# CSC 5450 Randomness and Computation

## Week 7: Markov Chain Monte Carlo

**Plan** ① Monte Carlo method, DNF counting, network reliability

② approximate counting & approximate sampling, Markov chain Monte Carlo method

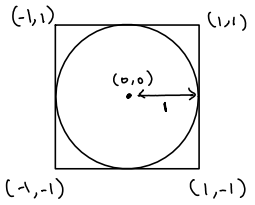
③ coupling: stationary distribution, shuffling cards, independent set, (random spanning tree).

---

### Monte Carlo method (MU 10.1)

It is a method to estimate values through sampling and simulation.

For example, to estimate the value of $\pi$, one can draw a circle of radius one and then repeatedly draw random points and count the fraction of points that fall inside the circle; see the picture.




The probability that a random point is in the circle is $\frac{\pi}{4}$.

So, if we choose $m$ random points, then the expected number of points in the circle is $m\pi/4$.

Hence, if there are $W$ points fall inside the circle in our experiment, then it is very natural to use $\frac{4W}{m}$ as an estimate of $\pi$.

How good is this estimation?

Intuitively, it is more accurate if we take more samples, and it is easy to prove it formally using the by-now standard Chernoff bound.

$$\Pr\left(\left|\frac{4W}{m} - \pi\right| \geq \varepsilon\pi\right) = \Pr\left(\left|W - \frac{m\pi}{4}\right| \geq \frac{\varepsilon m\pi}{4}\right)$$

$$= \Pr\left(|W - E[W]| \geq \varepsilon E[W]\right)$$

$$\leq 2e^{-\left(\frac{m\pi}{4}\right)\varepsilon^2/3} = 2e^{-\frac{m\pi\varepsilon^2}{12}}$$

By taking $m = \frac{12}{\pi\varepsilon^2} \cdot \ln(2/\delta)$, then this probability is at most $\delta$.

The same proof can be used to prove a more general statement.

**Theorem** Let $X_1, \ldots, X_m$ be independent and identically distributed indicator random

variables, with $\mu = E[X_i]$. If $m \geq \frac{3\ln(2/\delta)}{\varepsilon^2 \mu}$, then $\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| \geq \varepsilon\mu\right) \leq \delta$.

**Definition** A randomized algorithm gives an $(\varepsilon,\delta)$-approximation for the value $V$ if the output $X$ of the algorithm satisfies $\Pr(|X - V| \leq \varepsilon V) \geq 1-\delta$.

So, the above theorem gives an upper bound on the number of samples for an $(\varepsilon,\delta)$-approximation.
An important point to notice is that the algorithm is efficient if $\mu$ is large.
One can apply this theorem in different situations, e.g. how many samples for a survey on our chief.

**Definition** A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input $x$ and a parameter $0 < \varepsilon < 1$, the algorithm outputs an $(\varepsilon, 1/4)$-approximation to $V(x)$ in time that is polynomial in $1/\varepsilon$ and the size of the input $x$.

Note: It is easy to obtain an $(\varepsilon,\delta)$-approximation by using an $(\varepsilon, 1/4)$-approximation $O(\ln \frac{1}{\delta})$ times. (Hint: take the median as the estimation and apply Chernoff bound.)

So, to obtain an $(\varepsilon,\delta)$-approximation, we can just work with the above definition and do not worry about $\delta$.

---

## DNF counting (MU 10.2)

This is an example where straightforward application of the Monte Carlo method won't work, but a cleverer application would work.

Usually we consider a CNF (conjunctive normal form) formula which is a conjunction (AND) of the clauses where each clause is a disjunction (OR) of variables.
e.g. $(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_3 \vee \bar{x}_4)$

Here we consider a DNF (disjunctive normal form) formula which is a disjunction of the clauses where each clause is a conjunction of the variables.
e.g. $(x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (\bar{x}_1 \wedge x_3 \wedge x_4)$

We are interested in counting the number of satisfying assignments of a DNF formula.

How difficult is this problem?

Note that a DNF formula is just the negation of a CNF formula with the same number of clauses and variables, by using the DeMorgan's laws (see the above examples).

So, a DNF formula has $W$ satisfying assignments if the corresponding CNF formula has $2^n - W$ satisfying assignments.

If we can solve the exact DNF counting problem, one corollary is that we can *determine* whether a CNF formula is satisfiable (by checking whether $W < 2^n$), which is NP-hard.

Actually, the DNF counting problem (which is equivalent to #SAT) is a #P-complete problem, which is probably harder than NP-complete problems (i.e. counting vs YES/NO).

Thus we are interested in obtaining a FPRAS for the DNF counting problem.

A straightforward approach is to pick many random assignments and count the fraction of satisfying assignments among them.

Apply the Monte Carlo theorem, we get an $(\varepsilon, \delta)$-approximation if $m \geq \frac{3\ln(2/\delta)}{\varepsilon^2 \mu}$. ($m$: # of samples)

In estimating $\pi$, $\mu$ is $\frac{\pi}{4}$ and the number of samples required is small.

For the DNF counting problem, however, $\mu$ could be exponentially small (e.g. $\frac{n^2}{2^n}$), and thus the number of samples required for an $(\varepsilon, \delta)$-approximation is exponential in $n$.

For example, if the number of satisfying assignments is $n$ or $n^2$ or $n^3$, if we only take a polynomial number of samples, with high probability the answer is zero for all these cases.

To get an efficient estimation, the idea is to construct a sample space that include all the satisfying assignments, but at the same time the sample space is not too big.

First, note that finding a satisfying assignment of a DNF formula is easy: just pick a clause and satisfy it (since we only need to satisfy one clause).

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

Let $SC_i$ be the set of assignments that satisfy $C_i$. Note that if $C_i$ has $l_i$ variables, then $|SC_i| = 2^{n-l_i}$ (only one choice for the variables in $C_i$ and arbitrary values for the remaining variables).

Since each satisfying assignment must be contained in $SC_j$ for some $j$, and so $\left|\bigcup_{i=1}^{t} SC_i\right|$ is exactly the number we want to compute approximately.

$|\bigcup_{i=1}^{t} SC_i|$ is counting the number we want to compute approximately

Let $U = \{(i,a) \mid 1 \le i \le t, a \in SC_i\}$. So $|U| = \sum_{i=1}^{t} |SC_i|$. This is an over-estimate of $|\bigcup_{i=1}^{t} SC_i|$, since one satisfying assignment may satisfy more than one clause and is thus counted in $SC_i$ for different $i$.

To avoid the over-counting problem, we define a set $S$ with size $|\bigcup_{i=1}^{t} SC_i|$ as follows:

$S = \{(i,a) \mid 1 \le i \le t, a \in SC_i, a \notin SC_j \text{ for all } j<i\}$.

Given a member in $U$, one can check whether it belongs to $S$ efficiently.

The idea is to approximate $|S|$ by estimating $|S|/|U|$.

The point is that $|S|/|U| \ge 1/t$ since each assignment can satisfy at most $t$ clauses.

So, if we use $U$ as the sample space, then by the Monte Carlo theorem, one can get an $(\varepsilon,\delta)$-approximation of $|S|/|U|$ using at most $\frac{3t\ln(2/\delta)}{\varepsilon^2}$ samples.

Since we can compute $|U|$ efficiently, we can thus get an $(\varepsilon,\delta)$-approximation of $|S|$ as well.

So it remains to generate a uniform sample from $U$.

It is easy. Just pick $i$ with probability $\frac{|SC_i|}{\sum_{j=1}^{t}|SC_j|}$ and then just pick a uniform member of $SC_i$ by setting each variable not in clause $i$ randomly and independently with prob $1/2$ to be true and prob $1/2$ to be false.

So, $\Pr((i,a) \text{ is chosen}) = \Pr(i \text{ is chosen}) \cdot \Pr(a \text{ is chosen}) = \frac{|SC_i|}{|U|} \cdot \frac{1}{|SC_i|} = \frac{1}{|U|}$.

Putting together we have a FPRAS for the DNF counting problem.

---

## Network reliability [1,2]

This is an interesting application of the DNF counting result.

The problem is simple: given an undirected graph where each edge will fail with probability $p$, compute the probability that it is disconnected.

This "simple" problem turns out to be #P-complete, and thus we can only hope for a FPRAS.

The solution of this problem nicely combine several ideas we have learnt.

Let FAIL($p$) be the probability that the graph is disconnected when the failure probability of

an edge is $p$. Note that a graph is disconnected iff all edges of some cut fail.

Let $C$ be the size of a minimum cut in $G$.

First we consider an easy case.

Case 1: $p^C \geq n^{-4}$ This implies that $\text{FAIL}(p) \geq n^{-4}$.

Therefore, if we run the Monte Carlo simulation, then we can get an $(\varepsilon,\delta)$-approximation of $\text{FAIL}(p)$ using at most $\frac{3n^4\ln(2/\delta)}{\varepsilon^2}$ samples since $\mu = \text{FAIL}(p) \geq n^{-4}$.

The remaining case is the interesting case.

Case 2: $p^C < n^{-4}$ The idea is simple and elegant. The proof has two main steps.

① We know from homework 1 that an undirected graph has at most $O(n^{2\alpha})$ cuts with at most $\alpha C$ edges. Since $p^C$ is small, those cuts with more than $\alpha C$ edges are very unlikely to fail (with prob. $p^{\alpha C} \leq n^{-4\alpha}$). Therefore, we can set $\alpha$ to be small (but big enough) and focus only on the cuts with $\leq \alpha C$ edges. In fact, we can generate all of these small cuts in polynomial time using the randomized contraction algorithm.

② The key observation is that checking whether some small cuts will fail can be reduced to the DNF counting problem. Actually this is a more general version where each variable is set to true with probability $p$ (false with prob. $1-p$), and we'll sketch how to handle this.

Now we go into some details for each step.

Enumerating small cuts: Suppose $\alpha$ is given. As done in homework 1, the randomized contraction algorithm can be modified (see [1]) so that each cut with $\leq \alpha C$ edges is the output with probability $\geq n^{-2\alpha}$. If we run this algorithm for $n^{2\alpha}\ln(n^{2\alpha}/\delta)$ times, then a cut with $\leq \alpha C$ edges is not the output of one of these executions is at most $\left(1-\frac{1}{n^{2\alpha}}\right)^{n^{2\alpha}\ln(n^{2\alpha}/\delta)} \leq \frac{\delta}{n^{2\alpha}}$. Therefore, by the union bound, some small cut is not enumerated is at most $\delta$. So, with probability $1-\delta$, all small cuts have been enumerated and the total running time is $\tilde{O}(n^{2\alpha} \cdot T_{\text{contraction}})$ where $T_{\text{contraction}}$ is the running time of the randomized contraction algorithm, which is polynomial in $n$.

Ignoring large cuts  Let $p^c = n^{-(2+k)}$ for some $k > 2$.

$$\Pr(\text{some cut with } \geq \alpha C \text{ fails}) \leq \int_{\beta \geq \alpha} n^{2\beta} p^{\beta C} d\beta \leq \int_{\beta \geq \alpha} n^{-k\beta} d\beta = O(n^{-k\alpha}).$$

By setting $\alpha = 2 + \frac{\ln(\varepsilon / O(1))}{k \ln n}$, then $O(n^{-k\alpha}) = n^{-2k} \cdot \varepsilon \leq \varepsilon \cdot \text{FAIL}(p)$.

Therefore, ignoring all large cuts, this is still a $(1-\varepsilon)$-approximation of FAIL(p).

Furthermore, $\alpha = O(1)$ and thus the above enumeration only requires polynomial time.

Reduction to DNF counting  The reduction is actually straightforward.

Create a variable $x_e$ for each edge $e$.

For each small cut, create a clause $(x_{e_1} \wedge x_{e_2} \wedge \dots \wedge x_{e_m})$.

Set $x_e$ to be true if edge $e$ fails.

Then the formula is true if and only if all the edges in some cut fail.

Therefore, this is just the DNF counting problem when each variable is set to true with probability $p$.

The way to solve this more general DNF counting problem is very similar to the original one.

The algorithm is almost the same (see [1]).

There is one difference in the analysis, because each satisfying assignment may not have the same contribution to the answer. So we could not apply Chernoff's bound the way we did (using the above theorem for the Monte Carlo method). Instead, it is easier to compute the variance of the answer, and then apply Chebyshev's inequality which still gives a good upper bound on the number of samples required (see [1]).

Using a FPRAS for DNF counting can thus obtain a FPRAS for network reliability.

**Open Question**: Is there a FPRAS for computing the probability that $s$ and $t$ is disconnected in the resulting graph where each edge fails with probability $p$?

---

## Approximate sampling and approximate counting (MU 10.3)

We saw how to use sampling to do approximate counting.

We will show that approximate sampling is enough for approximate counting.

The reduction is quite general, but we will use the example of counting independent sets of a graph to illustrate the technique.

**Definition** Let $w$ be the random output of a sampling algorithm for a finite sample space $\Omega$. The sampling algorithm generates an $\varepsilon$-uniform sample of $\Omega$, if for any element $z$ in $\Omega$, $\frac{1}{2}\sum_{z\in\Omega}\left|\Pr(w=z)-\frac{1}{|\Omega|}\right| \le \varepsilon$, that is, the total variation distance between the output distribution and the uniform distribution is at most $\varepsilon$. A sampling algorithm is a <u>fully polynomial almost uniform sampler</u> (FPAUS) for a problem if, given an input $x$ and a parameter $\varepsilon>0$, it generates an $\varepsilon$-uniform sample of $\Omega(x)$ and runs in time that is polynomial in $\ln\varepsilon^{-1}$ and the size of the input $x$.

Our goal is to show that, given a FPAUS for independent sets, we can construct a FPRAS for counting the number of independent sets.

Let $e_1,\dots,e_m$ be the edges of $G$. Let $E_i$ be the first $i$ edges and $G_i=(V,E_i)$.

Let $\Omega(G_i)$ be the number of independent sets in $G_i$.

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|}\cdot\frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|}\cdot\ \dots\ \cdot\frac{|\Omega(G_1)|}{|\Omega(G_0)|}\cdot|\Omega(G_0)|$$

Since $G_0$ has no edges, every subset is an independent set, and so $|\Omega(G_0)|=2^n$.

Let $r_i=\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}$ for $1\le i\le m$. Then $|\Omega(G)|=2^n\prod_{i=1}^{m} r_i$

Later we will compute approximation $\tilde{r}_i$ for $r_i$, and output $2^n\prod_{i=1}^{m}\tilde{r}_i$ as approximation to $|\Omega(G)|$

To bound the error, we need to bound the ratio $R=\prod_{i=1}^{m}\frac{\tilde{r}_i}{r_i}$.

**Lemma** Suppose $\tilde{r}_i$ is an $(\varepsilon/2m,\ \delta/m)$-approximation to $r_i$ for all $1\le i\le m$.

Then $\Pr(|R-1|\le\varepsilon)\ge 1-\delta$. (This implies an $(\varepsilon,\delta)$-approximation to $|\Omega(G)|$.)

**Proof** Use the definition, union bound, simple calculations. See Lemma 10.3 of MU. ∎

To obtain a $(\varepsilon/2m,\ \delta/m)$-approximation for $r_i$, we estimate it by a Monte Carlo algorithm that uses the FPAUS for sampling independent sets.

The idea is to (approximately) sample independent sets in $G_{i-1}$ and compute the fraction of these sets that are also independent in $G_i$.

Let $e_i=(u,v)$. The only independent sets in $G_{i-1}$ that are not independent in $G_i$ are

those that contain $u$ and $v$.

An important point is that $r_i \geq \frac{1}{2}$. To see this, map each independent set $I$ in $\Omega(G_{i-1}) - \Omega(G_i)$ to the independent set $I - v$ in $G_i$. Note that each independent set in $\Omega(G_i)$ is mapped by at most one independent set in $\Omega(G_{i-1}) - \Omega(G_i)$. This implies that $|\Omega(G_i)| \geq |\Omega(G_{i-1}) - \Omega(G_i)|$, and thus

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

Therefore, if we have a uniform sampler for independent sets in $G_{i-1}$, then we can get a good approximation $\tilde{r}_i$ to $r_i$ using only a few samples by using the result about Monte Carlo method early in this notes.

The difference here is that we only have a FPAUS for sampling independent sets. But not surprisingly, using a good enough sampler will give us a good enough approximation to $r_i$.

There are two errors to bound:

- Since we are only using a FPAUS, the expected value may not equal to $r_i$. By using an $(\varepsilon/6m)$-uniform sampler, we can show that $|E[\tilde{r}_i] - r_i| \leq \frac{\varepsilon}{6m}$.
- By taking only some samples, we can only get an approximation to $E[\tilde{r}_i]$. Since $r_i$ is big ($\geq 1/2$), $E[\tilde{r}_i]$ is big and thus the standard Monte Carlo method would work to show that $\tilde{r}_i$ is very close to $r_i$ by taking $O(m^2 \varepsilon^{-2} \ln \frac{2m}{\delta})$ samples.

Combining these, we can show that $\tilde{r}_i$ is a good approximation to $r_i$.

For detailed calculations see Lemma 10.4 of MU.

**Theorem** Given a FPAUS for independent sets, one can construct a FPRAS for counting independent sets.

This reduction works for many "self-reducible" problems, e.g. counting number of graph colorings.

---

## Markov chain Monte Carlo method (MU 10.4)

If there is an approximate sampler, then we can do approximate counting.

But the hard part is to construct an approximate sampler.

Here is where Markov chain comes into the picture.

The basic idea is to define a finite, irreducible, aperiodic Markov chain whose set

of states is the sample space (e.g. each node corresponds to an independent set) and whose stationary distribution is the required distribution (e.g. uniform distribution).

Once we have constructed this Markov chain, we can do a random walk on the states. We know that it will converge to the stationary distribution. So, after a long enough time, the distribution will be close to the stationary distribution and we get an approximate sample.

For this approach to be efficient, we need to show that the convergence rate is fast and each step can be implemented efficiently. We will come back to these later.

Now we consider how to construct a Markov chain with the required stationary distribution. First we state a useful lemma.

**Lemma** Given a finite, irreducible, aperiodic Markov chain with transition matrix $P$. If $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ satisfies $\sum_{i=1}^{n} \pi_i = 1$ and $\pi_i P_{i,j} = \pi_j P_{j,i}$ for all $i, j$, then $\vec{\pi}$ is the stationary distribution according to $P$.

**Proof** $\sum_{i=1}^{n} \pi_i P_{i,j} = \sum_{i=1}^{n} \pi_j P_{j,i} = \pi_j$ for all $j$. So $\vec{\pi}$ must be the unique stationary distribution. ∎

The condition $\pi_i P_{i,j} = \pi_j P_{j,i}$ are called <u>time reversible</u>.

If the underlying graph of the Markov chain is an undirected graph, then we know that the stationary probability of a vertex is proportional to its degree. Therefore, for the stationary distribution to be the uniform distribution, we just need to make sure the graph is regular. If not regular we just add some self-loops to make it regular.

**Lemma** For a finite state space $\Omega$ and neighborhood structure $\{N(x) \mid x \in \Omega\}$, let $N = \max_{x \in \Omega} |N(x)|$ and $M \geq N$. Consider a Markov chain where

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

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

**Proof:** If $\pi_i = \pi_j$ for all $i, j$, then $\pi_i P_{i,j} = \pi_j P_{j,i}$. So it follows from the previous lemma. ∎

For the independent set problem, one can apply this lemma to prove that the following Markov chain has a uniform stationary distribution.

① Start from an arbitrary independent set $X_0$ (e.g. empty set)

② To compute $X_{i+1}$

ⓐ choose $v$ uniformly at random from $G$.

ⓑ If $v \in X_i$ then $X_{i+1} = X_i - v$

ⓒ If $v \notin X_i$ and $X_i + v$ is independent, then $X_{i+1} = X_i + v$.

This chain is irreducible because each state can reach every other state by first deleting vertices and then adding vertices. It is aperiodic because it has a self-loop if the graph has at least one edge. By the above lemma the stationary distribution is the uniform distribution.

## The Metropolis algorithm (MU 10.4.1)

This is a general construction of a Markov chain with any stationary distribution.

**Lemma** Under the same conditions as in the previous lemma, set

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

Then $\vec{\pi}$ is the stationary distribution.

**Proof:** Assume $\pi_x \leq \pi_y$. Then $P_{x,y} = 1$ and $P_{y,x} = \pi_x / \pi_y$. This is time-reversible. ∎

For example, suppose we want a Markov chain for independent sets with stationary probability proportional to $\lambda^{|I|}$ where $\lambda$ is a constant and $|I|$ is the size of the independent set.

By the above lemma we just need to modify the Markov chain as follows:

(b) if $v \in X_i$, set $X_{i+1} = X_i - v$ with probability $\min\{1, 1/\lambda\}$.

(c) if $v \notin X_i$, set $X_{i+1} = X_i + v$ if it is independent with probability $\min\{1, \lambda\}$.

Then it is easy to verify that the stationary probability is proportional to $\lambda^{|I|}$.

---

## Coupling (MU 11)

A powerful method to bound the convergence rate of Markov chains.

Last week we say that if two Markov chains go to the same state, then one cannot distinguish the two distributions afterwards because Markov chains don't remember the history.

Coupling is a method to make this argument formal.

Before stating the coupling method, let us first recall the definition of mixing time.

## Variation distance and mixing time (MU 11.1)

The total variation distance is to measure how close is two probability distributions.

**Definition** The total variation distance between two probability distributions $\vec{p}$ and $\vec{q}$ is defined as $\|\vec{p}-\vec{q}\| = \frac{1}{2}\sum_{x\in S}|p(x)-q(x)|$ where $S$ is the state space.

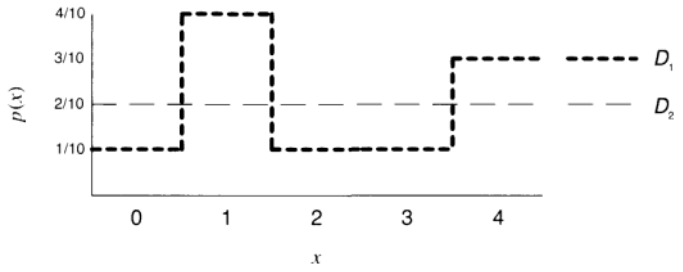
(Last week we used the notation $d_{TV}(\vec{p},\vec{q})$ to say the same thing, this week we follow the book.)

The following is an equivalent characterization of the total variation distance.

**Lemma** For any $A\subseteq S$, let $p(A)=\sum_{x\in A}p(x)$.

Then $\|\vec{p}-\vec{q}\| = \max_{A\subseteq S}|p(A)-q(A)|$

**Proof** (by picture from MU)




The area where $\vec{p}$ is "above" $\vec{q}$ is equal to the area where $\vec{p}$ is "below" $\vec{q}$.

The sum of these two areas is equal to $\sum_{x\in S}|p(x)-q(x)|$.

Let $A$ be the set of those values correspond to the area above.

Then $|p(A)-q(A)| = \frac{1}{2}\sum_{x\in S}|p(x)-q(x)| = \|\vec{p}-\vec{q}\|$. ∎

Mixing time is the time when the probability distribution is close to the stationary distribution regardless of the initial distribution.

**Definition** Let $\vec{\pi}$ be the stationary distribution. Let $\vec{p}_x^{\,t}$ be the probability distribution after $t$ time steps starting at state $x$.

Define $\Delta_x(t) = \|\vec{p}_x^{\,t}-\vec{\pi}\|$ and $\Delta(t)=\max_{x\in S}\Delta_x(t)$.

Also define $\tau_x(\varepsilon)=\min\{t:\Delta_x(t)\le\varepsilon\}$ and $\tau(\varepsilon)=\max_{x\in S}\tau_x(\varepsilon)$.

A Markov chain is rapidly mixing if $\tau(\varepsilon)$ is polynomial in $\ln(1/\varepsilon)$ and the size of the problem.

size of the problem.

## Coupling (MU 11.2)

**Definition** 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:

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

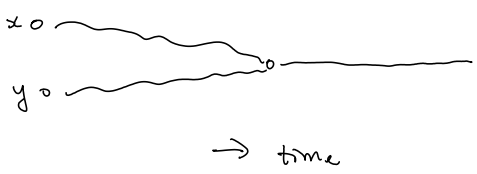
In words, coupling is a joint random process such that each Markov chain behaves exactly as the original Markov chain, although the moves of the two processes could be independent.

The power of the method is to allow us to design a joint process to

① bring the two copies to the same state quickly

② keep them in the same state by having the two chains make identical moves once they are in the same state.

Pictorially the method tries to do the following




(1) tries to minimize the time they merge

(2) makes sure once they merged they will not separate

Since both behave like the original Markov chains, we can thus argue that the distributions of $\vec{X}_t$ and $\vec{Y}_t$ are the same after they merged.

An upper bound of the time of merging is an upper bound on the mixing time.

**Lemma (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 $x, y \in S$,

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

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

**Proof** Consider the coupling where $Y_0$ is chosen according to the stationary distribution and $X_0$ takes on arbitrary value. For the given $T$ and $\varepsilon$ and for any $A \subseteq S$,

$$\begin{aligned}\Pr(X_T \in A) &\geq \Pr((X_T = Y_T) \cap (Y_T \in A)) \\ &= 1 - \Pr((X_T \neq Y_T) \cup (Y_T \notin A))\end{aligned}$$

$$\geq 1 - \Pr(X_T \neq Y_T) - \Pr(Y_T \in A) \quad \text{(by the union bound)}$$

$$= \Pr(Y_T \in A) - \Pr(X_T \neq Y_T)$$

$$\geq \Pr(Y_T \in A) - \varepsilon \quad \text{(by the assumption)}$$

$$= \pi(A) - \varepsilon$$

Similary, we can argue $\Pr(X_T \in A) \geq \pi(S-A) - \varepsilon$ which implies $\Pr(X_T \in A) \leq \pi(A) + \varepsilon$.

It follows that $\max_{x,A} \left| p_x^T(A) - \pi(A) \right| \leq \varepsilon$. This implies $\tau(\varepsilon) \leq T$. ∎

---

## Stationary distribution

Now we can explain why any finite, irreducible, aperiodic Markov chain will converge to the same distribution.

The coupling is easy: before they meet they run independently, and after they meet they always make the same move. It is easy to check that both chains behave like the original one.

Recall that for any finite, irreducible and aperiodic Markov chain, there exists $N$ such that

$P_{i,j}^n > 0$ for all $i,j$ and for all $n \geq N$.

Let $\delta = \min_{i,j} \{ P_{i,j}^N \}$.

Then with probability at least $\delta$, the two Markov chains will merge after $N$ steps.

Then, after $t = kN$ steps, the two Markov chains do not merge with probability $\leq (1-\delta)^k$.

This tends to zero when $t$ tends to infinity, and so any initial distribution will converge to the stationary distribution.

---

## Shuffling cards (MU 11.2.1)

Consider the following method for shuffling $n$ cards. In each step we pick a random card and put it on the top of the deck. How good is this shuffling process?

This is a Markov chain whose state space is the set of all permutations of the $n$ cards.

It is not difficult to check that this chain is irreducible and aperiodic, and also the stationary distribution is the uniform distribution.

To bound the mixing time, we consider the following coupling:

- choose a position $j$ uniformly at random from 1 to $n$, and then move the $j$-th card

to the top in the first chain. Denote the card value by C.

- move the card with value C to the top in the second chain.

This is a valid coupling, since the probability that a card is moved to top is $1/n$.

With this coupling, once a card C is moved to the top, then it will be in the same position in both chains.

So, the two Markov chains will be coupled if every card has been moved to the top at least once.

This is just the coupon collector problem.

Hence, after $n \ln n + n \ln(\frac{1}{\varepsilon})$ steps, the probability that the two chains have not coupled is at most $\varepsilon$.

---

## Random walk on the hypercube (MU 11.2.2)

Recall that an n-dimensional hypercube is a graph with $2^n$ nodes, each vertex corresponds to an n-bit string, and two nodes have an edge iff their corresponding bit strings differ in exactly one bit.

Starting from an arbitrary node, we do a random walk by choosing a random position and set its value to one with prob $1/2$ and zero with prob $1/2$.

It is easy to check that this Markov chain is irreducible and aperiodic, and that the stationary distribution is the uniform distribution.

To bound the mixing time, we consider a simple coupling: both chains choose the same position and set the same value. It is clear that it is a valid coupling.

With this coupling, once the i-th coordinate has been chosen, it will always be the same in the two chains.

Once every coordinate has been chosen at least once, the two chains coupled.

Again this is just the coupon collector problem, and thus $\tau(\varepsilon) \leq n \ln(n/\varepsilon)$.

---

## Independent sets of fixed size (MU 11.2.3)

This example is more interesting as the distance between two chains may increase or decrease.

Consider a Markov chain whose states are all independent sets of size exactly k in a graph:

① Choose a vertex $v$ in $X_t$ uniformly at random and a vertex $w \in V$ uniformly at random.

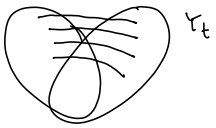
② if $w \notin X_t$ and $X_t - v + w$ is independent, then $X_{t+1} = X_t - v + w$; otherwise $X_{t+1} = X_t$.

Let $n$ be the number of vertices and $\Delta$ be the maximum degree of any vertex.

We will show that this Markov chain is rapidly mixing if $k \le n/(3\Delta+3)$.

It is an exercise to show that this chain is irreducible and aperiodic, and the stationary distribution is the uniform distribution.

The coupling will require an arbitrary bijection $M$ between the vertices of $X_t - Y_t$ and the vertices in $Y_t - X_t$.



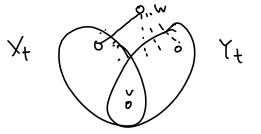

For the first chain, choose a random $v \in X_t$ and a random $w \in V$ and apply the move. So the first Markov chain behaves like the original one.

For the second chain, if $v \in Y_t$ then use the same pair of vertices $v$ and $w$ and make the same move; otherwise if $v \notin Y_t$, make the move with $M(v)$ and $w$.

This also behaves like the original Markov chain, since each pair of vertices with $v \in Y_t$ and $w \in V$ is chosen with probability $1/kn$.

Let $d_t = |X_t - Y_t|$ measures the difference between $X_t$ and $Y_t$. Clearly $d_t$ can change by at most one. We will show that $d_t$ is more likely to decrease than increase.

In order for $d_{t+1} = d_t + 1$. It must be the case that $v \in X_t \cap Y_t$, and that $w$ is adjacent to some vertex in $X_t - Y_t$ but not adjacent to any vertex in $Y_t - X_t$, or vice versa. In this case, a move is made in $Y$ but not in $X$.




Thus, $w$ must be a vertex or a neighbor of a vertex in the set $(X_t - Y_t) \cup (Y_t - X_t)$. It follows that $\Pr(d_{t+1} = d_t + 1 \mid d_t > 0) \le \frac{k - d_t}{k} \cdot \frac{2d_t(\Delta+1)}{n}$.

choosing $v$ the in $X_t \cap Y_t$ — choosing $w$ in $N(X_t \cup Y_t)$

Similarly, for $d_{t+1} = d_t - 1$, it is sufficient if $v \notin Y_t$ and $w$ is neither a vertex nor a neighbor of a vertex in $X_t \cup Y_t - \{v, M(v)\}$.


Note that $|X_t \cup Y_t| = k + d$, and thus

$$\Pr(d_{t+1} = d_t - 1 \mid d_t > 0) \ge \frac{d_t}{k} \cdot \frac{n - (k + d - 2)(\Delta+1)}{n}$$

So, $E[d_{t+1} \mid d_t] = \Pr(d_{t+1} = d_t + 1) \cdot (d_t + 1) + (d_t - 1) \cdot \Pr(d_{t+1} = d_t - 1)$

So, $E[d_{t+1} \mid d_t] = \Pr(d_{t+1} = d_t + 1) \cdot (d_t + 1) + (d_t - 1) \cdot \Pr(d_{t+1} = d_t - 1)$

$$\leq d_t + \left(\frac{k - d_t}{k}\right)\left(\frac{2 d_t (\Delta+1)}{n}\right) - \left(\frac{d_t}{k}\right)\left(\frac{n - (k + d_t - 2)(\Delta+1)}{n}\right)$$

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

$$\leq d_t \left(1 - \frac{n - (3k-3)(\Delta+1)}{kn}\right)$$

By induction, $E[d_t] \leq d_0 \left(1 - \frac{n - (3k-3)(\Delta+1)}{kn}\right)^t$

Since $d_0 \leq k$ and $d_t$ is a nonnegative integer, we have

$$\Pr(d_t \geq 1) \leq E[d_t] \leq k \left(1 - \frac{n - (3k-3)(\Delta+1)}{kn}\right)^t \leq k e^{-t(n-(3k-3)(\Delta+1))/kn}$$

Since $k \leq n/3(\Delta+1)$, $n - (3k-3)(\Delta+1) = n - (k-1)3(\Delta+1) \geq 1$ and thus $\Pr(d_t \geq 1) \to 0$.

It is easy to verify that $\tau(\varepsilon) \leq \frac{kn \ln(k\varepsilon^{-1})}{n - (3k-3)(\Delta+1)}$ which is polynomial in $n$ and $\ln(\varepsilon^{-1})$.

So the Markov chain is rapidly mixing

---

## Random spanning trees [3,4]

Last week we mentioned that one can generate a uniform spanning tree in cover time.

Here we use coupling to prove this result.

Let $G$ be the undirected graph. Define $\overleftrightarrow{G}$ be the directed graph such that each edge $uv$ in $G$ is replaced by two directed edges $uv$ and $vu$.

A directed spanning tree rooted at $r$ is a subset of $|V|-1$ edges where each vertex $v \neq r$ has exactly one outgoing edge, and there is a path from any vertex to $r$.


Note that there is a bijection between the set of spanning trees and the set of arborescences rooted at $r$.

The following Markov chain is defined on the set of arborescences.

① Start on an arborescence rooted at $r$. Start at vertex $r$.

② Let the current vertex be $u$. Choose a random vertex $v$. If $v \neq u$ and $uv \in \overleftrightarrow{G}$, then $X_{t+1} = X_t + uv - vw$ where $vw$ is the unique outgoing edge of $v$ in $X_t$.

(So, vertex $v$ becomes the root, and the current vertex is always the root.)

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

It can be shown that the chain is irreducible, aperiodic and the stationary distribution can be made to be the stationary distribution.

To bound the mixing time, we consider the coupling that the two chains always make the same move. In particular, the current vertex of the two chains will always be the same.

Once two chains reach a vertex $x$, then the outgoing edge of $x$ will always be the same in the two chains.

Therefore, once every vertex is visited (and return to $r$), then the two arborescences agree completely and will be the same.

The expected time to visit every node at least once is the cover time.

By Markov's inequality, $\Pr(X_t \neq Y_t \mid X_0, Y_0) \leq \frac{1}{4}$ in $4T_{cover}$.

There is also an extra factor due to the self-loops, but the overall running time is still a polynomial in $n$.

Forget about the directions gives an almost random spanning tree.

---

## Spectral gap, conductance, and more

We briefly mention something important that we have not discussed and hope to discuss in a later lecture.

Another technique to bound the mixing time of a Markov chain is to look at the eigenvalues of the matrix. It can be shown that a Markov chain is rapidly mixing if the difference between the first and the second eigenvalues, known as the spectral gap, is large.

The Cheeger's inequality shows that the spectral gap is large "iff" the conductance is large, where the conductance of a graph is defined as $\phi(G) = \min_{\substack{S \subseteq V \\ |S| \leq |V|/2}} \frac{|\delta(S)|}{d \cdot |S|}$,

assuming the graph is a $d$-regular undirected graph.

An important result in this area is a FPAUS for perfect matchings in bipartite graphs.

To show that the chain is rapidly mixing, the proof shows that the underlying graph has large conductance by showing that there is a multicommodity flow with "low" congestion. For details read Motwani-Raghavan 11.3.

---

## References


[1] Lecture notes by Eric Vigoda "On an FPRAS for network reliability".

[2] Karger. A randomized fully polynomial time approximation scheme for the all terminal network reliability problem.

[3] Lecture notes by Eric Vigoda "Bounding the mixing time via coupling"

[4] Propp, Wilson. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph