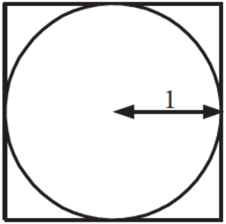
The Monte Carlo Method

Example: estimate the value of π.



- Choose X and Y independently and uniformly at random in $[0, 1]$.
- Let

$$Z = \begin{cases} 1 & \text{if } \sqrt{X^2 + Y^2} \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

- $\Pr(Z = 1) = \frac{\pi}{4}$.
- $4\mathbf{E}[Z] = \pi$.

- Let $Z_1, \dots, Z_m$ be the values of m independent experiments. $W = \sum_{i=1}^{m} Z_i$.
- $$\mathbf{E}[W] = \mathbf{E}\left[\sum_{i=1}^{m} Z_i\right] = \sum_{i=1}^{m} \mathbf{E}[Z_i] = \frac{m\pi}{4},$$
- $W' = \frac{4}{m} W$ is a natural estimate for π.
- $$\begin{aligned}\Pr(|W' - \pi| \geq \epsilon\pi) &= \Pr\left(|W - \frac{m\pi}{4}| \geq \frac{\epsilon m\pi}{4}\right) \\ &= \Pr\left(|W - \mathbf{E}[W]| \geq \epsilon \mathbf{E}[W]\right) \\ &\leq 2e^{-\frac{1}{12}m\pi\epsilon^2}. \text{(Chernoff bound, Cor. 4.6)}\end{aligned}$$

(ϵ, δ)-Approximation

Definition

A randomized algorithm gives an (ϵ, δ)-*approximation* for the value V if the output X of the algorithm satisfies

$$\Pr(|X - V| \leq \epsilon V) \geq 1 - \delta.$$

The method for approximating π gives an (ϵ, δ)-approximation as long as $\epsilon < 1$ and m is large enough to make

$$2e^{-m\pi\epsilon^2/12} \leq \delta$$

so we need

$$m \geq \frac{12 \ln(2/\delta)}{\pi\epsilon^2}$$

Theorem

Let $X_1, \ldots, X_m$ *be independent and identically distributed indicator random variables, with* $\mu = E[X_i]$*. If* $m \geq \frac{3 \ln \frac{2}{\delta}}{\epsilon^2 \mu}$*, then*

$$\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| \geq \epsilon\mu\right) \leq \delta.$$

That is, m *samples provide an* (ϵ, δ)*-approximation for* μ*.*

Approximate Counting

Example counting problems:

1. How many spanning trees in a graph?
2. How many perfect matchings in a graph?

DNF Counting

DNF = Disjunctive Normal Form.
Problem: How many satisfying assignments to a DNF formula?
A DNF formula is a disjunction of clauses.
Each clause is a conjunction of literals.

$$(\overline{x_1} \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_3 \wedge \overline{x_4})$$

Compare to CNF.

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_3}) \wedge \cdots$$

m clauses, n variables
Let's first convince ourselves that obvious approaches don't work!

DNF counting is hard

Question: Why?
We can reduce CNF satisfiability to DNF counting.
The negation of a CNF formula is in DNF.

1. CNF formula f
2. get the DNF formula $(\bar{f})$
3. count satisfying assignments to $\bar{f}$
4. This number is 2^n if and only if f is unsatisfiable.

DNF counting is #P complete

#P is the counting analog of NP.
Any problem in #P can be reduced (in polynomial time) to the DNF counting problem.
Example #P complete problems:

1. How many Hamilton circuits does a graph have?
2. How many satisfying assignments does a CNF formula have?
3. How many perfect matchings in a graph?

What can we do about a hard problem?

(ϵ, δ) FPRAS for DNF counting

FPRAS = "Fully Polynomial Randomized Approximation Scheme"
Notation:
U: set of all possible assignments to variables
$|U| = 2^n$.
$H \subset U$: set of satisfying assignments
Want to estimate $Y = |H|$
Give $\epsilon > 0, \delta > 0$, find estimate X such that

1. $\Pr[|X - Y| > \epsilon Y] < \delta$
2. Algorithm should be polynomial in $1/\epsilon$, $1/\delta$, n and m.

Monte Carlo method

Here's the obvious scheme (Algorithm 1, page 256 in book).

1. Repeat N times:
    1.1. Sample x randomly from U, that is, generate one of the 2^n possible assignments uniformly at random.
    1.2. Count a success if $x \in H$ (formula satisfied by x)
2. Return "fraction of successes" $\times |U|$.

Question: How large should N be?

We have to evaluate the probability of our estimate being good.

Let $\rho = \frac{|H|}{|U|}$.
Let the indicator random variable $Z_i = 1$ if i-th trial was successful. Then

$$Z_i = \begin{cases} 1 & \text{with probability} \quad \rho \\ 0 & \text{with probability} \quad 1 - \rho \end{cases}$$

$Z = \sum_{i=1}^{N} Z_i$ is a binomial random variable whose expected value is
$E[Z] = N\rho$
$X = \frac{Z}{N}|U|$ is our estimate of $|H|$

Probability that our algorithm succeeds

Recall: X denotes our estimate of $|H|$.

$$\begin{aligned} & \Pr[(1-\epsilon)|H| < X < (1+\epsilon)|H|] \\ = \; & \Pr[(1-\epsilon)|H| < Z|U|/N < (1+\epsilon)|H|] \\ = \; & \Pr[(1-\epsilon)N\rho < Z < (1+\epsilon)N\rho] \\ > \; & 1 - e^{-N\rho\epsilon^2/3} - e^{-N\rho\epsilon^2/2} \\ > \; & 1 - 2e^{-N\rho\epsilon^2/3} \end{aligned}$$

where we have used Chernoff bounds.
For an (ϵ, δ) approximation, this has to be greater than $1-\delta$,

$$2e^{-N\rho\epsilon^2/3} < \delta$$
$$N > \frac{3}{\rho\epsilon^2} \log \frac{2}{\delta}$$

Theorem

Let $\rho = |H|/|U|$. Then the Monte Carlo method is an (ϵ, δ) approximation scheme for estimating $|H|$ provided that $N > \frac{3}{\rho\epsilon^2} \log \frac{2}{\delta}$.

What's wrong?

How large could $\dfrac{1}{\rho}$ be?

ρ is the fraction of satisfying assignments.

1. The number of possible assignments is 2^n.
2. Maybe there are only a polynomial (in n) number of satisfying assignments.
3. So, $\dfrac{1}{\rho}$ could be exponential in n.

Question: An example where formula has only a few assignments?

The trick: Skewed sampling

Increase the hit rate (ρ)!

Sample from a different universe, ρ is higher, and all elements of H still represented.

What's the new universe?

Notation: H_i set of assignments that satisfy clause i.

$H = H_1 \cup H_2 \cup \ldots H_m$

Define a new universe

$$U = H_1 \biguplus H_2 \biguplus \ldots \biguplus H_m$$

$\biguplus$ means *multiset union*.

Example - Partition by clauses

$$(\overline{x_1} \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_3 \wedge \overline{x_4})$$

x_1	x_2	x_3	x_4	Clause
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
0	1	1	0	2
0	1	1	1	2
1	1	1	0	2
1	1	1	1	2
1	1	0	1	3
0	0	1	0	4
0	1	1	0	4
1	0	1	0	4
1	1	1	0	4

More about the universe U

1. U contains only the satisfying assignments.
2. U is a multiset (contains the same element many times).
3. Element of U is (v, i) where v is an assignment, i is the satisfied clause.
   $U = \{(v, i) | v \in H_i\}$
4. Each satisfying assignment v appears in as many clauses as it satisfies.

One way of looking at U

Partition by clauses.
m partitions, partition i contains H_i.

Another way of looking at U

Partition by assignments (one region for each assignment v).
Each partition corresponds to an assignment.
Can we count the different (distinct) assignments?

Example - Partition by assignments

$$(\overline{x_1} \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_3 \wedge \overline{x_4})$$

x_1	x_2	x_3	x_4	Clause
0	0	1	0	4
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	0	2
0	1	1	0	4
0	1	1	1	1
0	1	1	1	2
1	0	1	0	4
1	1	0	1	3
1	1	1	0	2
1	1	1	0	4
1	1	1	1	2

Canonical element

Crucial idea: For each assignment group, find a canonical element in U.
An element (v, i) is *canonical* if $f((v, i)) = 1$

$$f((v,i)) = \begin{cases} 1 & \text{if } i = \min\{j : v \in H_j\} \\ 0 & \text{otherwise} \end{cases}$$

For every assignment group, exactly one canonical element.
So, count the number of canonical elements!
Note: could use any other definition as long as exactly one canonical element per assignment

Count canonical elements

Reiterating:

1. Number of satisfying assignments $=$
   Number of canonical elements.
2. Count number of canonical elements.
3. Back to old random sampling method for counting!

What is ρ?

Lemma

$\rho \geq \frac{1}{m}$, *(pretty large).*

Proof.

$|H| = |\cup_{i=1}^{m} H_i|$, since H is a normal union.
So $|H_i| \leq |H|$
Recall $U = H_1 \uplus H_2 \uplus \ldots \uplus H_m$
$|U| = \sum_{i=1}^{m} |H_i|$, since U is a multiset union.
$|U| \leq m|H|$
$\rho = \frac{|H|}{|U|} \geq \frac{1}{m}$ □

How to generate a random element in U?

Look at the partition of U by clauses.
Algorithm Select:

1. Pick a random clause weighted according to the area it occupies.

$$\Pr[i] = \frac{|H_i|}{|U|} = \frac{|H_i|}{\sum_1^m |H_j|}$$

   $|H_i| = 2^{(n-k_i)}$ where k_i is the number of literals in clause i.

2. Choose a random satisfying assignment in H_i.
   - Fix the variables required by clause i.
   - Assign random values to the rest to get v

   (v, i) is the random element.

Running time: $O(n)$.

How to test if canonical assignment?

Or how to evaluate $f((v, i))$?

Algorithm Test:

1. Test every clause to see if v satisfies it.
   $cov(v) = \{(v, j) | v \in H_j\}$
2. If (v, i) the smallest in $cov(v)$, then $f(v, i) = 1$, else 0.

Running time: $O(nm)$.

Back to random sampling

Algorithm Coverage:

1. $s \leftarrow 0$ (number of successes)
2. Repeat N times:
   - Select (v, i) using **Select**.
   - if $f(v, i) = 1$ (check using **Test**) then success, increment s.
3. Return $s|U|/N$.

Number of samples needed is (from Theorem 3):

$$N = \frac{3}{\epsilon^2 \rho} \ln \frac{2}{\delta} \leq \frac{3m}{\epsilon^2} \ln \frac{2}{\delta}$$

Sampling, testing: polynomial in n and m
We have an FPRAS

Theorem

The Coverage algorithm yields an (ϵ, δ) approximation to $|H|$ provided that the number of samples $N \geq \frac{3m}{\epsilon^2} \log \frac{2}{\delta}$.

Counting Independent Sets

Input: a graph $G = (V, E)$. $|V| = n$, $|E| = m$.
Let $e_1, \ldots, e_m$ be an arbitrary ordering of the edges.

$$G_i = (V, E_i), \quad \text{where} \quad E_i = \{e_1, \ldots, e_i\}$$

$G = G_m$, $G_0 = (V, \emptyset)$ and G_{i-1} is obtained from G_i be removing a single edge.
$\Omega(G_i) =$ the set of independent sets in G_i.

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \times \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \times \frac{|\Omega(G_{m-2})|}{|\Omega(G_{m-3})|} \times \cdots \times \frac{|\Omega(G_1)|}{|\Omega(G_0)|} \times |\Omega(G_0)|.$$

$$r_i = \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}, \qquad i = 1, \ldots, m.$$

Algorithm

Estimating r_i
Input: Graphs $G_{i-1} = (V, E_{i-1})$ and $G_i = (V, E_i)$.
Output: $\tilde{r}_i =$ an approximation of r_i.

1. $X \leftarrow 0$.
2. Repeat for $M = \lceil 1296 m^2 \epsilon^{-2} \ln \frac{2m}{\delta} \rceil$ independent trials:
   1. Generate an uniform sample from $\Omega(G_{i-1})$;
   2. If the sample is an independent set in G_i, let $X \leftarrow X + 1$.
3. Return $\tilde{r}_i \leftarrow \frac{X}{M}$.

Lemma

$r_i \geq 1/2$.

Proof.

$$\Omega(G_i) \subseteq \Omega(G_{i-1}).$$

Suppose that G_{i-1} and G_i differ in the edge $\{u, v\}$.
An independent set in $\Omega(G_{i-1}) \setminus \Omega(G_i)$ contains both u and v. To bound the size of the set $\Omega(G_{i-1}) \setminus \Omega(G_i)$, we associate each $I \in \Omega(G_{i-1}) \setminus \Omega(G_i)$ with an independent set $I \setminus \{v\} \in \Omega(G_i)$. An independent set $I' \in \Omega(G_i)$ is associated with no more than one independent set $I' \cup \{v\} \in \Omega(G_{i-1}) \setminus \Omega(G_i)$, and thus $|\Omega(G_{i-1}) \setminus \Omega(G_i)| \leq |\Omega(G_i)|$. It follows that

$$r_i = \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} = \frac{|\Omega(G_i)|}{|\Omega(G_i)| + |\Omega(G_{i-1}) \setminus \Omega(G_i)|} \geq 1/2.$$

□

Lemma

When $m \geq 1$ and $0 < \epsilon \leq 1$, the procedure for estimating r_i yields an estimate $\tilde{r}_i$ that is $(\epsilon/2m, \delta/m)$-approximation for r_i.

- Our estimate is $2^n \prod_{i=1}^{m} \tilde{r}_i$
- The true number is $|\Omega(G)| = 2^n \prod_{i=1}^{m} r_i$.
- To evaluate the error in our estimate we need to bound the ratio

$$R = \prod_{i=1}^{m} \frac{\tilde{r}_i}{r_i}.$$

Lemma

Suppose that for all i, $1 \leq i \leq m$, $\tilde{r}_i$ is an $(\epsilon/2m, \delta/m)$-approximation for r_i. Then

$$\Pr(|R - 1| \leq \epsilon) \geq 1 - \delta.$$

Proof: For each $1 \leq i \leq m$, we have

$$\Pr\left(|\tilde{r}_i - r_i| \leq \frac{\epsilon}{2m} r_i\right) \geq 1 - \frac{\delta}{m}.$$

Equivalently,

$$\Pr\left(|\tilde{r}_i - r_i| > \frac{\epsilon}{2m} r_i\right) < \frac{\delta}{m}.$$

By the union bound the probability that $|\tilde{r}_i - r_i| > \frac{\epsilon}{2m} r_i$ for any i is at most δ, and hence $|\tilde{r}_i - r_i| \leq \frac{\epsilon}{2m} r_i$ for all i with probability at least $1 - \delta$. Equivalently,

$$1 - \frac{\epsilon}{2m} \leq \frac{\tilde{r}_i}{r_i} \leq 1 + \frac{\epsilon}{2m}$$

holds for all i with probability at least $1 - \delta$. When these bounds hold for all i, we can combine them to obtain

$$1 - \epsilon \leq \left(1 - \frac{\epsilon}{2m}\right)^m \leq \prod_{i=1}^{m} \frac{\tilde{r}_i}{r_i} \leq \left(1 + \frac{\epsilon}{2m}\right)^m \leq (1 + \epsilon),$$

Estimating r_i

Input: Graphs $G_{i-1} = (V, E_{i-1})$ and $G_i = (V, E_i)$.

Output: $\tilde{r}_i =$ an approximation of r_i.

1. $X \leftarrow 0$.
2. Repeat for $M = \lceil 1296 m^2 \epsilon^{-2} \ln \frac{2m}{\delta} \rceil$ independent trials:
   1. Generate an uniform sample from $\Omega(G_{i-1})$;
   2. If the sample is an independent set in G_i, let $X \leftarrow X + 1$.
3. Return $\tilde{r}_i \leftarrow \frac{X}{M}$.

Definition

Let w be the (random) output of a sampling algorithm for a finite sample space Ω. The sampling algorithm generates an *ϵ-uniform sample* of Ω if, for any subset S of Ω,

$$\left| \Pr(w \in S) - \frac{|S|}{|\Omega|} \right| \leq \epsilon.$$

A sampling algorithm is a *fully polynomial almost uniform sampler (FPAUS)* for a problem if, given an input x and a parameter $\epsilon > 0$, it generates an ϵ-uniform sample of $\Omega(x)$, and it runs in time polynomial in $\ln \epsilon^{-1}$ and the size of the input x.

Estimating r_i
Input: Graphs $G_{i-1} = (V, E_{i-1})$ and $G_i = (V, E_i)$.
Output: $\tilde{r}_i =$ an approximation of r_i.

1. $X \leftarrow 0$.
2. Repeat for $M = \lceil 1296 m^2 \epsilon^{-2} \ln \frac{2m}{\delta} \rceil$ independent trials:
   1. Generate an $\frac{\epsilon}{6m}$-uniform sample from $\Omega(G_{i-1})$;
   2. If the sample is an independent set in G_i, let $X \leftarrow X + 1$.
3. Return $\tilde{r}_i \leftarrow \frac{X}{M}$.

Lemma

When $m \geq 1$ and $0 < \epsilon \leq 1$, the procedure for estimating r_i yields an $(\epsilon/2m, \delta/m)$-approximation for r_i

How do we Generate an $\frac{\epsilon}{6m}$-uniform sample from $\Omega(G_{i-1})$?

From Approximate Sampling to Approximate Counting

Theorem

Given a fully polynomial almost uniform sampler (FPAUS) for independent sets in any graph, we can construct a fully polynomial randomized approximation scheme (FPRAS) for the number of independent sets in a graph G with maximum degree at most Δ.

The Markov Chain Monte Carlo Method

Idea: define an ergodic Markov chain whose stationary distribution is the desired probability distribution.
Let $X_0, X_1, X_2, \ldots, X_n$ be the run of the chain.
The Markov chain converges to its stationary distribution from any starting state X_0 so after some sufficiently large number r of steps, the distribution at of the state X_r will be close to the stationary distribution π of the Markov chain.
Now, repeating with X_r as the starting point we can use X_{2r} as a sample etc.
So $X_r, X_{2r}, x_{3r}, \ldots$ can be used as almost independent samples from π.

$N(x)-$ set of neighbors of x. Let $M \geq \max_{x \in \Omega} |N(x)|$.

Lemma

Consider a Markov chain where for all x and y with $y \neq x$, $P_{x,y} = \frac{1}{M}$ if $y \in N(x)$, and $P_{x,y} = 0$ otherwise. Also, $P_{x,x} = 1 - \frac{|N(x)|}{M}$. If this chain is irreducible and aperiodic, then the stationary distribution is the uniform distribution.

Proof.

We show that the chain is time-reversible, and apply Theorem 7.10. For any $x \neq y$, if $\pi_x = \pi_y$, then

$$\pi_x P_{x,y} = \pi_y P_{y,x},$$

since $P_{x,y} = P_{y,x} = 1/M$. It follows that the uniform distribution $\pi_x = 1/|\Omega|$ is the stationary distribution. $\square$

Sampling a uniform distribution on the independent sets

Consider a Markov chain whose states are independent sets in a graph $G = (V, E)$:

1. X_0 is an arbitrary independent set in G.
2. To compute X_{i+1}:
   1. Choose a vertex v uniformly at random from V.
   2. If $v \in X_i$ then $X_{i+1} = X_i \setminus \{v\}$;
   3. if $v \notin X_i$, and adding v to X_i still gives an independent set, then $X_{i+1} = X_i \cup \{v\}$;
   4. otherwise, $X_{i+1} = X_i$.

- The chain is irreducible
- The chain is aperiodic (as G has at least one edge)
- For $y \neq x$, $P_{x,y} = 1/|V|$ or 0.

The lemma implies that the stationary distribution is the uniform distribution.

The Metropolis Algorithm

Assuming that we want to sample with non-uniform distribution.
For example, we want the probability of an independent set of size i to be proportional to λ^i.
Consider a Markov chain on independent sets in $G = (V, E)$:

1. X_0 is an arbitrary independent set in G.
2. To compute X_{i+1}:
    1. Choose a vertex v uniformly at random from V.
    2. If $v \in X_i$ then set $X_{i+1} = X_i \setminus \{v\}$ with probability $\min(1, 1/\lambda)$;
    3. if $v \notin X_i$, and adding v to X_i still gives an independent set, then set $X_{i+1} = X_i \cup \{v\}$ with probability $\min(1, \lambda)$;
    4. otherwise, set $X_{i+1} = X_i$.

Lemma

For a finite state space Ω, let $M \geq \max_{x \in \Omega} |N(x)|$. For all $x \in \Omega$, let $\pi_x > 0$ be the desired probability of state x in the stationary distribution. Consider a Markov chain where for all x and y with $y \neq x$,

$$P_{x,y} = \frac{1}{M} \min\left(1, \frac{\pi_y}{\pi_x}\right)$$

if $y \in N(x)$, and $P_{x,y} = 0$ otherwise. Further, $P_{x,x} = 1 - \sum_{y \neq x} P_{x,y}$. Then if this chain is irreducible and aperiodic, the stationary distribution is given by the probabilities π_x.

Proof.

We show the chain is time-reversible. For any $x \neq y$, if $\pi_x \leq \pi_y$, then $P_{x,y} = 1$ and $P_{y,x} = \pi_x/\pi_y$. It follows that $\pi_x P_{x,y} = \pi_y P_{y,x}$. Similarly, if $\pi_x > \pi_y$, then $P_{x,y} = \pi_y/\pi_x$ and $P_{y,x} = 1$, and it follows that $\pi_x P_{x,y} = \pi_y P_{y,x}$. □

Note that the Metropolis Algorithm only needs the ratios π_x/π_y's. In our construction, the probability of an independent set of size i is λ^i/B for $B = \sum_x \lambda^{size(x)}$ although we may not know B.