

## Math 350 - Homework 5 - Solutions

1. A deck of 100 cards—numbered  $1, 2, \dots, 100$ —is shuffled (i.e., a random permutation is applied to the cards in the deck) and then turned over one card at a time. Say that a “hit” occurs whenever card  $i$  is the  $i$ th card to be turned over,  $i = 1, \dots, 100$ . Write a simulation program to estimate the expectation and variance of the total number of hits. Run the program. Find the exact answers and compare them with your estimates.

This is known as the *matching problem*. Let us first obtain the exact values. We let  $n$  denote the number of cards. (In the problem  $n = 100$ , but the answer does not depend of the exact value as we shall see.) Let  $S_{n,i}$  be the set of permutations that fix  $i$  (i.e., for which  $i$  is a hit). In other words,  $S_{n,i}$  consists of the bijections  $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  such that  $\sigma(i) = i$ . Note that this set has exactly  $(n - 1)!$  elements. Let  $I_i$  be the random variable taking values in  $\{0, 1\}$  such that  $I_i(\sigma) = 1$  if  $\sigma(i) = i$  and  $I_i(\sigma) = 0$  if  $\sigma(i) \neq i$ . In other words,  $I_i$  is the indicator function of the set  $S_{n,i}$ . Therefore,

$$E[I_i] = \frac{|S_{n,i}|}{|S_n|} = \frac{(n-1)!}{n!} = \frac{1}{n}.$$

The number of hits (or fixed points) is the random variable

$$X = I_1 + \dots + I_n.$$

The expectation of  $X$  is:

$$E[X] = \sum_{i=1}^n E[I_i] = n \times \frac{1}{n} = 1.$$

To obtain the variance, first observe that

$$E[I_i I_j] = \begin{cases} \frac{1}{n} & \text{if } i = j \\ \frac{1}{n(n-1)} & \text{if } i \neq j. \end{cases}$$

In fact, if  $i = j$ , then  $I_i I_j = I_i^2 = I_i$ ; and if  $i \neq j$ , then  $I_i I_j$  is 1 on the subset of  $S_n$  consisting of all the  $\sigma$  that fix both  $i$  and  $j$  (and 0 on the complement of that set). Now this set has  $(n - 2)!$  elements, so  $E[I_i I_i] = (n - 2)!/n! = 1/n(n - 1)$ . Also notice that the number of pairs  $(i, j)$  with  $i \neq j$  is  $n(n - 1)$ . Therefore,

$$\sum_{i=1}^n \sum_{j=1}^n E[I_i I_j] = \sum_{i=1}^n E[I_i^2] + \sum_{i \neq j} E[I_i I_j] = 1 + n(n - 1) \times \frac{1}{n(n - 1)} = 2.$$

Thus we finally get

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2 = E[X^2] - 1 = \sum_{i=1}^n \sum_{j=1}^n E[I_i I_j] - 1 = 2 - 1 = 1.$$

We now wish to confirm this by simulation. A program that generates a random permutation  $\sigma$  in  $S_n$  can

be as follows:

```
function b=rand_perm(a)
%a is a given vector of the form [a(1) a(2) ... a(n)]
%b is the vector of same length as a obtained
%by applying a random permutation to the indices of a.
b=a;
n=length(b);
for i=n:-1:2
    j = floor(i*rand(1))+1;
    k = b(j);
    l = b(i);
    b(j) = l;
    b(i) = k;
end
```

We can now use this program to calculate the expected value and variance of the number of hits with the following script (here I use a total of 1000 trials):

```
a=1:100;
m=1000;
N=zeros(1,m);
for j=1:m
    b = rand_perm(a);
    N(j) = sum(a==b);
end
Exp_value = sum(N)/m
Variance = (sum(N.^2)/m)-(sum(N)/m)^2
```

A typical run gave the values

```
Exp_value = 0.9760
Variance = 0.9134
```

2. This problem is about the acceptance-rejection method.

- (a) Write a program that implements the acceptance-rejection method to obtain a random variable  $X$  taking values in  $\{1, 2, \dots, n\}$  with probabilities  $P\{X = j\} = p_j$ . Assume that the random variable  $Y$  (which is accepted or rejected to obtain  $X$ ) is uniform with values in  $\{1, 2, \dots, n\}$ . (As input variables, take the vector of probabilities  $p = (p_1, \dots, p_n)$ , where  $n$  is arbitrary, and as output variable the sample value of  $X$ .)

```
function X=acc_rej_unif(p,m)
%Inputs: p=(p_1, ..., p_n) vector of probabilities, where X=j with
%         probability p_j.
%         m is the number of independent sample values of X.
```

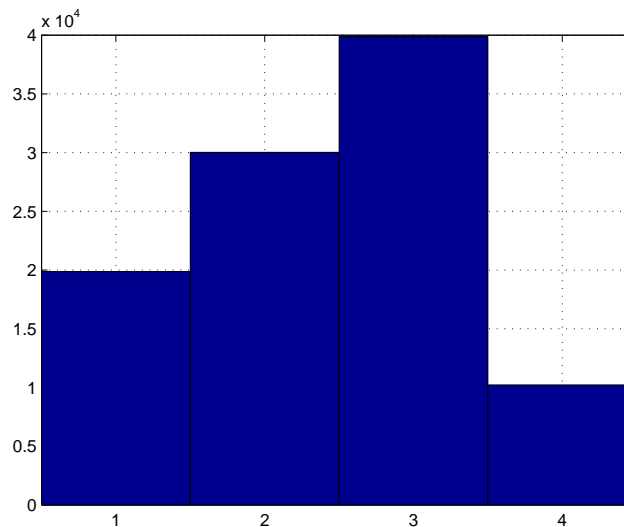
```

%Output: X a vector of length m containing the sampled values of
%         the random variable with distribution p, obtained by the
%         acceptance rejection method. The random variable that is
%         accepted or rejected is discrete uniform.
l=length(p);
c=max(l*p);
for i=1:m
    r=0;
    U=1;
    while U>=r
        U=rand(1);
        j=floor(l*rand(1))+1;
        r=l*p(j)/c;
    end
    X(i)=j;
end

```

- (b) Suppose now that  $n = 4$  and  $p_1 = 0.2$ ,  $p_2 = 0.3$ ,  $p_3 = 0.4$ ,  $p_4 = 0.1$ . Test that your program is sound by generating a sequence  $X_1, X_2, \dots, X_k$ , for some large  $k$ , and check that the frequency of occurrence of  $j = 2$  is approximately 0.3.

We can see this by drawing a histogram. The following graph gives the number of occurrences of  $j = 1, 2, 3, 4$  out of  $10^5$  trials. The frequencies are exactly as expected.



3. Suppose that  $p = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$  and  $q = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$  are two probability vectors and  $\alpha = 1/3$ . Write a program based on the composition method of section 4.5 that generates a random variable  $X$  with probabilities

$$P(X = j) = \alpha p_j + (1 - \alpha)q_j.$$

Do a similar test as in the previous problem to check that your program does what is expected. (The

composition method amounts to the following: Let  $Y$  be a random variable with probability vector  $p$  and  $Z$  a random variable with probability vector  $q$ , both taking values in  $\{1, 2, 3, 4\}$  in the present case. Then simulate a Bernoulli random variable  $B$  with parameter  $\alpha$ . If  $B = 1$  generate a sample value of  $Y$  and set  $X = Y$ ; if  $B = 0$ , generate a sample value of  $Z$  and set  $X = Z$ .)

The following generates  $10^5$  independent values of  $X$  using the composition method.

```
m=100000;
P=[1/2 1/4 1/8 1/8;1/4 1/4 1/4 1/4];
a=1/3;
F=cumsum(P,2);
for j=1:m
    f = F(1+(rand(1)>a),:);
    u = rand(1);
    X(j) = 1*(u<=f(1))+2*(u>f(1)&u<=f(2))+3*(u>f(2)&u<=f(3))+4*(u>f(3));
end
```

We can check that it is doing what it is supposed to do by finding the frequency of each value. The combined distribution of  $X$  is  $(1/3, 1/4, 5/24, 5/24) = (0.3333, 0.2500, 0.2083, 0.2083)$ . The script

```
for s=1:4
    q(s)=sum(X==s)/m;
end
q
```

gave the values  $(0.3343, 0.2498, 0.2084, 0.2075)$ .

4. A (discrete time) Markov chain consists of a sequence of random variables  $X_0, X_1, X_2, \dots$ , not necessarily independent or equally distributed, characterized by the following properties:

- Each  $X_j$  takes values in a set  $S = \{s_1, s_2, \dots\}$  (finite or infinite), which we call the set of states (of a system whose time evolution is being modeled by the chain);
- The initial state  $X_0$  has a probability distribution  $P(X_0 = j) = \pi_j$ . We call  $(\pi_j)$  the initial distribution of the chain.
- For each  $n = 1, 2, \dots$ , the conditional probability  $P(X_n = j | X_{n-1} = i) = p_{ij}$  is given. These are called the transition probabilities of the chain.

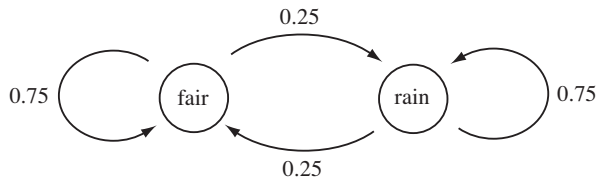
As a simple example, consider the following crude model of weather forecasting. The town of Markoville has only two possible weather conditions:  $s_1 = \text{“fair”}$  and  $s_2 = \text{“rainy”}$ . Empirical observation has shown that the best predictor of Markoville’s weather tomorrow is today’s weather, with the following day-to-day transition probabilities:

For example, if today’s weather is rainy, the probability that tomorrow’s is fair is  $p_{21} = 0.25$ , and that tomorrow’s is also rainy is  $p_{22} = 0.75$ .

Write a program that simulates Markoville’s weather for the next 1000, assuming that the weather today is “fair.”

A general program for simulating a Markov chain is given in the next program. I will use the same program to simulate Markoville’s weather. The states are 1 for rainy and 2 for fair. We set

		tomorrow	
		rain	fair
today	rain	0.75	0.25
	fair	0.25	0.75



```

P=[0.75 0.25; 0.25 0.75];
p=[0 1];
n=10^5;
X=markov_chain(p,P,n);
%Let us find the frequency of rainy days:
rainy=sum(X==ones(1,n))/n

```

I obtained the value 0.5001 for the frequency of rainy days. The exact value turns out to be  $1/2$ . (This is easy to show. We may return to general properties of Markov chains later.)

5. A more general Markov chain program.

- Write a general program that simulates a Markov chain  $X_1, \dots, X_n$  with the data:  $\pi = (\pi_1, \dots, \pi_k)$  the initial distributions;  $P = (p_{ij})$  the transition probabilities matrix; and  $n$ , the number of random variables in the chain.
- Suppose that  $\pi = (0.2 \ 0.5 \ 0.3)$  and

$$P = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.4 & 0.1 & 0.5 \end{pmatrix}.$$

Use your program (for a large enough  $n$ ) to find the frequency of occurrence of each of the three states in the chain.

Here is the program which I have already used for problem 4. The solution to problem 5 is after this script.

```

function X=markov_chain(p,P,n)
%Inputs - p probability distribution of initial state
%       - P transition probabilities matrix
%       - n number of iterates
%Output - X sample chain of length n
%
k=length(p);
X=zeros(1,n); %Initialize X.
F=cumsum(P,2); %The ith row of F has the cumulative distribution
               %of P(i,:).
G=cumsum(p); %Cumulative distributison function of p.
%Generate the initial state, X(1), with distribution p.
u=rand(1);

```

```

x=(u<G(1));
for i=2:k
    x=x+i*(G(i-1)<=u & u<G(i));
end
X(1)=x;
%We now generate X(j) inductively.
for j=2:n
    q=P(x,:);
    G=cumsum(q);
    u=rand(1);
    x=(u<G(1));
    for i=2:k
        x=x+i*(G(i-1)<=u & u<G(i));
    end
    X(j)=x;
end

```

Similarly to problem 4, we set

```

P=[.8 .1 .1; .3 .4 .3; .4 .1 .5];
p=[.2 .5 .3];
n=10^5;
X=markov_chain(p,P,n);

```

We can find the frequencies of each state as follows:

```

f1=sum(X==1*ones(1,n))/n
f2=sum(X==2*ones(1,n))/n
f3=sum(X==3*ones(1,n))/n

```

for which I got the following values:

```

f1 =0.6427
f2 =0.1443
f3 =0.2130

```

Think about the following remark: the matrix power  $P^{10000}$  is given by

```

P^10000=
    0.6429    0.1429    0.2143
    0.6429    0.1429    0.2143
    0.6429    0.1429    0.2143

```

It is clear that the frequencies we obtained by simulation are the row entries in this matrix. Can you explain why this is so? This requires interpreting the  $n$ th power of  $P$  in terms of the probabilities of each state at the  $n$ th iterate of the process.