Assignment 10

15-359 Probability and Computing
Name: Shashank Singh
Email: sss1@andrew.cmu.edu
Section: B
Due: Wednesday, April 16, 2012

Problem 9.7 Walks on Undirected Weighted Graphs

Let V be the set of edges in the graph, and let E be the set of edges in the graph. For each node i, let

$$x_i = \frac{\sum_{j \in V} w_{ij}}{2 \sum_{(u,v) \in E} w_{uv}}$$

(where $w_{ij} = 0$ when $(i, j) \notin E$). Since each edge in E corresponds to two vertices, and the numerator of each x_i is the sum of the weights of its incident edges, $\sum_{i \in V} x_i = 1$. Furthermore, $\forall i, j \in V$,

$$x_i P_{ij} = \frac{w_{ij}}{\sum_{(i,j) \in E} w_{i,j}} = x_j P_{ji}.$$

Thus, the Markov chain is time-reversible, so that, if $\vec{\pi}$ is the limiting distribution of the Markov chain, then

$$\vec{\pi} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{|V|} \end{bmatrix}. \quad \blacksquare$$

Problem 9.8 Randomized Chess

(a) (i) Given a square on the chessboard, a king can move horizontally until he is in the same column as that square, and then move vertically until he is on that square, so that the Markov chain of the king's movements is irreducible. $\blacksquare$

(ii) Consider the following two methods by which a king can return to its current square: (a) move one square diagonally and then move one square diagonally in the opposite direction, and (b) move one square diagonally, move one square vertically, and then move square horizontally, so as to return to the initial square. Since one method takes 2 moves and the other takes 3 moves and $\gcd(2, 3) = 1$, the king's moves are aperiodic. $\blacksquare$

(b) (i) The bishop can only move to squares of the same color, so that, if it starts on a black square, white squares are unreachable, and, if it starts on a white square, black squares are unreachable. Therefore, the Markov chain of the bishop's movements is not irreducible. $\blacksquare$

(ii) Consider the following two methods by which a bishop can return to its current square: (a) move one square diagonally and then move one square diagonally in the opposite direction, and (b) move two squares diagonally, and then move one square in the opposite direction twice. Since one method takes 2 moves and the other takes 3 moves and $\gcd(2, 3) = 1$, the bishop's moves are aperiodic. ■

(c) (i) Any 3×3 section of the chessboard is clearly irreducible. Thus, since the chessboard can be covered by 3×3 sections between which the knight can move, the knight can move to any square on the chessboard from any other square on the chessboard. Thus, the Markov chain of the knight's moves is irreducible. ■

(ii) Since the knight can move only from white squares to black squares and vice versa, any sequence of moves from a square to itself must have an even number of moves, so that a knight's moves are not aperiodic. ■

(d) Connect two squares on the chessboard with an edge if and only if the king can move between them (i.e., they are adjacent or diagonally adjacent). Since the king chooses a move uniformly at random, each edge has equal weight, so we can assign a weight of 1 to each edge. There are 210 edges, of which 3 are adjacent to the corner, so that, by the result of Problem 9.7 above,

$$\pi_{corner} = \frac{3}{2 \cdot 210} = \frac{1}{140}.$$

Therefore, the expected number of moves before returning to the corner is

$$m_{corner,corner} = \frac{1}{\pi_{corner}} = \boxed{140.}$$

(e) Connect two squares on the chessboard with an edge if and only if they are the same color as the corner and the bishop can move between them (i.e., they are in the same diagonal). Since the bishop chooses a move uniformly at random, each edge has equal weight, so we can assign a weight of 1 to each edge. There are 140 edges, of which 7 are adjacent to the corner, so that, by the result of Problem 9.7 above,

$$\pi_{corner} = \frac{7}{2 \cdot 140} = \frac{1}{40}.$$

Therefore, the expected number of moves before returning to the corner is

$$m_{corner,corner} = \frac{1}{\pi_{corner}} = \boxed{40.}$$

(f) Connect two squares on the chessboard with an edge if and only if the knight can move between them (i.e., they are in an 'L' shape). Since the knight chooses a move uniformly at random, each edge has equal weight, so we can assign a weight of 1 to each edge. There are 168 edges, of which 2 are adjacent to the corner, so that, by the result of Problem 9.7 above,

$$\pi_{corner} = \frac{2}{2 \cdot 168} = \frac{1}{168}.$$

Therefore, the expected number of moves before returning to the corner is

$$m_{corner,corner} = \frac{1}{\pi_{corner}} = \boxed{168.}$$

Problem 11.4 Practice with Definition of Poisson Process

(a) (i) Let $\lambda = 50$. By Theorem 11.8 (Poisson splitting), yellow packets arrive according to a Poisson process with rate 0.95λ, and, furthermore, this is independent of the number of green packets received. Thus, the expected number of yellow packets received is $0.95\lambda = \boxed{47.5}$ ($E[X]$, where $X \sim \text{Poisson}(0.95\lambda)$).

(ii) By the same reasoning as in part (i) above, for $\lambda = 50$, if $X \sim \text{Poisson}(0.95\lambda)$, the probability of 2000 yellow packets arrive is

$$P(X = 2000) = \frac{(0.95\lambda)^{2000}}{2000!} e^{-0.95\lambda} \approx \boxed{1.73 \times 10^{-2403}.}$$

(b) Let R be a random variable denoting the number of red packets which arrived during the second, let B be a random variable denoting the number of black packets which arrived during the second, let P be a random variable denoting the total number of packets which arrived during the second. Note that $P = B + R$. Then, by Theorem 11.7 (Merging independent Poisson processes), since $R \sim \text{Poisson}(30)$ and $B \sim \text{Poisson}(10)$, $P \sim \text{Poisson}(40)$. Therefore, by Bayes Rule,

$$\begin{aligned}
P(40 \text{ red packets}|60 \text{ packets}) &= \frac{P(60 \text{ packets}|40 \text{ red packets}) \cdot P(40 \text{ red packets})}{60P(\text{ packets})} \\
&= \frac{P(20 \text{ black packets}) \cdot P(40 \text{ red packets})}{60P(\text{ packets})} \\
&= \frac{\left(\frac{10^{20}e^{-10}}{20!}\right)\left(\frac{30^{40}e^{-30}}{40!}\right)}{\frac{40^{60}e^{-40}}{60!}} = \boxed{\frac{\left(\frac{10^{20}}{20!}\right)\left(\frac{30^{40}}{40!}\right)}{\frac{40^{60}}{60!}}.}
\end{aligned}$$

(c) By Theorem 11.9, each packet has a $\frac{10}{30} = \frac{1}{3}$ probability of arriving within the first ten seconds and a $\frac{20}{30} = \frac{2}{3}$ probability of arriving in the last 20 seconds. Thus the probability that 20 packets arrived during the first 20 seconds is the number of ways of choosing 20 of the 100 packets, times the probability that those 20 packets arrive in the first 10 seconds and the remaining 80 packets arrive in the last 20 seconds, i.e.

$$\boxed{\binom{100}{20}\left(\frac{1}{3}\right)^{20}\left(\frac{2}{3}\right)^{80}.}$$

Problem 11.6 Malware and honeypots

(a) This is the same as the probability that the lag for that infection is less than $t-s$, so that we evaluate the Cumulative Density Function of $\text{Exp}(\mu)$ at $t-s$, giving $\boxed{1-e^{-\mu(t-s)}.}$

(b) The probability is the average over all $s \in [0,t]$ of the probability density that the infection occurs at time s (which is uniform on $[0,1]$ times the probability that the infection is found by time t given that it occurs at time s (which was derived in part (b) above):

$$\int_0^t \frac{1}{t}(1-e^{-\mu(t-s)})\,ds = 1 - \frac{1-e^{-\mu t}}{\mu t} = \boxed{\frac{\mu t - 1 + e^{-\mu t}}{\mu t}.}$$

(c) By Theorem 11.9 (Uniformity), if $N(t)$ denotes the number of infected hosts at time t infections, then

$$\lambda \approx \frac{N(t)}{t}.$$

Thus, if $p(t)$ is the probability that an infection occuring before time t is detected by the Honeypot by time t (as computed in part (b) above),

$$\lambda \approx \frac{N(t)}{t} = \boxed{\frac{N_1(t)}{tp(t)}.}$$

(d) For $N(t)$ as defined in part (c) above, $N_2(t) = N(t) - N_1(t)$. Therefore, for $p(t)$ as defined in part (c) above,

$$N_2(t) = \boxed{\frac{N_1(t)}{p(t)} - N_1(t).}$$

Problem 11.7 Sum of Geometric number of Exponentials

For $\delta > 0$, consider flipping a biased coin with probability $p\mu\delta$ of getting heads on each flip, once every timestep of length δ. Let X be a random variable denoting the number of timesteps until a heads, so that $X \sim \text{Geometric}(p\mu\delta)$. Then, as explained in Section 11.3, as $\delta \to 0$, the distribution of δX approaches $\text{Exp}(p\mu)$. Furthermore, as $\delta \to 0$, the distribution of δX approaches that of S_N, because flipping a coin with probability $p\mu\delta$ of getting a heads is the same as flipping two coins, one with probability $\mu\delta$ of getting a heads and the other with probability p of getting a heads) until both are heads, so that, as $\delta \to 0$, each heads occurs at time $t \sim \text{Exp}(\mu)$, and, each time we get a heads, we stop with probability p, consistent with the fact that $N \sim \text{Geometric}(p)$. Therefore, $S_N \sim \text{Exp}(p\mu)$. ■

Problem 12.1 Converting a CTMC to a DTMC

The DTMC is as follows:

If $\vec{\pi}$ is the limiting distribution of this Markov chain, then the balance equations give:

$$(\lambda_{12}\delta + o(\delta))\pi_1 = (\lambda_{21}\delta + o(\delta))\pi_2 + \lambda_{31}\delta\pi_3,$$

$$(\lambda_{21}\delta + \lambda_{23}\delta + o(\delta))\pi_2 = (\lambda_{12}\delta + o(\delta))\pi_1 + (\lambda_{32}\delta + o(\delta))\pi_3,$$

$$\lambda_{31}\delta\pi_3 = (\lambda_{23}\delta + o(\delta))\pi_2.$$

Dividing by δ and taking the limit as $\delta \to 0$ gives:

$$\boxed{\begin{gathered} \lambda_{12}\pi_1 = \lambda_{21}\pi_2 + \lambda_{31}\pi_3, \\ (\lambda_{21} + \lambda_{23})\pi_2 = \lambda_{12}\pi_1 + \lambda_{32}\delta\pi_3, \\ \lambda_{31}\pi_3 = \lambda_{23}\pi_2. \end{gathered}}$$

Problem 13.1 Bathroom Queue

By the derived formula for the mean length of an M/M/1 queue,

$$\begin{aligned} \frac{E[T_Q]^{women}}{E[T_Q]^{men}} &= \frac{\frac{\lambda/\mu}{\mu-\lambda}}{\frac{\lambda/(2\mu)}{2\mu-\lambda}} \\ &= \frac{2\mu-\lambda}{2(\mu-\lambda)} \\ &= \frac{2\mu}{2(\mu-\lambda)} + \frac{\lambda}{2(\lambda-\mu)} \\ &= \frac{1}{1-\lambda/\mu} + \frac{1}{2(1-\mu/\lambda)} \\ &= \boxed{\frac{1}{1-\rho} + \frac{1}{2(1-1/\rho)}.} \end{aligned}$$

Problem 13.2 Server Farm + Extra Credit

By Theorem 11.8 (Poisson splitting), jobs arrive at Server 1 according to a Poisson process with mean $p\lambda$, and jobs arrive at Server 2 according to a Poisson process with mean $(1-p)\lambda$. Therefore, for those jobs going to Server 1, the mean response time is $\frac{1}{\mu_1 - p\lambda}$, and, for those jobs going to Server 2, the mean response time is $\frac{1}{\mu_2-(1-p)\lambda}$. Taking the weighted average over both servers, the mean response time is

$$\frac{p}{\mu_1 - p\lambda} + \frac{1-p}{\mu_2 - (1-p)\lambda}.$$

In order to minimize mean response time, we find the minimum of this quantity with respect to p (by differentiating and finding zeros), and then allocate the minimizing fraction p of the jobs to Server 1 and the fraction $(1-p)$ of the jobs to Server 2. ■

Problem 13.4 M/M/1 Number in Queue

By Little's Law,

$$E[N_Q] = \lambda E[T_Q].$$

Thus, by the derived formula for $E[T_Q]$,

$$E[N_Q] = \lambda \frac{\rho}{\mu - \lambda} = \frac{\rho}{\mu/\lambda - 1} = \frac{\rho}{1/\rho - 1} = \frac{\rho^2}{1-\rho}. \quad \blacksquare$$

Problem 13.5 M/M/1/FCFS with Finite Capacity

(a) The CTMC appears as follows:

(b) Let $\vec{\pi}$ be the limiting distribution of the CTMC, and let $\rho = \lambda/\mu$. Then, the balance equations are identical to those in the example on page 300, so that, for $0 \le i \le N$,

$$\pi_i = \rho^i \pi_0.$$

Since $1 = \sum_{i=0}^{N} \pi_i = \sum_{i=0}^{N} \rho^i \pi_0$,

$$\pi_0 = \frac{1-\rho}{1-\rho^{N+1}}, \quad \text{so} \quad \pi_i = \rho^i \frac{1-\rho}{1-\rho^{N+1}}.$$

(c) The utilization of the system is the time when the queue is not empty:

$$1 - \pi_0 = 1 - \boxed{\frac{1-\rho}{1-\rho^{N+1}}.}$$

(d) By PASTA, the fraction of jobs turned away is the same as the fraction of time during which the queue is full,

$$\pi_N = \boxed{\rho^N \frac{1-\rho}{1-\rho^{N+1}}.}$$

(e) The rate at which jobs are turned away is the fraction of jobs which are turned away times the rate at which jobs arrive,

$$\pi_N \lambda = \boxed{\rho^N \frac{1-\rho}{1-\rho^{N+1}} \lambda.}$$

(f) We can model this finite capacity queue as an M/M/1 queue where the rate of job arrivals is $(1-\pi_N)\lambda$, the fraction of jobs which are not turned away. Therefore, by the formula derived on page 302,

$$E[\text{Number in system}] = \frac{(1-\pi_N)\rho}{1-(1-\pi_N)\rho} = \boxed{\frac{(1-\rho^N \frac{1-\rho}{1-\rho^{N+1}})\rho}{1-(1-\rho^N \frac{1-\rho}{1-\rho^{N+1}})\rho}.}$$

(g) By the reasoning from part (f) above and the formula derived on page 303,

$$E[T] = \frac{1}{\mu-(1-\pi_N)\lambda} = \boxed{\frac{1}{\mu-(1-\rho^N \frac{1-\rho}{1-\rho^{N+1}})\lambda}.}$$

(h) Evaluating the expression derived in part (d) at $N = 10, \rho = .4$, gives a loss probability of 6.29×10^{-5}, whereas evaluating the same expression at $N = 5, \rho = 0.2$ (since doubling the CPU speed doubles μ and thus halves ρ) gives a loss probability of 2.56×10^{-4}, so that it is more effective to $\boxed{\text{double the buffer size.}}$

(i) Evaluating the expression derived in part (d) at $N = 10, \rho = .8$, gives a loss probability of 2.35×10^{-2}, whereas evaluating the same expression at $N = 5, \rho = 0.4$ (since doubling the CPU speed doubles μ and thus halves ρ) gives a loss probability of 6.17×10^{-3}, so that it is more effective to $\boxed{\text{double the CPU speed.}}$

(j) The results of parts (h) and (i) follow intuitively from the plot of $E[N]$ against ρ (Figure 13.3) when ρ is small, reducing ρ has little effect on the expected number of jobs in the queue, whereas when ρ is large, reducing ρ dramatically reduces the expected number of jobs in the queue, and thus the chance that the queue is full. ■