TMA4265 - Stochastic Processes

Semester Project

Candidate number
111111

October 29, 2012

Problem One

a.

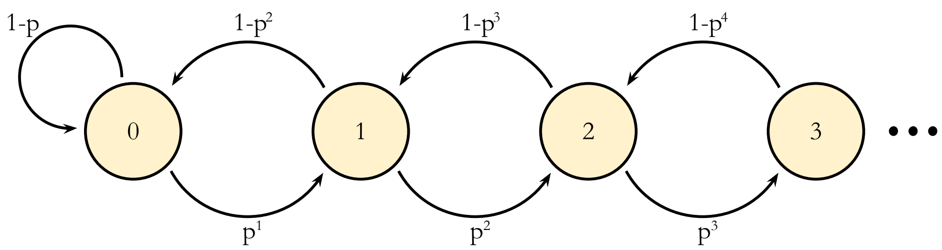

Figure 1: The Markov chain P_{ij} illustrated as a digraph.

The Markov chain P_{ij} in fig. 1 describes a random walk. Starting in any state, you can reach any other state - i.e. all states communicate. Thus, P_{ij} is **strongly connected**, has **one equivalence class** and is **irreducible**. Let X_n and X_{n+s} be two states such that there are $s-1$ intermediate states between them. To get from state i to $j = i + s$ you have to visit all intermediate states at least once[1]. Another trait of P_{ij} is that with growing i, the probability of moving from state i to $j = i+1$ diminishes exponentially, while the probability of moving back to $j' = i-1$ grows exponentially. Starting in state i, it is therefore reasonable to claim that it is always possible to return to state i in a finite number of steps. State i

[1]Equivalently, s is the minimum number of steps you must preform to make the transition from state i to $i+s$

is then said to be positive recurrent, and since positive recurrence is a class property, the whole chain is **positive recurrent**.

Starting in state i, let n be the number that allows you to end up in state i after n transitions with the probability $P_{ii}^n > 0$. Let N_i be the set of all such numbers for our chain. The greatest common divisor of the elements in N_i is called the period d of state i. Since $P_{00}^n > 0$ for any n, $N_i = 0, 1, 2, ...$ and so P_{00} has a period of $d = 1$. Since periodicity is a class property, the chain as a whole has period one. A Markov chain with $d = 1$ is called **aperiodic**.

A positive recurrent aperiodic chain is called **ergodic**. Theorem 4.1 in [Ross 2010] states that for an irreducible ergodic Markov chain the limit $\pi_j = \lim_{n\to\infty} P_{ij}^n$, $j \geq 0$ exists,

$$\pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij} \tag{1}$$

and

$$\sum_{j=0}^{\infty} \pi_j = 1. \tag{2}$$

In this case eq. (1) becomes

$$\pi_i = \begin{cases} \pi_{i-1}P_{i-1,i} + \pi_{i+1}P_{i+1,i} & = \pi_{i-1}p^i + \pi_{i+1}(1-p^{i+2}) \qquad i > 0 \\ \pi_i P_{i,i} + \pi_{i+1}P_{i+1,i} & = \pi_0(1-p) + \pi_1(1-p) \qquad i = 0 \end{cases} \tag{3}$$

The π's are called the limiting probabilities that the process will be in state i at time n, and it can be shown that π_i also equals the long-run proportion of time that the process will be in state i. Below the limiting probabilities are calculated for the first 3 states.

$$\pi_1 = \frac{p}{1-p^2}\pi_0; \quad \pi_2 = \frac{p^2}{1-p^3}\pi_1; \quad \pi_3 = \frac{p^3}{1-p^4}\pi_2$$

By induction this yields the general formula (recurrence relation)

$$\pi_{i+1} = \frac{p^{i+1}}{1-p^{i+2}}\pi_i = \frac{p^{i+1}p^i p^{i-1}\cdots p^2 p^1}{(1-p^{i+2})(1-p^{i+1})(1-p^i)\cdots(1-p^3)(1-p^2)}\pi_0$$
$$= \frac{p^{1+2+\ldots+(i-1)+i+(i+1)}}{\prod_{i=2}^{k=i+2}(1-p^k)}\pi_0 = \frac{p^{\frac{1}{2}(i+1)(i+2)}}{\prod_{i=2}^{k=i+2}(1-p^k)}\pi_0$$

or, equivalently,

$$\pi_i = \pi_0 \cdot \prod_{k=1}^{k=i+1} \frac{p^k}{1-p^{k+1}} \tag{4}$$

Now eq. (2) becomes

$$\begin{aligned} \sum_i \pi_i = \pi_0 \left(1 + \sum_i \cdot \prod_{k=1}^{k=i+1} \frac{p^k}{1-p^{k+1}}\right) = 1 \\ \implies \pi_0 = \left(1 + \sum_i \cdot \prod_{k=1}^{k=i+1} \frac{p^k}{1-p^{k+1}}\right)^{-1} \end{aligned} \tag{5}$$

Substituting eq. (5) into eq. (4) gives an expression for the limiting probabilities

$$\pi_i = \frac{\prod_{k=1}^{k=i+1} \frac{p^k}{1-p^{k+1}}}{1 + \sum_i \cdot \prod_{k=1}^{k=i+1} \frac{p^k}{1-p^{k+1}}} \tag{6}$$

The distribution is positive so the limiting probabilities exist and the states are indeed positive recurrent.

b.

Let the set ξ contain the first 10 states, and let T_i be the time spent to reach state 10, starting in state $X_0 = i \in \xi$.

$$\begin{aligned} \mu_i &= E[T_i] \\ &= E[T_i|X_1 = i-1]P(X_1 = i-1) + E[T_i|X_1 = i+1]P(X_1 = i+1) \\ &= (1 + E[T_{i-1}])(1-p^{i+1}) + (1 + E[T_{i+1}])p^{i+1} \\ &= 1 + \mu_{i-1} + (\mu_{i+1} - \mu_{i-1})p^{i+1} \end{aligned} \tag{7}$$

Note the special cases for $\mu_0 = 1 + \mu_0 + (\mu_1 - \mu_0)p$, and $\mu_9 = 1 + \mu_8(1 - p^{10})$

eq. (7) constitutes a set of 10 equations that can be solved with MATLAB. In table ¡TABLE¿ the solutions are listed for $p = 0.75$ and $p = 0.90$.

The exponetial decay for the probabilities for moving on to the next state clearly manifests itself for the case of $p = 0.75$; no matter what state you start in, you'll have to wait for a very long time to reach state 10. The transition from state 9 to 10 is taking very long time compared to the other one-step transitions.

c.

The probability for ever reaching state 100 is 1, no matter what state you start in, since all limiting probabilities are positive. If you start in state $X_0 < 100$, you will have to wait for a very long time to reach state 100, and if $X_0 > 100$ you will only have to wait close to $X_0 - 100$ steps (at least if not $p \approx 1$).

d.

Problem Two

In the following X_n is the number of individuals in the n'th generation, X_0 is the initial population, $\mu = E[\textit{children per individual}]$, $p_j = P\{\textit{an individual has } j \textit{ offspring}\}$. The distributions that are analyzed are listed in table 1

	p_0	p_1	p_2	p_3
I	0.6	0.05	0.15	0.2
II	0.25	0.60	0.10	0.05

Table 1: The probability distributions given in Problem Two.

Chapter 4.7 in [Ross 2010] presents some results that are used in this problem. Below is some properties of branching processes discussed.

Generally, the mean number, and the variance, of offspring of a single individual is

$$\mu = \sum_{j=0}^{\infty} jP_j \qquad \text{and} \qquad \sigma^2 = \sum_{j=0}^{\infty} (j-\mu)^2 P_j \tag{8}$$

respectively.

For both our distributions $\mu = 0.95$, and since $\mu < 1$ the population will eventually die out. Defining Z_i to be the number of offspring of the i'th individual for the $(n-1)$th generation, one can find

$$X_n = \sum_{i=1}^{X_{n-1}} Z_i \tag{9}$$

in the edge case where $X_0 = 1$. One can then obtain

$$E[X_n] = E[E[X_n|X_{n-1}]] = E\left[E\left[\sum_{i=1}^{X_{n-1}} Z_i|X_{i-1}\right]\right] = E[X_{n-1}]\mu \tag{10}$$

which leads to the result

$$E[X_1] = \mu, \quad E[X_2] = \mu E[X_1] = \mu^2, \quad \cdots, \quad E[X_n] = \mu^n. \tag{11}$$

It can then be shown that the variance is

$$Var(X_n) = \begin{cases} \sigma^2\mu^{n-1} \cdot \frac{1-\mu^n}{1-\mu}, & \mu \neq 1 \\ n\sigma^2, & \mu = 1 \end{cases} \tag{12}$$

When doing simulations, the estimated mean value and standard deviation is given by

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad S^2 = \frac{1}{n-1}\sum_{i=1}^{n} X_i - \bar{X}, \tag{13}$$

respectively.

a.

The mean and standard deviation is found analytically from eq. (8), eq. (11) and eq. (12), and is presented in table 2.

The branching process was simulated a hundred thousand times for each of the probability distributions. The values in table 3 were computed by running

n	$E[X_n]$	$SD_I[X_n]$	$SD_{II}[X_n]$
10	0.5987	2.7977	2.7692
100	0.0059	0.4379	0.0678
1000	5.2912e-23	4.1529e-11	2.4697e-11

Table 2: Analytical values of the mean $E[X_n]$ and the standard deviation, SD ($= Var^{\frac{1}{2}}[X_n]$) for distribution I and II.

```
% No. of simulations
N = 100000;
% Size of the first population
init = 1;
% Distribution I = p1 and II = p2
p1 = [0.60 0.05 0.15 0.20];
p2 = [0.25 0.60 0.10 0.05];
% Run the simulations for I and II
branchtrials(N, init, p1);
branchtrials(N, init, p2);
```

n	$E_I[X_n]$	$E_{II}[X_n]$	$E_{III}[X_n]$	$SD_I[X_n]$	$SD_{II}[X_n]$	$SD_{III}[X_n]$
10	0.6007	0.5989	0.0554	2.8099	1.6708	0.4662
100	0.0060	0.0050	0	0.4583	0.2392	0
1000	0	0	0	0	0	0

Table 3: Simulated values of the mean $E[X_n]$ and the standard deviation, SD ($= Var^{\frac{1}{2}}[X_n]$) for distribution I and II.

The simulated values is indeed very close to the ones found analytically for both distributions. Given that the MATLAB rand()-function is uniform, the simulated values would probably become even more accurate if the number of simulations is increased. No simulation ever reached $n = 1000$, but this is only natural considering the low probability for that to happen.

Case (i), (ii) and (iii) are presented in table 4 for both distributions.

	Distribution I		Distribution II		Distribution III	
Case	$E[\cdot]$	$SD[\cdot]$	$E[\cdot]$	$SD[\cdot]$	$E[\cdot]$	$SD[\cdot]$
(i)	4.5197	7.1409	8.2460	10.8820	3.5185	2.5587
(ii)	19.9991	109.4967	19.9830	65.9630	3.9787	7.0344
(iii)	2.9699	5.4466	2.3411	2.9429	1.5572	1.1611

Table 4: Simulated values of the mean $E[X_n]$ and the standard deviation, SD ($= Var^{\frac{1}{2}}[X_n]$) for distribution I and II.

b.

```
% Defining a third distribution
p3 = [0.50 0.30 0.15 0.05];
% Run the simulations for III
branchtrials(N, init, p3);
```

The three different cases are presented in figs. 2 to 4 as histograms, grouped by probability distribution.

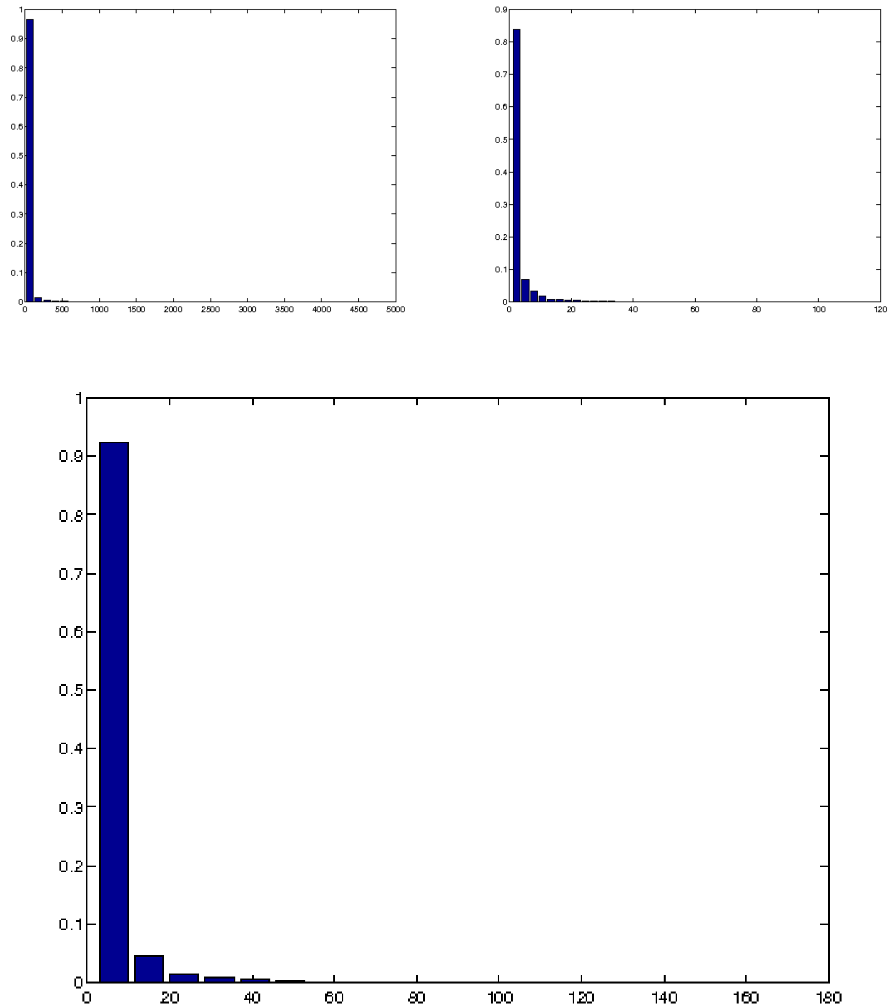

Figure 2: Case (i), (ii) and (iii) for distribution I

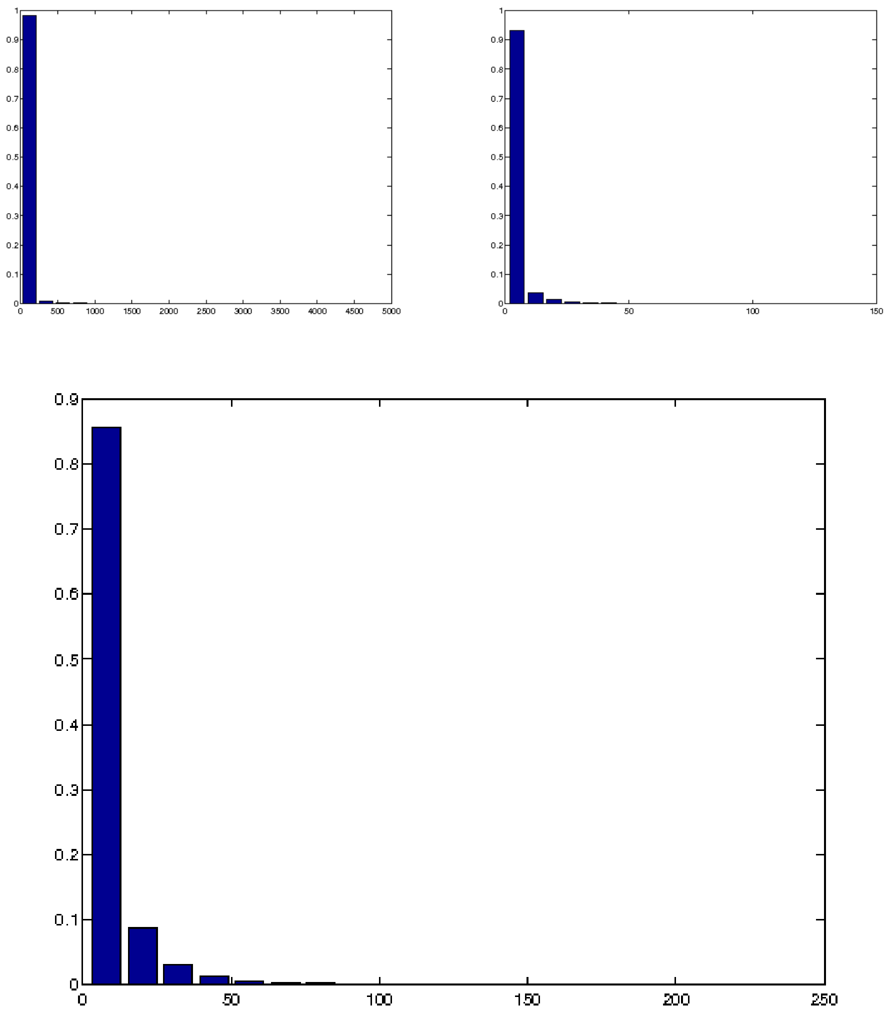

Figure 3: Case (i), (ii) and (iii) for distribution II

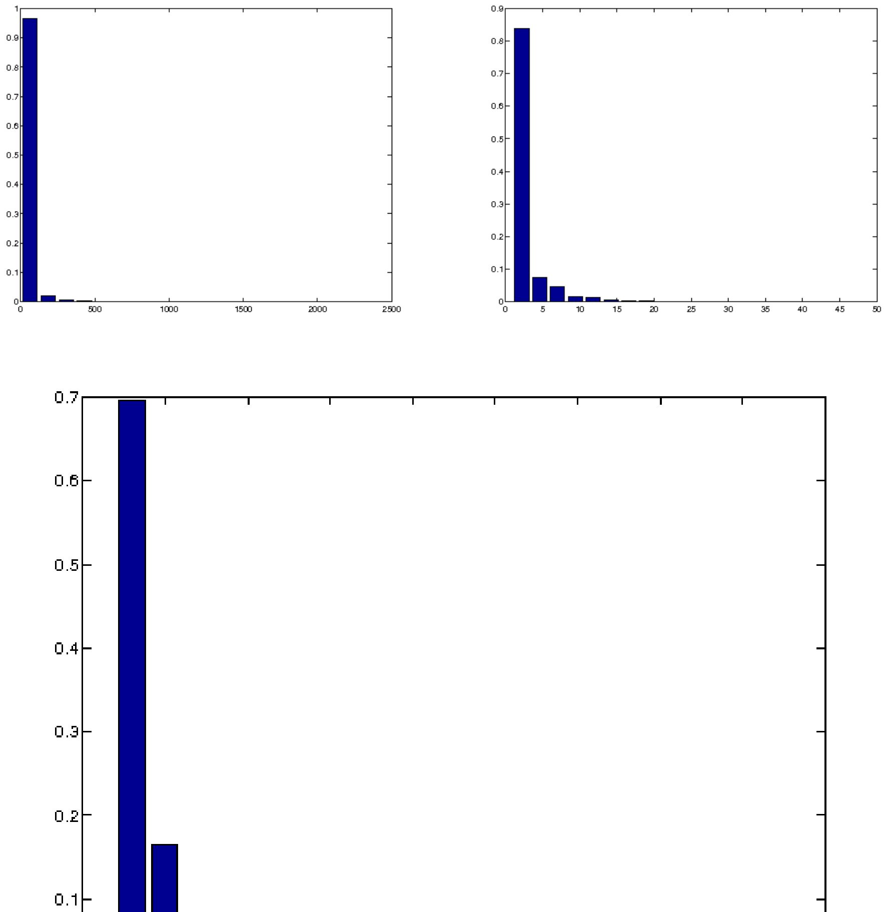

Figure 4: Case (i), (ii) and (iii) for distribution III

References

[Ross 2010] Sheldon M. Ross (2010), Introduction to Probability Models.

Problem One

```
function W = randwalk(N,i,p)

% ----// randwalk.m //----
%
% Simulates a random walk
%
%
%  Input:
%    N - int, number of transitions
%    i - int, initial state
%    p - double, 0<p<1, probability.
%
%  Output:
%    w - vector, with transtions.
%
% -------------------------

W = zeros(1,N+1);
W(1) = i;
for j=1:N
  r=rand(1);
  % Walk to the right
  if W(j) >= 0 && r < p^(W(j)+1)
    W(j+1) = W(j) + 1;
  % Walk to the left
  elseif W(j) > 0
    W(j+1) = W(j)-1;
  else
    W(j+1) = 0
  end
end
end
```

Problem Two

```
function [popu, n, tot_people, largest_gen] = branchsim(init, dist)

% ----// branchsim.m //----
%
% Simulates a branching process
```

```
%
%
%  Input:
%    init - int, initial size of the population
%    dist - vector, offspring probability
%           distribution [p0, ..., pn]
%
%  Output:
%    popu - vector, population for each generation.
%    n    - int, number of generations
%    tot_people  - int, total number of people lived.
%    largest_gen - int, size of the largest generation.
%
% -------------------------

popu = zeros(1,2000);
% The population of the 1st generation
popu(1) = 1;
% Accumulated probabilities to be used w. rand(1)
cumdist = cumsum(dist);
len = length(dist);
tot_people = init;
largest_gen = init;

% Generation number n
n = 1;
while popu(n) > 0
  popu(n+1) = 0;
  for j=1:popu(n)
    r = rand(1);
    if r < cumdist(1)
      popu(n+1) = popu(n+1) + 0;
    elseif r < cumdist(2)
      popu(n+1) = popu(n+1) + 1;
    elseif r < cumdist(3)
      popu(n+1) = popu(n+1) + 2;
    else
      popu(n+1) = popu(n+1) + 3;
    end
  end
  tot_people = tot_people + popu(n+1);
      if largest_gen < popu(n+1)
        largest_gen = popu(n+1);
      end
```

```
        n=n+1;
    end
    end
```