Artificial Intelligence

CSC 665

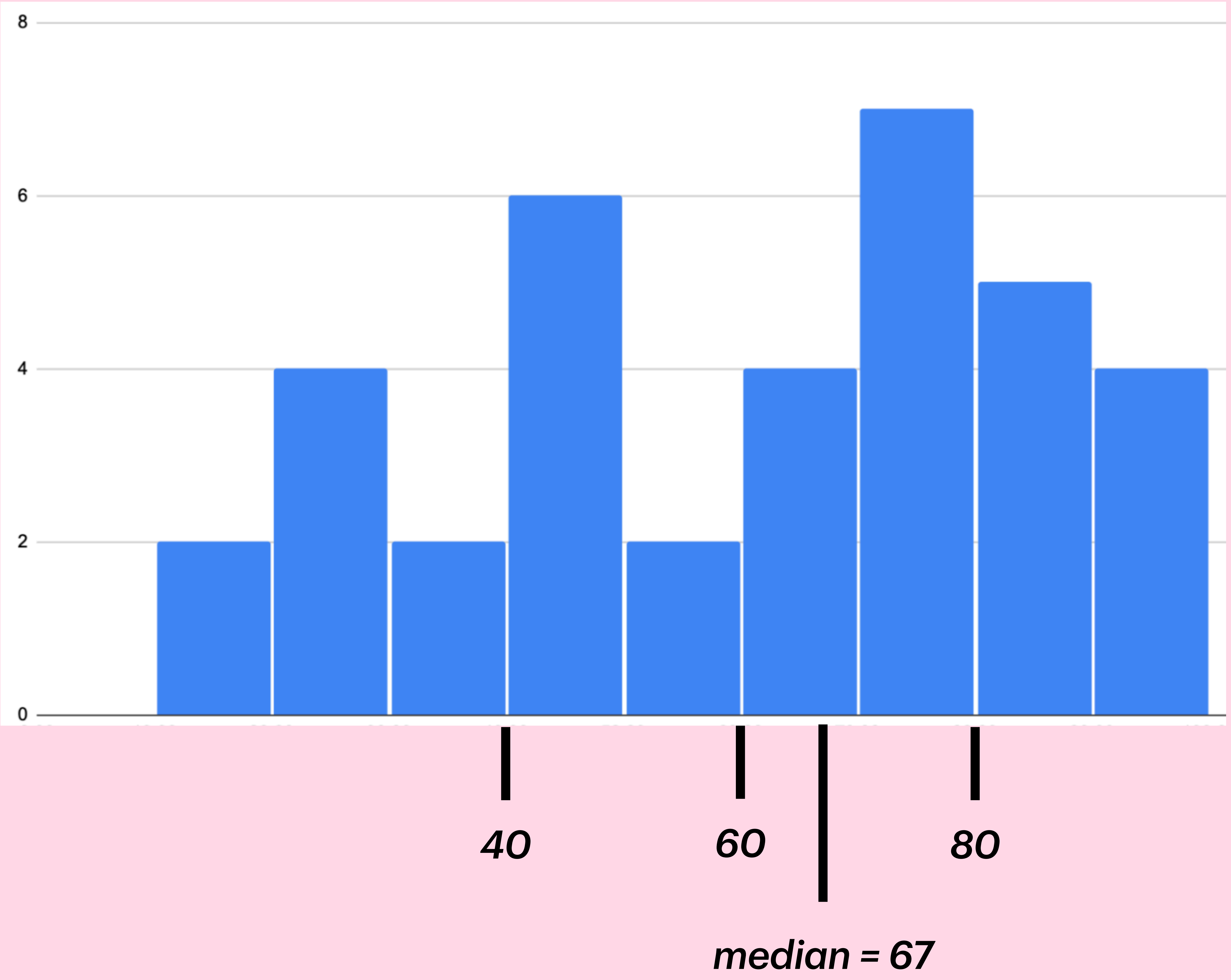
tyler dae devlin

PGMs V

10.24.2023

Administrivia

- HW2 grades released
- HW4 due Monday 10/30 (sorry)
- Late waivers will be applied near the end of the course
- Homeworks 1–4 + midterm 1 account for 50% of your grade
- Other 50%: HW5, HW6, midterm 2, final exam

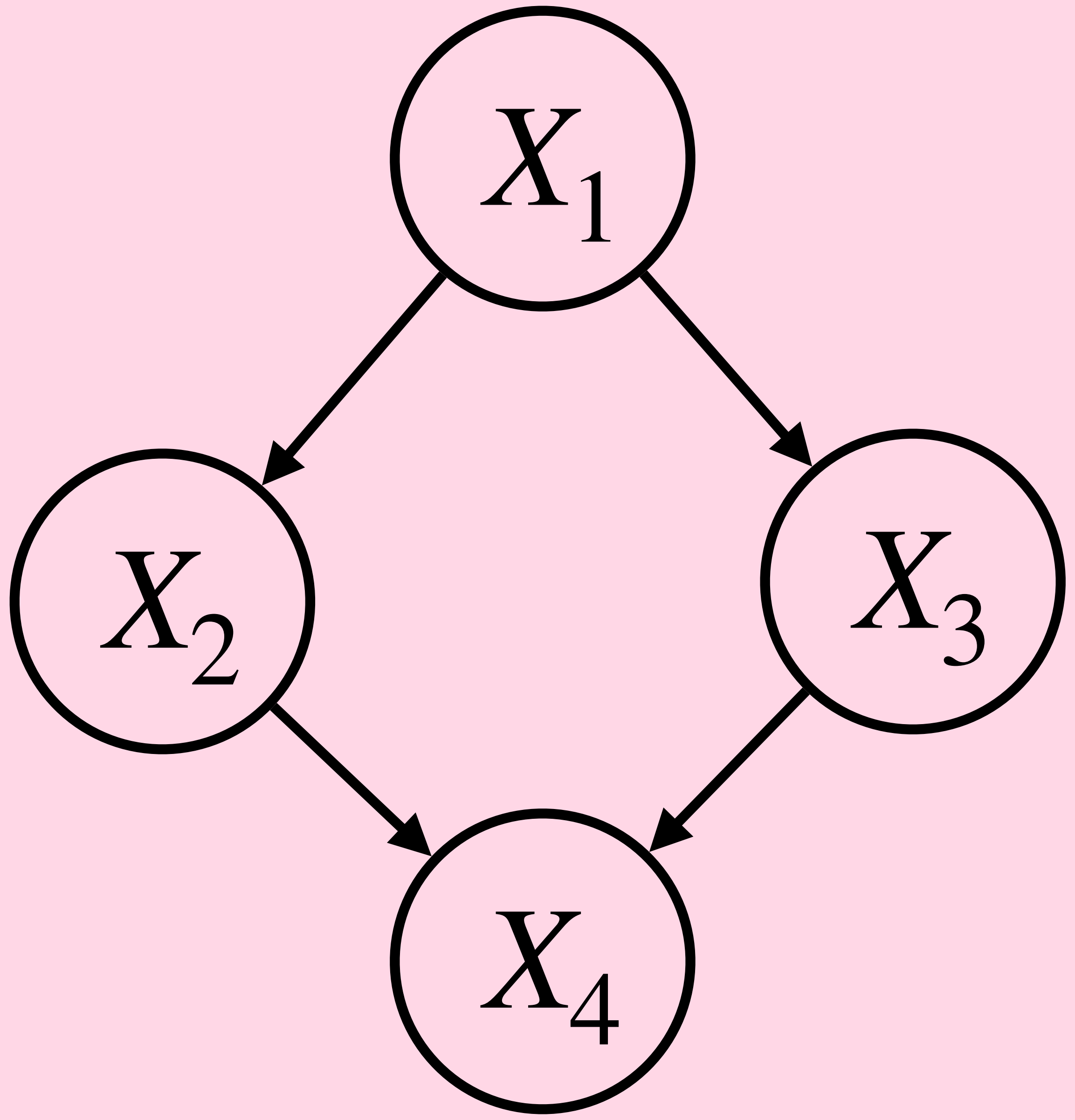


- **Search:** make decisions by looking ahead
- **Logic:** deduce new facts from existing facts
- **Constraints:** find a way to satisfy a given specification
- **Probability:** reason quantitatively about uncertainty
- **Learning:** make future predictions from past observations

Modeling

Bayesian networks

- Let $X = (X_1, \ldots, X_n)$ be random variables
- A Bayesian network is a directed acyclic graph (DAG) where each node is a random variable
- The Bayesian network specifies a joint distribution over X as a product of local conditional distributions, one for each node
- $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i))$$



$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2, X_3)$$

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

Query variables X, evidence variables E, other variables Y

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

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

Marginalization is exponential in the number of variables in the model

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

Forward sampling from 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

[forward sampling example]

Estimating the joint distribution

- Can we estimate $P(X_1, \ldots, X_n)$ if we can sample from it?
- **Yes!**
  - Let $n(x_1, \ldots, x_n)$ denote the number of times we observe the sample $(x_1, \ldots, x_n)$
  - Let N denote the total number of samples
  - Then $\dfrac{n(x_1, \ldots, x_n)}{N} \approx P(x_1, \ldots, x_n)$
- Is this useful?
- **No!**
- We can already exactly compute $P(x_1, \ldots, x_n)$ by multiplying local conditional distributions
- But these ideas are useful when designing techniques to estimate conditional distributions $P(x \mid e)$ or marginal distributions $P(x)$

Approximate inference: rejection sampling

- Can we turn the previous algorithm into a recipe for estimating $P(X = x \mid E = e)$?
- Yes! Just toss out any samples where $E \neq e$
- This is known as rejection sampling

[rejection sampling example]

Approximate inference: rejection sampling

- Can we turn the previous algorithm into a recipe for estimating $P(X = x \mid E = e)$?
- Yes! Just toss out any samples where $E \neq e$
- This is known as rejection sampling
- In most practical applications, the number of samples where $E = e$ is small, so you end up throwing away most samples...

Approximate inference: importance sampling

- Let $z = (x, y)$, i.e. all the variables other than the evidence variables
- We want to sample from $P(z \mid e)$, but we don't know how (other than rejection sampling so far)
- Suppose we have a distribution $Q(z)$ that's easy to sample from
- Idea: sample from Q, then re-weight to account for the difference between P and Q
- $\dfrac{n_Q(z)}{N} \dfrac{P(z \mid e)}{Q(z)}$

Approximate inference: importance sampling

- We want to sample from $P(x, y \mid e)$, but we don't know how (other than rejection sampling so far)
- Suppose we have a distribution $Q(x, y)$ that's easy to sample from
- Idea: sample from Q, then re-weight to account for the difference between P and Q
- $\dfrac{n_Q(z)}{N} \dfrac{P(z \mid e)}{Q(x, y)} \approx Q(z) \dfrac{P(z \mid e)}{Q(z)}$

Approximate inference: importance sampling

- We want to sample from $P(x, y \mid e)$, but we don't know how (other than rejection sampling so far)
- Suppose we have a distribution $Q(x, y)$ that's easy to sample from
- Idea: sample from Q, then re-weight to account for the difference between P and Q
- $\frac{n_Q(z)}{N} \frac{P(z \mid e)}{Q(z)} \approx Q(z) \frac{P(z \mid e)}{Q(z)} = P(z \mid e)$

Approximate inference: importance sampling

- Importance sampling can be much more **sample-efficient** than rejection sampling
- But still possible for samples to have samples to have **zero or near-zero weight**
- Also possible for samples to have **arbitrarily large weight**, causing erratic behavior
- Both cases are frequent when there is a **large mismatch** between $P(z \mid e)$ and $Q(z)$

Approximate inference: likelihood weighting

- Likelihood weighting is a type of importance sampling method
- Define $Q(z) = \prod_{i=1}^{m} P(z_i \mid \text{parents}(Z_i))$
- I.e., forward sample the unobserved variables
- The weight of a sample z is

$$\frac{P(z \mid e)}{Q(z)} = \alpha \frac{P(z, e)}{Q(z)}$$

$$= \alpha \frac{\prod_i P(z_i \mid \text{parents}(Z_i)) \prod_j P(e_j \mid \text{parents}(E_j))}{\prod_i P(z_i \mid \text{parents}(Z_i))}$$

$$= \alpha \prod_j P(e_j \mid \text{parents}(E_j))$$

Approximate inference: likelihood weighting

function likelihoodWeighting(N) :

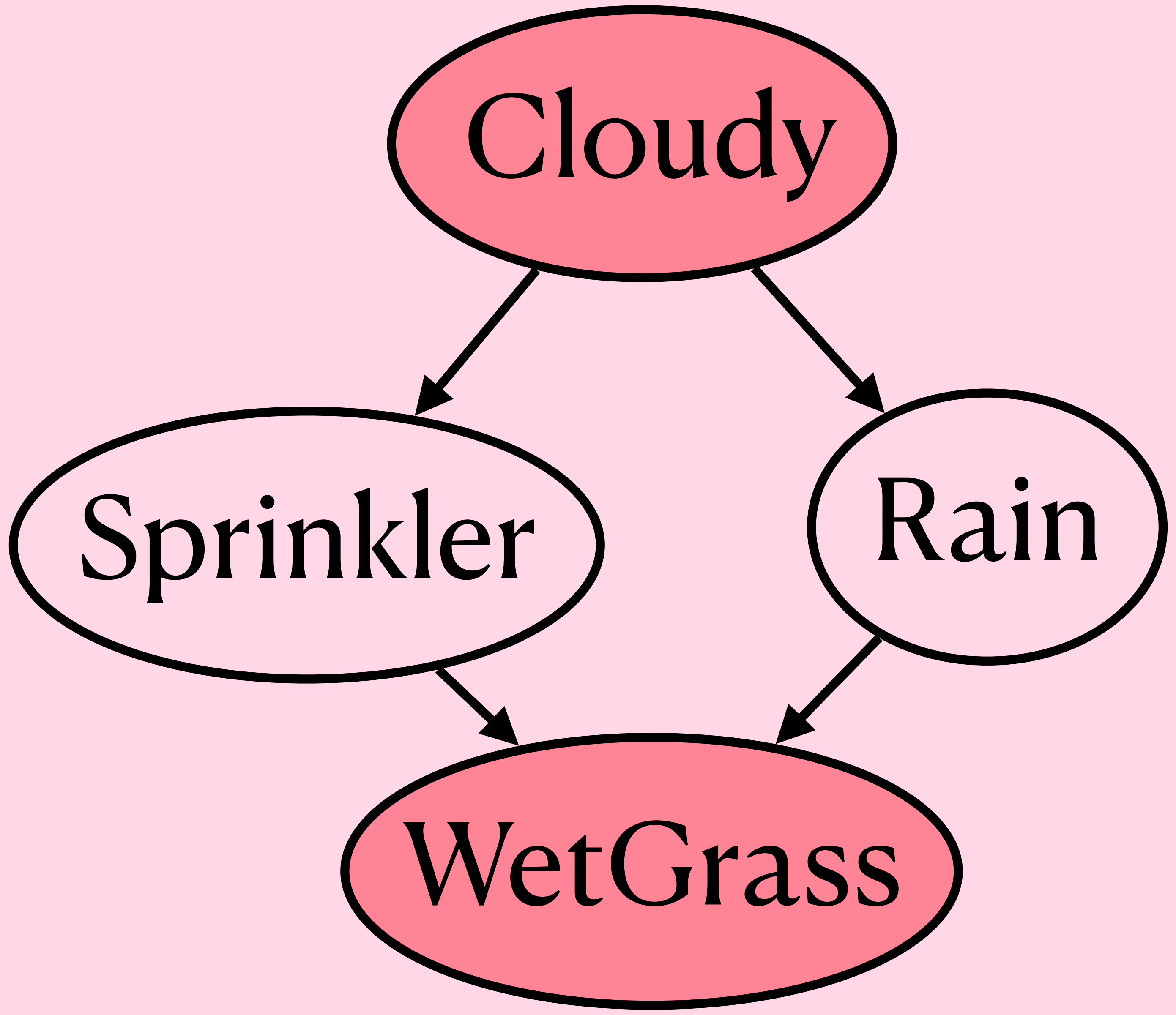
- **for** $j = 1,\ldots,N$:
  - $(z, e), w \leftarrow$ weightedSample()
  - $W[z] \leftarrow W[z] + w$
- **return** normalize(W)

function weightedSample() :

- $w \leftarrow 1$
- **for** each variable X in the Bayesian network in topological order:
  - **if** $X \in E$ with observed value e :
    - $w \leftarrow w \cdot P(X = e \mid \text{parents}(X))$
  - **else**: sample from $P(X \mid \text{parents}(X))$
- **return** $(z, e), w$

Likelihood weighting example

- Observe Cloudy $= 1$, WetGrass $= 1$
- To generate one sample:
  - **Cloudy:** $w \leftarrow w \cdot P(\text{Cloudy} = 1) = 1 \cdot 0.5$
  - **Sprinkler:** sample from $P(\text{Sprinkler} \mid \text{Cloudy} = 1)$. Suppose we sample Sprinkler $= 0$.
  - **Rain:** sample from $P(\text{Rain} \mid \text{Cloudy} = 1)$. Suppose we sample Rain $= 1$.
  - **WetGrass:** $w \leftarrow w \cdot P(\text{WetGrass} = 1 \mid \text{Sprinkler} = 0, \text{Rain} = 1) = 0.5 \cdot 0.9$
- **Final sample:** (Sprinkler $= 0$, Rain $= 1$) with weight 0.45



Summary

- (Exact) **inference by enumeration**: exponential in number of variables
- (Approximate) **forward sampling** from $P(x, y, e)$: useful for computing marginals $P(x)$
- (Approximate) **rejection sampling** from $P(x \mid e)$: simple but wasteful
- (Approximate) **importance sampling** from $P(x, y \mid e)$ by re-weighting $Q(x, y)$: more efficient than rejection sampling but suffers when distributions are mismatched
- (Approximate) **likelihood weighting** to sample from $P(x, y \mid e)$ using a Q inspired by forward sampling: same pros and cons as importance sampling in general

More approximate inference

- Markov chain Monte Carlo (MCMC) methods including
  - Gibbs sampling
  - Metropolis-Hastings method
- Variational methods
- Message passing and belief propagation