Stat 102C HW3: Answer Key

Muzhou Liang, Jonathan Arfa

May 21, 2014

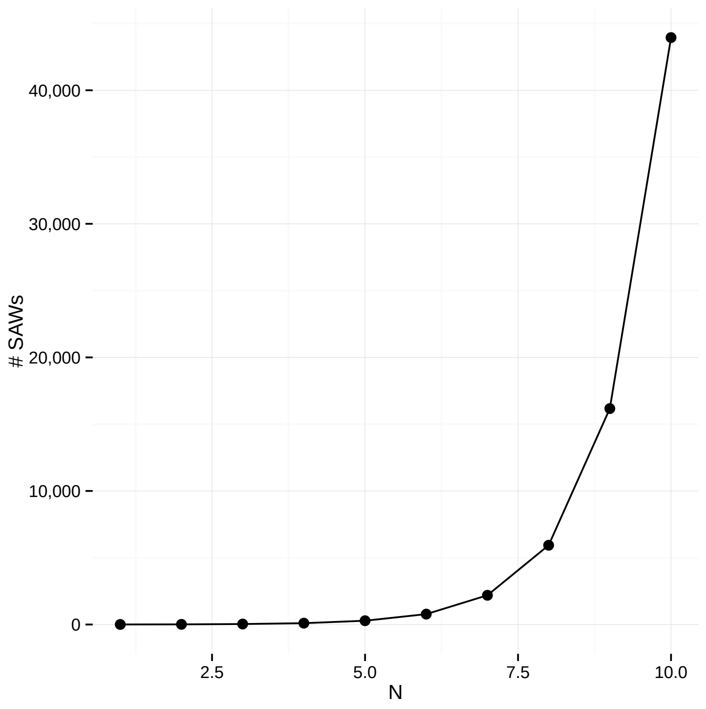
Problem 1

```
> SAW = function(n = 5){
+   #create a map, every cell is FALSE because it's unexplored
+   map = matrix(FALSE, nrow = 2*n+1, ncol = 2*n+1)
+   #initially we're at the center
+   r = c = n+1
+   w = 1
+   for (i in 1:n) { #for every step of our path
+     #set our current position to TRUE
+     map[r, c] = TRUE
+     opt = c('up', 'right', 'down', 'left')
+     #Note that r,c==1,1 denotes the uppermost left corner of the map.
+     # if up is explored already, delete "1" in options
+     if (map[r-1, c] == TRUE) opt = opt[-which(opt == 'up')]
+     if (map[r, c+1] == TRUE) opt = opt[-which(opt == 'right')]
+     if (map[r+1, c] == TRUE) opt = opt[-which(opt == 'down')]
+     if (map[r, c-1] == TRUE) opt = opt[-which(opt == 'left')]
+     nopt = length(opt)
+     if (nopt == 0) {
+       w = 0
+       break
+       #so if our path is stuck, end the walk and give this walk a weight of 0.
+     }
+     #w is the 1/(probability of a path), which is the product of 1/probability
+     #of each step.
+     w = w * nopt
+     dir = sample(opt, 1) #take one direction from the available ones
+     if (dir == 'up'){
+       r = r - 1
+     } else if (dir == 'right'){
+       c = c + 1
+     } else if (dir == 'down'){
+       r = r + 1
+     } else {
```

```
+       c = c - 1
+     }
+   }
+   pathlength = sum(map)
+   #pathlenth isn't necessary for this problem, but the distributions
+   #of path lenghts might be cool to look at later
+   res = list("w" = w, "length" = pathlength)
+   return(res)
+ }
> mean(replicate(1e4, SAW(10)$w))

[1] 44033.06

> #if you don't understand what the above line does, research
> #what the replicate functon does
```

How does the number of walks change with N?

```
> N = 1:10
> walks = numeric(length(N))
> Nsim = 1e4
> for(n in N) #If you don't have much RAM, this could take some time. Reduce Nsim or N
+ {
+   w = replicate(Nsim, SAW(n)$w)
+   walks[n] = mean(w)
+ }
> library(ggplot2)
> library(scales)
> ggplot(data.frame(N, walks), aes(x=N, y=walks)) + geom_point(size=3) +
+   geom_line() + theme_minimal() + scale_y_continuous('# SAWs', labels=comma)
```



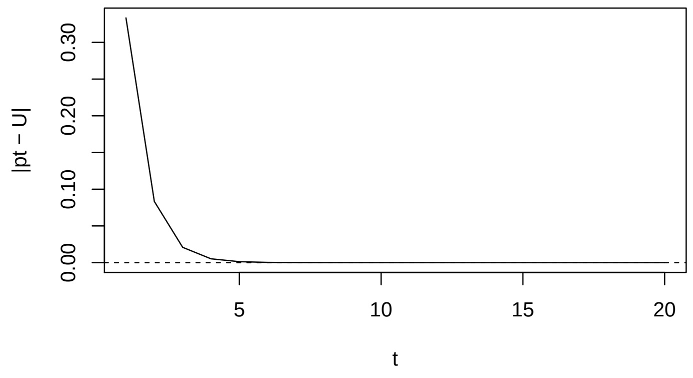
Problem 2

$$p^{(t+1)}(y) = \sum_x p^{(t)}(x,y) = \sum_x p^{(t)}(x)p(y|x) = \sum_x p^{(t)}(x)K(x,y)$$

```
> RW3 = function(t = 1e2, p = c(1, 0, 0), k11 = 0.5, k12 = 0.25,
+                k13 = 0.25, k21 = 0.25, k22 = 0.5, k23 = 0.25,
+                k31 = 0.25, k32 = 0.25, k33 = 0.5) {
+   states = c(1, 2, 3)
+   K = matrix(c(k11, k12, k13, k21, k22, k23, k31, k32, k33), 3, byrow=TRUE)
+   if(!(sum(K[1,]) == 1 & sum(K[2,]) == 1 & sum(K[3,]) == 1)) {
+     print('Row sum is not 1!')
+     break
+   }
+   for (i in 1:t) {
+     p1 = sum(p * K[, 1])
+     p2 = sum(p * K[, 2])
+     p3 = sum(p * K[, 3])
+     p = c(p1, p2, p3)
+   }
+   return(p)
+ }
```

We can set different $p^{(0)}$ as (1, 0, 0), (0, 1, 0), (0, 0, 1). The results are shown as below. We find that no matter what $p^{(0)}$ is, $p^{(t)}$ is always close to the uniform distribution.

```
> RW3(p = c(1, 0, 0))

[1] 0.3333333 0.3333333 0.3333333

> RW3(p = c(0, 1, 0))

[1] 0.3333333 0.3333333 0.3333333

> RW3(p = c(0, 0, 1))

[1] 0.3333333 0.3333333 0.3333333

> RW3Unif = function(t = 1e2,  p = c(1, 0, 0), k11 = 0.5, k12 = 0.25,
+                    k13 = 0.25, k21 = 0.25, k22 = 0.5, k23 = 0.25,
+                    k31 = 0.25, k32 = 0.25, k33 = 0.5) {
+   z = matrix(0, nrow = t, ncol = 3)
+   for (i in 1:t) {
+     z[i, ] = RW3(i, p, k11, k12, k13, k21, k22, k23, k31, k32, k33)
+   }
+   diff = apply(z, 1, FUN = function(x){sum(abs(x - 1/3))})
+   return(diff)
+ }
> t = 20
> abdiff = RW3Unif(t)
> plot(1:t, abdiff, type = 'l', xlab = 't', ylab = '|pt - U|')
> abline(h = 0, lty = 2)
```



The result below suggets that $p^{(t)}$ will be close to another distribution if you change the transition matrix.

```
> RW3(t = 1e2, k12 = 1/6, k13 = 1/3, k31 = 1/12,
+     k32 = 1/6, k33 = 3/4)

[1] 0.2142857 0.2500000 0.5357143
```

Problem 3

```
> RWM = function(t = 1e2, M = 1e4, p0 = 1, k11 = 0.5, k12 = 0.25,
+                k13 = 0.25, k21 = 0.25, k22 = 0.5, k23 = 0.25,
+                k31 = 0.25, k32 = 0.25, k33 = 0.5) {
+   states = c(1, 2, 3)
+   K = matrix(c(k11, k12, k13, k21, k22, k23, k31, k32, k33), 3, byrow=TRUE)
+   if(!(sum(K[1,]) == 1 & sum(K[2,]) == 1 & sum(K[3,]) == 1)) {
+     print('Row sum is not 1!')
+     break
+   }
+   people = rep(p0, M)
+   for (i in 1:t) {
+     # get the number of people at 1
+     n1 = sum(people == 1)
+     n2 = sum(people == 2)
+     n3 = sum(people == 3)
+     # get the index of people at 1
+     index1 = which(people == 1)
+     index2 = which(people == 2)
+     index3 = which(people == 3)
+     # draw the next states for those at 1
+     move1 = sample(states, n1, replace = TRUE, prob = K[1,])
+     move2 = sample(states, n2, replace = TRUE, prob = K[2,])
+     move3 = sample(states, n3, replace = TRUE, prob = K[3,])
+     people[index1] = move1
+     people[index2] = move2
+     people[index3] = move3
+   }
+   p1 = sum(people == 1)/M
+   p2 = sum(people == 2)/M
+   p3 = sum(people == 3)/M
+   p = c(p1, p2, p3)
+   return(p)
+ }
```

The estimated $p^{(t)}$ is shown as below.

```
> RWM(t = 1e2, M = 1e5)

[1] 0.33425 0.33247 0.33328
```

The estimated $p^{(t)}$ will be different if we change the transition probabilities.

```
> RWM(t = 1e2, M = 1e5, k12 = 1/6, k13 = 1/3, k31 = 1/12,
+     k32 = 1/6, k33 = 3/4)

[1] 0.21396 0.25078 0.53526
```