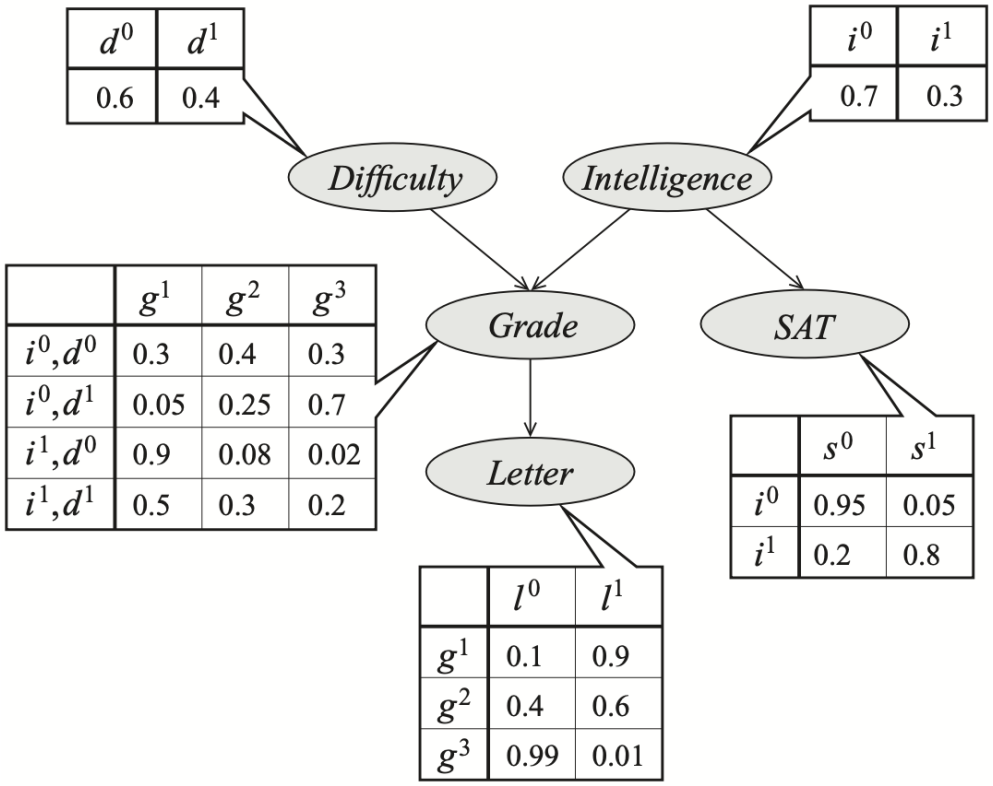
Sampling for graphical models

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"



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^3	0.99	0.01

$$P(L, G, S, D, I) = P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination:**

$$P(L, G, S, D, I) = P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

$$P(I = i^1, G = g^2, S = s^0) = \sum_{l,d} P(L = l, G = g^2, S = s^0, D = d, I = i^1)$$

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination:**

$$P(L, G, S, D, I) = P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

$$P(I = i^1, G = g^2, S = s^0) = \sum_{l,d} P(L = l, G = g^2, S = s^0, D = d, I = i^1)$$

$$= \sum_{l,d} P(L = l \mid G = g^2)P(G = g^2 \mid D = d, I = i^1)P(S = s^0 \mid I = i^1)P(D = d)P(I = i^1)$$

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination:**

$$P(L, G, S, D, I) = P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

$$P(I = i^1, G = g^2, S = s^0) = \sum_{l,d} P(L = l, G = g^2, S = s^0, D = d, I = i^1)$$

$$= \sum_{l,d} P(L = l \mid G = g^2)P(G = g^2 \mid D = d, I = i^1)P(S = s^0 \mid I = i^1)P(D = d)P(I = i^1)$$

$$= \left(\sum_{d} P(G = g^2 \mid D = d, I = i^1)P(D = d)\right)P(S = s^0 \mid I = i^1)P(I = i^1)$$

$$P(G = g^2, S = s^0) = \sum_{i} \left(\sum_{d} P(G = g^2 \mid D = d, I = i)P(D = d)\right)P(S = s^0 \mid I = i)P(I = i)$$

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination**
  - Straightforward, but can be computationally prohibitive
- **Variational inference**
  - Approximate, biased
- **Sampling strategies**
  - Rejection sampling
  - Importance weighted sampling
  - MCMC

Rejection sampling

Rejection sampling

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Algorithm 12.1 Forward Sampling in a Bayesian network

Procedure Forward-Sample (
  $\mathcal{B}$ // Bayesian network over $\mathcal{X}$
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 **for** $i = 1, \ldots, n$
3 $\boldsymbol{u}_i \leftarrow \boldsymbol{x}\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
4 Sample x_i from $P(X_i \mid \boldsymbol{u}_i)$
5 **return** $(x_1, \ldots, x_n)$

Rejection sampling

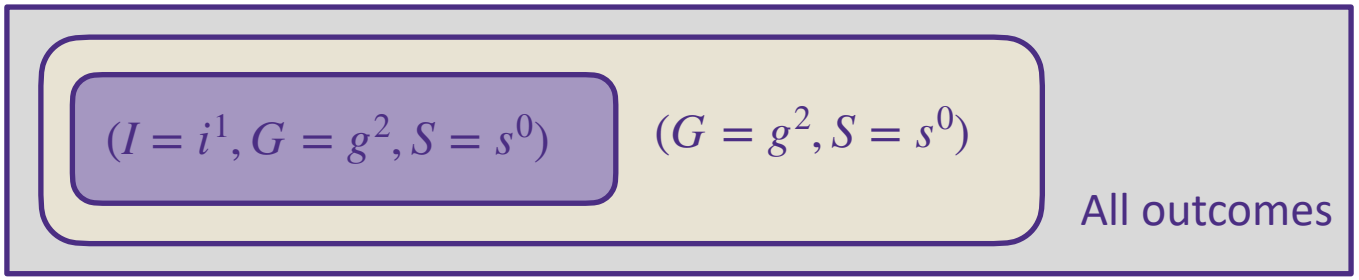
Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Draw M samples $\{(L_k, G_k, S_k, D_k, I_k)\}_{k=1}^{M}$ using forward sampling and

Set $\chi = \{k : G_k = g^2, S_k = s^0\}$ and output

$$\widehat{P}(I = i^1 \mid G = g^2, S = s^0) = \frac{1}{|\chi|} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\}$$



Rejection sampling

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Draw M samples $\{(L_k, G_k, S_k, D_k, I_k)\}_{k=1}^{M}$ using forward sampling and

Set $\chi = \{k : G_k = g^2, S_k = s^0\}$ and output

$$\widehat{P}(I = i^1 \mid G = g^2, S = s^0) = \frac{1}{|\chi|} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\}$$

$$\mathbb{E}\left[\widehat{P}(I = i^1 \mid G = g^2, S = s^0)\right] = \mathbb{E}\left[\frac{1}{|\chi|} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\}\right]$$

$$= \sum_{n=1}^{\infty} \mathbb{E}\left[\frac{1}{n} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\} \,\middle|\, |\chi| = n\right] \mathbb{P}(|\chi| = n)$$

$$= \sum_{n=1}^{\infty} \frac{n P(I = i^1 \mid G = g^2, S = s^0)}{n} \mathbb{P}(|\chi| = n)$$

Unbiased estimator of $P(I = i^1 \mid G = g^2, S = s^0)$!

Rejection sampling

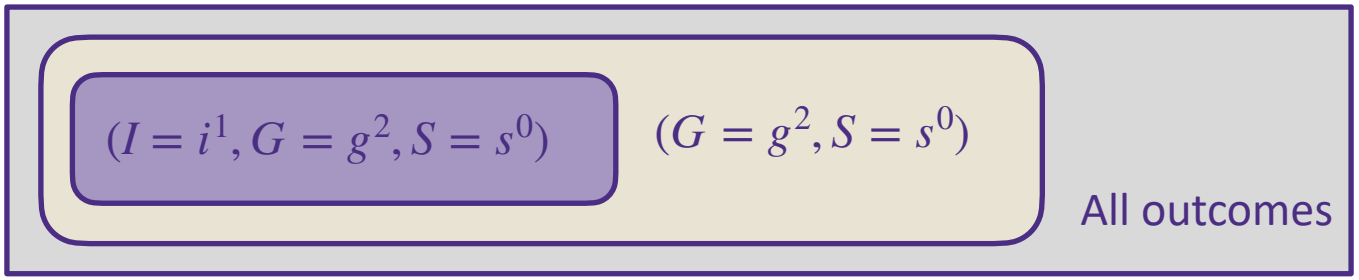
Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Draw M samples $\{(L_k, G_k, S_k, D_k, I_k)\}_{k=1}^{M}$ using forward sampling and

Set $\chi = \{k : G_k = g^2, S_k = s^0\}$ and output

$$\widehat{P}(I = i^1 \mid G = g^2, S = s^0) = \frac{1}{|\chi|} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\}$$



How big goes M need to be to get an accurate estimate?

Rejection sampling

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Draw M samples $\{(L_k, G_k, S_k, D_k, I_k)\}_{k=1}^{M}$ using forward sampling and

Set $\chi = \{k : G_k = g^2, S_k = s^0\}$ and output

$$\widehat{P}(I = i^1 \mid G = g^2, S = s^0) = \frac{1}{|\chi|} \sum_{k \in \chi} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\}$$

Rejection sampling takeaways:

- Very simple to implement
- May require an enormous amount of samples if the conditional statement is rare.
    - Consider P(disease | symptoms). Any precise set of symptoms is going to be rare.

Importance sampling

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination**
  - Straightforward, but can be computationally prohibitive
- **Variational inference**
  - Approximate, biased
- **Sampling strategies**
  - Rejection sampling 
  - Importance weighted sampling
  - MCMC

Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\widehat{\mu}_P := \frac{1}{M} \sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\widehat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\hat{\mu}_P := \frac{1}{M}\sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\hat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\hat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\hat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

$$\mathbb{E}_Q\left[\frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}\right] =$$

Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\hat{\mu}_P := \frac{1}{M}\sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\hat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\hat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\hat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

$$\mathbb{E}_Q\left[\frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}\right] = \mathbb{E}_Q\left[f(Y_1)\frac{P(Y_1)}{Q(Y_1)}\right]$$

$$= \sum_x Q(x)\cdot f(x)\frac{P(x)}{Q(x)}$$

$$= \sum_x P(x)\cdot f(x)$$

$$= \mathbb{E}_P[f(X)]$$

Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\hat{\mu}_P := \frac{1}{M}\sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\hat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\hat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\hat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

Moreover, as $M \to \infty$ we have $\frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)} \sim \mathcal{N}\left(\mathbb{E}_P[f(X)], \sigma_Q^2/M\right)$ where

$$\sigma_Q^2 = \mathbb{E}_Q[\left(f(Y)\frac{P(Y)}{Q(Y)}\right)^2] - \mathbb{E}_Q[f(Y)\frac{P(Y)}{Q(Y)}]^2$$

$$= \mathbb{E}_P[f(X)^2\frac{P(X)}{Q(X)}] - \mathbb{E}_P[f(X)]^2$$

σ_Q^2 is minimizes when

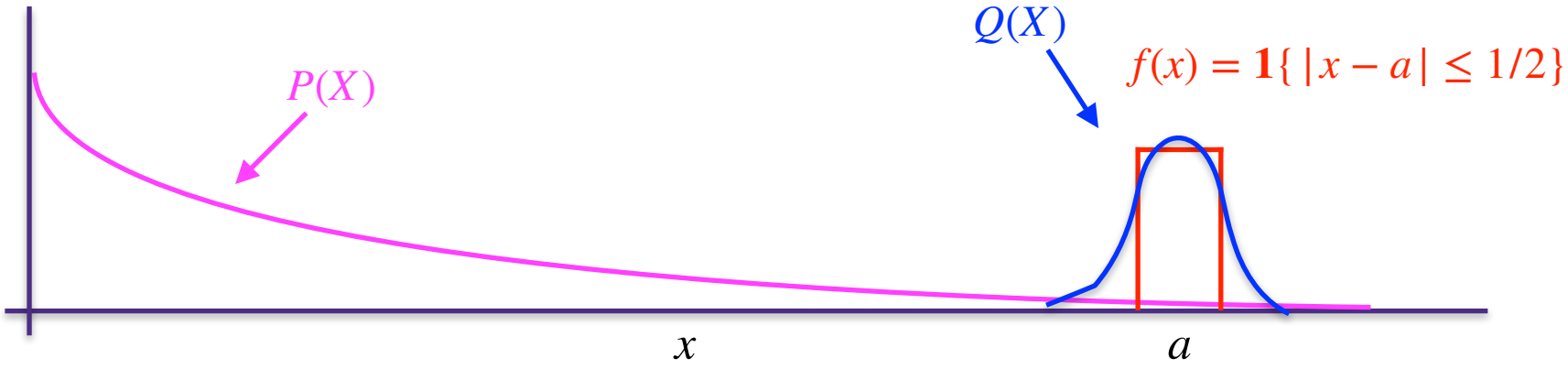
$Q(x) \propto |f(x)|P(x)$

Importance sampling

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\widehat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\widehat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

Example: rare event sampling

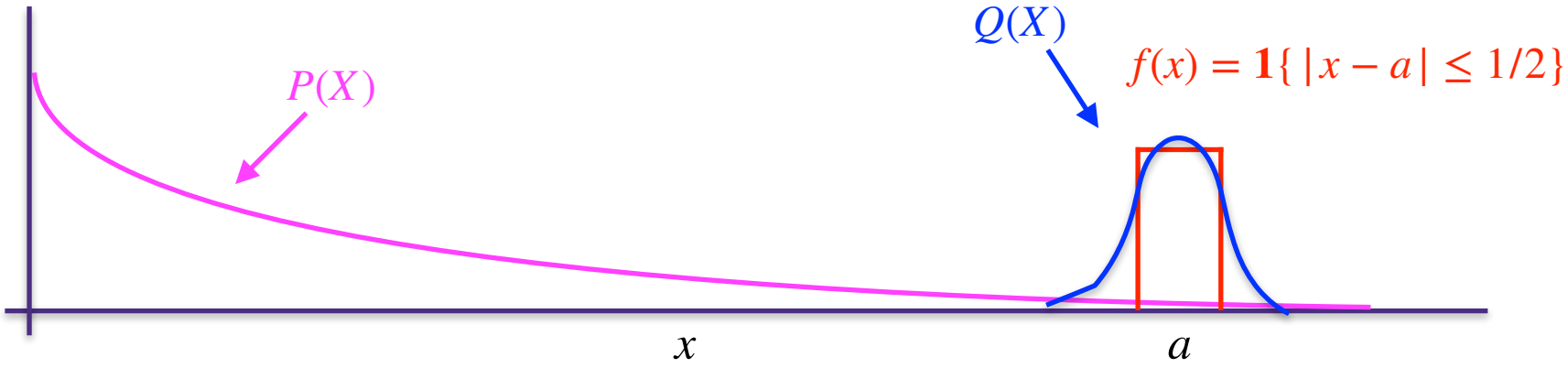


Importance sampling

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\widehat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\widehat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

Example: rare event sampling



$$\mathbb{E}_P[f(X)] = \int_x P(x)\mathbf{1}\{\,|x-a| \le 1/2\}dx \approx P(a)$$

$\sigma_Q^2 \approx P(a)^2 \implies$ sampling from Q requires $M \ge \epsilon^{-2}$ for relative error $\frac{\widehat{\mu}_Q - \mu}{\mu} \le \epsilon$

$\sigma_P^2 \approx P(a) \implies$ sampling from P requires $M \ge \frac{\epsilon^{-2}}{P(a)}$ for relative error $\frac{\widehat{\mu}_P - \mu}{\mu} \le \epsilon$

Self-normalized Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\widehat{\mu}_P := \frac{1}{M}\sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\widehat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\widehat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\widehat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

Self-normalized Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\widehat{\mu}_Q^{sn} := \frac{\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}}{\sum_{i=1}^{M}\frac{P(Y_i)}{Q(Y_i)}}$ then $\mathbb{E}_Q\left[\widehat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

Biased $\mathbb{E}_Q[\widehat{\mu}_Q^{sn}] \neq \mu$ but is asymptotically consistent since $\mathbb{E}_Q[\frac{P(Y_i)}{Q(Y_i)}] = 1$

Self-normalized Importance sampling

Fix any function $f : \mathcal{X} \to [0,1]$ and suppose we wish to estimate $\mu = \mathbb{E}_P[f(X)]$

If I draw $X_1, \ldots, X_M \sim P$ and define $\hat{\mu}_P := \frac{1}{M}\sum_{i=1}^{M} f(X_i)$ then $\mathbb{E}_P\left[\hat{\mu}_P\right] = \mathbb{E}_P[f(X)]$

Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\hat{\mu}_Q := \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}$ then $\mathbb{E}_Q\left[\hat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

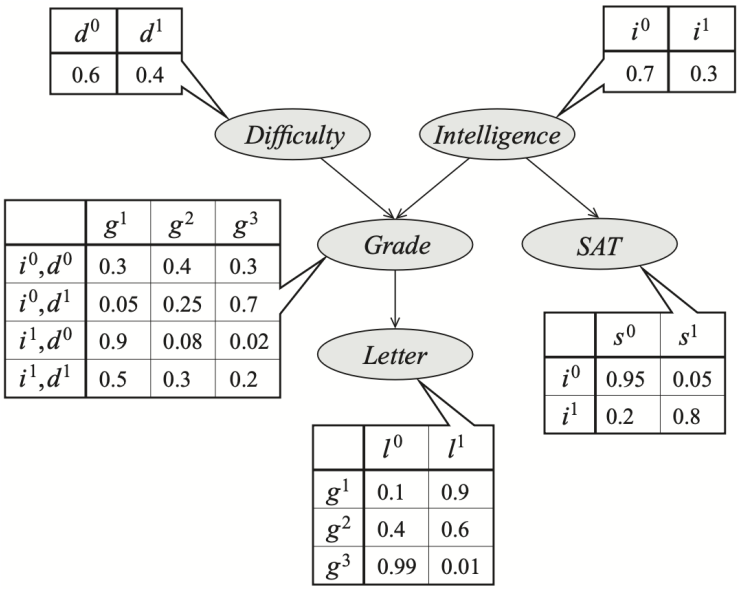
Self-normalized Importance sampling with candidate distribution Q:

If I draw $Y_1, \ldots, Y_M \sim Q$ and define $\hat{\mu}_Q^{sn} := \frac{\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}}{\sum_{i=1}^{M}\frac{P(Y_i)}{Q(Y_i)}}$ then $\mathbb{E}_Q\left[\hat{\mu}_Q\right] = \mathbb{E}_P[f(X)]$

If we only know P up to a normalizing constant such that we have $\widetilde{P}(x) = ZP(x)$, we can use $\widetilde{P}$ to obtain the same estimator:

$$\hat{\mu}_Q^{sn} = \frac{\sum_{i=1}^{M} f(Y_i)\frac{\widetilde{P}(Y_i)}{Q(Y_i)}}{\sum_{i=1}^{M}\frac{\widetilde{P}(Y_i)}{Q(Y_i)}} = \frac{\sum_{i=1}^{M} f(Y_i)\frac{P(Y_i)}{Q(Y_i)}}{\sum_{i=1}^{M}\frac{P(Y_i)}{Q(Y_i)}}$$

Bayesian Network Inference with Importance Sampling



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

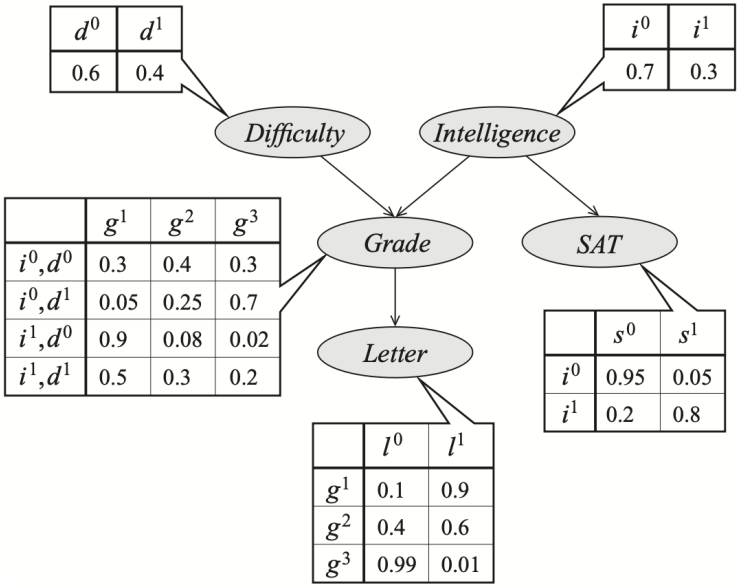
	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

$$P(X) := P(L, G, S, D, I)$$
$$= P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

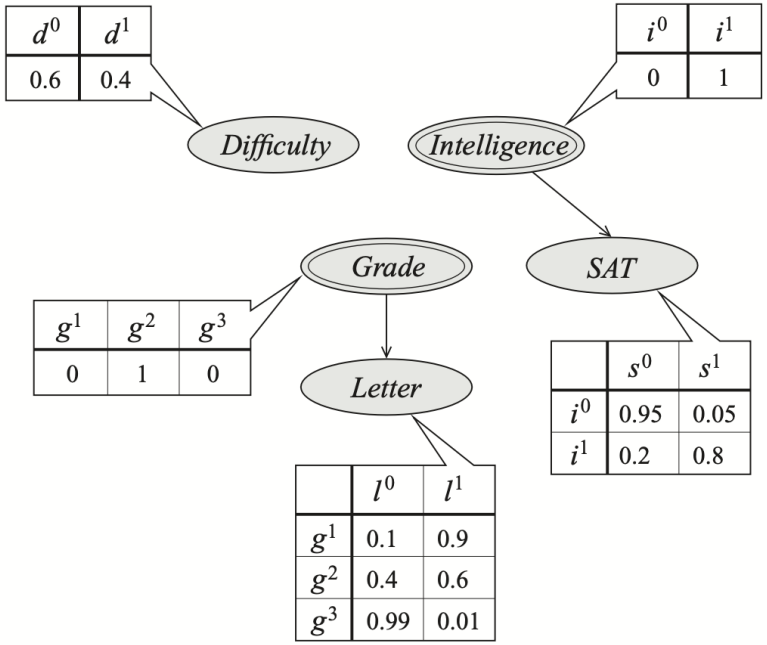
If $X := (L, G, S, D, I)$ and $P(X)$ denotes a Bayesian network then as we saw earlier, sampling from conditionals of P directly is awkward and inefficient. Can we define a convenient Q and use important sampling?

Bayesian Network Inference with Importance Sampling

Original network $\mathcal{B}$



"Mutilated" network $\mathcal{B}_{I=i^1,G=g^2}$



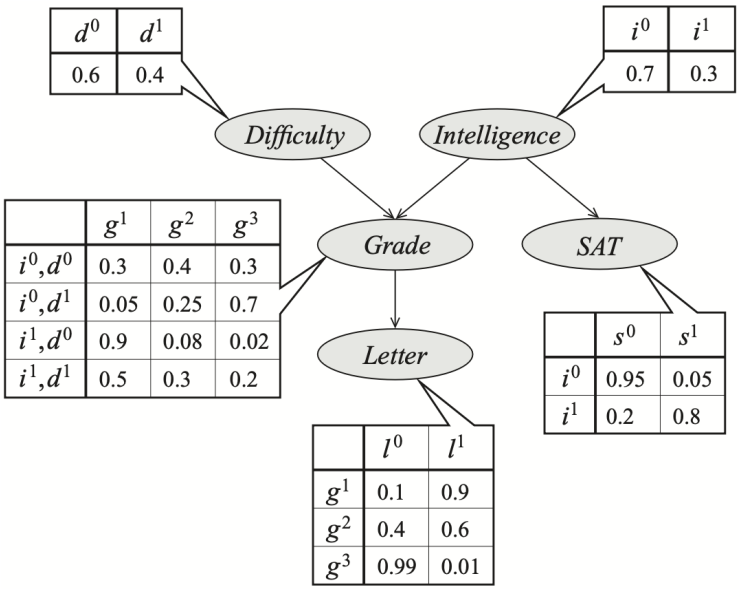
$$P(X) := P(L,G,S,D,I)$$
$$= P(L|G)P(G|D,I)P(S|I)P(D)P(I)$$

$$Q_{I=i^1,G=g^2}(Y) = P(L|G=g^2)P(S|I=i^1)P(D)$$

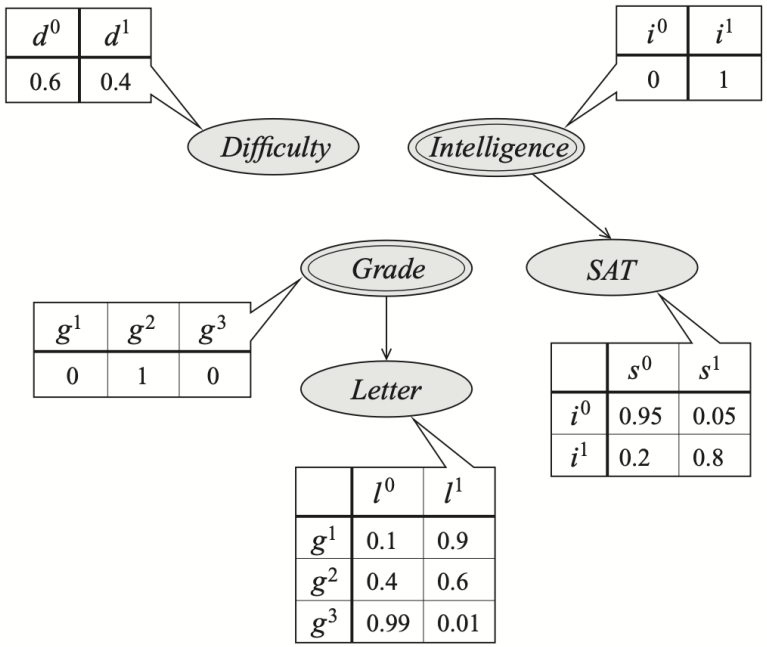
If $X := (L,G,S,D,I)$ and $P(X)$ denotes a Bayesian network then as we saw earlier, sampling from conditionals of P directly is awkward and inefficient. Can we define a convenient Q and use important sampling?

Bayesian Network Inference with Importance Sampling

Original network $\mathscr{B}$



"Mutilated" network $\mathscr{B}_{I=i^1,G=g^2}$



$$P(X) := P(L,G,S,D,I)$$
$$= P(L|G)P(G|D,I)P(S|I)P(D)P(I)$$

$$Q_{I=i^1,G=g^2}(Y) = P(L|G=g^2)P(S|I=i^1)P(D)$$

Note that $P(I=i^1, G=g^2) = \mathbb{E}_P[\mathbf{1}\{I=i^1, G=g^2\}]$

$$= \mathbb{E}_Q[\frac{1}{M}\sum_{k=1}^{M}\mathbf{1}\{I_k=i^1, G_k=g^2\}\frac{P(Y_k)}{Q(Y_k)}]$$

If $\{I=i^1, G=g^2\}$ is a rare event, this could require substantial fewer samples!

Bayesian Network Inference with Importance Sampling

How do we compute $\frac{P(Y)}{Q(Y)}$? Let $Y = (L = l, g = g^2, S = s, D = d, I = i^1)$

Then $$\frac{P(Y)}{Q(Y)} = \frac{P(L = l, g = g^2, S = s, D = d, I = i^1)}{Q_{I=i^1, G=g^2}(L = l, g = g^2, S = s, D = d, I = i^1)}$$

$$P(X) := P(L, G, S, D, I)$$
$$= P(L|G)P(G|D, I)P(S|I)P(D)P(I)$$

$$Q_{I=i^1, G=g^2}(Y) = P(L|G = g^2)P(S|I = i^1)P(D)$$

Note that $P(I = i^1, G = g^2) = \mathbb{E}_P[\mathbf{1}\{I = i^1, G = g^2\}]$

$$= \mathbb{E}_Q[\frac{1}{M}\sum_{k=1}^{M} \mathbf{1}\{I_k = i^1, G_k = g^2\}\frac{P(Y_k)}{Q(Y_k)}]$$

If $\{I = i^1, G = g^2\}$ is a rare event, this could require substantial fewer samples!

Bayesian Network Inference with Importance Sampling

How do we compute $\frac{P(Y)}{Q(Y)}$? Let $Y = (L = l, g = g^2, S = s, D = d, I = i^1)$

$$\text{Then } \frac{P(Y)}{Q(Y)} = \frac{P(L = l, g = g^2, S = s, D = d, I = i^1)}{Q_{I=i^1, G=g^2}(L = l, g = g^2, S = s, D = d, I = i^1)}$$

$$= \frac{P(G = g^2 \mid D = d, I = i^1)P(I = i^1)}{1}$$

Note: all but the "mutilated" terms cancel.

$$P(X) := P(L, G, S, D, I)$$
$$= P(L \mid G)P(G \mid D, I)P(S \mid I)P(D)P(I)$$

$$Q_{I=i^1, G=g^2}(Y) = P(L \mid G = g^2)P(S \mid I = i^1)P(D)$$

Note that $P(I = i^1, G = g^2) = \mathbb{E}_P[\mathbf{1}\{I = i^1, G = g^2\}]$

$$= \mathbb{E}_Q[\frac{1}{M}\sum_{k=1}^{M} \mathbf{1}\{I_k = i^1, G_k = g^2\}\frac{P(Y_k)}{Q(Y_k)}]$$

If $\{I = i^1, G = g^2\}$ is a rare event, this could require substantial fewer samples!

Bayesian Network Inference with Importance Sampling

Algorithm 12.2 Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $\boldsymbol{Z} = \boldsymbol{z}$ // Event in the network
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $\boldsymbol{u}_i \leftarrow \boldsymbol{x}\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin \boldsymbol{Z}$ **then**
6 Sample x_i from $P(X_i \mid \boldsymbol{u}_i)$
7 **else**
8 $x_i \leftarrow \boldsymbol{z}\langle X_i \rangle$ // Assignment to X_i in $\boldsymbol{z}$
9 $w \leftarrow w \cdot P(x_i \mid \boldsymbol{u}_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

Theorem: If the above algorithm is run on the mutilated network wrt to a set of set variables **z** and $Q(Y)$ represents its probability, then $w = \dfrac{P(Y)}{Q(Y)}$.

Bayesian Network Inference with Importance Sampling

Algorithm to estimate $P(\chi(X) = z)$:

1. Execute algorithm on previous slide M times with event $\{\chi(X) = z\}$
2. Get $(Y_1, w_1), \ldots, (Y_M, w_M)$ back
3. Set $\widehat{P}(\chi(X) = z) = \frac{1}{M} \sum_{k=1}^{M} \mathbf{1}\{\chi(Y_k) = z\} w_k$

Takeaway: $\widehat{P}(\chi(X) = z)$ is an unbiased estimator of $P(\chi(X) = z)$.
By the properties of importance sampling estimators of the previous slides, if Q is close to P, the critical size of M may not even depend on $P(\chi(X) = z)$, even if it is very rare. But if Q is very far, this could be worse than rejection sampling.

Theorem: If the above algorithm is run on the mutilated network wrt to a set of set variables **z** and $Q(Y)$ represents its probability, then $w = \frac{P(Y)}{Q(Y)}$.

Bayesian Network Inference with Importance Sampling

How do we use estimators like $\widehat{P}(\chi(X) = z) = \frac{1}{M}\sum_{k=1}^{M} \mathbf{1}\{\chi(Y_k) = z\} w_k$ to compute conditional queries like $P(I = i^1 \mid G = g^2, S = s^0)$?

Ratio method

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Use M samples to compute

$$\widehat{P}(I = i^1, G = g^2, S = s^0)$$

Use M' samples to compute

$$\widehat{P}(G = g^2, S = s^0)$$

and take the ratio!

Numerator and denominator unbiased.

Self-normalized method

Collect data $(Y_1, w_1), \ldots, (Y_M, w_M)$ using event $\{G = g^2, S = s^0\}$

Output

$$\frac{\sum_{k=1}^{M} \mathbf{1}\{I_k = i^1, G_k = g^2, S_k = s^0\} w_k}{\sum_{k=1}^{M} w_k}$$

Numerator and denominator unbiased:

$$\mathbb{E}_Q[w_k] = \mathbb{E}_Q[w_k \mathbf{1}\{G_k = g^2, S_k = s^0\}]$$
$$= P(G = g^2, S = s^0)$$

MCMC

Inference in Bayesian Networks

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Approaches:

- **Exact inference by variable elimination**
  - Straightforward, but can be computationally prohibitive
- **Variational inference**
  - Approximate, biased
- **Sampling strategies**
  - Rejection sampling 
  - Importance weighted sampling 
  - MCMC

Markov Chain Monte Carlo (MCMC)

Goal: compute queries like $P(I = i^1 \mid G = g^2, S = s^0)$ or in words "the probability of high intelligence given medium grade and low SAT?"

$$P(I = i^1 \mid G = g^2, S = s^0) = \frac{P(I = i^1, G = g^2, S = s^0)}{P(G = g^2, S = s^0)}$$

Sampling from the "mutilated" network will give you sample such that $\{G = g^2, s = s^0\}$. The problem is that this sample is drawn from $Q_{I=i^1, G=g^2}(Y)$ and not $P(I = i^1 \mid G = g^2, S = s^0)$. Importance sampling tries to "correct" the misalignment by weighting the sample appropriately.

MCMC starts with the candidate distribution $\pi^{(0)} := Q$ and then *evolves* the distribution $\pi^{(t)}$ slowly over time until it converges $\pi^{(t)} \to \pi_* := P$. Sampling from this distribution results in a sample from P

Gibbs sampling

Algorithm 12.4 Generating a Gibbs chain trajectory

Procedure Gibbs-Sample (
  $\boldsymbol{X}$ // Set of variables to be sampled
  Φ // Set of factors defining P_Φ
  $P^{(0)}(\boldsymbol{X})$, // Initial state distribution
  T // Number of time steps
)

1 Sample $\boldsymbol{x}^{(0)}$ from $P^{(0)}(\boldsymbol{X})$
2 **for** $t = 1, \ldots, T$
3 $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)}$
4 **for each** $X_i \in \boldsymbol{X}$
5 Sample $x_i^{(t)}$ from $P_\Phi(X_i \mid \boldsymbol{x}_{-i})$
6 // Change X_i in $\boldsymbol{x}^{(t)}$
7 **return** $\boldsymbol{x}^{(0)}, \ldots, \boldsymbol{x}^{(T)}$

$$P(X_i \mid x_{-i}) = P(X_i \mid \text{MarkovBlanket}(i))$$
$$= P(X_i \mid \text{parents}(i)) \prod_{j \in \text{child}(i)} P(x_j \mid \text{parents}(j))$$

Markov blanket of *Cloudy* is
Sprinkler and *Rain*
Markov blanket of *Rain* is
Cloudy, *Sprinkler*, and *WetGrass*

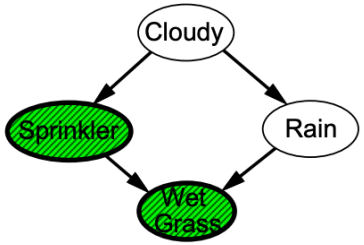


Figure credit: Pedro Domingos

Gibbs sampling

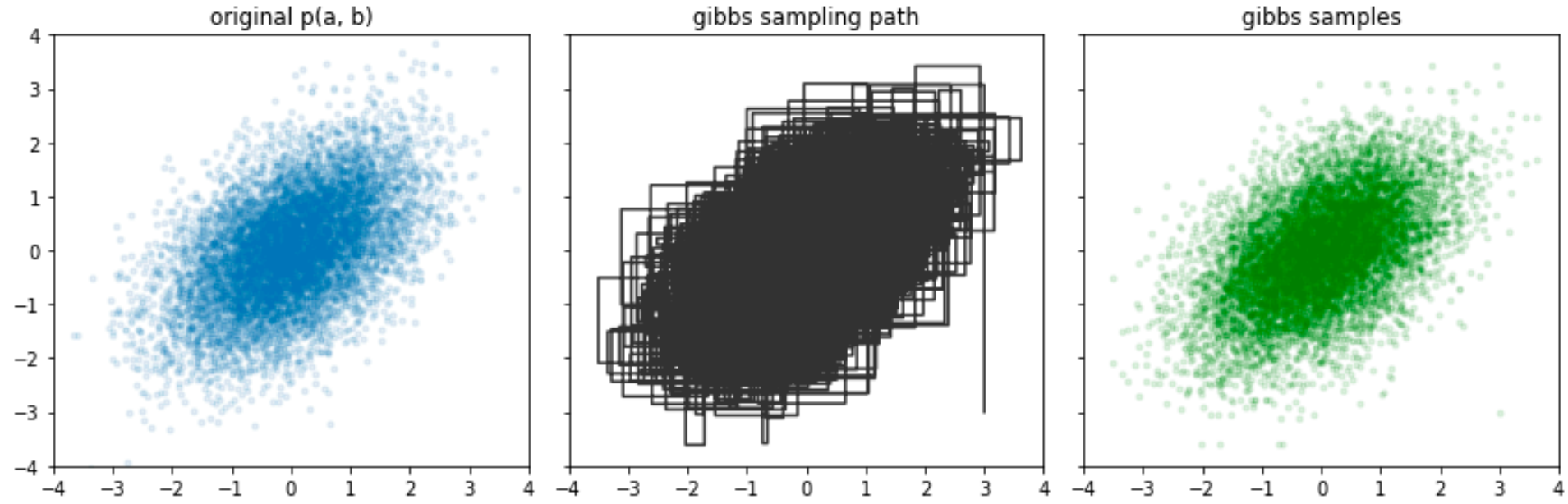


Figure credit: https://jessicastringham.net/2018/05/09/gibbs-sampling/

Markov Chain Monte Carlo (MCMC)

Gibb's sampling is just one example of a MCMC algorithm.

In general, the approach to sampling from distribution π_*:

- Construct a Markov chain transition kernel $T : \mathcal{X} \to \mathcal{X}$ whose stationary distribution is equal to π_*
- Start with a realization $x^{(0)} \sim \pi^{(0)}$ drawn from an arbitrary starting distribution $\pi^{(0)}$
- Run Markov chain to evolve $x^{(t)} \mapsto x^{(t+1)}$ (equiv. $\pi^{(t)} \to \pi^{(t+1)}$) and return sample once convergence $\pi^{(t)} \approx \pi_*$

Is it guaranteed to converge? How fast does it converge?

Markov Chain Monte Carlo (MCMC)

Consider a **Markov chain** defined by: $T_{xy} = \mathbb{P}(x_{t+1} = y \,|\, x_t = x)$

We say a Markov chain is **reversible** if there exists a distribution π that satisfies the **detailed balance equation:**

$$\pi_x T_{xy} = \pi_y T_{yx} \qquad \text{for all } x, y$$

If such a π exists, then it is a **stationary distribution** satisfying

$$\pi^\top T = \pi^\top.$$

Markov Chain Monte Carlo (MCMC)

Consider a **Markov chain** defined by: $T_{xy} = \mathbb{P}(x_{t+1} = y \,|\, x_t = x)$

We say a Markov chain is **reversible** if there exists a distribution π that satisfies the **detailed balance equation:**

$$\pi_x T_{xy} = \pi_y T_{yx} \qquad \text{for all } x, y$$

If such a π exists, then it is a **stationary distribution** satisfying

$$\pi^\top T = \pi^\top.$$

Proof: $[\pi^\top T]_y = \sum_x \pi_x T_{xy} = \sum_x \pi_y T_{yx} = \pi_y$

Idea: construct a reversible Markov chain T with respect to target distribution π, and run the chain until it mixes to the stationary distributor π

Markov Chain Monte Carlo (MCMC)

Consider a **Markov chain** defined by: $T_{xy} = \mathbb{P}(x_{t+1} = y \,|\, x_t = x)$

We say a Markov chain is **reversible** if there exists a distribution π that satisfies the **detailed balance equation:**

$$\pi_x T_{xy} = \pi_y T_{yx} \qquad \text{for all } x, y$$

If such a π exists, then it is a **stationary distribution** satisfying

$$\pi^\top T = \pi^\top.$$

A chain is **regular** if there exists a $k \in \mathbb{N}$ such that $[T^k]_{x,y>0}$.

Theorem: If a chain is regular then its stationary distribution is unique.

If $T_{xx} > 0$ and the chain is irreducible, then the chain is regular.

Markov Chain Monte Carlo (MCMC)

Revisit **Gibb's sampling** as MCMC:

$$T_{xy} = \frac{1}{d}\sum_{i=1}^{d} \mathbb{P}(y_i \,|\, y_{-i} = x_{-i}) \mathbf{1}\{y_{-i} = x_{-i}\}$$

$$\begin{aligned}
\pi_x T_{xy} &= \frac{1}{d}\sum_{i=1}^{d} \mathbb{P}(y_i \,|\, y_{-i} = x_{-i}) \mathbf{1}\{y_{-i} = x_{-i}\} \pi_x \\
&= \frac{1}{d}\sum_{i=1}^{d} \pi_y \mathbf{1}\{y_{-i} = x_{-i}\} \mathbb{P}(x_i \,|\, x_{-i} = y_{-i}) \\
&= \pi_y T_{yx}
\end{aligned}$$

Gibb's sampling is **reversible** but not necessarily **regular** (no guarantee of converging to π)

Metropolis-Hastings Algorithm

The problem with Gibbs is that the chain wasn't guaranteed to traverse the entire space. Can we construct a chain that is **reversible** and **regular**?

Idea: construct a random walk chain that is guaranteed to traverse everywhere, then bias it towards the target distribution π

Let K_{xy} be a **regular** Markov chain whose support includes support of π

Define $T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$

Metropolis-Hastings Algorithm

The problem with Gibbs is that the chain wasn't guaranteed to traverse the entire space. Can we construct a chain that is **reversible** and **regular**?

Idea: construct a random walk chain that is guaranteed to traverse everywhere, then bias it towards the target distribution π

Let K_{xy} be a **regular** Markov chain whose support includes support of π

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

Implementation: Given x_t draw y with probability $K_{x_t y}$. Set $x_{t+1} = y$ with probability $\min\{1, \frac{\pi_y K_{y x_t}}{\pi_{x_t} K_{x_t y}}\}$, otherwise set $x_{t+1} = x_t$.

Metropolis-Hastings Algorithm

The problem with Gibbs is that the chain wasn't guaranteed to traverse the entire space. Can we construct a chain that is **reversible** and **regular**?

Idea: construct a random walk chain that is guaranteed to traverse everywhere, then bias it towards the target distribution π

Let K_{xy} be a **regular** Markov chain whose support includes support of π

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

Note:

- The MH transition kernel T_{xy} satisfies $\pi_x T_{xy} = \pi_y T_{yx}$ by construction, thus **reversible**.
- Since K_{xy} is **regular**, so is T_{xy}. Thus, **this chain converges** to π!

Metropolis-Hastings Algorithm

The problem with Gibbs is that the chain wasn't guaranteed to traverse the entire space. Can we construct a chain that is **reversible** and **regular**?

Idea: construct a random walk chain that is guaranteed to traverse everywhere, then bias it towards the target distribution π

Let K_{xy} be a **regular** Markov chain whose support includes support of π

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

If $\mathcal{X} \subset \mathbb{R}^d$ then natural to take $K_{xy} = \frac{1}{\sqrt{2\pi\sigma^d}} \exp(-\|x - y\|^2 / 2\sigma^2)$

Warning: if σ too small, exploration is slow and takes a very long time to traverse space. If σ is too large, you'll leave the support of π and samples will keep getting rejected.

Metropolis-Hastings Algorithm

The problem with Gibbs is that the chain wasn't guaranteed to traverse the entire space. Can we construct a chain that is **reversible** and **regular**?

Idea: construct a random walk chain that is guaranteed to traverse everywhere, then bias it towards the target distribution π

Let K_{xy} be a **regular** Markov chain whose support includes support of π

Define $T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$

You only need to evaluate $\frac{\pi_y}{\pi_x}$ so you don't need to know normalization constant of Markov Networks $\pi(x) = \frac{1}{Z} \prod_{(i,j) \in E} \phi_{i,j}(x_i, x_j)$

MCMC in continuous domains

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

If $\mathcal{X} \subset \mathbb{R}^d$ then natural to take $K_{xy} = \frac{1}{\sqrt{2\pi\sigma^d}} \exp(-\|x - y\|^2 / 2\sigma^2)$

Warning: if σ too small, exploration is slow and takes a very long time to traverse space.
If σ is too large, you'll leave the support of π and samples will keep getting rejected.

MH Algorithm with Gaussian candidate distribution:

Init: $\theta_0 \in \mathbb{R}^d$

For $t = 1, 2, \ldots$

Draw $\tilde{\theta}_{t+1} \sim \mathcal{N}(\theta_t, \sigma^2 I)$

Set $\theta_{t+1} = \tilde{\theta}_{t+1}$ with probability $\min\{1, \frac{\pi(\tilde{\theta}_{t+1})}{\pi(\theta_t)}\}$, otherwise $\theta_{t+1} = \theta_t$

MCMC in continuous domains

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

If $\mathcal{X} \subset \mathbb{R}^d$ then natural to take $K_{xy} = \frac{1}{\sqrt{2\pi\sigma^d}} \exp(-\|x - y\|^2 / 2\sigma^2)$

Warning: if σ too small, exploration is slow and takes a very long time to traverse space.
If σ is too large, you'll leave the support of π and samples will keep getting rejected.

MH Algorithm with Gaussian candidate distribution:

Init: $\theta_0 \in \mathbb{R}^d$

For $t = 1, 2, \ldots$

Draw $\tilde{\theta}_{t+1} \sim \mathcal{N}(\theta_t, \sigma^2 I)$

Set $\theta_{t+1} = \tilde{\theta}_{t+1}$ with probability $\min\{1, \frac{\pi(\tilde{\theta}_{t+1})}{\pi(\theta_t)}\}$, otherwise $\theta_{t+1} = \theta_t$

This is hill climbing in a random direction.
Very similar to derivative free optimization.
EXTREMELY inefficient in large dimensions

MCMC in continuous domains

$$\text{Define } T_{xy} = \begin{cases} K_{xy} \min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} & \text{if } x \neq y \\ K_{xx} + \sum_{y \neq x} K_{xy}(1 - K_{xy}) & \text{otherwise.} \end{cases}$$

If $\mathcal{X} \subset \mathbb{R}^d$ then natural to take $K_{xy} = \frac{1}{\sqrt{2\pi\sigma^d}} \exp(-\|x-y\|^2/2\sigma^2)$

Langevin Monte Carlo (LMC) takes $K_{xy} = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp(-\|x + \frac{\sigma^2}{2} \nabla_z \log(\pi(z))|_{z=x} - y\|^2/2\sigma^2)$

As $\sigma \to 0$ the acceptance probability $\min\{1, \frac{\pi_y K_{yx}}{\pi_x K_{xy}}\} \to 1$ and the chain just becomes

$$x_{t+1} = x_t + \frac{\sigma^2}{2} \nabla_z \log(\pi(z))|_{z=x_t} + \sqrt{\sigma^2}\eta_t \qquad \eta_t \sim \mathcal{N}(0, I)$$

which is what people use in practice. Converges much faster than traditional MH.

Hit and run sampling

Frequently we have a set $K \subset \mathbb{R}^d$ and we would like to sample from a uniform measure over this set. Useful in its own right, but can also be used for a candidate distribution to sample from.

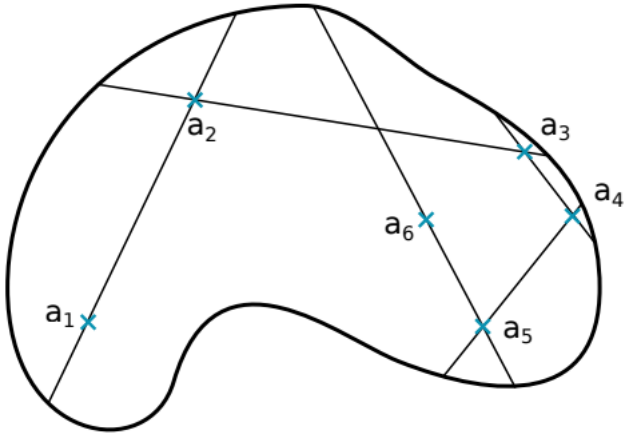
Hit and run sampling:

Initialize: $a_1 \in K$

For $t = 1,2,\ldots$

Pick a uniformly distributed random line ℓ through a_t

Set a_{t+1} to a uniform random point along $\ell \cap K$



Claim: Under benign smoothness conditions on K, Hit and run converges to a uniform stationary distribution.

Hit and run sampling

Frequently we have a connected set $K \subset \mathbb{R}^d$ and we would like to sample from a uniform measure over this set. Useful in its own right, but can also be used for a candidate distribution to sample from.

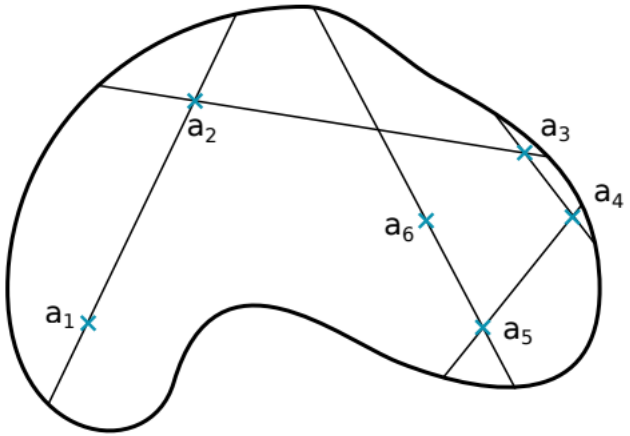
Hit and run sampling:

Initialize: $a_1 \in K$

For $t = 1,2,\ldots$

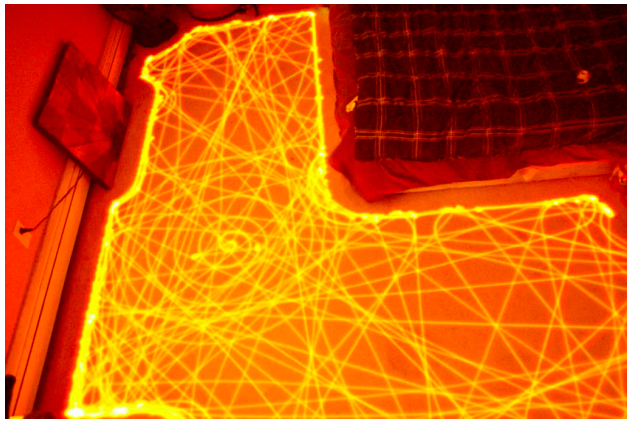
Pick a uniformly distributed random line ℓ through a_t

Set a_{t+1} to a uniform random point along $\ell \cap K$



Claim: Under benign smoothness conditions on K, Hit and run converges to a uniform stationary distribution.

Proof by vacuum:



MCMC Convergence

Okay, so we now have a **reversible** and **regular** Markov chain T_{xy} that is guaranteed to converge to our target distribution π as its **stationary distribution**. How long do we have to wait for convergence?

The ϵ-**mixing time** of chain T is the smallest time such that for all $t > T_{\text{min}}(\epsilon)$

$$\|\pi^{(0)}T^t - \pi\|_{TV} \leq \epsilon.$$

Theorem: $T_{\text{mix}}(\epsilon) = O(\frac{1}{1-\lambda_2})$ where $\lambda_2 < 1$ is the second largest eigenvalue of T.

Proof sketch: consider special case where T is diagonalizable so that $T = U\text{diag}(\lambda)U^{-1}$

Fact: The largest eigenvalue of T is $\lambda_1 = 1$ with $\pi^\top T = \pi^\top$ and $T\mathbf{1} = \mathbf{1}$.

$$\|\pi_0^\top T^t - \pi\|_{TV} = \|(\pi_0^\top - \pi^\top)T^t\|_{TV} \leq \sqrt{n}\|(\pi_0^\top - \pi^\top)T^t\|_2 = \sqrt{n}\|(\pi_0^\top - \pi^\top)\sum_{i=2}^{n} u_i v_i^\top \lambda_i^t\|_2 \leq \sqrt{2n}\lambda_2^t$$

$$T_{\text{mix}}(\epsilon) \leq \frac{\log(\sqrt{2n}/\epsilon)}{\log(1/\lambda_2)} \leq \frac{\log(\sqrt{2n}/\epsilon)}{1-\lambda_2}$$

Deterministic methods

Core sets

We wish to estimate $\mathbb{E}_P[\mathbf{1}\{X \in A\}] = \sum_i P(x_i)\mathbf{1}\{x_i \in A\}$.

If mass is concentrated on just a small number of heavy hitters s.t. $\mathbb{P}(X \in \{x_i\}_{i=1}^k) \geq .99$

Then we just need to cover these elements to answer many queries.

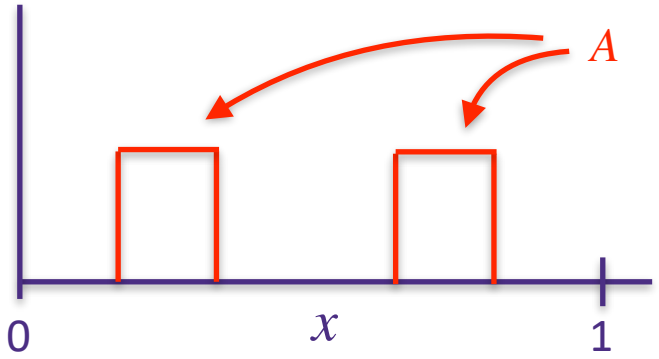
Identifying these heavy hitters is an inference task in itself, but may dwarf the computation of sampling. Very heuristic-y but could be effective

Quasi Monte Carlo

At the end of the day we wish to estimate $\mathbb{E}_P[\mathbf{1}\{X \in A\}] = \int_x P(x)\mathbf{1}\{x \in A\}dx.$

Suppose we knew P. Why are we sampling instead of approximating the integral?

Example: Let P = uniform($[0,1]$) and $A \subset [0,1]$, set $\mu = \mathbb{E}_P[\mathbf{1}\{X \in A\}]$



Draw $x_1, \ldots, x_n \sim P$, set $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \in A\}$

$$\mathbb{E}[\,|\,\hat{\mu} - \mu\,|\,] \approx \frac{1}{\sqrt{n}}$$

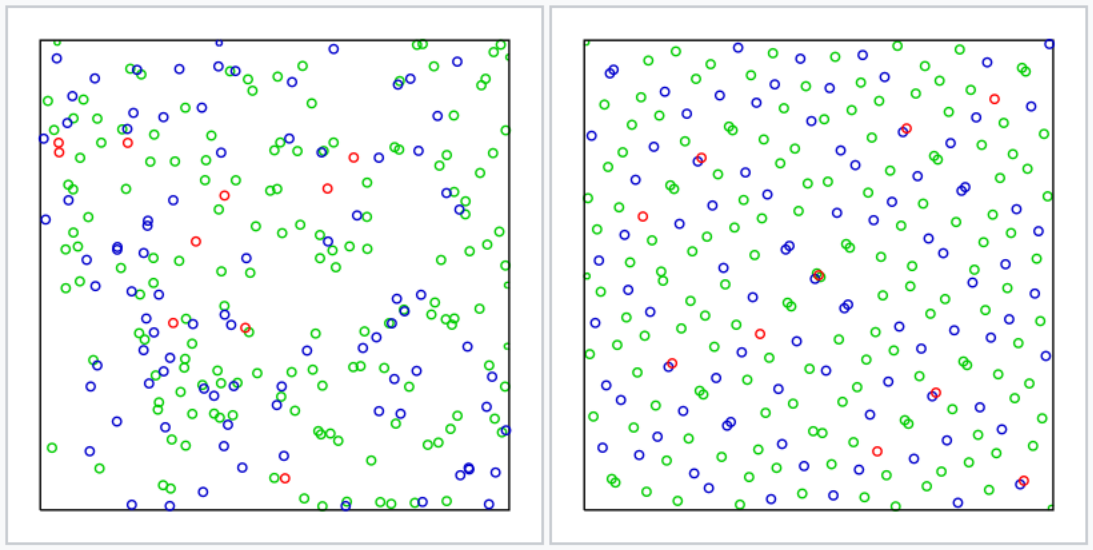
Set $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\frac{i-1/2}{n} \in A\}$

$$\mathbb{E}[\,|\,\tilde{\mu} - \mu\,|\,] \leq \frac{1}{n}$$

Quasi Monte Carlo

At the end of the day we wish to estimate $\mathbb{E}_P[\mathbf{1}\{X \in A\}] = \int_x P(x)\mathbf{1}\{x \in A\}dx.$

Suppose we knew P. Why are we sampling instead of approximating the integral?



Random Uniform

Sobol Sequence

$$\mathbb{E}[\,|\,\widehat{\mu} - \mu\,|\,] \approx \frac{1}{\sqrt{n}}$$

$$\mathbb{E}[\,|\,\widetilde{\mu} - \mu\,|\,] \leq \frac{\log(n)^{d-1}}{n}$$