Assignment 3 for CS203: Probability for Computer Science

Abhyuday Pandey
170039

Problem 1: Subpart 1

Strategy

For given position $i \in [1, N-1]$ we make decision whether to head in a direction till house is reached. Performed *iters* times which is by default set to 10^6.

Source Code Dependencies

- Standard Library for C (cstdlib)
- GNUplot

Reproducibility

- *rand*() modulo 2 used to make a decision.
- Set iters equal to 10^6 for $n = 20$
- The probability distribution must lie in the interval **[0,1)**.

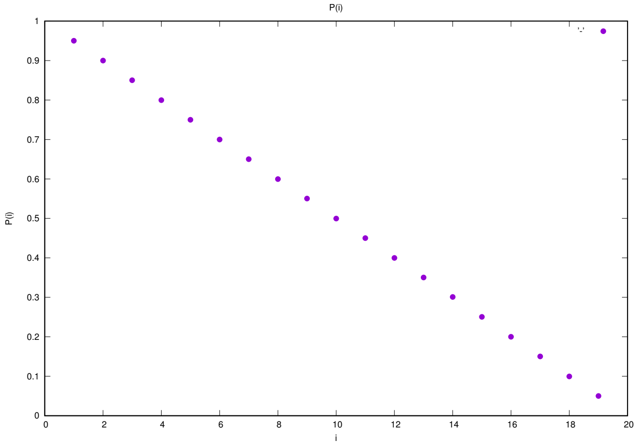

Table 1: Probability distribution of Z

Z	$P(Z)$	Z	$P(Z)$
1	0.950120	11	0.450064
2	0.900023	12	0.399925
3	0.850117	13	0.350256
4	0.799820	14	0.300984
5	0.749972	15	0.250554
6	0.699899	16	0.199942
7	0.650132	17	0.150257
8	0.599658	18	0.099600
9	0.550316	19	0.049927
10	0.499695		

Conclusion

The following conclusions were drawn:

- For each i, probability gets decremented by about 5% and the plot obtained is linear.
- In agreement with the fact that $p(i) + p(20 - i) = 1$.

Problem1:Subpart 2

Strategy

For given position $i \in \{1, 10\}$ we make decision whether to head in a direction till house is reached. Performed *iters* times which is by default set to 10^4. Also, in this case we need to limit number of steps (*steps*) to make the running time of this program finite.

Source Code Dependencies

- Standard Library for C (cstdlib)

Reproducibility

- *rand*() modulo 2 used to make a decision.
- Set iters equal to 10^4 for $steps \in \{10^8, 10^7...10^1\}$
- The probability distribution must lie in the interval **[0,1)**.

Table 1

$steps$	$P(i = 1)$	$P(i = 10)$
10^8	1.0000	0.9991
10^7	0.9999	0.9977
10^6	0.9997	0.9933
10^5	0.9982	0.9742
10^4	0.9913	0.9196
10^3	0.9768	0.7504
10^2	0.9205	0.3155
10^1	0.7534	0.0008

Conclusion

The following conclusions were drawn:

- Both the probabilities converge to 1 as number of steps increase.
- For $steps = 10^8$ $P(i = 1) = 1.0000$ and $P(i = 10) = 0.9991$

Problem 2

Strategy

For given position $(x, y) = (1, 0)$ we make decision whether to head in a direction of all four in 2 dimensions till house is reached. Performed *iters* times which is by default set to 10^4. Also, in this case we need to limit number of steps (*steps*) to make the running time of this program finite.

Source Code Dependencies

- Standard Library for C (cstdlib)

Reproducibility

- *rand*() modulo 4 used to make a decision.
- Set iters equal to 10^4 for $steps \in \{10^7, 10^6...10^1\}$
- The probability distribution must lie in the interval **[0,1)**.

Table 1

$steps$	$P((x, y))$
10^7	0.8366
10^6	0.8094
10^5	0.7827
10^4	0.7396
10^3	0.6785
10^2	0.5840
10^1	0.4263

Conclusion

The following conclusions were drawn:

- The successive probabilities have less difference as *steps* increases.
- For $steps = 10^7$ $P((x, y)) = 0.8366$

Problem 3

Strategy

For given position $(x, y, z) = (1, 0, 0)$ we make decision whether to head in a direction of all six in 3 dimensions till house is reached. Performed *iters* times which is by default set to 10^4. Also, in this case we need to limit number of steps (*steps*) to make the running time of this program finite.

Source Code Dependencies

- Standard Library for C (cstdlib)

Reproducibility

- *rand*() modulo 6 used to make a decision.
- Set iters equal to 10^4 for $steps \in \{10^7, 10^6 ... 10^1\}$
- The probability distribution must lie in the interval **[0,1)**.

Table 1

$steps$	$P((x, y, z))$
10^7	0.3346
10^6	0.3429
10^5	0.3363
10^4	0.3371
10^3	0.3323
10^2	0.3165
10^1	0.2563

Conclusion

The following conclusions were drawn:

- The successive probabilities have less difference as *steps* increases and converges near a value.
- For $steps = 10^7$ $P((x, y, z)) = 0.3346$

1 Problem 1: Subpart 1

```
//Abhyuday Pandey Codes September 3rd 2018
//This code is for Random walk for finite distance in 1-D
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;

int main(){
    //clock will give running time
    clock_t start=clock();
    //Set N=20 and iters=10^6, p[N] will store p(i)*iters for each i
    int N=20,iters=1000000;
    int p[N];
    //Initialize all p[i]'s to zero
    for(int i=0;i<N;i++) p[i]=0;
    //This loop will run exactly iters times
    while(iters--){
        //For each 'i' we compute probability
        for(int i=1;i<N;i++){
            int steps=i;
            //This loop will terminate when house/bar is reached
            while(1){
                int random=rand()%2;
                if(random) steps++;
                else steps--;
                if(steps==0){p[i]++; break;}
                if(steps==N){break;}
            }
        }
    }
    printf("---------%f--------\n",(double)(clock()-start)/CLOCKS_PER_SEC);
    //--------------------PLOTTING PART--------------------------------
    //Graph plotting
    long long int sum=0;
    FILE * gnuplotPipe = popen ("gnuplot -persistent", "w");
    fprintf(gnuplotPipe, "set terminal postscript eps enhanced color font
         'Helvetica,10'\n");
    fprintf(gnuplotPipe, "set output 'graph31.eps'\n");
    fprintf(gnuplotPipe, "set xlabel 'i'\nset ylabel 'P(i)\nset title 'P(i)'\n");
    fprintf(gnuplotPipe, "plot '-' pointtype 7\n");
    for( int idx = 1 ; idx < N ; idx++ ){
            fprintf(gnuplotPipe,"%lf %lf\n", (double)(idx),
                (double)(p[idx])/1000000.0);
            printf("For i=%d, p(i)=%lf\n",idx, (double)(p[idx])/1000000.0);
    }
    fprintf(gnuplotPipe, "e\n");
    //Graph end
    return 0;
}
```

2 Problem 1: Subpart 2

```
//Abhyuday Pandey Codes September 3rd 2018
//This code is for Random walk in infinite distance in 1-D
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
//Helper function to perform experiment iters times and given threshold on steps
void call(int iters, int steps, int i){
    int sum=0;
    //This loop runs iters times
    while(iters--){
        int pos=i;
        int s=steps;
        //This loop terminates after origin is reached or threshold steps are
            reached
        while(s--){
            if(rand()%2) pos++;
            else pos--;
            if(pos==0){sum++; break;}
        }
    }
    //print the probability for given i, steps and iters=10,000
    printf("For %d steps P(i=%d)=%lf\n",steps,i,(double)(sum)/10000.0);
}
int main(){
    //clock will give running time
    clock_t start=clock();
    //iters is number of times experiment is run (default 10,000)
    int iters=10000;
    //Each step in {10^8,10^7,10^6,10^5...10}
    for(int steps=100000000;steps>=10;steps/=10) call(iters,steps,1);
    for(int steps=100000000;steps>=10;steps/=10) call(iters,steps,10);
    printf("---------%f--------\n",(double)(clock()-start)/CLOCKS_PER_SEC);
    return 0;
}
```

3 Problem 2

```
//Abhyuday Pandey Codes September 3rd 2018
//This code is for Random walk in infinite distance in 2-D
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
//Helper function to perform experiment iters times and given threshold on steps
void call(int iters, int steps,int x,int y){
    int sum=0;
    //This loop runs iters times
    while(iters--){
        int pos1=x,pos2=y;
        int s=steps;
        //This loop terminates after origin is reached or threshold steps are
             reached
        while(s--){
            int randint=rand()%4;
            if(randint==3) pos1++;
            else if(randint==2) pos1--;
            else if(randint==1) pos2++;
            else pos2--;
            if(pos1==0&&pos2==0){sum++; break;}
        }
    }
    //print the probability for given (x,y), steps and iters=10,000
    printf("P(i)=%lf\n",(double)(sum)/10000.0);
}
int main(){
    //clock will give running time
    clock_t start=clock();
    //iters is number of times experiment is run (default 10,000)
    int iters=10000;
    //Each step in {10^7,10^6,10^5...10}
    for(int steps=10000000; steps>=10;steps/=10) call(iters,steps,1,0);
    printf("---------%f--------\n",(double)(clock()-start)/CLOCKS_PER_SEC);
    return 0;
}
```

4 Problem 3

```
//Abhyuday Pandey Codes September 3rd 2018
//This code is for Random walk in infinite distance in 3-D
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
//Helper function to perform experiment iters times and given threshold on steps
void call(int iters, int steps,int x,int y,int z){
    int sum=0;
    //This loop runs iters times
    while(iters--){
        int pos1=x,pos2=y,pos3=z;
        int s=steps;
        //This loop terminates after origin is reached or threshold steps are
             reached
        while(s--){
            int randint=rand()%6;
            if(randint==5) pos3++;
            else if(randint==4) pos3--;
            else if(randint==3) pos1++;
            else if(randint==2) pos1--;
            else if(randint==1) pos2++;
            else pos2--;
            if(pos1==0&&pos2==0&&pos3==0){sum++; break;}
        }
    }
    //print the probability for given (x,y,z), steps and iters=10000
    printf("Given %d steps,P(i)=%lf\n",steps,(double)(sum)/10000.0);
}
int main(){
    //clock will give running time
    clock_t start=clock();
    //iters is number of times experiment is run (default 10,000)
    int iters=10000;
    //Each step in {10^7,10^6,10^5...10}
    for(int steps=10000000;steps>=10;steps/=10) call(iters,steps,1,0,0);
    printf("---------%f--------\n",(double)(clock()-start)/CLOCKS_PER_SEC);
    return 0;
}
```