Bayesian Networks

Doug Downey
EECS 349 Machine Learning

Answering Queries: Summing Out

		Intelligence = i^1		Intelligence=i^2	
		Time=t^1	Time=t^2	Time=t^1	Time=t^2
Grade	g^1	0.05	0.02	0.15	0.03
	g^2	0.14	0.14	0.05	0.0
	g^3	0.10	0.25	0.01	0.02

P(*Grade*| *Time* = t^1)?

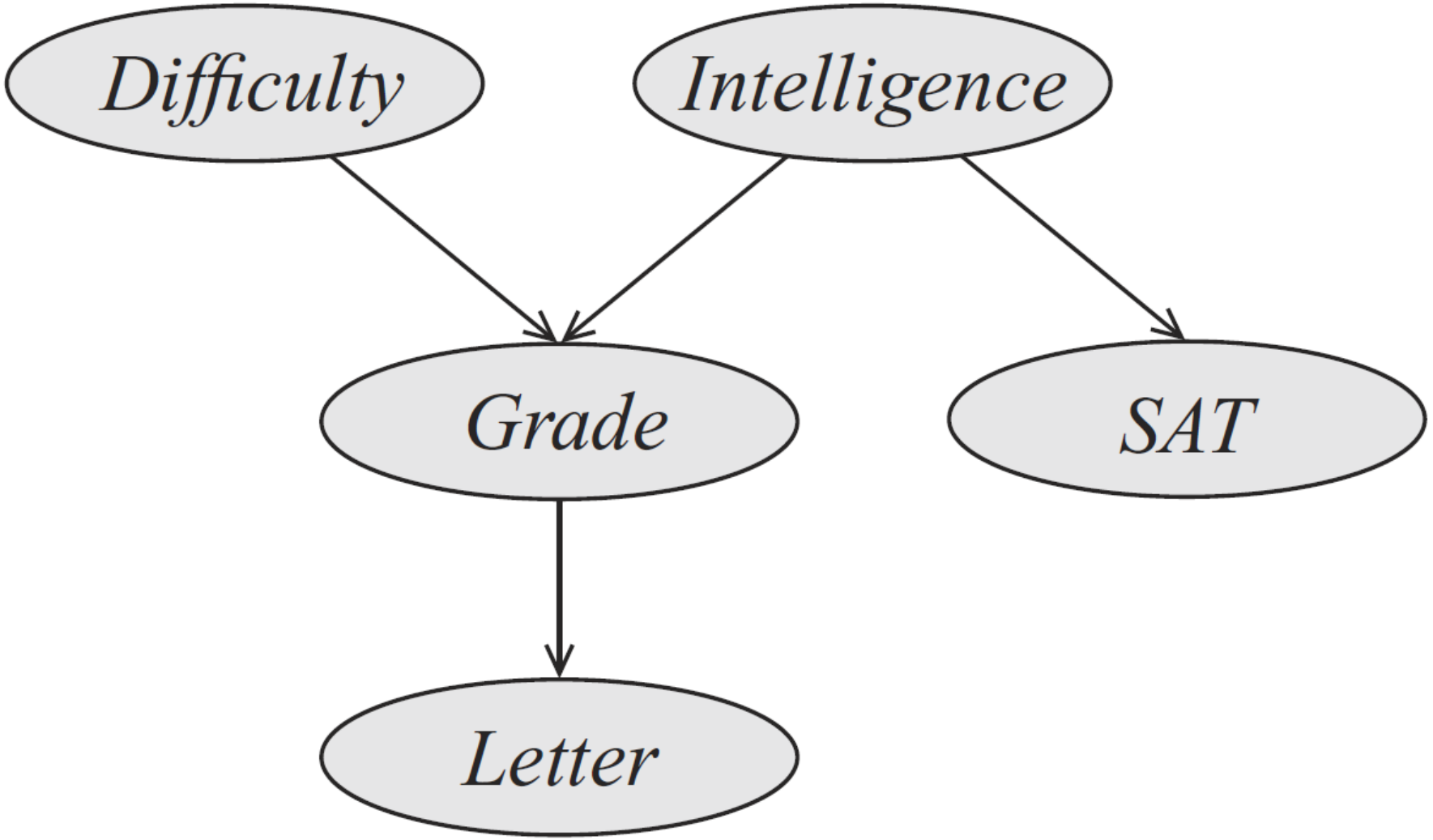
$$\sum_{v \in Val(Intelligence)} P(Grade, Intelligence = v \mid Time = t^1)$$

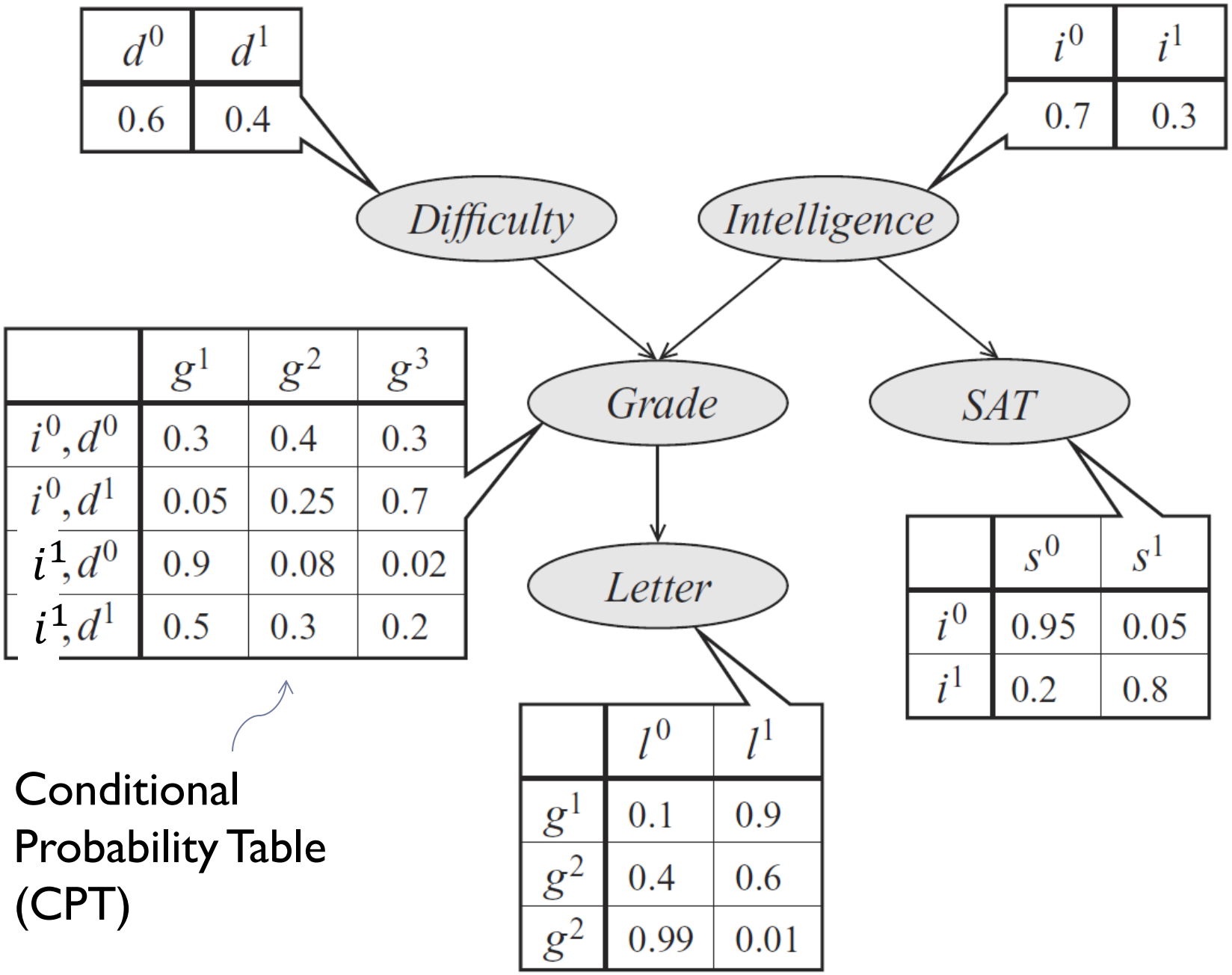
Answering Queries: Solved?

- Given the joint distribution, we can answer any query by summing
- …but, joint distribu Boolean variables has 2^500 -1 paramete)^150)
- For non-trivial pro boolean r.v.s or more), using the joint dist uires
  - Way too much **con** compute the sum
  - Way too many **observations** to learn the parameters
  - Way too much **space** to store the joint distribution

Bayesian Networks

- A general framework for modeling probability distributions
  - Expresses conditional independencies
- Begin with a graph
  - **Nodes**: Random variables (e.g. attributes, classes)
  - Directed **Edges**: Causal relationships


Difficulty
Intelligence
Grade
SAT
Letter



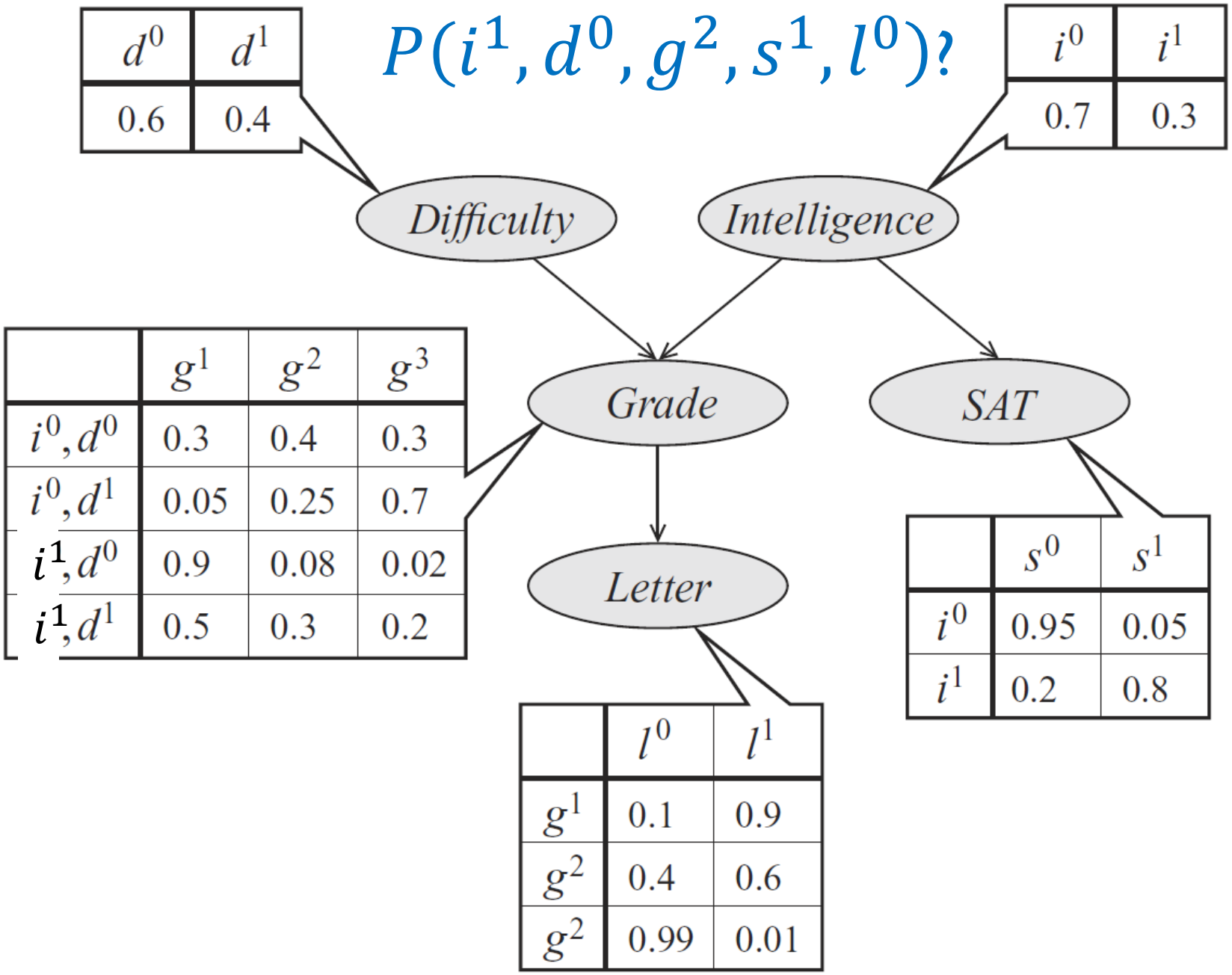
Conditional Probability Table (CPT)

What does this wacky thing do?

- BNs represent the joint distribution compactly
- You can obtain the BN's probabilities for an event by multiplying the relevant values from each CPT:

$$P(i^1, d^0, g^2, s^1, l^0) = \cdots$$

$P(i^1, d^0, g^2, s^1, l^0)$?



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

What does this wacky thing do?

- BNs represent the joint distribution compactly

- You can obtain the BN's probabilities for an event by multiplying the relevant values from each CPT:

$$P(i^1, d^0, g^2, s^1, l^0)$$
$$= P(i^1)P(d^0)P(g^2|i^1, d^0)P(s^1|i^1)P(l^0|g^2)$$
$$= 0.3 \cdot 0.6 \cdot 0.08 \cdot 0.8 \cdot 0.4 = 0.004608$$

Building a Bayes Net

- Create a node for each attribute or class variable
- Connect nodes with causal edges
  - How? Domain knowledge
    (or learn from data – more on this in 395/495 PGMs course)
- Obtain CPTs
  - How? Use **data**, or write from domain knowledge

Bayes Net Advantages

- Compactness
  - Our "student" network has **15** independent parameters
  - Vs. how many for a full joint distribution table?
- Ease of inference
  - (more on this later)

Computational Complexity

- How does training time and testing time complexity compare between decision trees and nearest-neighbor?

Think/Pair/Share

What's an upper-bound on the number of parameters in a Bayes Net?

Think

Start

End

Think/Pair/Share

What's an upper-bound on the number of parameters in a Bayes Net?

Pair

Start

End

Think/Pair/Share

What's an upper-bound on the number of parameters in a Bayes Net?

P(A | B, C, D) matrixL
= 8 independent parameters

but if A is conditionally independent of C and D given B:
P(A | B) matrixL
= 2 independent parameters

Share

once we know the value of B, the prob of A is unchanged if we also learn the values of C and D

Conditional Independence
P(A (upsidedownT) B | C) <=> P(A | C) * P(B | C) = P(A, B | C)
<=> P(A | B, C) = P(A | C)

k = max number of parents
v = max number of variable values
n = number of random variables

~Bayes Net~
UB(#params) = n*v^k*(v-1) < n*v^(k+1)

^^ where v^k = parent configs
^^ where (v-1) = for child distr.

^ not very big compared to the joint:

~Joint~
v^n - 1

~Bayes Net~
500 binary max 2 parents
< 500 * 2^3 = 4000

~Joint~
2^500 - 1

From Graphs to Independencies

- The Bayes Net encodes **independencies**
  - Independencies are what allow BN compactness

- Question:

  Which independencies are encoded in a given BN graph?

Global Semantics

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \mathrm{Pa}(X_i))$$

Local Independences

- Each node is conditionally independent of its non-descendants given its parents.

- Theorem:
  Local Independences ⇔ Global Semantics

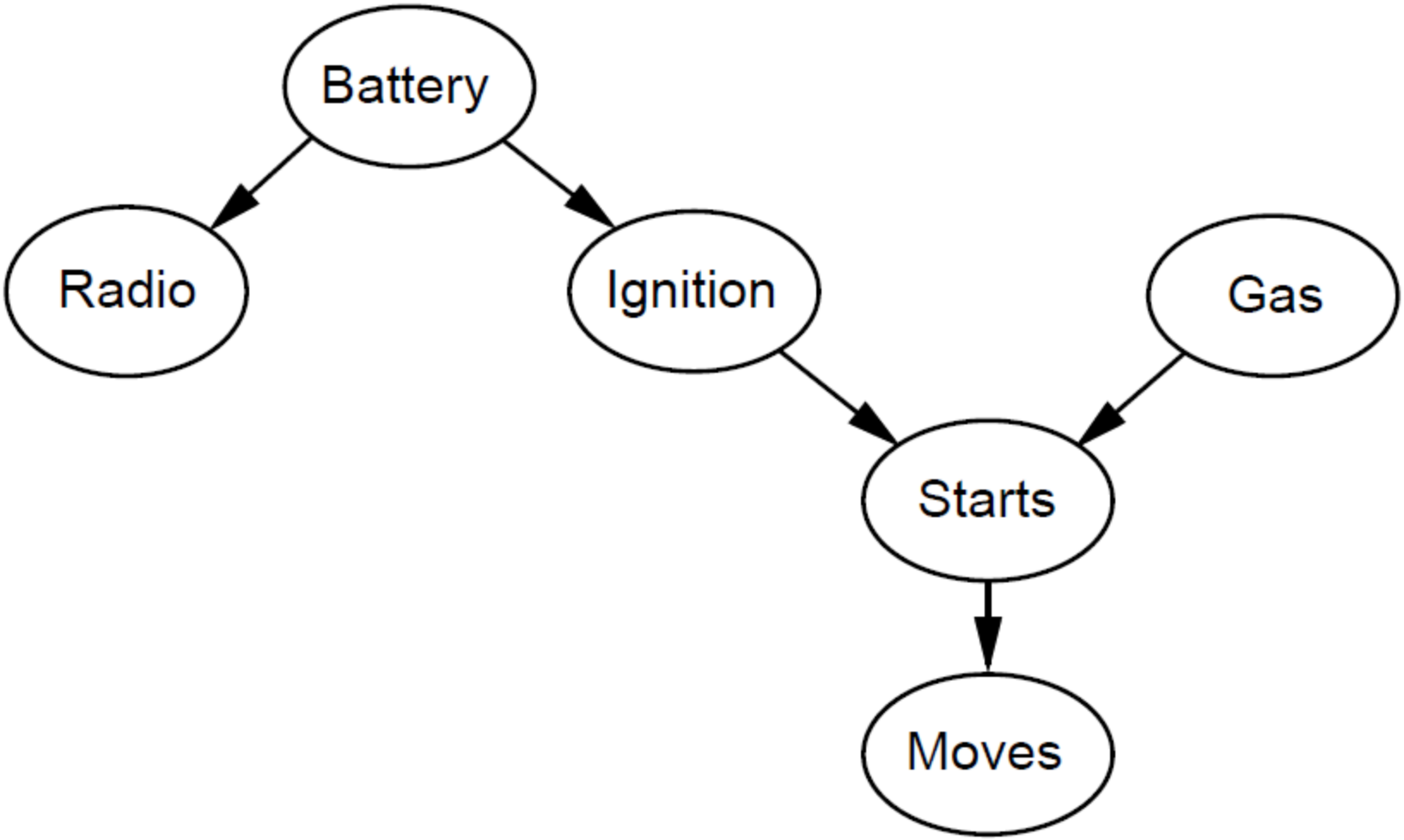
What does the graph look like…

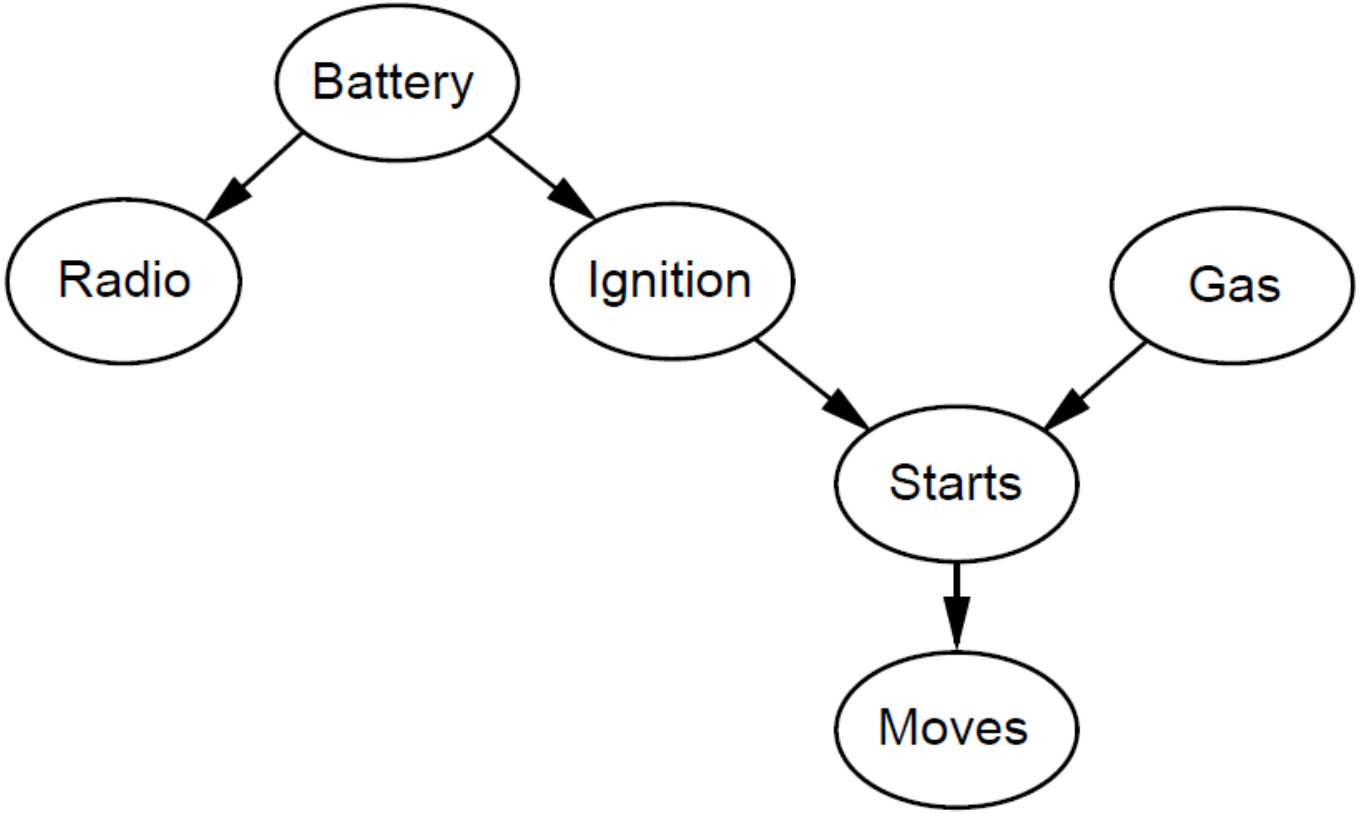
- No independence?
- All variables independent?
- Common Cause? Common Effect?
  - Correlation != causation
  - “Explaining away”

Active Trails and D-separation

- Two nodes in *G* are d-separated unless there is an active trail between them

- An *Active Trail* between nodes *X* and *Y* given evidence nodes ***E*** is any path between *X* and *Y* such that
  - For any v-structure (*A* => *C* <= *B*) on the path, either *C* or one of its descendents is in ***E***
  - No other nodes on the path are in ***E***


Battery
Radio
Ignition
Gas
Starts
Moves

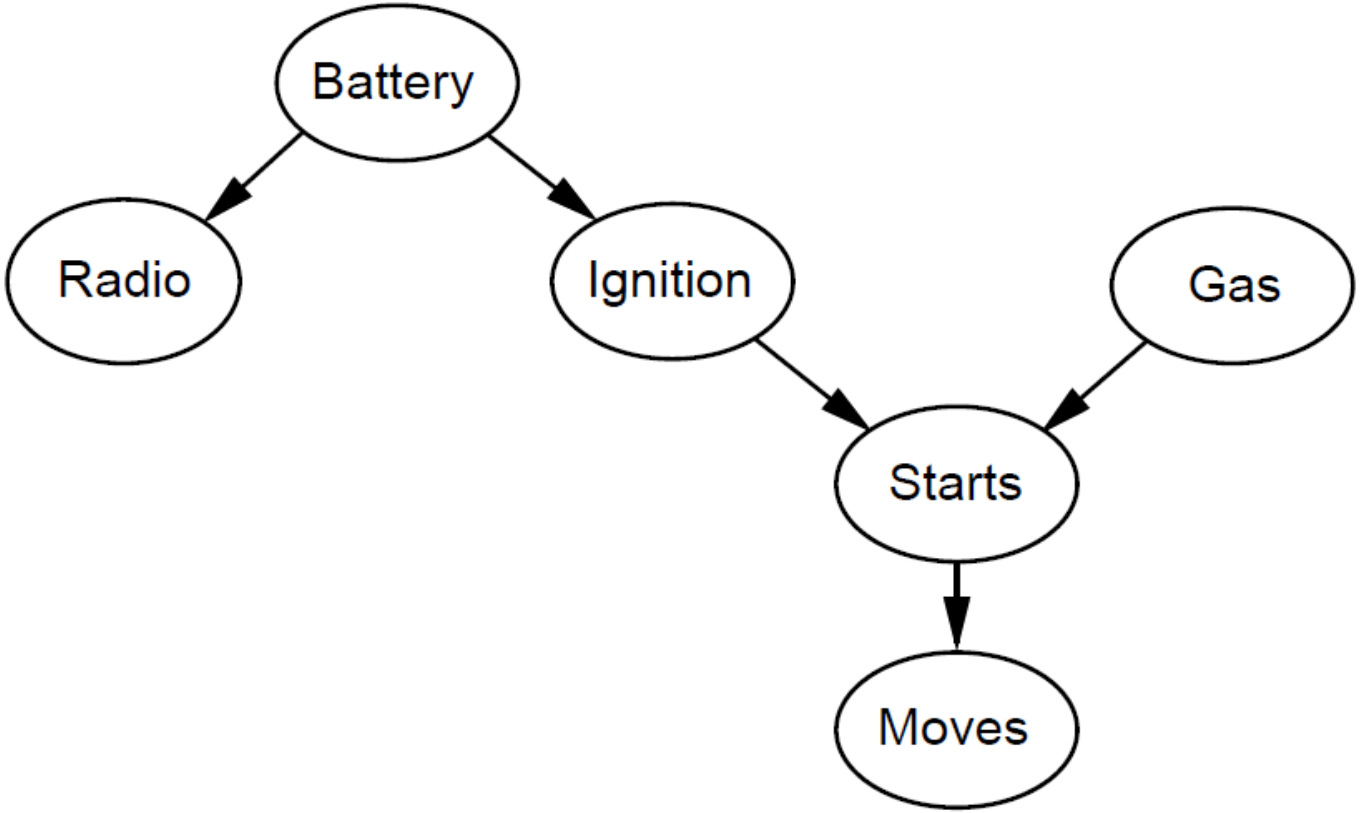


(Gas ⊥ Radio)? (Radio ⊥ Ignition)?
(Radio ⊥ Ignition | Battery)? (Gas ⊥ Radio | Moves)?

Think

Start

End

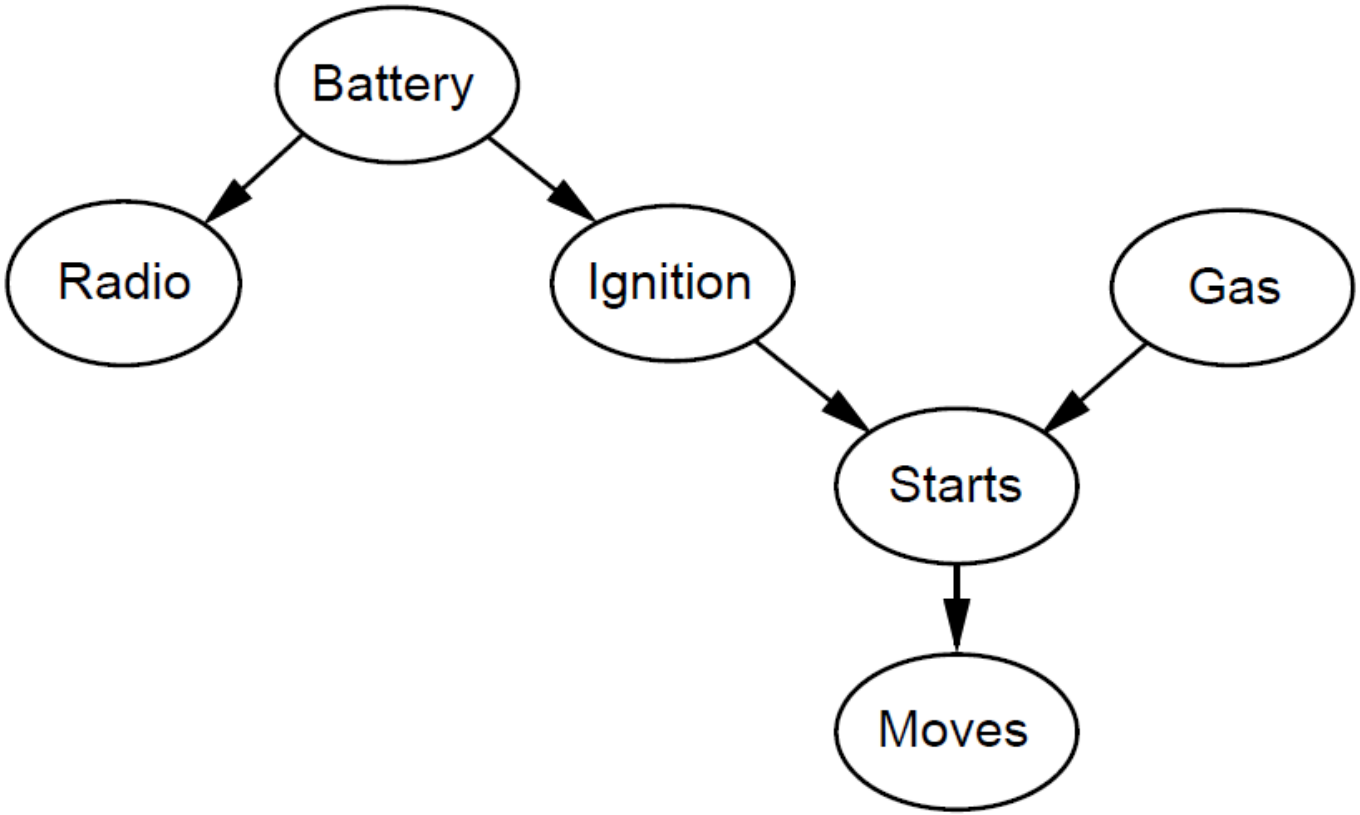


(Gas ⊥ Radio)? (Radio ⊥ Ignition)?
(Radio ⊥ Ignition | Battery)? (Gas ⊥ Radio | Moves)?

Pair

Start

End



(Gas ⊥ Radio)? (Radio ⊥ Ignition)?
(Radio ⊥ Ignition | Battery)? (Gas ⊥ Radio | Moves)?

Share