ECS 256 Homework 1.1

Bingxi Li, Qiwei Li, Jiaping Zhang*
University of California, Davis

January 2016

1 The Problem

Items come in that need to be packed in boxes, in order of arrival. The maximum allowable weight per box is W_{max}; if an item would cause a box to go overweight, a new box must be started.

The weights $W_1, W_2, W_3, ...$ of the items are independent and identically distributed (i.i.d.) with some distribution on $1, 2, ..., w_{max}$. Let X_n denote the total weight of items in the current box at ẗimen̈, i.e. after the nth item has been packed. Due to the i.i.d. assumption, the X_n obey the Markov property.

Do the following, with π denoting the stationary distribution of the chain:

- Derive an expression for the long-run mean number of items per box in terms of π.
- Derive an expression for the long-run mean weight per box in terms of π.
- Let Q denote the weight of the first item put into any box. Derive an expression for the probability mass function of Q, i.e. the values P(Q = i), i = 1,2,...,r in terms of π.
- Take w_{max} to be 10, and P(W = i) = ci/10, i = 1,2,...,10 for suitable c, i.e. the c that makes the probabilities sum to 1. Find the values of the above expressions, and check via simulation.

2 The Solution

Notation and Settings

The maximum weight is w_{max}. The weight of items $W_1, W_2, W_3, \ldots, W_n$ are independent and identically distributed on $\{1, 2, 3, \ldots, w_{max}\}$ with probability being:

$$I_i = P(w = i), \forall i \in \{1, 2, 3, \ldots, w_{max}\} \tag{1}$$

The average weight of item, $\bar{w}$ is:

$$\overline{w} = \sum_{i=1}^{w_{max}} I_i \cdot i \tag{2}$$

When the box have weight w_i at n^{th} state and w_j at $(n+1)^{th}$ state, the transition probability between the two states is defined in the following matrix as the P_{ij} for $i, j \in \{1, 2, 3, ..., w_{max}\}$.

We want to point out an important note here. When computing each P_{ij}, one needs to consider the probability of achieving weight j in the current and the probability of achieving weight j in the next box because the current one cannot take the item weight.

*bxli@ucdavis.edu, qwli@ucdavis.edu, jpzhang@ucdavis.edu

$$\begin{bmatrix}
0 & I_1 & I_2 & \dots & I_{\frac{w_{max}}{2}-2} & I_{\frac{w_{max}}{2}-1} & I_{\frac{w_{max}}{2}} & \cdots & I_{w_{max}-2} & I_{w_{max}-1}+I_{w_{max}} \\
0 & 0 & I_1 & \dots & I_{\frac{w_{max}}{2}-3} & I_{\frac{w_{max}}{2}-2} & I_{\frac{w_{max}}{2}-1} & \cdots & I_{w_{max}-3}+I_{w_{max}-1} & I_{w_{max}-2}+I_{w_{max}} \\
0 & 0 & 0 & \dots & I_{\frac{w_{max}}{2}-4} & I_{\frac{w_{max}}{2}-3} & I_{\frac{w_{max}}{2}-2} & \cdots & I_{w_{max}-4}+I_{w_{max}-1} & I_{w_{max}-3}+I_{w_{max}} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & I_1 & I_2 & \cdots & I_{\frac{w_{max}}{2}}+I_{w_{max}-1} & I_{\frac{w_{max}}{2}+1}+I_{w_{max}} \\
0 & 0 & 0 & \dots & 0 & 0 & I_1+I_{\frac{w_{max}}{2}+1} & \cdots & I_{\frac{w_{max}}{2}-1}+I_{w_{max}-1} & I_{\frac{w_{max}}{2}}+I_{w_{max}} \\
0 & 0 & 0 & \dots & I_{\frac{w_{max}}{2}-1} & I_{\frac{w_{max}}{2}} & 0 & \cdots & I_{\frac{w_{max}}{2}-2}+I_{w_{max}-1} & I_{\frac{w_{max}}{2}-1}+I_{w_{max}} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\
0 & I_2 & I_3 & \dots & I_{\frac{w_{max}}{2}-1} & I_{\frac{w_{max}}{2}} & I_{\frac{w_{max}}{2}+1} & \cdots & I_{w_{max}-1} & I_1+I_{w_{max}} \\
I_1 & I_2 & I_3 & \dots & I_{\frac{w_{max}}{2}-1} & I_{\frac{w_{max}}{2}} & I_{\frac{w_{max}}{2}+1} & \cdots & I_{w_{max}-1} & I_{w_{max}}
\end{bmatrix}$$

With the matrix P, we can derive the stationary distribution of box's weight on the Markov Chain,

$$\pi = [\pi_1, \pi_2, \pi_3, \cdots, \pi_i, \cdots, \pi_{w_{max}-1}, \pi_{w_{max}}] \tag{3}$$

where the π_i represents the stationary probability for the box weight being i.

Suppose the current weight for a box is i, the possible weights for the next coming itemto be beyond the remaining capacity of current box can be denoted as $w_{max-i+1}, w_{max-i+2}, \dots, w_{max}$. Thus the probability of using a new box to accommodate the new item can be expressed as

$$p(w_{box} = i) = \pi_i \cdot (\sum_{j=w_{max}-i+1}^{w_{max}} I_j) \tag{4}$$

where the π_i is the probability of current box's weight being i and $\sum_{j=w_{max}-i+1}^{w_{max}} I_j$ is probability that the coming item's weight j is beyond the remaining capacity of current box. The latter is namely the probability of using a new box to accommodate the new item, which means the weight of current box would be unchanged.

The relative probability of different final weights in a box is,

$$p_i = \frac{\pi_i \cdot (\sum_{j=w_{max}-i+1}^{w_{max}} I_j)}{\sum_{i=1}^{w_{max}} \pi_i \cdot (\sum_{j=w_{max}-i+1}^{w_{max}} I_j)} \tag{5}$$

Thus the average weight per box is:

$$\overline{w}_{box} = \sum_{i=1}^{w_{max}} i \cdot p_i = \sum_{i=1}^{w_{max}} i \cdot \frac{\pi_i \cdot (\sum_{j=w_{max}-i+1}^{w_{max}} I_j)}{\sum_{i=1}^{w_{max}} \pi_i \cdot (\sum_{j=w_{max}-i+1}^{w_{max}} I_j)} \tag{6}$$

2.1 Mean Number of Items

The average number of items in a box, $\overline{N}$, can be evaluated by dividing the average weight of box with the average weigh of item,

$$\overline{N} = \frac{\overline{w}_{box}}{\overline{w}} \tag{7}$$

2.2 Mean Weight of Box

The weight of a box has already been derived with equation **??**.

2.3 Probability Mass Function of First Item Weight

The weight of first item in a box is denoted with Q. By applying the "Law of Total Probability", the weight of first item being i into any box can be expressed as the probability sum of the events in which the final weight of its previous box is j:

$$P(Q=i) = \sum_{j=w_{max}-i+1}^{w_{max}} P(\text{1st item weight} = i|\text{previous box weight} = j) \cdot P(\text{previous box weight} = j) \quad (8)$$

Here the j should be large than $w_{max} - i + 1$ in order to use a new box to package the new coming item with weight i. The weight of box and the weight of coming item are two independent terms. The formula 8 can be further clarified with the expressions below,

$$P(\text{first item weight} = i|\text{previous box weight} = j) = I_i \quad (9)$$

$$P(\text{previous box weight} = j) = \pi_j \quad (10)$$

where the P_{ij} is the transition probability of different weight of box i,j in the transition matrix P and p_j is the probability of final weight of a box being j.

Combining the formulas 8,9,10, we have the following probability mass function of first item weight i,

$$\begin{aligned} P(Q=i) &= \sum_{j=w_{max}-i+1}^{w_{max}} P(\text{first item weight} = i|\text{previous box weight} = j) \cdot P(\text{previous box weight} = j) \\ &= \sum_{j=w_{max}-i+1}^{w_{max}} I_i \cdot \pi_j \end{aligned} \quad (11)$$

The relative probability of the first items with different weights is then,

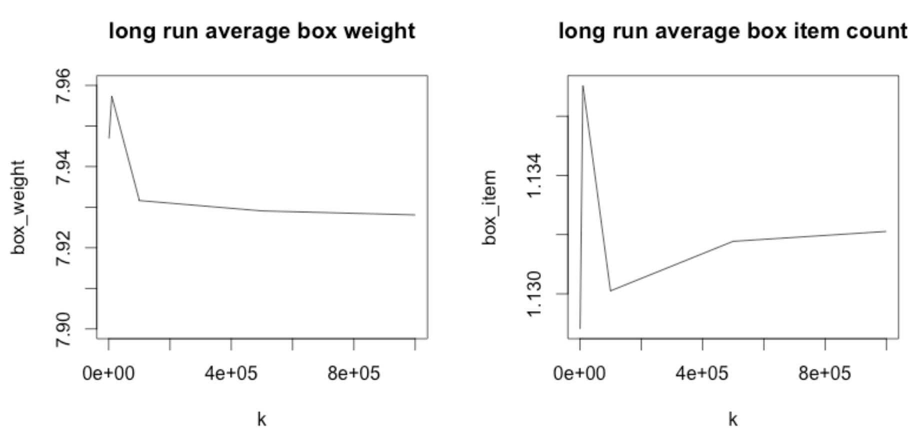
$$P(Q=i) = \frac{\sum_{j=w_{max}-i+1}^{w_{max}} I_i \cdot \pi_j}{\sum_{i=1}^{w_{max}} \sum_{j=w_{max}-i+1}^{w_{max}} I_i \cdot \pi_j} \quad (12)$$

2.4 Simulation

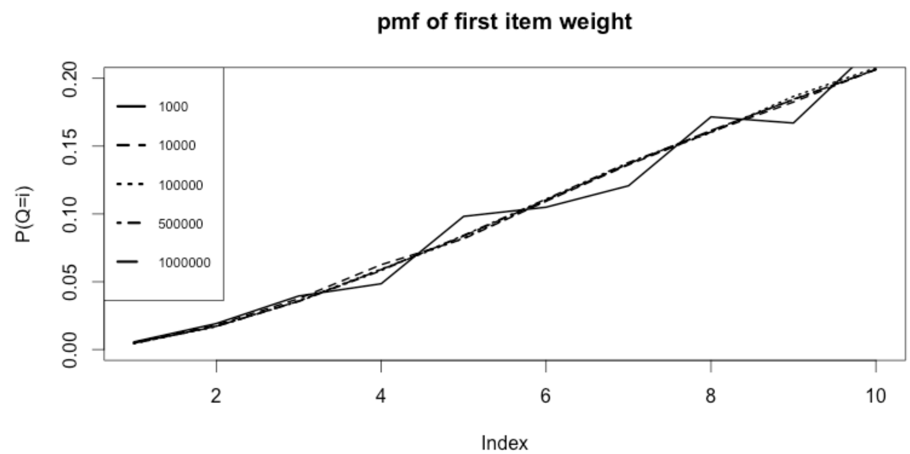
2.4.1 Simulation result

Using the code in Reference section, we can calculate long term average of item count, box weight, and first item weight in 1k, 10k, 100k,500k,1000k steps simulation, which separately represents the packaging of 1k, 10k, 100k, 500k and 1000k items.

From the following figure, we can see that long run average box weight and long run average box item count converge. Long run average box weight converges approximately to 7.928084 and long run average box item count converges approximately to 1.132108.



From the following figure, we can see that long run probability mass function for the first item weight converges. $P(Q = i), \forall i \in \{1, 2, ..., 10\}$ converges to $0.004741268, 0.017198984, 0.035708949, 0.058954521,$ $0.083682023, 0.110335240, 0.136964683, 0.161583502, 0.184244907, 0.206585925$



2.4.2 Analytical result

With our above formulas, we calculate the relative properties on the Markov Chain, the average of final box weight, the average number of items in a box and the probability distribution of first item weight.

```
> Box_Final_Weight_Average
[1] 7.927308
> ITEM_NUM
[1] 1.132473
> first_item_probability
          1           2           3           4           5
0.004650895 0.017247546 0.035881214 0.058755486 0.084176755
          6           7           8           9          10
0.110598922 0.136706700 0.161525751 0.184552633 0.205904098
```

In conclusion, our simulation verified that our solutions are correct!

3 Reference (code)

3.1 Simulation of Packaging Items

```
isGoodNumber<-function(X,n){
  ifelse(X==n, TRUE, FALSE)
}

stat<-function(X,n){
        t=list()
        len=length(X)
        for (i in 1:n){
                t[i]=length(X[isGoodNumber(X,i)])/len
        }
        return(t)
}
stat1<-function(X,n){
        t=rep(1,n)
        X=X[!is.na(X)]
        len=length(X)
        for (i in 1:n){
                t[i]=length(X[isGoodNumber(X,i)])/len
        }
        return(t)
}

Item_Box_Simulation<-function(n){
        c=1/(sum(1:10)/10)
        p=0.1*c*(1:10)
        wt_max=10

        itm_wt=sample(x=1:10,size=1,prob=p)
        bx_cnt=c(1,rep(NA,n-1))
        bx_crnt_wt=c(itm_wt,rep(0,n-1))
        bx_fnl_wt=rep(NA,n)
        bx_itm_num=rep(NA,n)
        frst_itm_wt=c(itm_wt,rep(NA,n))
        count=2
        itm_num=1
        while(count<=n){
                itm_wt=sample(x=1:10,size=1,prob=p)
                if ((itm_wt+bx_crnt_wt[count-1])>wt_max){
                        bx_cnt[count]=1
                        bx_crnt_wt[count]=itm_wt
                        bx_fnl_wt[count-1]=bx_crnt_wt[count-1]
                        bx_itm_num[count-1]=itm_num
                        itm_num=1
                        frst_itm_wt[count]=itm_wt
                }
                else{
                        bx_crnt_wt[count]=itm_wt+bx_crnt_wt[count-1]
                        itm_num=itm_num+1
                }
                count=count+1
```

```
        }
        cwt_sta_dis=stat(bx_crnt_wt,10)
        fwd_sta_dis=stat1(bx_fnl_wt,10)
        fiw_sta_dis=stat1(frst_itm_wt,10)
        bx_fnl_wt_av=mean(bx_fnl_wt,na.rm=T)
        bx_itm_num_av=mean(bx_itm_num,na.rm=T)
        frst_itm_wt_av=mean(frst_itm_wt,na.rm=T)

        #t<-list('current_weight_distribution'=cwt_sta_dis,'final_weight_distribution'=fwd
        t<-list('Box_final_Weight'=bx_fnl_wt_av,'Box_Item_Number'=bx_itm_num_av,'First_Item
        #t=list()
        #t['current_weight_distribution']=cwt_sta_dis
        #t['final_weight_distribution']=fwd_sta_dis
        #return(list(Box_Weight_Average=bx_fnl_wt_av,Box_Item_Number=bx_itm_num_av,First_Ite
        return(t)
}
```

3.2 Analytically Computed Values

```
############ ITEM WEIGHT & PROBABILITY ############
I<-c(1:10)/10*10/55
SUM<-sum(I)
#I<-c(1:10)
W_I<-c(1:10)*1.0
W_I_av<-sum(I*W_I)

################ TRANSITION MATRIX ################
pijdef<-function(I){
        n<-length(I)
        p<-diag(0,n)
        for (i in 1:n){
                for (j in 1:n){
                        if (i>j){
                                if ((i+j)>n) p[i,j]<-0+I[j]
                                else p[i,j]<-0
                        }
                        if (i<j){
                                if ((i+j)>n) p[i,j]<-I[j-i]+I[j]
                                else p[i,j]<-I[j-i]
                        }
                        if (i==j & (i+j)>n) p[i,j]<-I[j]
                }
        }
        return(p)
}

P<-pijdef(I)

#################    MARKOV CHAIN    #################
fsd<-function(p) {
        n<-nrow(p)
        nwmtrx<-diag(n)-t(p)
        nwmtrx[n,]<-rep(1,n)
        rt<-c(rep(0,n-1),1)
```

```
        sol<-solve(nwmtrx,rt)
        return(sol)
}

PI<-fsd(P)

############# BOX WEIGHT & PROBABILITY ############
###### BOX PROBABILITY ######
P_box<-function(I,PI){
        n<-length(I)
        pbox=rep(1,length(I))
        for (i in 1:n){
                tot<-sum(I[(n-i+1):n])
                pbox[i]<-PI[i]*tot
        }
        return(pbox)
}

pbox<-P_box(I,PI)
pbox<-pbox/sum(pbox)

######## BOX WEIGHT ########
W_B<-c(1:length(I))*1.0
Box_Final_Weight_Average<-sum(pbox*W_B)

####### BOX ITEM NUM #######
ITEM_NUM<-Box_Final_Weight_Average/W_I_av

###### FIRST ITEM NUM ######
P_first_weight_item<-function(PI,I){
        n<-length(I)
        pfirstitem<-rep(1,n)
        for (i in 1:n){
                tot1<-0
                for (j in (n-i+1):n){
                        tot1<-tot1+PI[j]*I[i]
                }
                pfirstitem[i]<-tot1
        }
        return(pfirstitem)
}

first_item_probability<-P_first_weight_item(PI,I)
first_item_probability<-first_item_probability/sum(first_item_probability)
names(first_item_probability)<-c(1:length(I))
Weight_First_Item_average<-sum(first_item_probability*W_I)
```