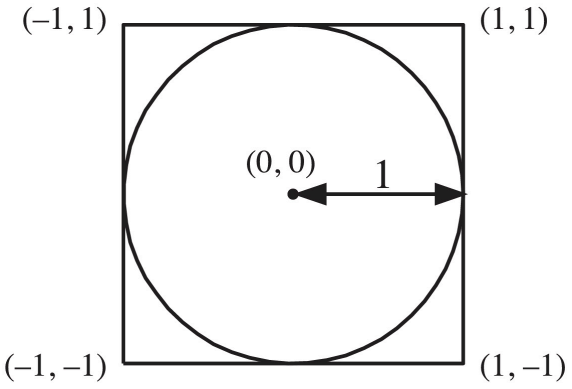
• The Monte Carlo Method. M-U Ch 11

- Refers to a collection of tools for estimating values through sampling and simulation.

§ Approach to estimate π :



Let (X, Y) be a point chosen uniformly at random in $[-1,1] \times [-1,1]$.

[equiv. $X \sim U[-1,1], Y \sim U[-1,1]$].

Define.

$$Z = \begin{cases} 1 & \text{if } x^2+y^2 \le 1 \\ 0 & \text{otherwise.} \end{cases}$$

Hence, $\mathbb{P}[Z=1] = \frac{\pi \cdot 1^2}{2^2} = \pi/4$.

Assume, we run the experiment m times, Z_i being the value of Z at the ith run.

let $W = \sum_{i=1}^{m} Z_i$, then

$$\mathbb{E}[W] = \mathbb{E}\left[\sum_{i=1}^{m} Z_i\right] = \sum_{i=1}^{m} \mathbb{E}[Z_i] = \frac{m\pi}{4}.$$

Then $W' = (4/m)W$ is a natural estimate for π.

Applying chernoff bounds,

$$\begin{aligned} \mathbb{P}[|W' - \pi| \ge \varepsilon\pi] &= \mathbb{P}\left[\left|W - \frac{m\pi}{4}\right| \ge \frac{\varepsilon m\pi}{4}\right] \\ &= \mathbb{P}[|W - \mathbb{E}[W]| \ge \varepsilon\, \mathbb{E}[W]] \\ &\le 2e^{-\mathbb{E}W \cdot \varepsilon^2/3} = 2e^{-m\pi\varepsilon^2/12}. \end{aligned}$$

Hence, large number of samples will imply good approximation of π.

Definition 11.1: *A randomized algorithm gives an* (ε, δ)-approximation *for the value* V *if the output* X *of the algorithm satisfies*

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

The above method for π gives (ε,δ)-approx by choosing $\varepsilon < 1$ and $2e^{-m\pi\varepsilon^2/12} \leq \delta$, i.e. $m \geq 12 \ln(2/\delta)/\pi\varepsilon^2$.

- Chernoff bound for (ε,δ)-approximation.

Theorem 11.1: *Let* $X_1, \ldots, X_m$ *be independent and identically distributed indicator random variables, with* $\mu = \mathbf{E}[X_i]$. *If* $m \geq (3\ln(2/\delta))/\varepsilon^2\mu$, *then*

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

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

say indepset

some graph

Definition 11.2: *A* fully polynomial randomized approximation scheme (FPRAS) *for a problem is a randomized algorithm for which, given an input* x *and any parameters* ε *and* δ *with* $0 < \varepsilon, \delta < 1$, *the algorithm outputs an* (ε, δ)*-approximation to* $V(x)$ *in time that is polynomial in* $1/\varepsilon$, $\ln \delta^{-1}$, *and the size of the input* x.

max indepset

- Monte Carlo method require an efficient process that generates a sequence of i.i.d. RVs $X_1, \ldots, X_n$ s.t. $\mathbb{E}[X_i] = V$, the value we want to approximate.

We then take enough samples to get an (ε,δ)-approximation to V.

→ Generating a good sequence of samples is often a nontrivial task.

§ Application: DNF counting problem.

Disjunctive normal form:

- A boolean formula F
- Disjunction (OR) of clauses
- Each clause is conjunction (AND) of literals.

$$(x_1 \wedge \overline{x}_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (\overline{x}_1 \wedge x_3 \wedge x_4).$$
$$\quad\; c_1 \qquad\qquad\quad c_2 \qquad\qquad\quad c_3$$

- Easy to check satisfiability.
- How hard is is counting $c(F)$, the number of satisfying assignments of F?

Note, CNF formula H $\xrightarrow{\text{deMorgan's law}}$ DNF formula $\overline{H}$

$$\neg(P \vee Q) \iff (\neg P) \wedge (\neg Q),$$
and
$$\neg(P \wedge Q) \iff (\neg P) \vee (\neg Q)$$

- A CNF formula H has satisfying assignment iff there is some assignment for the variables (which) doesnot satisfy $\overline{H}$.

i.e. H is satisfiable $\iff c(\overline{H}) < 2^n$.

So, finding $c(\overline{H})$ is at least as hard as solving NP-complete problem SAT.

- Complexity class #P (sharp-P): ← class of function problems, not decision problems.

Informally, is the set of counting problems assoc. with the decision problems in NP.

formally, a problem $\Pi \in \#P$ if there is a polytime nondeterministic turing machine s.t. for any input I, the number of accepting computations equals the number of different solutions associated with the input I.

computing $c(F)$ is #P-complete: as hard as any problem in #P.

(examples of other #P-complete problems: counting no. of HAM-cycles in a graph, counting no. of perfect matchings in bipartite graph,..)

- So we don't expect to exactly compute $c(F)$, but will go for FPRAS.

- Naive approach:

DNF Counting Algorithm I:

Input: A DNF formula F with n variables.

Output: $Y =$ an approximation of $c(F)$.

1. $X \leftarrow 0$.
2. For $k = 1$ to m, do:
   (a) Generate a random assignment for the n variables, chosen uniformly at random from all 2^n possible assignments.
   (b) If the random assignment satisfies F, then $X \leftarrow X + 1$.
3. Return $Y \leftarrow (X/m)2^n$.

w.l.o.g. assume $c(F) > 0$ as it is easy to check $c(F) = 0$ or not.

Define RV $X_K = \begin{cases} 1, & \text{if } k\text{'th iteration generates a satisfying assignment,} \\ 0, & \text{otherwise,} \end{cases}$

Let $X = \sum_{K=1}^{m} X_K$, then $\mathbb{E}[X] = m\,\mathbb{E}[X_K] = m \cdot \frac{e(F)}{2^n}$.

$\therefore \mathbb{E}[Y] = \frac{\mathbb{E}[X] \cdot 2^n}{m} = e(F)$. ← good!

Applying Theorem 11.1, Y gives (ε, δ)-approx of $e(F)$, when $m \geq 3 \cdot 2^n \ln(2/\delta) / (\varepsilon^2 e(F))$.

μ is small

So if $e(F) = O(n^t)$ for $t \in \mathbb{Z}$, then m is not polynomial.

#var = n, each clause has k literals ($1 \to n$)
#clauses = t, $\binom{2n}{k}^t$

Intuitively, in this case, w.h.p. we must sample an exponential number of assignments before finding the first satisfying assignment.

So for our strategy to work, we need to construct a sample space containing all satisfying assignments and these assignments are sufficiently dense in the sample space to allow efficient sampling.

§ FPRAS for DNF Counting:

Let $F = C_1 \vee C_2 \vee \cdots \vee C_t$.

So, F is satisfied if any clause C_i is satisfied.

If clause C_i has l_i literals, then there are 2^{n-l_i} satisfying assignments for C_i.

Let SC_i be the set of all satisfying assignments for C_i.

Define, $U = \{(i,a) \mid i \in [t],\ a \in SC_i\}$.

As $|U| = \sum_{i=1}^{t} |SC_i|$, and $|SC_i| = 2^{n-l_i}$, we know $|U|$.

can be exponential if $n - l_i$ is $\Omega(n)$. in that case $c(F)$ is $2^{\Omega(n)}$. a naive approach will work.

But, say $n - l_i = o(n)$ then naive will not work.

We want to compute $c(F) = |\bigcup_{i=1}^{t} SC_i| \leq |U|$.

can be strict, as an assignment can satisfy more that one clause.

To estimate $c(F)$, we construct $S \subseteq U$ with $|S| = c(F)$.

$$S = \{(i,a) \mid i \in [t],\ a \in SC_i,\ a \notin SC_j \text{ for } j < i\}$$

– assigning each satisfying assignment to a unique (i,a) tuple.

To estimate $|S|$, we want to estimate $|S|/|U|$. known

Advantage: S is relatively dense in U.

[As each assignment can satisfy at most t different clauses, $|S|/|U| \geq 1/t$.]

Cute trick: two-stage sampling

- How to sample S uniformly from U?

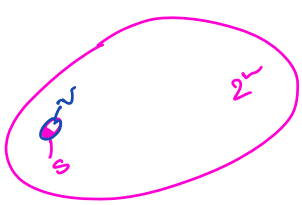
To choose (i,a), we choose i with probability

$$\frac{|SC_i|}{\sum_{i=1}^{t} |SC_i|} = \frac{|SC_i|}{|U|}.$$

Then to select 'a', we choose T/F uniformly at random for each literal not in clause i. Then

$$\mathbb{P}[(i,a) \text{ is chosen}] = \mathbb{P}[i \text{ is chosen}] \cdot \mathbb{P}\left[\begin{matrix} a \text{ is} \\ \text{chosen} \end{matrix} \,\middle|\, \begin{matrix} i \text{ is} \\ \text{chosen} \end{matrix}\right]$$

$$= \frac{|SC_i|}{|U|} \cdot \frac{1}{|SC_i|} = \frac{1}{|U|}.$$



DNF Counting Algorithm II:

Input: A DNF formula F with n variables.

Output: $Y =$ an approximation of $c(F)$.

1. $X \leftarrow 0$.

2. For $k = 1$ to m, do:

(a) With probability $|SC_i| / \sum_{i=1}^{t} |SC_i|$ choose, uniformly at random, an assignment $a \in SC_i$.

(b) If a is not in any SC_j, $j < i$, then $X \leftarrow X + 1$.

3. Return $Y \leftarrow (X/m) \sum_{i=1}^{t} |SC_i|$.

- **Theorem:** Above algo is a FPRAS when

$$m = \lceil (3t/\varepsilon^2) \ln(2/\delta) \rceil.$$

→ poly $(t, \varepsilon, \ln(1/\delta))$.

Proof:

Step 2a, selects an element $(i,a) \in U$, uniformly at random.

Define $X_i = \begin{cases} 1 & \text{if } (i,a) \in S \\ 0 & \text{otherwise.} \end{cases}$, Then $X = \sum X_i$.

Now, $\mathbb{P}[(i,a) \in S] = \frac{c(F)}{|U|} \geq 1/t$. $\Rightarrow \mathbb{E}[X_i] = 1/t$.

→ put $\mu = \frac{c(F)}{|U|}$.

Then by Theorem 11.1, with these samples X/m (resp. Y) gives an (ε, δ)-approximation of $c(F)/|U|$ (resp. $c(F)$).

§ From approximate sampling to approximate counting

Definition 11.3: *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,*

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

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

E.g. for FPAUS for independent sets (IS)
input := a graph $G = (V, E)$ and a parameter ε,
sample space := all independent sets in G,
output := ε-uniform sample of the indep. sets
in time poly$(|V|, \ln \varepsilon^{-1})$.

Q. Given an FPAUS for IS, can we construct an FPRAS for counting number of IS.

Let $E(G) = \{e_1, \ldots, e_m\}$.

Let $E_i := \{e_1, \ldots, e_i\}$, $G_i := \{V, E_i\}$. So, $G = G_m$.

Let $\Omega(G_i)$ denote the set of IS of G_i.

Now, $|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \times \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \times \ldots \times \frac{|\Omega(G_1)|}{|\Omega(G_0)|} \times |\Omega(G_0)|$

Clearly, G_0 has no edges, so every subset of V is an IS $\Rightarrow |\Omega(G_0)| = 2^n$.

Now we need good estimate $\widetilde{r_i}$ for the ratios:

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

Then, estimate of $|\Omega(G)|$ will be $2^n \prod_{i=1}^{m} \tilde{r}_i$.

Whereas $|\Omega(G)| = 2^n \prod_{i=1}^{m} r_i$.

To evaluate error in the estimate, we need to bound the ratio:

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

Thus to have an (ε, δ)-approximation (Defn 11.1), we want $\mathbb{P}(|R-1| \leq \varepsilon) \geq 1-\delta$.

- **Lemma:** If $\forall i \in [m]$, $\tilde{r}_i$ is an $(\varepsilon/2m, \delta/2m)$-approx for r_i, then $\mathbb{P}(|R-1| \leq \varepsilon) \geq 1-\delta$.

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

- **Lemma:** When $m \geq 1$, $0 < \varepsilon \leq 1$, above algo yields an $(\varepsilon/2m, \delta/m)$-approx for r_i.

- **Theorem:** Given an FPAUS for IS in any graph G, we can construct an FPRAS for #IS in G.

Main takeaway: Construct a sequence of refinements of the problem, starting with an instance that is easy to count and ending with actual counting problem, s.t. the ratio between the counts in successive instances is at most poly(input).

§ The Markov Chain Monte Carlo Method.

Why MC to sample?
– Think of IS problem. There are exponential possible items from which we want to sample. How to make sure they follow a specific distribution?

– general approach to sample from a desired probability distribution.

Idea: Define an ergodic Markov chain whose set of states is the sample space, and whose stationary distribution is the required sampling distribution.

Let $X_0, X_1, \ldots$ be a run of the chain.

After a sufficiently large number of steps r, the distribution of the state $X_r \approx$ stationary distribution (indep of X_0).

We can repeat the same starting from X_r.

Thus, $X_r, X_{2r}, X_{3r}, \ldots$ can be thought as almost independent samples from the stationary distr. of the Markov chain.

- Efficiency of this approach depends on:

(a) how large r must be to ensure a suitably good sample,

(b) how much computation is required for each step of the Markov chain.

→ Simplest case: construct a Markov chain with a stationary distr. that is uniform over the state space Ω.

– Need to design a set of moves that ensures the state space is irreducible under the Markov chain.

Consider state space to be all the independent sets. Two indep sets x, y are neighbors of each other, if they differ in just one vertex.

This makes the state space to be irreducible, as all indep. sets can reach the empty indep. set by a sequence of vertex deletions.

Note: just doing a random walk won't give stationary distribution to be uniform,
[$\because$ it converges to $\pi_v = d(v)/2|E|$ and the graph may not be regular].

Lemma 11.7: *For a finite state space Ω and neighborhood structure $\{N(X) \mid x \in \Omega\}$, let $N = \max_{x\in\Omega} |N(x)|$. Let M be any number such that $M \geq N$. Consider a Markov chain where*

$$P_{x,y} = \begin{cases} 1/M & \text{if } x \neq y \text{ and } y \in N(x), \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x), \\ 1 - N(x)/M & \text{if } x = y. \end{cases}$$

← self-loop

If this chain is irreducible and aperiodic, then the stationary distribution is the uniform distribution.

Consider now the following simple 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}:
   - **(a)** choose a vertex v uniformly at random from V;
   - **(b)** if $v \in X_i$ then $X_{i+1} = X_i \setminus \{v\}$;
   - **(c)** if $v \notin X_i$ and if adding v to X_i still gives an independent set, then $X_{i+1} = X_i \cup \{v\}$;
   - **(d)** otherwise, $X_{i+1} = X_i$.

This chain has the property that the neighbors of a state X_i are all independent sets that differ from X_i in just one vertex. Since every state can reach and is reachable from the empty set, the chain is irreducible. Assuming that G has at least one edge (u, v), then the state $\{v\}$ has a self-loop ($P_{v,v} > 0$), and the chain is aperiodic. Further, when $y \neq x$, it follows that $P_{x,y} = 1/|V|$ or 0. Lemma 11.7 therefore applies, and the stationary distribution is the uniform distribution.

- The Metropolis Algorithm:

– Generalizes to sample from a chain with a nonuniform stationary distribution.

- Say, we want to construct a Markov chain with stationary distr. $\pi_x = b(x)/B$, where $\forall x \in \Omega \quad b(x) > 0$, and $B = \sum_{x \in \Omega} b(x)$ is finite.

Lemma 11.8: *For a finite state space Ω and neighborhood structure $\{N(X) \mid x \in \Omega\}$, let $N = \max_{x \in \Omega} |N(x)|$. Let M be any number such that $M \geq N$. 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*

$$P_{x,y} = \begin{cases} (1/M)\min(1, \pi_y/\pi_x) & \text{if } x \neq y \text{ and } y \in N(x), \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x), \\ 1 - \sum_{y \neq x} P_{x,y} & \text{if } x = y. \end{cases}$$

Then, if this chain is irreducible and aperiodic, the stationary distribution is given by the probabilities π_x.

outgoing influx: say $\pi_y < \pi_x$. $\pi_x \cdot \frac{1}{M}\min(1, \frac{\pi_y}{\pi_x}) = \frac{\pi_y}{M}$.

ingoing influx: $\pi_y \cdot \frac{1}{M}\min(1, \frac{\pi_x}{\pi_y}) = \pi_y \cdot \frac{1}{M} \cdot 1 = \pi_y/M$.

For example, say we need to sample each indep. set with prob. proportional to $\lambda^{|I|}$ for constant $\lambda > 0$.
So we need $\pi_x = \lambda^{|I_x|}/B$, $B = \sum_x \lambda^{|I_x|}$.

$\lambda = 1 \Rightarrow$ uniform distr.
$\lambda > 1 \Rightarrow$ large indep. sets have higher prob.
$\lambda < 1 \Rightarrow$ " " " smaller prob.

Consider now the following variation on the previous Markov chain for independent sets in a graph $G = (V, E)$.

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

It is a two-step approach:

① Propose a move by choosing a vertex v to add or delete, where each vertex is chosen with probability $1/M$, here $M = |V|$. (step 2a).

② This choice is then accepted with prob. $\min(1, \pi_y/\pi_x)$.

Here, π_y/π_x $\begin{cases} = \lambda, & \text{if the chain attempts to add a vertex.} \\ = 1/\lambda, & \text{if the chain attempts to delete a vertex.} \end{cases}$

Thus transition probability:

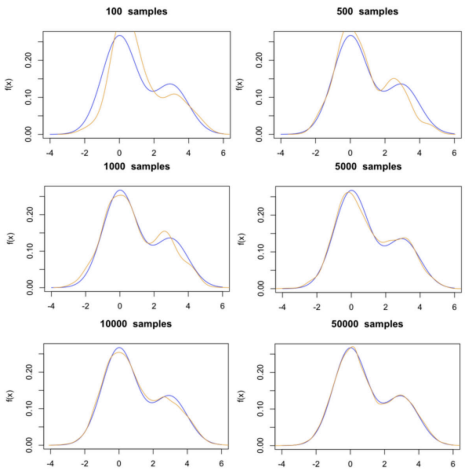
$$P_{x,y} = \frac{1}{M} \min(1, \pi_y/\pi_x). \quad \text{for } y \in N(x),\ x \neq y.$$

So lemma 11.8 applies.

flux transfer: say $\pi_y < \pi_x$

$$\pi_x \cdot \frac{\pi_y}{\pi_x} \cdot \frac{1}{M} = \frac{\pi_y}{M} \cdot 1.$$

Note: even for $\lambda = 1$, B is number of IS which can be $\exp(n)$. But we don't need to know B! Just π_y/π_x is sufficient.



← Convergence of Metropolis algorithm.

§ Coupling of Markov Chains. Ch12 M-U

– a powerful method for bounding the rate of convergence of Markov chains.

§ Variation distance & mixing time.

Definition 12.1: *The* variation distance *between two distributions* D_1 *and* D_2 *on a countable state space* S *is given by*

makes it in [0,1].

$$\|D_1 - D_2\| = \frac{1}{2}\sum_{x\in S} |D_1(x) - D_2(x)|.$$

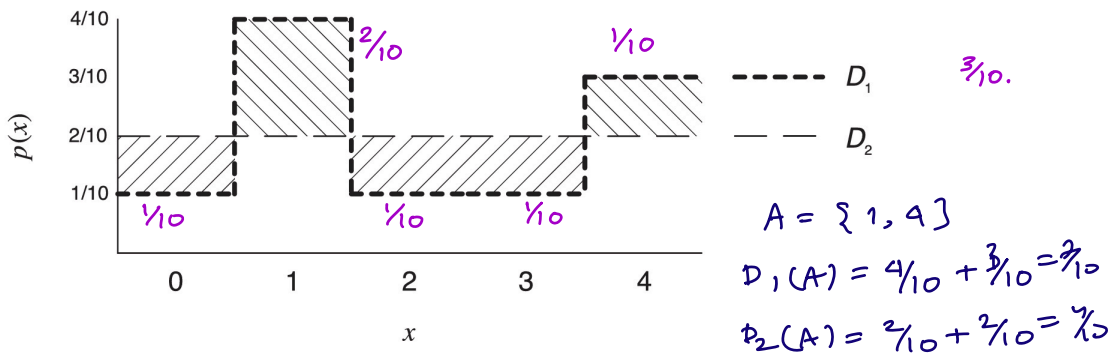


Figure 12.1: Example of variation distance. The areas shaded by upward diagonal lines correspond to values x where $D_1(x) < D_2(x)$; the areas shaded by downward diagonal lines correspond to values x where $D_1(x) > D_2(x)$. The total area shaded by upward diagonal lines must equal the total area shaded by downward diagonal lines, and the variation distance equals one of these two areas.

Lemma 12.1: *For any* $A \subseteq S$, *let* $D_i(A) = \sum_{x\in A} D_i(x)$ *for* $i = 1, 2$. *Then*

$$\|D_1 - D_2\| = \max_{A\subseteq S} |D_1(A) - D_2(A)|.$$

A careful examination of Figure 12.1 helps make the proof of this lemma transparent.

- Lemma: A sampling algorithm returns an ε-uniform sample on Ω iff $\|D - U\| \le \varepsilon$.
  (output distr. / uniform distr.)

Definition 12.2: *Let $\bar{\pi}$ be the stationary distribution of an ergodic Markov chain with state space S. Let p_x^t represent the distribution of the state of the chain starting at state x after t steps. We define*

$$\Delta_x(t) = \|p_x^t - \bar{\pi}\|; \qquad \Delta(t) = \max_{x \in S} \Delta_x(t).$$

That is, $\Delta_x(t)$ is the variation distance between the stationary distribution and p_x^t, and $\Delta(t)$ is the maximum of these values over all states x.

We also define

$$\tau_x(\varepsilon) = \min\{t : \Delta_x(t) \leq \varepsilon\}; \qquad \tau(\varepsilon) = \max_{x \in S} \tau_x(\varepsilon).$$

That is, $\tau_x(\varepsilon)$ is the first step t at which the variation distance between p_x^t and the stationary distribution is less than ε, and $\tau(\varepsilon)$ is the maximum of these values over all states x.

called mixing time of Markov chain.

- A chain is rapidly mixing if $\tau(\varepsilon) = \text{poly}(\text{input}, \log 1/\varepsilon)$.

§ Coupling:

- A general technique for bounding mixing time.

Definition 12.3: *A* coupling *of a Markov chain M_t with state space S is a Markov chain $Z_t = (X_t, Y_t)$ on the state space $S \times S$ such that:*

$$\Pr(X_{t+1} = x' \mid Z_t = (x, y)) = \Pr(M_{t+1} = x' \mid M_t = x);$$
$$\Pr(Y_{t+1} = y' \mid Z_t = (x, y)) = \Pr(M_{t+1} = y' \mid M_t = y).$$

Lemma 12.2 [Coupling Lemma]: *Let $Z_t = (X_t, Y_t)$ be a coupling for a Markov chain M on a state space S. Suppose that there exists a T such that, for every $x, y \in S$,*

$$\Pr(X_T \neq Y_T \mid X_0 = x, Y_0 = y) \leq \varepsilon.$$

useful to show faster sampling.

Then

$$\tau(\varepsilon) \leq T.$$

That is, for any initial state, the variation distance between the distribution of the state of the chain after T steps and the stationary distribution is at most ε.

- We are interested in couplings that
  - bring the two copies of the chain to the same state and then
  - keep them in the same state by having the two chains move identically.

→ When two copies of the chain reach the same state, they are said to have coupled.

EXAMPLE 5.1. A simple random walk on the segment $\{0, 1, \ldots, n\}$ is a Markov chain which moves either up or down at each move with equal probability. If the walk attempts to move outside the interval when at a boundary point, it stays put. It is intuitively clear that $P^t(x, n) \leq P^t(y, n)$ whenever $x \leq y$, as this says that the chance of being at the "top" value n after t steps does not decrease as you increase the height of the starting position.

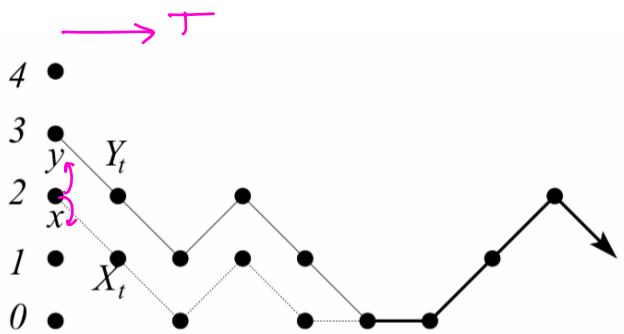


FIGURE 5.1. Coupled random walks on $\{0, 1, 2, 3, 4\}$. The walks stay together after meeting.

Proof of $p^t(x,n) \leq p^t(y,n)$.

→ Let $\Delta_i = \begin{cases} +1 & \text{w.p. } 1/2 \\ -1 & \text{w.p. } 1/2. \end{cases}$, $\Delta_1, \Delta_2, \cdots$ iid RV

X_t, Y_t are random walks on $\{0, 1, \cdots, n\}$ starting at x, and y, respectively.

$\Delta_t = +1$, move both up if possible
$= -1$, move both down if possible.

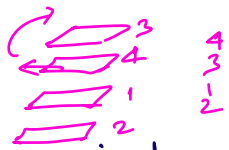
Clearly, if $x \leq y$, then by this coupling $X_t \leq Y_t$. $\forall t$.

Hence, if $X_t = n$, then $Y_t = n$ as well.

$\therefore \; p^t(x,n) = \mathbb{P}(X_t = n) \leq \mathbb{P}(Y_t = n) = p^t(y,n)$.

– This shows power of coupling: building two simultaneous copies of a Markov chain using a shared randomness, can be useful to obtain bounds on the distance to stationary.

§ Example: Shuffling cards.



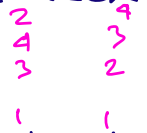
– n cards are being shuffled.

– At each step, a card is chosen independently and uniformly at random, and put on the top of the deck.

$n!$ states

Edge = Ulam distance 1.

Can be modeled as Markov chain, where the state is the current order of the deck.



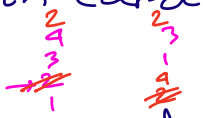
Consider the following coupling:

Given two copies X_t and Y_t of the chain in different states, choose a position $j \in [n]$ uniformly at random and simultaneously choose the jth card from the top & move it to the top.

– This is a valid coupling, as each chain individually acts as the original shuffling Markov chain.

However, as the chains start from diff state, jth card might be different in them. So we may not bring both the chains towards same state.

Alternate coupling: Choose $j \in [n]$ unif. at random. Obtain X_{t+1} from X_t by moving jth card on top. let the top card by c.



To obtain Y_{t+1} from Y_t move the card w. value c to the top.

- This is a valid coupling, as in both chains prob. a specific card is moved to top $= 1/n$.

- With this coupling, if same card C moves to the top, it remains in the same position in both the chains.

- So, two copies are coupled once every card has been moved to the top at least once.

- This is coupon collector's problem.

Now $\mathbb{P}$[a specific card is not moved to top after $n \ln n + cn$ step]

$\leq \left(1 - \frac{1}{n}\right)^{n \ln n + cn} \leq e^{-(\ln n + c)} = e^{-c}/n.$

Union bound $\Rightarrow$ $\mathbb{P}$[$\exists$ a card not moved to top at least once] $\leq e^{-c}$.

Take $c = \ln(1/\varepsilon) \Rightarrow e^{-c} = \varepsilon.$

So after $n \ln n + n \ln(1/\varepsilon)$ steps, prob. that the chains have not coupled is $\leq \varepsilon$.

So, coupling lemma imply the variation dist. between the uniform distribution and the distr. of the state of the chain after $n \ln(1/\varepsilon)$ steps is bounded above by ε.

Message: Quick coupling = Good mixing.

§ Application: Independent sets of fixed size.

Consider Markov chain whose states are indep. sets of size exactly k.

Define: move (v, w, X_t):

choose $v \in X_t$ and $w \in V$, indep. & unif. at random.

If $w \notin X_t$, and $(X_t - \{v\}) \cup \{w\}$ is indep. set then

$$X_{t+1} = (X_t - \{v\}) \cup \{w\}.$$ $X_t - v + w$.

Otherwise, $X_{t+1} = X_t$.

HW: Show this chain is ergodic if $k \leq n/(2\Delta+2)$, [Ex 12.11].

We show that this chain is rapidly mixing whenever $k \leq n/(3\Delta+3)$.

If $k = n/2$, then for $K_{n/2, n/2}$ it is disconnected.

Coupling on $Z_t = (X_t, Y_t)$:

For coupling, choose $v \in X_t$, $w \in V$ unif at random and perform move $m(v, w, X_t)$.

$\frac{|X_t \cap Y_t|}{k} \cdot \frac{1}{|X_t \cap Y_t|}$

for transition of Y_t,

if $v \in Y_t$, perform move $m(v, w, Y_t)$.

if $v \notin Y_t$, perform move $m(v', w, Y_t)$, where v' is unif. chosen at random from $Y_t - X_t$.

$\frac{|Y_t - X_t|}{k} \cdot \frac{1}{|Y_t - X_t|}$

Let $d_t = |X_t - Y_t|$ measure the difference between the two independent sets after t steps.

We'll see d_t changes by at most 1 and d_t is more likely to decrease than increase.

$v \in X_t$. Case 1. $v \in X_t \cap Y_t$.

Then d_t remains same if no chain moves or if both moves (then v gets deleted & w is added in both). $\rightarrow w \notin X_t, w \notin Y_t$

Then d_t increases only if one of the chains move & the other does not.

Say X_t moves & Y_t don't.

- either $w \in Y_t$ but $w \notin X_t$.

  or $w \in N(Y_t - v)$ but $w \notin N(X_t - v)$,

when Y_t moves & X_t don't, it is analogus.

X_t doesnot move — Y_t doesnot move

Thus w must be a vertex or a neighbor of a vertex in set $(X_t - Y_t) \cup (Y_t - X_t)$.

Case 2: $v \in X_t \setminus Y_t$.

i.e. $v \notin Y_t$. moves: $m(v, w, X_t)$; $m(v', w, Y_t)$. (v' from $Y_t - X_t$)

As $v \notin Y_t$, $v' \notin X_t$, if both moves d_t decreases.
(as both will have w in it)

This happens when w is not in $(X_t \cup Y_t) - v - v'$ or their neighbors.

If both does not move, d_t remains same.

If one moves (say X_t) & other does not (Y_t) then w is in $(Y_t - v')$ or its neighbors.
So d_t can not increase in this case.
(same or decrease.)

Suppose $d_t > 0$. Now $\boxed{d_{t+1} = d_t + 1}$ means at t, v (must be) chosen from $X_t \cap Y_t$, and w (must be) chosen s.t. there is a transition in exactly one of the chains.

Thus w must be a vertex or a neighbor of a vertex in set $(X_t - Y_t) \cup (Y_t - X_t)$. ($X_t$ does not move; Y_t does not move; $|X_t \Delta Y_t|$)

$$\therefore \mathbb{P}(d_{t+1} = d_t + 1 \mid d_t > 0) \leq \frac{K - d_t}{K} \cdot \frac{2d_t(\Delta+1)}{n}$$

($|X_t \cap Y_t|$; degree)

$\mathbb{P}[v \in X_t \cap Y_t]$

$\mathbb{P}[w \in \text{Nbr}(X_t - Y_t \cup Y_t - X_t)]$

Case 2

If $d_{t+1} = d_t - 1$, then at time t, $v \in X_t$, $v \notin Y_t$, it is sufficient to consider the case when w is neither a vertex nor a neighbor of a vertex in $X_t \cup Y_t - \{v, v'\}$.

$$\mathbb{P}(d_{t+1} = d_t - 1 \mid d_t > 0) \geq \frac{d_t}{K} \cdot \left(\frac{n - (K + d_t - 2)(\Delta+1)}{n}\right)$$

Hence, for $d_t > 0$. ($v \in X_t - Y_t$)

$$\mathbb{E}[d_{t+1} \mid d_t] = d_t + \mathbb{P}(d_{t+1} = d_t + 1) - \mathbb{P}(d_{t+1} = d_t - 1)$$

$$\leq d_t + \frac{K - d_t}{K} \cdot \frac{2d_t(\Delta+1)}{n} - \frac{d_t}{K} \cdot \frac{n - (K + d_t - 2)(\Delta+1)}{n}$$

$$= d_t \left(1 - \frac{n - (3K - d_t - 2)(\Delta+1)}{Kn}\right)$$

$$\leq d_t \left(1 - \frac{n - (3K-3)(\Delta+1)}{Kn}\right). \quad (\because d_t > 0)$$

i.e. $d_t \geq 1$

Once $d_t = 0$, both chains follow same path.
so, $\mathbb{E}[d_{t+1} \mid d_t = 0] = 0$.

Using property of conditional expectation,

$$\mathbb{E}[d_{t+1}] = \mathbb{E}[\mathbb{E}[d_{t+1} \mid d_t]]$$

$$\leq \mathbb{E}\left[d_t \left(1 - \frac{n - (3K-3)(\Delta+1)}{Kn}\right)\right]$$

$$= \mathbb{E}[d_t]\left(1 - \frac{n(3K-3)(\Delta+1)}{Kn}\right].$$

By induction,

$$\mathbb{E}[d_t] \le d_0 \left(1 - \frac{n(3K-3)(\Delta+1)}{Kn}\right)^t \quad \cdots (*)$$

Since, $d_0 \le K$, $d_t \ge 0$.

$$\mathbb{P}[d_t \ge 1] \le \mathbb{E}[d_t] \qquad \text{(By Markov)}$$

$$\le K\left(1 - \frac{n(3K-3)(\Delta+1)}{Kn}\right)^t \quad [\text{By } (*)]$$

$$\le K e^{-t(n-(3K-3)(\Delta+1))/Kn}.$$

$$\left[\because K e^{-t(n-(3K-3)(\Delta+1))/Kn} \le \varepsilon \Rightarrow e^{t(n-(3K-3)(\Delta+1))/Kn} \ge K\varepsilon^{-1}\right.$$
$$\left.\Rightarrow t[n-(3K-3)(\Delta+1)] \le Kn \ln(K\varepsilon^{-1}) \Rightarrow \tau(\varepsilon) \le \frac{Kn \ln(K\varepsilon^{-1})}{n-(3K-3)(\Delta+1)}.\right]$$

So, whenever $n - (3K-3)(\Delta+1) \ge 0$

i.e. $K \le n/(3\Delta+3)$, RHS or the variation distance converges to 0.

$$\therefore \tau(\varepsilon) \le \frac{Kn \ln(K\varepsilon^{-1})}{n-(3K-3)(\Delta+1)}.$$

→ The chain is rapidly mixing.

- Check out lecture notes of Aspnes for practice problems.