15-780: Grad AI
Lecture 18: Probability, planning, graphical models

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

Admin

- Reminder: project milestone reports due 2 weeks from today

Review: probability

- Independence, correlation
- Expectation, conditional e., linearity of e., iterated e., independence & e.
- Experiment, prior, posterior
- Estimators (bias, variance, asymptotic behavior)
- Bayes Rule
- Model selection

Review: probability & AI

$$\mathbb{Q}_1 X_1 \ \mathbb{Q}_2 X_2 \ \mathbb{Q}_3 X_3 \ \ \ldots \ \ F(X_1, X_2, X_3, \ldots)$$

each quantifier is max, min, or mean

- PSTRIPS
- QBF and "QBF+"
- PSTRIPS to QBF+ translation

Example: got cake?

- $\neg have_1 \wedge gatebake_1 \wedge bake_2 \Leftrightarrow Cbake_2$
- $have_1 \wedge gateeat_1 \wedge eat_2 \Leftrightarrow Ceat_2$
- $have_1 \wedge eat_2 \Leftrightarrow Ceat'_2$
- $[Cbake_2 \Rightarrow have_3] \wedge [Ceat_2 \Rightarrow eaten_3] \wedge [Ceat'_2 \Rightarrow \neg have_3]$
- $0.8{:}gatebake_1 \wedge 0.9{:}gateeat_1$

Example: got cake?

- $have_3 \Rightarrow [Cbake_2 \vee (\neg Ceat'_2 \wedge have_1)]$
- $\neg have_3 \Rightarrow [Ceat'_2 \vee (\neg Cbake_2 \wedge \neg have_1)]$
- $eaten_3 \Rightarrow [Ceat_2 \vee eaten_1]$
- $\neg eaten_3 \Rightarrow [\neg eaten_1]$

Example: got cake?

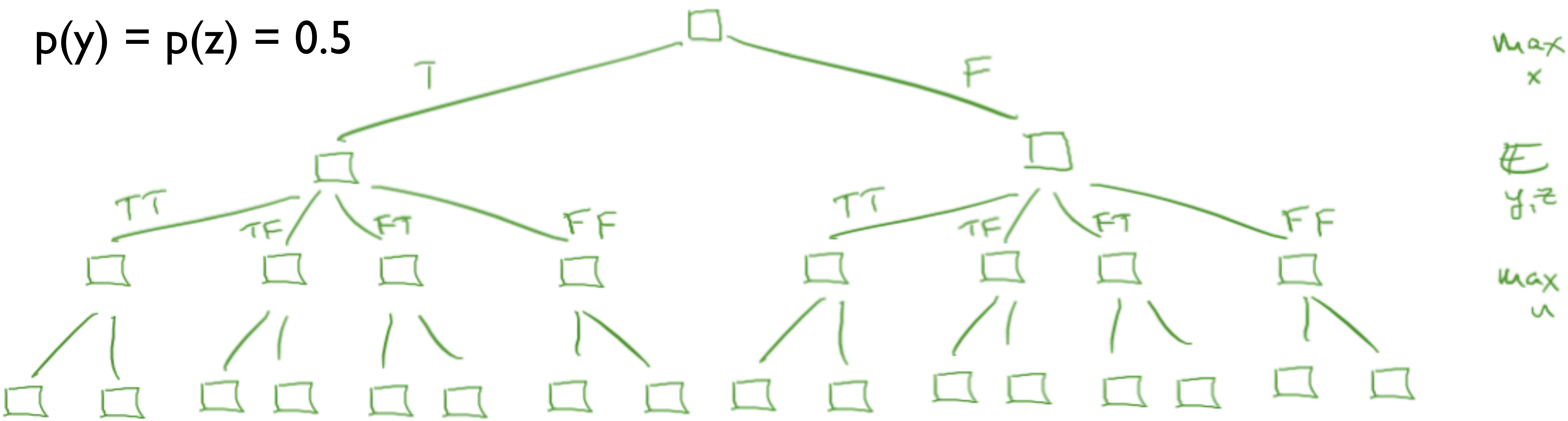
- $\neg bake_2 \vee \neg eat_2$
- (pattern from past few slides is repeated for each action level w/ adjacent state levels)

Example: got cake?

- $\neg have_I \wedge \neg\ eaten_I$
- $have_T \wedge eaten_T$

Simple QBF+ example

p(y) = p(z) = 0.5



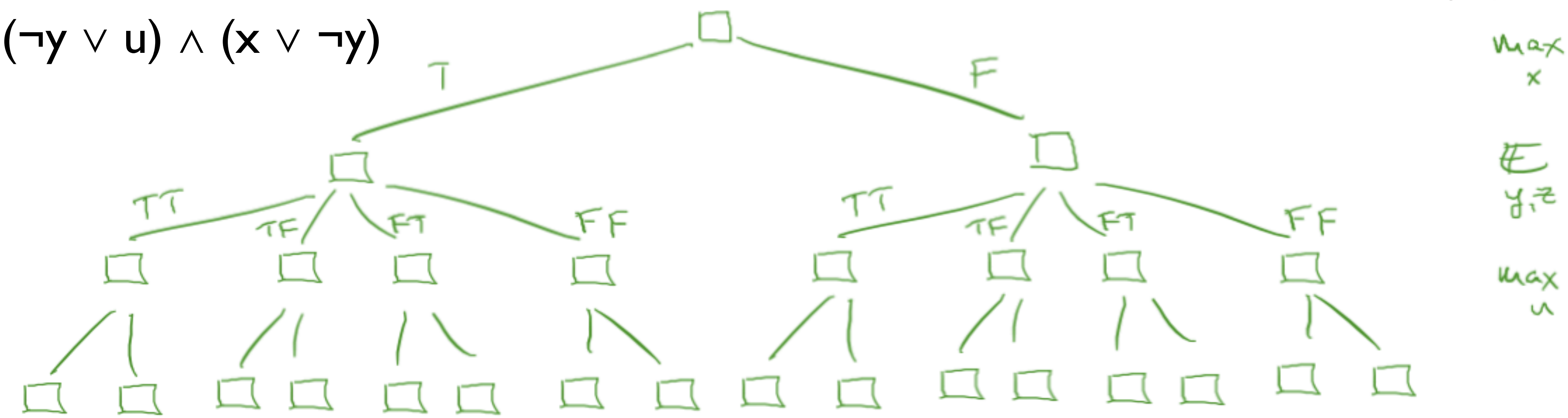
$$\max_x \; \mathbb{E}_{y,z} \; \max_u \; (\bar{x} \vee z) \wedge (\bar{y} \vee u) \wedge (x \vee \bar{y})$$

How can we solve?

- Scenario trick
  - transform to PBI or 0-1 ILP
- Dynamic programming
  - related to algorithms for SAT, #SAT
  - also to belief propagation in graphical models (next)

Solving exactly by scenarios

$(\neg x \vee z) \wedge (\neg y \vee u) \wedge (x \vee \neg y)$

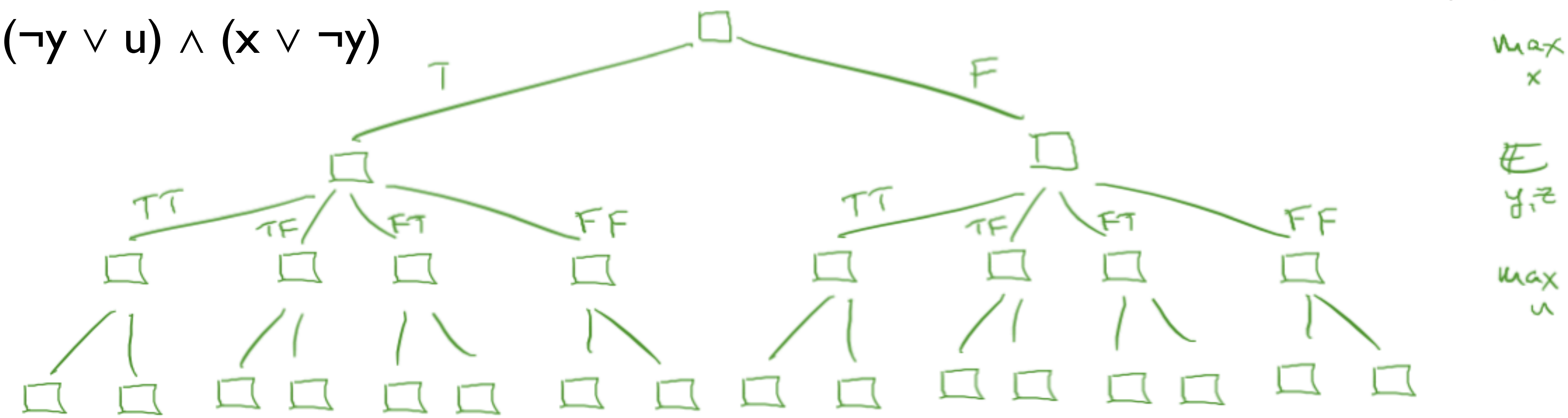


- Replicate u to uYZ: u00, u01, u10, u11
- Replicate clauses: share x; set y, z by index; replace u by uYZ; write aYZ for truth value
- $a00 \Leftrightarrow [(\neg x \vee 0) \wedge (\neg 0 \vee u00) \wedge (x \vee \neg 0)] \wedge$

  $a01 \Leftrightarrow [(\neg x \vee 1) \wedge (\neg 0 \vee u01) \wedge (x \vee \neg 0)] \wedge \ldots$
- add a PBI: $a00 + a01 + a10 + a11 \geq 4 * \text{threshold}$

Solving by sampling scenarios

$(\neg x \vee z) \wedge (\neg y \vee u) \wedge (x \vee \neg y)$



- Sample a subset of the values of y, z (e.g., {11, 01}):
  - $a11 \Leftrightarrow [(\neg x \vee 1) \wedge (\neg 1 \vee u11) \wedge (x \vee \neg 1)] \wedge$

    $a01 \Leftrightarrow [(\neg x \vee 1) \wedge (\neg 0 \vee u01) \wedge (x \vee \neg 0)]$
- Adjust PBI: $a11 + a10 \geq 2 * \text{threshold}$

Combining PSTRIPS w/ scenarios

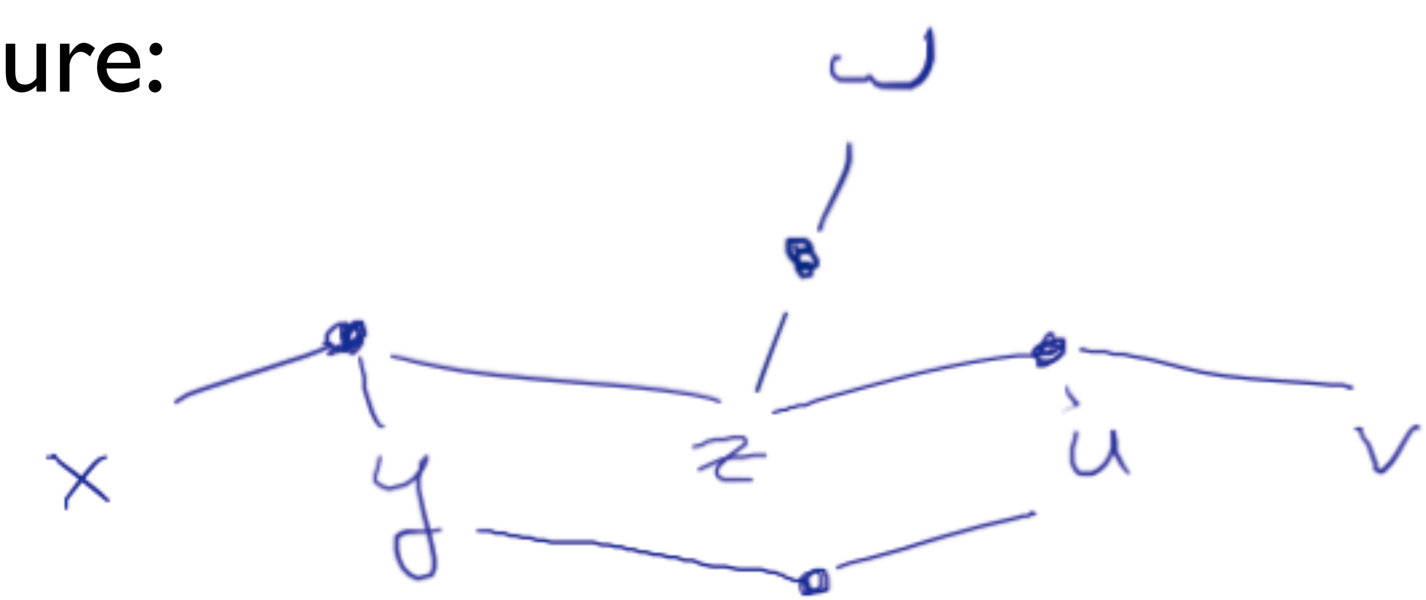
- Generate M samples of Nature ($gatebake_1$, $gateeat_1$, $gatebake_3$, $gateeat_3$, $gatebake_5$, …)
- Replicate state-level vars M times
- One copy of action vars $bake_2$, eat_2, $bake_4$, …
- Replicate clauses M times (share actions)
- Replace goal constraints w/ constraint that all goals must be satisfied in at least y% of scenarios (a PBI)
- Give to MiniSAT+ (fixed y) or CPLEX (max y)

Dynamic programming

- Consider the simpler problem (all p=0.5):

$$\mathbb{E}_{xyzuvw} \, (x \vee y \vee \bar{z}) \wedge (\bar{y} \vee \bar{u}) \wedge (z \vee w) \wedge (z \vee u \vee v)$$

- This is essentially an instance of #SAT
- Structure:



Dynamic programming for variable elimination

$$\sum_x \sum_y \sum_z \sum_u \sum_v \sum_w A(xyz)\, B(yu)\, C(zw)\, D(zuv)$$

	A
TTT	1
TTF	1
TFT	1
TFF	1
FTT	1
FTF	1
FFT	0
FFF	1

	B
TT	0
TF	1
FT	1
FF	1

	C
TT	1
TF	1
FT	1
FF	0

	D
TTT	1
TTF	1
TFT	1
TFF	1
FTT	1
FTF	1
FFT	1
FFF	0

Variable elimination

In general

- Pick a variable ordering
- Repeat: say next variable is z
  - move sum over z inward as far as it goes
  - make a new table by multiplying all old tables containing z, then summing out z
  - arguments of new table are “neighbors” of z
- Cost: O(size of biggest table * # of sums)
  - sadly: biggest table can be exponentially large
  - but often not: low-treewidth formulas

Connections

- Scenarios are related to your current HW
- DP is related to belief propagation in graphical models (next)
- Can generalize DP for multiple quantifier types (not just sum or expectation)
  - handle PSTRIPS

Graphical models

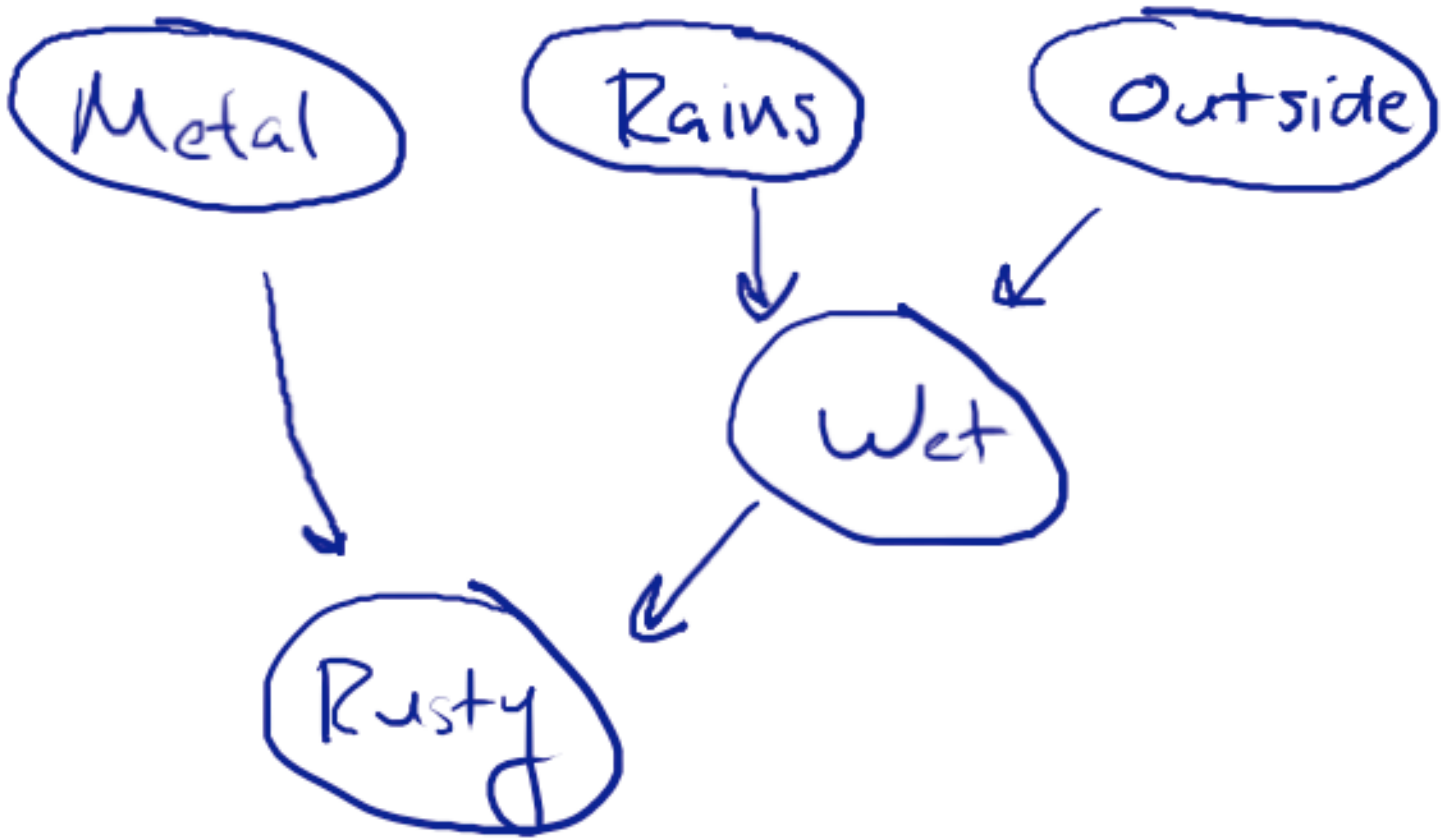
Why do we need graphical models?

- So far, only way we've seen to write down a distribution is as a big table
- Gets unwieldy fast!
  - E.g., 10 RVs, each w/ 10 settings
  - Table size = 10^{10}
- Graphical model: way to write distribution compactly using diagrams & numbers
- Typical GMs are huge (10^{10} is a small one), but we'll use tiny ones for examples

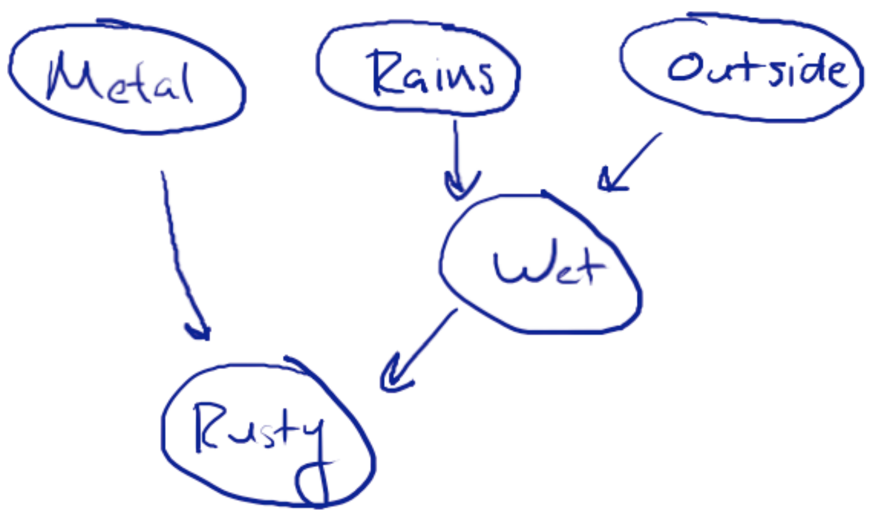
Bayes nets

- Best-known type of graphical model
- Two parts: DAG and CPTs

Rusty robot: the DAG



Rusty robot: the CPTs



- For each RV (say X), there is one CPT specifying P(X | pa(X))

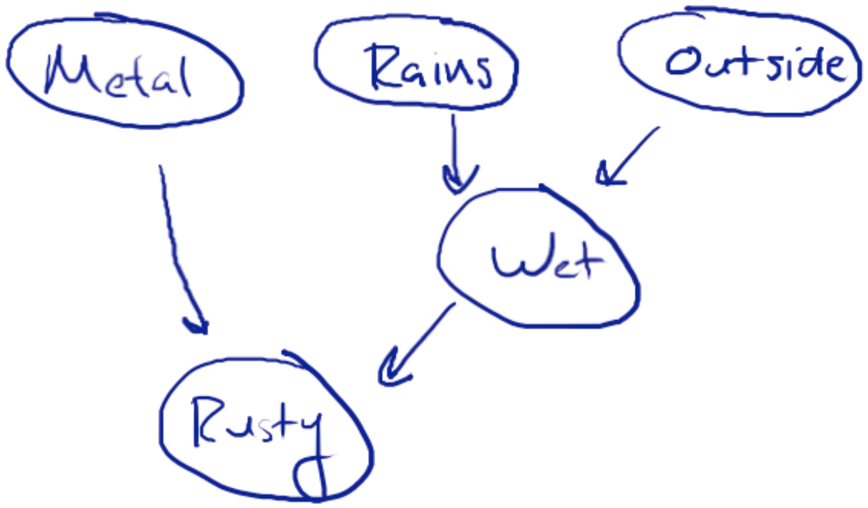
P(Metal) = 0.9
P(Rains) = 0.7
P(Outside) = 0.2
P(Wet | Rains, Outside)

TT: 0.9	TF: 0.1
FT: 0.1	FF: 0.1

P(Rusty | Metal, Wet) =

TT: 0.8	TF: 0.1
FT: 0	FF: 0

Interpreting it



Benefits

- 11 v. 31 numbers
- Fewer parameters to learn
- Efficient ***inference*** = computation of marginals, conditionals ⇒ posteriors

Comparison to prop logic + random causes

- Can simulate any Bayes net w/ propositional logic + random causes—one cause per CPT entry
- E.g.:

Inference Qs

- Is $Z > 0$?
- What is $P(E)$?
- What is $P(E_1 \mid E_2)$?
- Sample a random configuration according to $P(.)$ or $P(. \mid E)$
- Hard part: taking sums over r.v.s (e.g., sum over all values to get normalizer)

Inference example

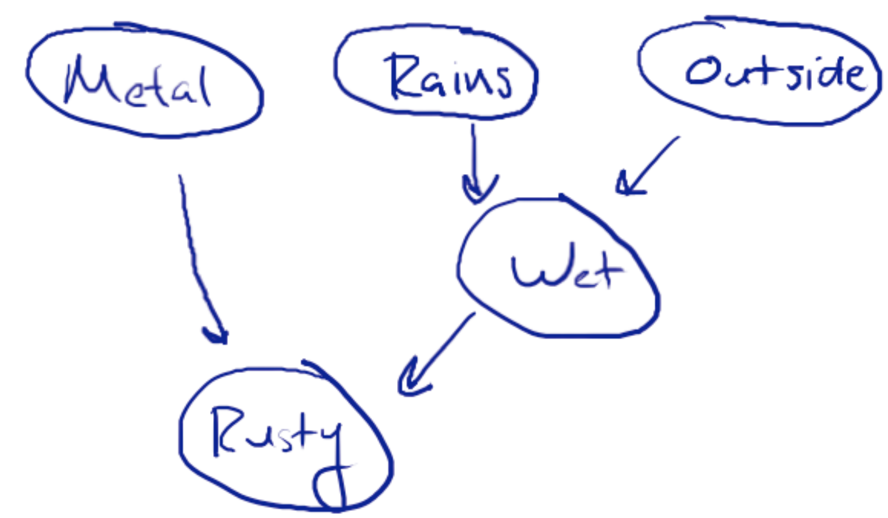
- $P(M, Ra, O, W, Ru) =$

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

- Find marginal of M, O

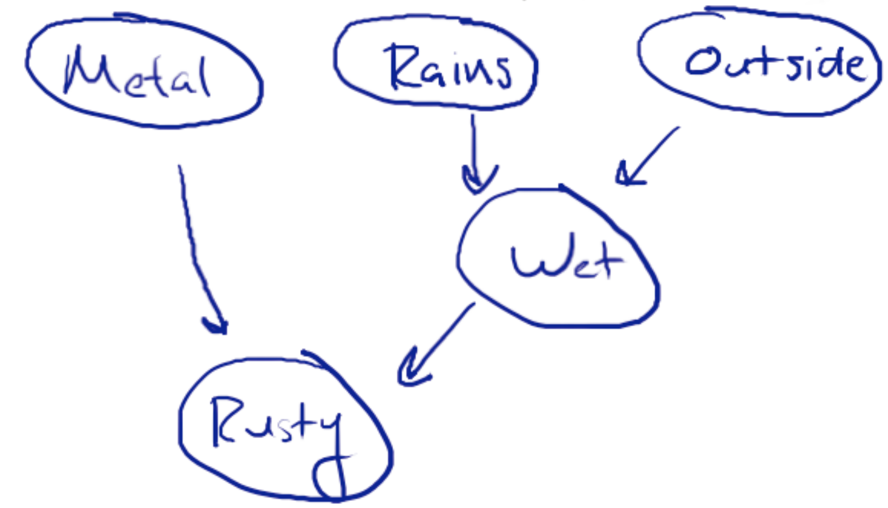
Independence

- Showed M ⊥ O
- Any other independences?



- Didn't use CPTs: some independences depend only on graph structure
- May also be "accidental" independences
  - i.e., depend on values in CPTs

Conditional independence



- How about O, Ru? O Ru
- Suppose we know we're not wet
- P(M, Ra, O, W, Ru) =

  P(M) P(Ra) P(O) P(W|Ra,O) P(Ru|M,W)
- Condition on W=F, find marginal of O, Ru

Conditional independence

- This is generally true
  - conditioning can make or break independences
  - many conditional independences can be derived from graph structure alone
  - accidental ones often considered less interesting
- We derived them by looking for factorizations
  - turns out there is a purely graphical test
  - one of the key contributions of Bayes nets

Blocking

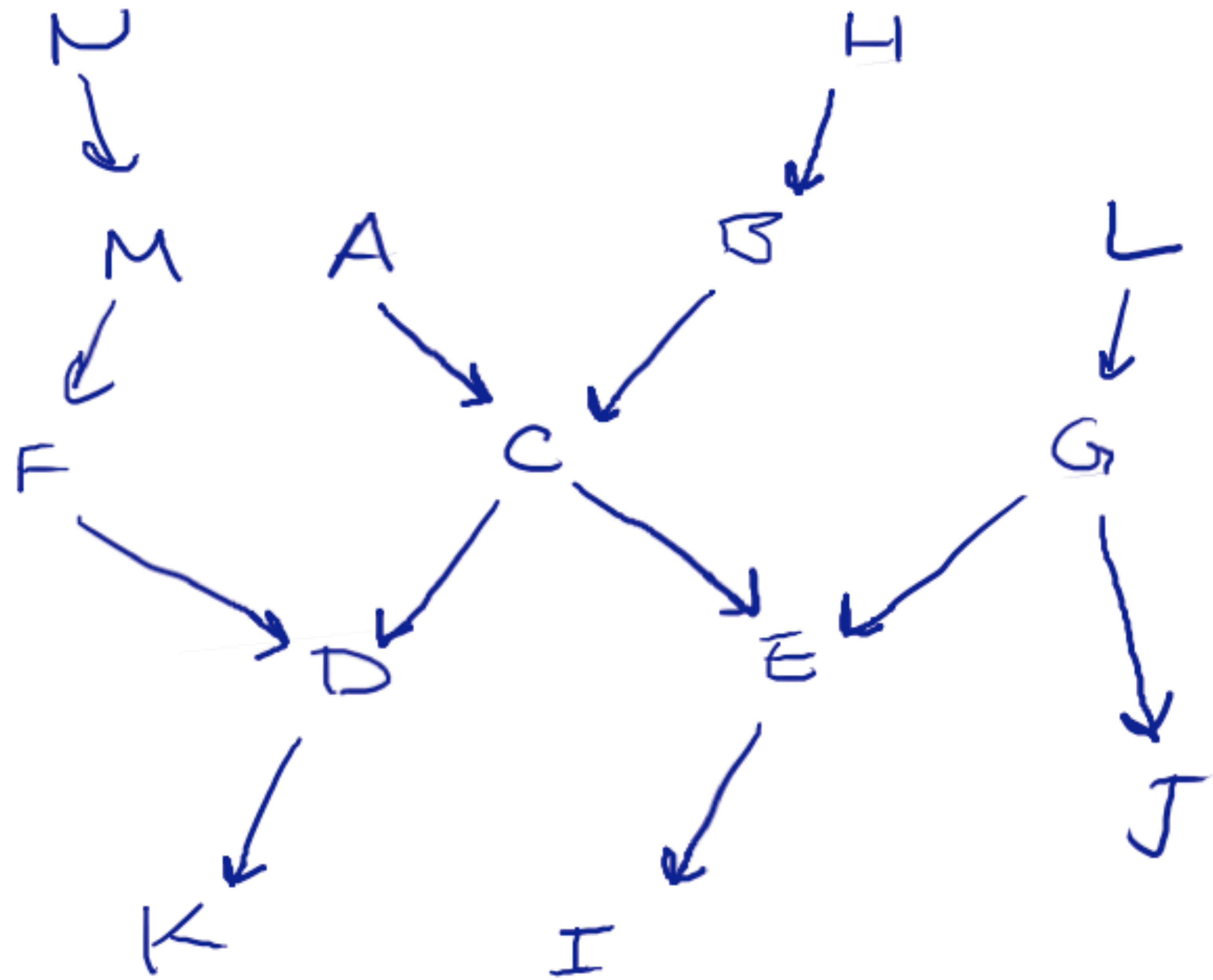
- Shaded = observed (by convention)

Example: explaining away

- Intuitively:

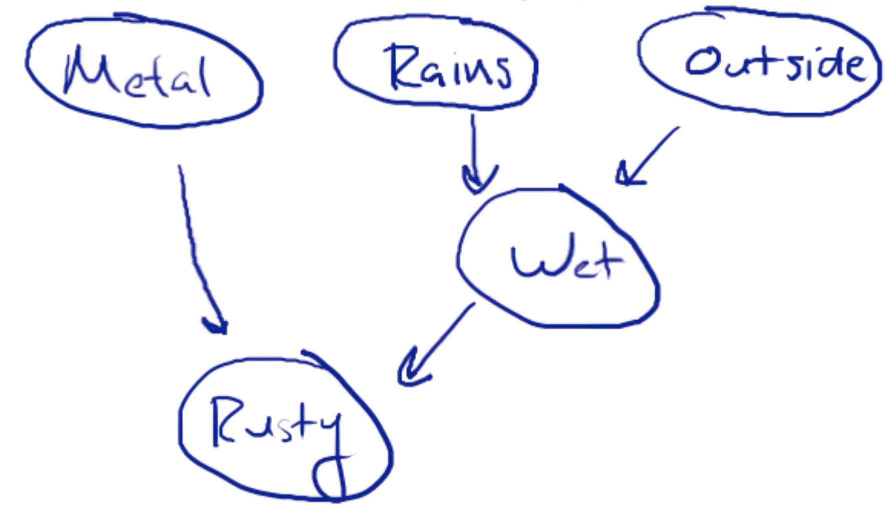
Markov blanket

Markov blanket of C = minimal set of obs'ns to make C independent of rest of graph



Learning Bayes nets

(see 10-708)



P(M) =

P(Ra) =

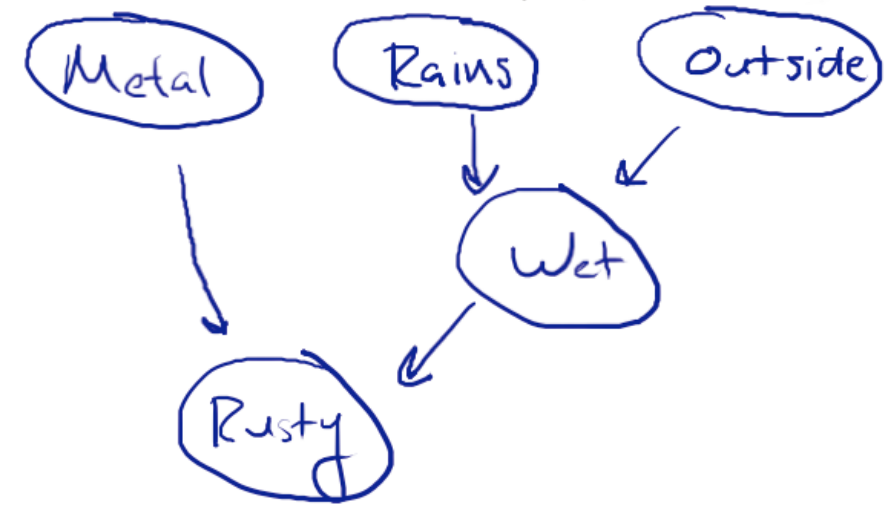
P(O) =

P(W | Ra, O) =

P(Ru | M, W) =

M	Ra	O	W	R
T	F	T	T	F
T	T	T	T	T
F	T	T	F	F
T	F	F	F	T
F	F	T	F	T

Laplace smoothing



P(M) =

P(Ra) =

P(O) =

P(W | Ra, O) =

P(Ru | M, W) =

M	Ra	O	W	R
T	F	T	T	F
T	T	T	T	T
F	T	T	F	F
T	F	F	F	T
F	F	T	F	T

Advantages of Laplace

- No division by zero
- No extreme probabilities
  - No near-extreme probabilities unless lots of evidence

Limitations of counting and Laplace smoothing

- Work ***only*** when all variables are observed in all examples
- If there are ***hidden*** or ***latent*** variables, more complicated algorithm—see 10-708
  - or just use a toolbox!