Artificial Intelligence

CSC 665

tyler dae devlin

PGMs IV

4.4.2024

- **Search:** make decisions by looking ahead

- **Logic:** deduce new facts from existing facts

- **Constraints:** find a way to satisfy a given specification

- **Probability:** reason quantitatively about uncertainty

- **Learning:** make future predictions from past observations

Modeling

Factored representations

- Not just good for compactness!

- Factored representations make it easy to construct **complex models** from **simple parts**

- Saw this with **propositional** vs. **first-order** logic

- Propositional formulas are **somewhat decomposable**

- E.g., to understand $(p \wedge \neg q) \vee (q \wedge \neg p)$, examine each disjunct separately

Factored representations

- But propositional symbols are **atomic**, which **limits composability**

- E.g., "every CSC 665 student knows AI" is **awkward** to express in propositional logic: JuanKnowsAI ∧ JamesKnowsAI ∧ NealKnowsAI ∧ …

- Yet the statement in English is a **simple**, relating a set of students to a disciplinary field

- We needed to **extend our language** to first-order logic in order to get this additional level of compositionality

Factored representations

- Bayesian networks represent a joint distribution as a product of **simple conditional distributions**

- Imagine writing down the joint distribution for the 6 variables on the right one row at a time...

- But specifying the conditional distributions and then multiplying is (relatively) **straightforward**

- From **simple parts**, a **complex whole**
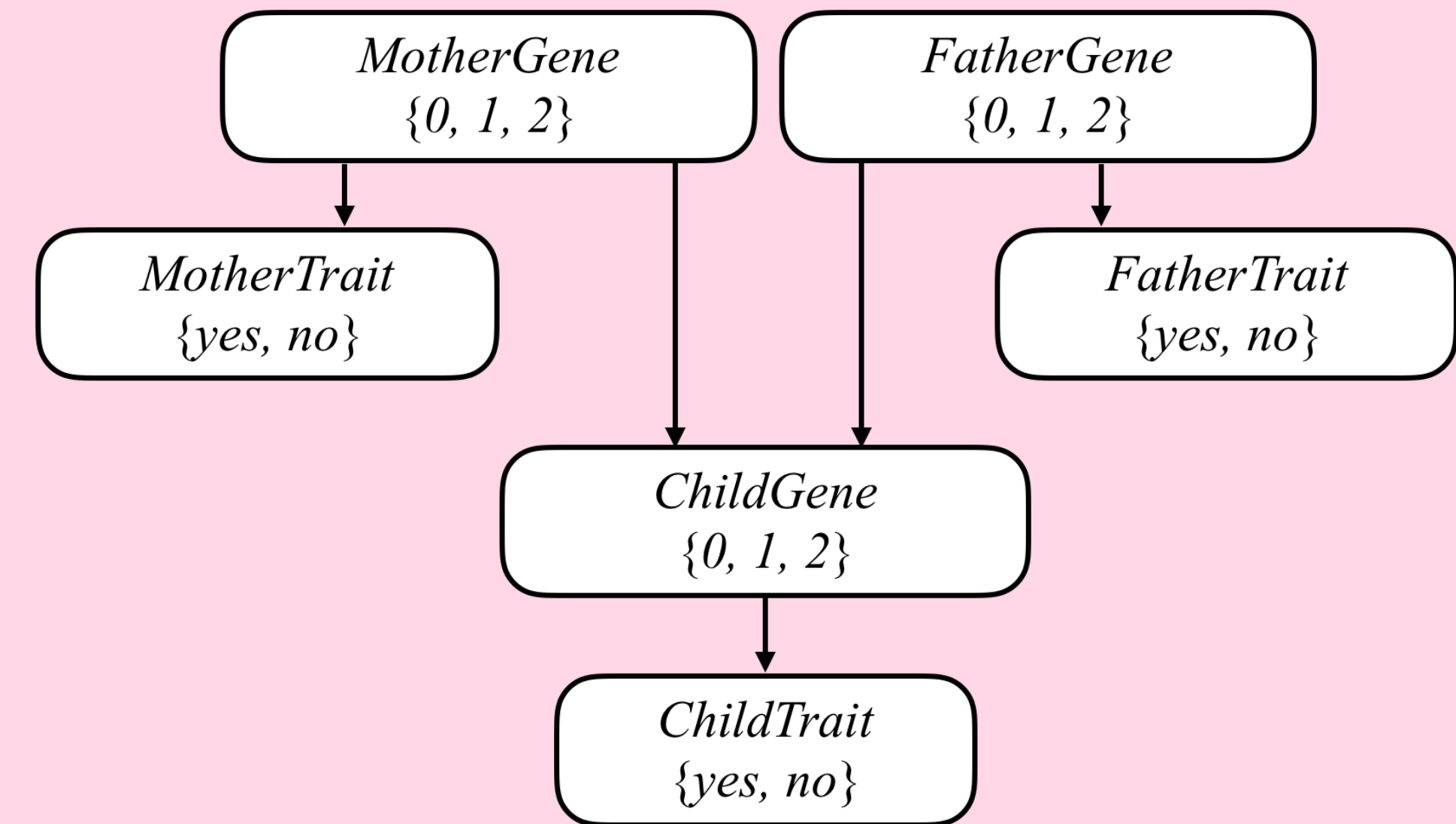
Inference

Types of inference

- Exact inference

  - Compute $P(X \mid E)$ exactly

  - Only tractable for small models with no continuous variables

- Approximate inference

  - Approximate $P(X \mid E)$

  - There's a chance the approximation is bad, and you have no way of knowing for sure

Exact inference by enumeration

- Divide set of all variables into

  - Query variables X

  - Evidence variables E

  - Other variables Y

$$P(x \mid e) = \frac{P(x, e)}{P(e)}$$
$$= \alpha P(x, e)$$
$$= \alpha \sum_y P(x, y, e)$$

We know how to compute $P(x, y, e)$ from the Bayesian network
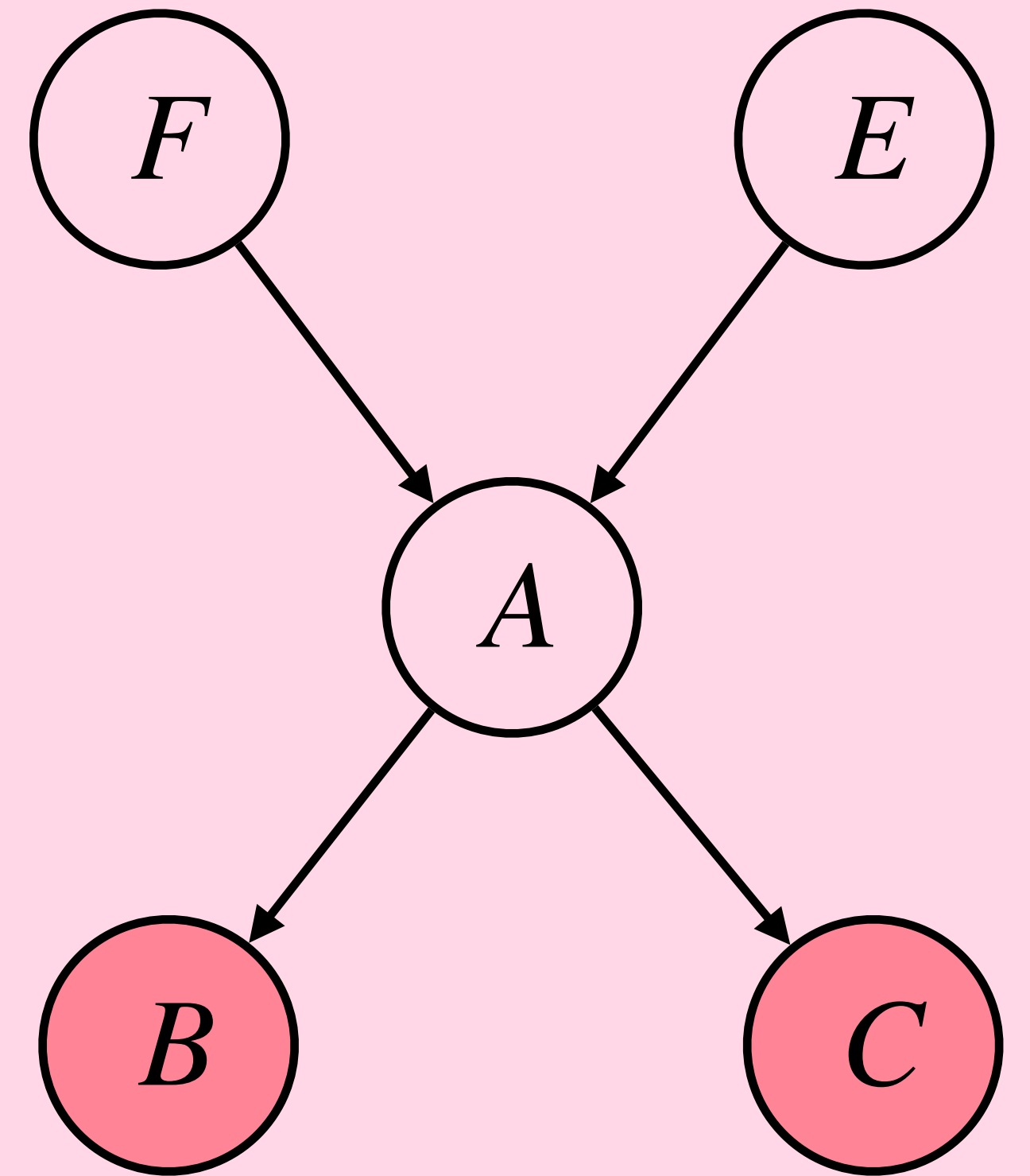
Exact inference by enumeration

- Query variables F

- Evidence variables B, C

- Other variables A, E

$$P(f \mid b, c) = \alpha P(f, b, c)$$

$$= \alpha \sum_a \sum_e P(f, e, a, b, c)$$

$$= \alpha \sum_a \sum_e P(f)P(e)P(a \mid f, e)P(b \mid a)P(c \mid a)$$

This requires us to sum 4 products of 5 factors each

In general, $O(2^n)$ products of $O(n)$ terms $\implies$ $O(n2^n)$ overall time complexity

Can get $O(2^n)$ with backtracking-like algorithm, but can't do any better

Exact inference by enumeration

function doInferenceByEnumeration(X, e)

- $Q(X) \leftarrow$ empty distribution over X

- **for** each value x of X:

  - $Q(x) \leftarrow$ jointProb($\{X, E, Y\}, \{X = x, E = e\}$) $\longleftarrow$ $P(X = x, E = e) = \sum\limits_{y} P(X = x, E = e, Y = y)$

- **return** normalize($Q(X)$)

function jointProb(vars, assignments)

- **if** vars is empty: **return** 1.0

- $V \leftarrow$ first(vars)

- **if** V has an assignment v in assignments:

  - **return** $P(v \mid$ parents(V)) $\cdot$ jointProb(rest(vars), assignments)

- **else**: **return** $\sum\limits_{v} P(v \mid$ parents(V)) $\cdot$ jointProb(rest(vars), assignments $\cup \{V = v\}$)

Sampling

- Suppose you have a coin $C \in \{h, t\}$

- You don't know if it's fair or biased, or what the bias parameter is

- How would you estimate $P(C = h)$?

- **Answer:** sample!

- Flip the coin N times. If there are n heads, estimate $P(C = h) \approx n/N$

- Is this a good estimator?

- Yes, in the sense that $n/N \to P(C = h)$ as $N \to \infty$

- The more samples we collect, the better the estimate

Estimating the joint distribution

- Can we sample from $P(X_1, \ldots, X_n)$ if we have its Bayesian network?

- Yes! As long as we can sample each conditional distribution (easy for discrete distributions)

- Algorithm:

  - assume $X_1, \ldots, X_n$ are in topological order

  - **for** $i = 1, \ldots, n$:

    - sample $x_i \sim P(X_i \mid \text{parents}(X_i))$, where the parents are assigned values from previous samples $x_1, \ldots, x_{i-1}$

  - **return** sample $(x_1, \ldots, x_n)$

- The relative frequency of a given assignment $(x_1, \ldots, x_n)$ approaches $P(x_1, \ldots, x_n)$ as more samples are generated