[Solution of homework 3]: More on Markov chains

Problem 1 (FTMC for countably infinite chains)

Recall that the fundamental theorem of Markov chains states that [F] + [A] + [I] implies [S]+[U]+[C]. In this problem, we will develop its generalization, namely, [PR] + [A] + [I] implies [S]+[U]+[C] (Please refer to our lecture notes for the meaning of these abbreviations). We assume Ω is the state space of the Markov chain, which can be finite or countably infinite. Let $P \in [0,1]^{\Omega\times\Omega}$ be the transition function of the chain. That is, for every $i, j \in \Omega$, $P(i,j) = \mathbf{Pr}\left[X_{t+1} = j \mid X_t = i\right]$. Assume P has the properties of [PR], [A] and [I].

1. Prove that this is indeed a generalization. That is, "[PR] + [A] + [I] implies [S]+[U]+[C]" implies "[F] + [A] + [I] implies [S]+[U]+[C]".
   Proof.
   We only need to prove that "[F]+[I]" implies "[PR]".
   Let $h_{ji}^m = \mathbf{P}_j\left(X_t = i \text{ for some } t \in [m]\right)$ be the probability that the chain visited i in m steps. Because of "[F]" and "[I]", there exists a $n \in \mathbb{N}$ and a $\delta > 0$ such that $h_{ji}^n \geq \delta$ for any $j \in \Omega$. Moreover,
   $$1 - h_{ii}^t \leq (1-\delta)^{\lfloor t/n \rfloor}, \forall t \in \mathbb{N}.$$
   Hence
   $$\mathbf{E}_i\left[T_i\right] = \sum_{t=1}^{\infty} \mathbf{P}_i(T_i \geq t) = \sum_{t=1}^{\infty} 1 - h_{ii}^t \leq \sum_{t=0}^{\infty} (1-\delta)^{\lfloor t/n \rfloor} = \frac{n}{\delta} < \infty.$$
   Therefore, "[F]+[A]+[I]" $\Rightarrow$ "[PR]+[A]+[I]" $\Rightarrow$ "[S]+[U]+[C]".
2. We now define another Markov chain on Ω^2. For each pair $(i,j) \in \Omega^2$, the chain moves to $(i',j') \in \Omega^2$ following P coordinate-wise and independently. (That is, $\mathbf{Pr}\left[(X_{t+1}, Y_{t+1}) = (i',j') \mid (X_t, Y_t) = (i,j)\right] = P(i,i') \cdot P(j,j')$.) Prove that in this chain, for any $i, j, k \in \Omega$, it holds that $\mathbf{Pr}_{(i,j)}\left[T_{(k,k)} < \infty\right] = 1$.
   Proof.
   Let Q be the product chain.
   First we need to prove that Q is [I]. Since P is [A]+[I], for any $i, j \in \Omega$, there exists an n

such that $P^t(i,j) > 0, \forall t \geq n$. Thus for any i,j,k,l, we denote by n_{ik} and n_{jl} the smallest number such that $P^{n_{ik}}(i,k) > 0$ and $P^{n_{jl}}(j,l) > 0$ respectively; then let $n = \max\{n_{ik}, n_{jl}\}$ and we have $Q^n((i,j),(k,l)) = P^n(i,k) \cdot P^n(j,l) > 0$.
Now we prove Q is [PR]. Let π denote the unique stationary distribution of P. We have

$$\sum_{i,j} \pi(i)\pi(j) Q((i,j),(k,l)) = \sum_{i,j} \pi(i)P(i,k) \cdot \pi(j)P(j,l) = \pi(k)\pi(l)$$

and

$$\sum_{i,j} \pi(i)\pi(j) = 1$$

which means $\pi \otimes \pi$ is a stationary distribution for Q. Since Q is [I] and has a stationary distribution, thus $\pi(i)\pi(j) = \frac{1}{\mathbf{E}_{(i,j)}[T_{(i,j)}]} > 0$. So $\mathbf{E}_{(i,j)}\left[T_{(i,j)}\right] < \infty$, which implies $\mathbf{Pr}_{(i,j)}\left[T_{(k,k)} < \infty\right] = 1$.

3. Use above to prove the FTMC for countably infinite chains.
Proof.
We have proved that [PR]+[I]$\Rightarrow$ [S]+[U] in lecture notes, and now we need to prove the Markov chain will converge.
Let π be the stationary distribution for P and μ_0 is an arbitrary distribution over Ω. $\{X_t\}$ and $\{Y_t\}$ are two Markov chains with transition kernel P, and $X_0 \sim \pi, Y_0 \sim \mu$.
Now construct a coupling ω_{t+1} for π and $\mu_{t+1} = \mu_0^\top P^{t+1}$.
If $X_t = Y_t$, then $X_{t'} = Y_{t'}, t' > t$; else X_{t+1} and Y_{t+1} evolve independently according to P.
Therefore,

$$D_{TV}(\mu_t, \pi) \leq \mathbf{Pr}[X_t \neq Y_t].$$

Since $\forall i,j,k$, $\mathbf{P}_{(X_0=i, Y_0=j)}(T_{(k,k)} < \infty) = 1$(by the above result),

$$\lim_{t\to\infty} \mathbf{Pr}_{(X_t,Y_t)\sim\omega_t}[X_t \neq Y_t] = 0,$$

which means $\lim_{t\to\infty} D_{TV}(\mu_t, \pi) = 0$. So [PR]+[A]+[I]$\Rightarrow$ [S]+[U]+[C].

Problem 2 (A Randomized Algorithm for 3-SAT)

Recall the randomized algorithm we developed for 2-SAT. In the problem, we apply it to those 3-SAT instances. Since 3-SAT is NP-complete in general, we cannot expect it to terminate in polynomial-time and output the correct answer with high probability. However, it is still better than the brute-force algorithm sometimes. Let n be the number of variables of the input formula.

1. In the 2-SAT algorithm shown in the class, if we repeat the random flipping operation for $100n^2$ times, then the algorithm outputs the correct answer with probaiblity at least $(1-\frac{1}{100})$. Consider the following way to boost the correct probability. The algorithm only repeats the random flipping operation for $2n^2$ times. If it outputs a satisfying assignment, then we just output as it is. Otherwise, we run the algorithm again. Repeat this for 50 times (So the total number of iterations is still $100n^2$). If all these algorithms claim the formula is not satisfiable, then we output "not satisfiable". What is the probability of correctness of our new algorithm?
   Solution.
   For $i \in [50]$, let $\{X_t^i\}$ and $\{Y_t^i\}$ denote the i-th Markov chains similar to $\{X_t\}$ and $\{Y_t\}$ in the lecture note, respectively.
   Then

$$\begin{aligned}\mathbf{Pr}\left[\text{the algorithm is correct}\right] &= \mathbf{Pr}\left[\exists i \in [50], t \in [2n^2] \cup \{0\} s.t. X_t^i = n\right] \\ &\geq \mathbf{Pr}\left[\exists i \in [50], t \in [2n^2] \cup \{0\} s.t. Y_t^i = n\right]\end{aligned}$$

   Thus

$$\begin{aligned}1-\mathbf{Pr}\left[\exists i \in [50], t \in [20^2] \cup \{0\} s.t. Y_t^i = n\right] &= \Pi_i \mathbf{Pr}\left[T_{Y_0^i \to n} > 20n^2\right] \\ &\leq \left(\frac{\mathbf{E}\left[T_{Y_0^1 \to n}\right]}{2n^2}\right)^{50} = \frac{1}{2^{50}}.\end{aligned}$$

   So the probability of correctness of our new algorithm is $1-\frac{1}{2^{50}}$.
2. Now we apply our algorithm on a 3-SAT instance (Now in each step, if σ_t is not satisfiable, we choose an unsatisfied clause, pick one of its three literals uniformly at random, and flip its value). Assume the same notations in the class. Prove that $\mathbf{Pr}\left[X_{t+1} = X_t + 1\right] \geq \frac{1}{3}$ and $\mathbf{Pr}\left[X_{t+1} = X_t - 1\right] \leq \frac{2}{3}$.
   Solution.
   WLOG assume we choose the clause $c = x \vee y \vee z$ in round t and $\sigma_t(x) = \sigma_t(y) = \sigma_t(z) = \text{false}$. Because c is satisfying under σ, we consider the following three conditions:

- Only one literal is assinged true, WOLG let $\sigma(x)$=true and $\sigma(y) = \sigma(z)$=false. Then

$$\mathbf{Pr}\left[X_{t+1} = X_t + 1\right] = \mathbf{Pr}\left[\text{flip } x\right] = \frac{1}{3}$$

- There are two literals are assigned true. Then $\mathbf{Pr}\left[X_{t+1} = X_t + 1\right] = \frac{2}{3}$.

- There are three literals are assigned true. Then $\mathbf{Pr}\left[X_{t+1}=X_t+1\right]=1$.
  Because $\mathbf{Pr}\left[X_{t+1}=X_t\right]=0$, thus
  $\mathbf{Pr}\left[X_{t+1}=X_t-1\right]=1-\mathbf{Pr}\left[X_{t+1}=X_t+1\right]\leq\frac{2}{3}$.

3. Prove that in order for our algorithm to be correct with probability 0.99, we need to repeat the random flipping operations for $\Theta(2^n)$ times.
   Solution.
   We use the same $\{X_t\}_{t\geq 0}$ as in the lecture note. Define the 1-D random walk $\{Y_t\}_{t\geq 0}$ on $[n]\cup\{0\}$ that $Y_0=X_0$ and for $Y_t\notin\{0,1\}$,

$$Y_{t+1}=\begin{cases}Y_t+1, & \text{w.p. } \frac{1}{3}\\ Y_t-1, & \text{w.p. } \frac{2}{3}\end{cases}$$

   If $Y_t=0, Y_{t+1}=Y_t+1$ w.p. 1 and if $Y_t=n$, then $Y_{t+1}=Y_t-1$ w.p. 1.
   Now we need to calculate the expectation of $T_{i\to n}$, the first hitting time of n from i.
   For $i>0$, we have

$$\mathbf{E}\left[T_{i\to i+1}\right]=1+\frac{2}{3}\mathbf{E}\left[(T_{i-1\to i}+T_{i\to i+1})\right].$$

   Thus $\mathbf{E}\left[T_{i\to i+1}\right]=3+2\mathbf{E}\left[T_{i-1\to i}\right]\Rightarrow\mathbf{E}\left[T_{i\to i+1}\right]+3=2\left(\mathbf{E}\left[T_{i-1\to i}\right]+3\right)$.
   Note that $T_0\to 1=1$, so

$$\mathrm{E}\left[T_{i\to n}\right]=\sum_{k=i}^{n-1}\mathbf{E}\left[T_{k\to k+1}\right]=\sum_{k=i}^{n-1}2^k\times 4-3=4(2^n-2^i)-3(n-i)\leq 4\times 2^n.$$

   Therefore, if we repeat the random flipping operations for 400×2^n times, our algorithm will be correct with probability 0.99.

4. Suppose we start with $X_0=n-i$ for some $i>0$. Can you find a good lower bound for the probability $\mathbf{Pr}\left[\exists t\in[1,3n]:X_t=n\right]$?
   Solution.
   Define the 1-D random walk $\{Z_t\}_{t\geq 0}$ on $\mathbb{Z}$ that $Z_0=X_0=n-i$ and

$$Z_{t+1}=\begin{cases}Z_t+1, & \text{w.p. } \frac{1}{3}\\ Z_t-1, & \text{w.p. } \frac{2}{3}\end{cases}$$

   Here we can easily use the same randomness to couple the distribution of $X_t, Y_t \text{ and } Z_t$ such that if $\{Z_t\}$ never hits n , then $Z_t\leq Y_t\leq X_t\leq n$.
   Hence

$$\begin{aligned}\mathbf{Pr}\left[\exists t \in [1,3n] : X_t = n\right] &\geq \mathbf{Pr}\left[\exists t \in [1,3n] : Y_t = n\right] \geq \mathbf{Pr}\left[\exists t \in [1,3n] : Z_t = n\right] \\ &\geq \mathbf{Pr}\left[Z_{3i} = n\right] = \binom{3i}{i}\left(\frac{1}{3}\right)^{2i}\left(\frac{2}{3}\right)^{i} \approx \frac{a \cdot 2^{-i}}{\sqrt{i}},\end{aligned}$$

where a is a constant.

5. Now we consider how to improve the performance of the algorithm. Suppose the input formula is satisfiable. The following observation is crucial. The X_t is more likely to decrease, and therefore, when t is large, it is more likely to be close to 0. This observation suggests us that we should not repeat the flipping process for too long. As a result, we only repeat the flipping process for $3n$ times. Suppose we start with some σ_0 which is uniform at random from all 2^n assignments of the variables. What is the probability that the algorithm outputs a satisfying assignment?
   Solution.
   Let $p_i = \frac{a \cdot 2^{-i}}{\sqrt{i}}$. Then

$$\begin{aligned}\mathbf{Pr}\left[\text{ Algorithm outputs a satisfying assignment}\right] &\geq \sum_{i=1}^{n} \Pr[X_0 = n - i] \cdot p_i \\ &= \sum_{i=1}^{n} \binom{n}{i} 2^{-n} \cdot \frac{a \cdot 2^{-i}}{\sqrt{i}} \\ &\geq \frac{a}{\sqrt{n}} 2^{-n} \sum_{i=0}^{n} \binom{n}{i} 2^{-i} \\ &= \frac{a}{\sqrt{n}} \left(\frac{3}{4}\right)^{n}\end{aligned}$$

6. Design an algorithm to find a solution for 3-SAT with probability 0.99. Determine the smallest constant $c \in (1,2]$ so that your algorithm terminates in $n^{O(1)} \cdot c^n$ time.
   Solution.
   Repeat the process in “5” for m times, and we have

$$\mathbf{Pr}\left[\text{Algortihm outputs a wrong results}\right] \leq \left(1 - \frac{a}{\sqrt{n}}\left(\frac{3}{4}\right)^{n}\right)^{m} = 0.01.$$

We can calculate $m \approx \ln 100 \times \frac{\sqrt{n}}{a}\left(\frac{4}{3}\right)^{n}$. Thus the algorithm terminates in $3 \ln 100 \times \frac{n^{3/2}}{a}\left(\frac{4}{3}\right)^{n}$ and the constant $c = \frac{4}{3}$.