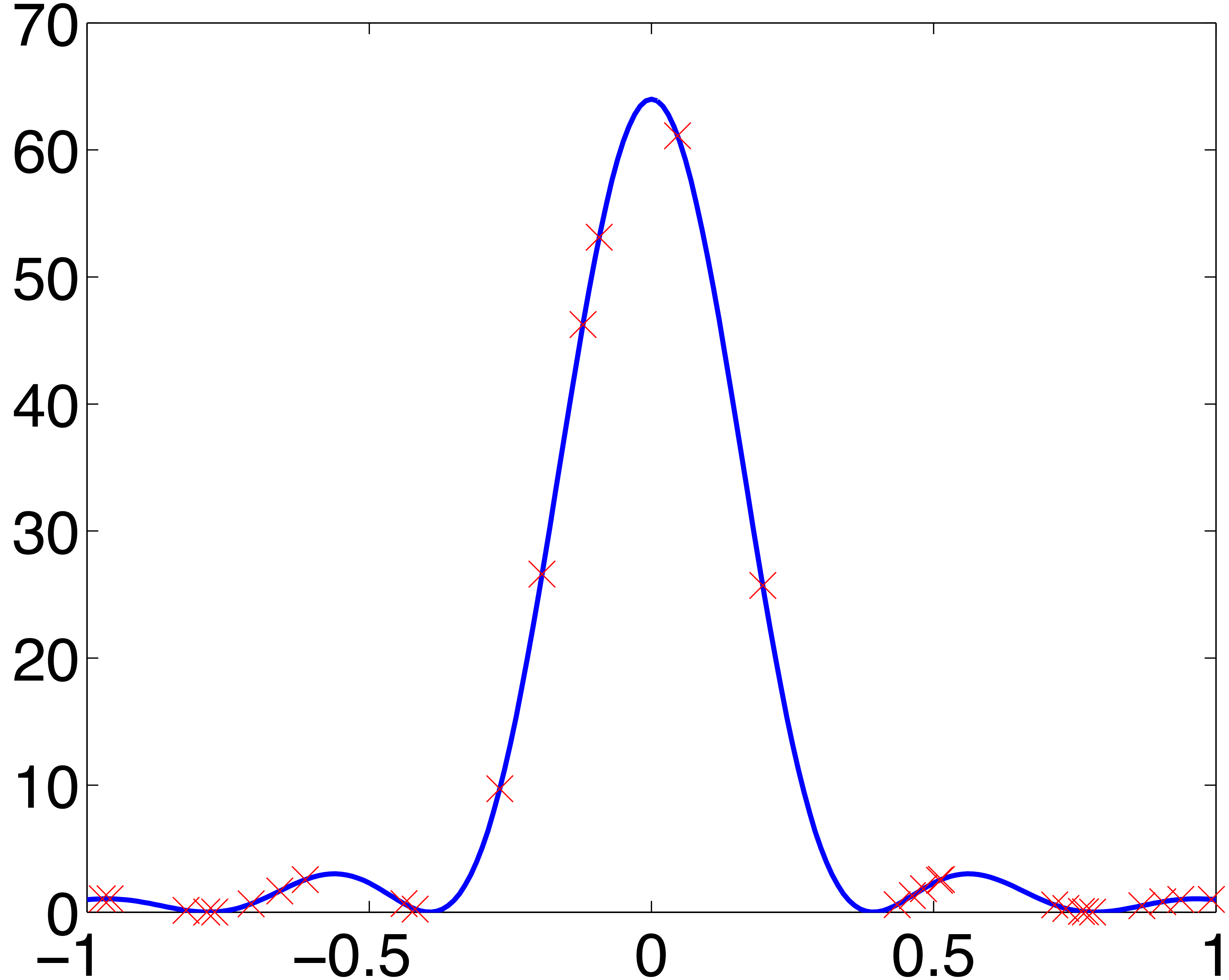
Integration in 1000s of dims

Eliazar and Parr, IJCAI-03

Simple 1D problem



Uniform sampling



$$\frac{2}{N}\sum_i f(x_i)$$

Uniform sampling

$$\begin{aligned} E(f(X)) &= \int P(x)f(x)dx \\ &= \frac{1}{V}\int f(x)dx \end{aligned}$$

- So, V E(f(X)) is desired integral
- But standard deviation can be big
- Can reduce it by averaging many samples
- But only at rate 1/sqrt(N)

Importance sampling

- Instead of x ~ uniform, use x ~ Q(x)
- Q = importance distribution
- Should have Q(x) large where f(x) is large
- Problem:

$$E_Q(f(X)) = \int Q(x) f(x) dx$$

Importance sampling

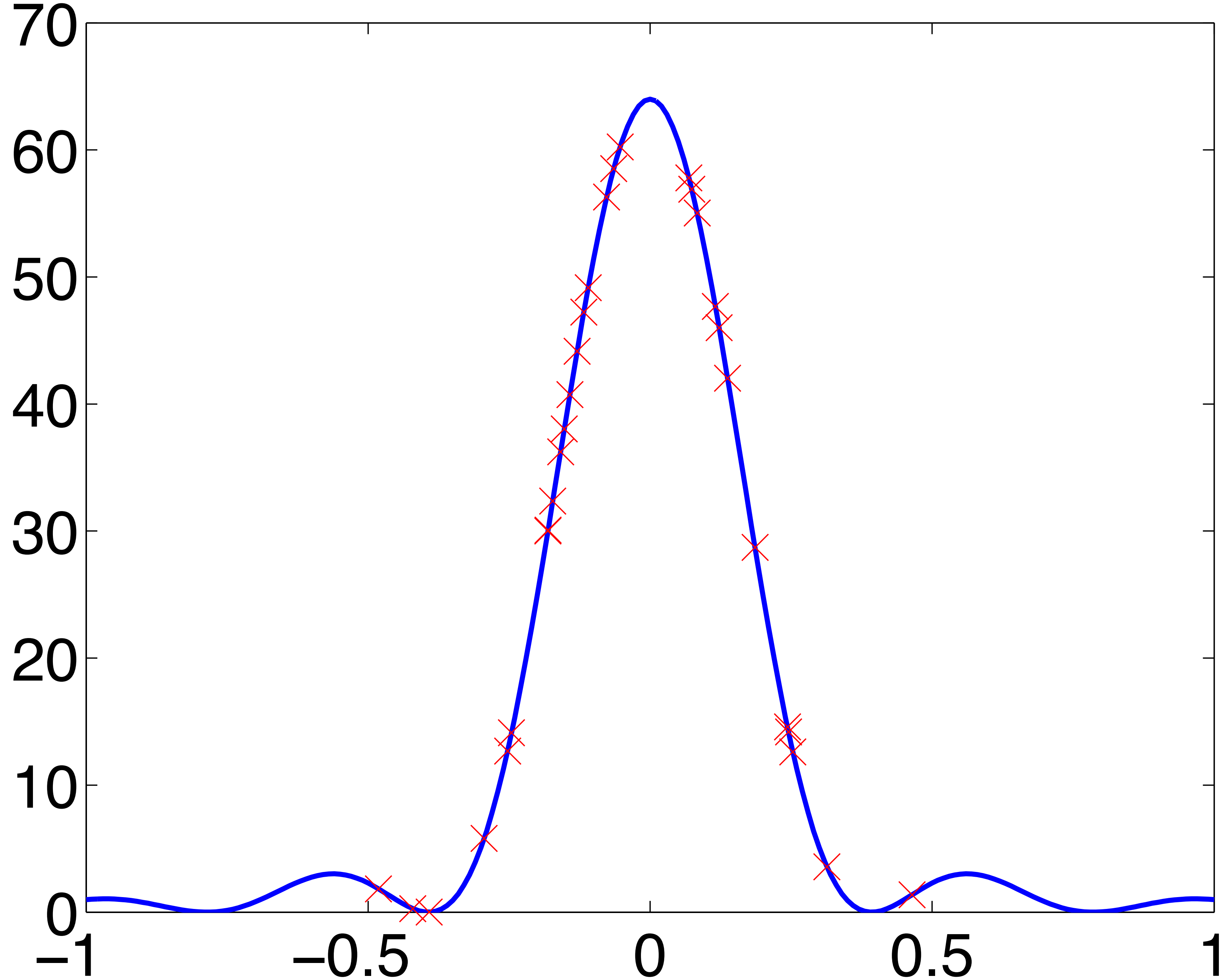
$$h(x) \equiv f(x)/Q(x)$$

$$\begin{aligned} E_Q(h(X)) &= \int Q(x)h(x)dx \\ &= \int Q(x)f(x)/Q(x)dx \\ &= \int f(x)dx \end{aligned}$$

Importance sampling

- So, take samples of h(X) instead of f(X)
- $w_i = 1/Q(x_i)$ is **importance weight**
- $Q = 1/V$ yields uniform sampling

Importance sampling



Variance

- How does this help us control variance?
- Suppose f big ==> Q big
- And Q small ==> f small
- Then h = f/Q never gets too big
- Variance of each sample is lower ==> need fewer samples
- A good Q makes a good IS