○ 실험보고서

확률과통계 HW#1

제출일 : 05월 09일

2018008559 컴퓨터소프트웨어학부 신상윤

1. Introduction

1.1 실험목적

0~9 정수를 Random으로 생성하고, 0이 발생한 경우를 사건으로 하여 0이 발생한 횟수와, 0이 발생한 횟수 사이의 시간 간격을 확률분포로 구하여 수학적으로 정의된 Poisson, Exponential, Erlang-2 확률분포와 비교하고, 수학적 확률분포와 난수 발생으로부터 구한 히스토그램 사이의 차이를 분석하고, 수학적 확률분포와 얼마나 잘 맞는지를 검토한다. 얼마나 많이 어떤 방식으로 난수 발생을 하는지, 시간 간격을 어떻게 설정해야 할지 생각해본다.

1.2 배경 이론

1.2.1 이항 분포

연속된 n번의 독립적 시행에서 각 시행이 확률 p를 가질 때의 이산 확률분포이다. 각 시행이 실행될 횟수를 확률변수로 놓으면, 각 시행이 x번 실행될 확률은

$P(x = X) = \binom{n}{x} p^x (1-p)^{n-x}$이다. 이때의 기댓값과 분산은 각각

$E(X) = np$, $V(X) = np(1-p)$이다.

1.2.2 Poisson 분포

푸아송분포는 단위 시간 안에 어떤 사건이 몇 번 발생할 것인지를 표현하는 이산 확률분포이다. 정해진 시간 안에 어떤 사건이 일어날 횟수에 대한 기댓값을 λ라고 했을 때, 그 사건이 n회 일어날 확률을

$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$로 정의한다. 이때 평균과 분산은 모두 λ이다.

이항 분포의 기댓값을 λ라 하자. $np = \lambda$, $p = \frac{\lambda}{n}$ 즉,

$P(X = x) = \binom{n}{x} (\frac{\lambda}{n})^x (1-\frac{\lambda}{n})^{n-x} = \frac{\lambda^k}{k!} \frac{n(n-1)\cdots(n-k+1)}{n^k} (1-\frac{\lambda}{n})^{(-\frac{n}{\lambda})(-\lambda)}$에서

n이 ∞로 갈 때 $P(X = x) = \frac{\lambda^k}{k!} e^{-\lambda}$가 되고, 이는 푸아송분포와 같다.

1.2.3 Exponential 분포(지수분포)

일정시간 동안 발생하는 사건의 횟수가 푸아송분포를 따른다면, 다음 사건이 일어날 때까지 대기시간은 지수분포를 따른다. 확률 밀도함수는

$P(k) = \lambda e^{-\lambda k}$ 평균과 분산은 각각 $E(X) = \frac{1}{\lambda}$, $V(X) = \frac{1}{\lambda^2}$이다.

1.2.4 Erlang 분포

지수분포의 확장이다. 어떤 사건과 k번째에 따라오는 사건의 시간 간격이 확률변수이다. 확률 밀도함수는

$P(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}$ (k=1,2,3...), $E(X) = \frac{k}{\lambda}$, $V(X) = \frac{k}{\lambda^2}$

2. Process

2.1 Poisson 분포

시간 간격을 1000으로 잡고 그 안에서 일어난 사건(0 발생)의 횟수를 구하는 코드이다. 1000번 반복. 0~9 정수를 Random으로 생성하고, 0이 발생한 경우를 사건으로 하므로 이는 이항분포를 따른다. 이때 사건의 시행횟수인 n이 커질수록 이항분포는 푸아송분포에 가까워질 것이다.

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int main(){
    srand(time(NULL));                  -> 랜덤하게 숫자 생성
    float a[1001] = {0};                -> 사건의 횟수를 저장 cnt번 발생했으면 a[cnt]++가 된다.
    for(int i=0; i<1000; i++){          -> 실험 1000번 진행
        int cnt = 0;                    -> 사건의 횟수를 저장하는 변수 각 실험마다 0으로 초기화
        for(int j=0; j<1000; j++){      -> 시간간격을 1000으로 잡았다
            if(rand()%10 == 0)          -> 사건이 발생하면
                cnt++;                      사건의 횟수를 저장하는 변수를 증가시킨다. 이를 반복
        }
        a[cnt]++;
    }
    int cnt1 = 0;                       -> n = 1000, p = 0이 나올 확률 = 0.1이므로 λ=100인
    for(int i=60; i<140; i++){              푸아송분포가 나올 것이다. λ=100인 푸아송분포에서
        printf("%.3f\n",a[i]/1000);         x가 60이하 이거나 140이상일 확률은 0에 가까우므로
        cnt1 += a[i];                       60에서 140까지만 나오도록 설정했다.
```

```
    }
    printf("%d",cnt1);
    return 0;
}
```

이때 60이하 140이상인 사건이 발생할 수는 있으므로 정확도를 따지는 변수 cnt1을 선언하고, 출력하도록 했다. 현재 실험에서는 cnt가 1000이면 모든 표본이 60과 140 사이에 있다고 할 수 있다.

이 코드를 n 값에 따라 3번씩 실행했다.

2.2 Exponential 분포

현재 실험은 이항분포를 따르고 있고 n이 커질수록 푸아송분포와 비슷해진다. 따라서 사건이 일어나고 다음 사건이 일어날 때까지 대기시간은 지수분포를 따를 것이다. 이때 대기시간은 0과 0사이의 숫자개수로 할 것이다. 즉 0이 나오는 사건이 일어나고 다음에 바로 0이 나온다면 대기시간은 0이다.

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int main(){
    srand(time(NULL));
    float a[1001] = {0};
    int cnt1 = 0;                       -> 푸아송 코드에서는 for문으로 실험횟수를 조절했지만 지수
    for(int i=0; ;i++){                    분포에서는 시간 간격을 측정하는 것이므로 시간간격을
        int cnt = 0;                       cnt1 = 1000번 측정할 때까지 for문을 무한번 돌린다.
        if(rand()%10 == 0){             -> 0인 사건이 일어났으니 간격을 이제 측정할 것이다.
            for(int j=0; ;j++){         -> 다음 0인 사건이 일어날 때까지 실행
                if(rand()%10 == 0){     -> 일어나면 그때의 횟수를 저장하고 for문을 탈출한다.
                    a[cnt]++;
                    break;
                }
                cnt++;
            }
            cnt1++;
        }
        if(cnt1 == 1000)                -> 1000번 측정을 완료하면 for문을 탈출
            break;
    }
    int cnt2 = 0;
    for(int i=0; i<70; i++){            -> x가 0일 때 즉, 0이 나오고 다시 0이 나올 확률은
        printf("%.3f\n",a[i]/1000);        0.1이다. 따라서 P(x = 0) = 0.1 = λe^(-λx)이고
        cnt2 += a[i];                      λ는 결국 0.1이다. λ가 0.1인 지수분포가 나올 것이다.
    }                                      λ가 0.1인 지수분포 그래프에서 x가 70이상일 확률이
    printf("%d",cnt2);                     0에 가까우므로 0에서 70까지만 나오도록 설정했다.
    return 0;                              70이상인 경우가 나올수는 있으므로 정확도를 따지는
}                                          변수 cnt2를 선언하고, 출력하도록 했다.
```

이 코드도 n 값에 따라 3번씩 실행했다.

2.3 Erlang-2 분포

지수분포와 비슷하게 구현하면 되는데 0을 세는 변수를 하나 추가해주면 된다. 0을 세다가 2번째 셌을 때 배열에 저장하고 for 문을 탈출한다.

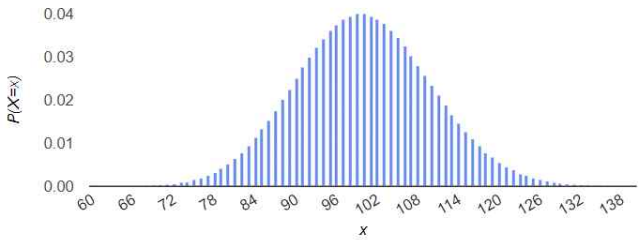
```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int main(){
    srand(time(NULL));
    float a[1001] = {0};
    int cnt1 = 0;
    for(int i=0; ;i++){
        int cnt = 0;
        int cnt0 = 0;                    -> 0이 몇 번 나왔는지 세는 변수
        if(rand()%10 == 0){              -> 0인 사건이 일어나고 측정 시작, 현재 cnt0 = 0
            for(int j=0; ;j++){
                if(rand()%10 == 0){      -> 다음 0이 나왔을 때 cnt0 = 1이면 0이 2번째 나오는 것이므로
                    if(cnt0 == 1){          배열에 저장하고 break
                        a[cnt]++;           cnt0이 1이 아니면 첫 번째 0인 것이므로 cnt0++하고
                        break;              for문 이어서 진행
                    }
                    cnt0++;
                }
                cnt++;
            }
            cnt1++;
        }
        if(cnt1 == 1000)                 -> 1000번 측정을 완료하면 for문을 탈출
            break;
    }
    int cnt2 = 0;
    for(int i=0; i<80; i++){             -> 감마분포에서 2번째에 따라오는 사건의 시간 간격이므로 α = 2
        printf("%.3f\n",a[i]/1000);         위의 지수분포에서 구한 λ가 0.1이었으므로 β = 0.1
        cnt2 += a[i];                    따라서 α가 2이고, β가 0.1인 감마 분포가 나올 것이다.
    }                                       감마분포 그래프에서 x가 90이상일 확률이 0에 가까우므로
    printf("%d",cnt2);                      0에서 90까지만 나오도록 설정하였다.
    return 0;                            90이상인 경우가 나올수는 있으므로 정확도를 따지는
}                                        변수 cnt2를 선언하고, 출력하도록 했다.
```

이 코드도 n 값에 따라 3번씩 실행했다.

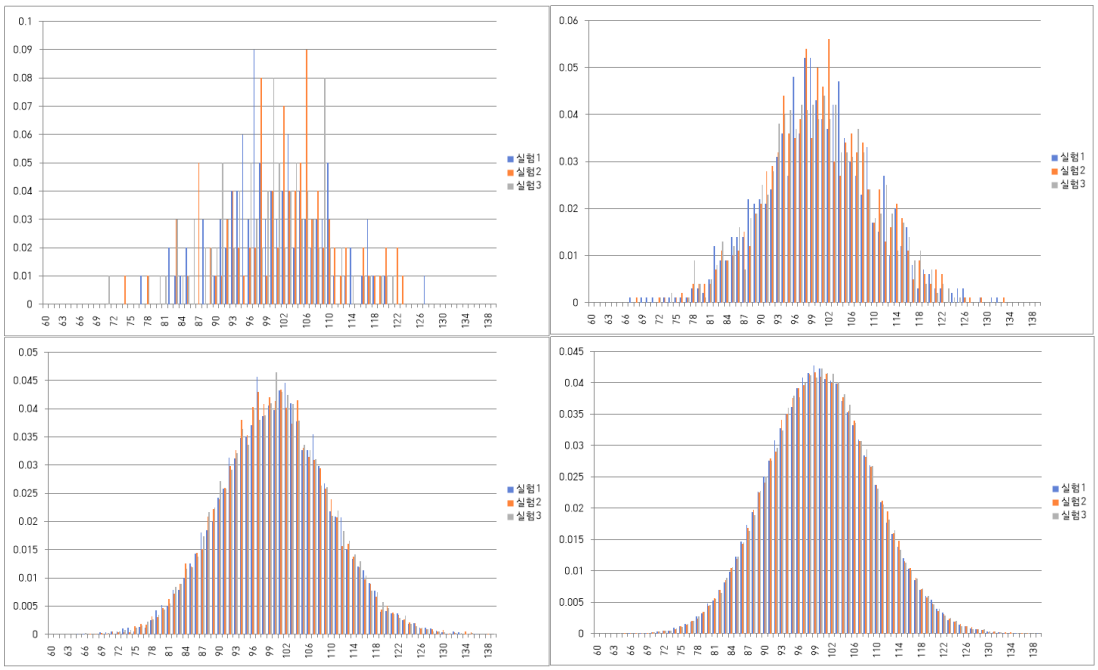
3. Result

3.1 Poisson 분포

λ가 100일 때 즉, n = 1000일 때 푸아송분포이다.

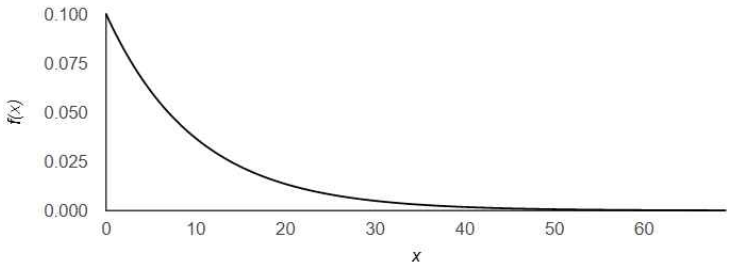


구간 n인 난수 발생 실험을 각각 100, 1000, 10000, 100000회 진행했을 때 히스토그램이다.



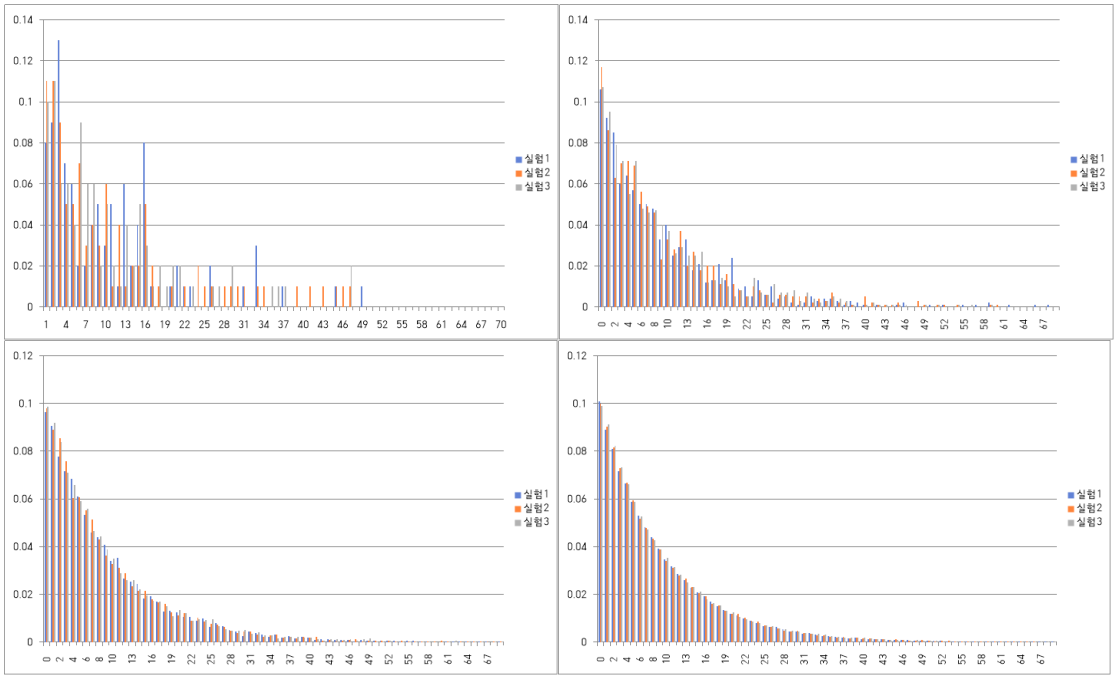
3.2 Exponential 분포

λ가 0.1인 지수분포이다.



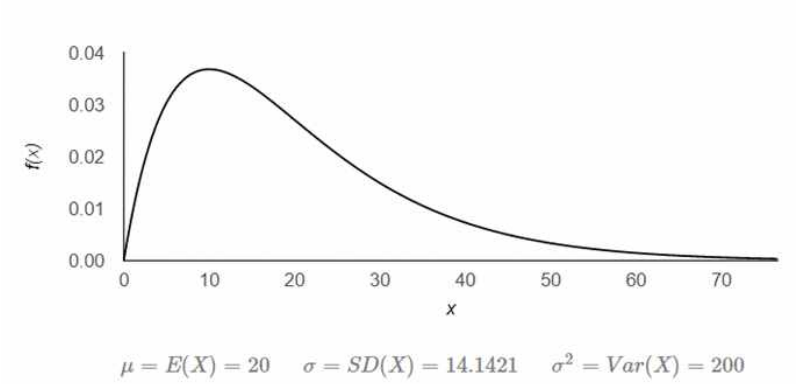
$$\mu = E(X) = 10 \quad \sigma = SD(X) = 10 \quad \sigma^2 = Var(X) = 100$$

시간 간격 측정을 각각 100, 1000, 10000, 100000회 진행했을 때 히스토그램이다.

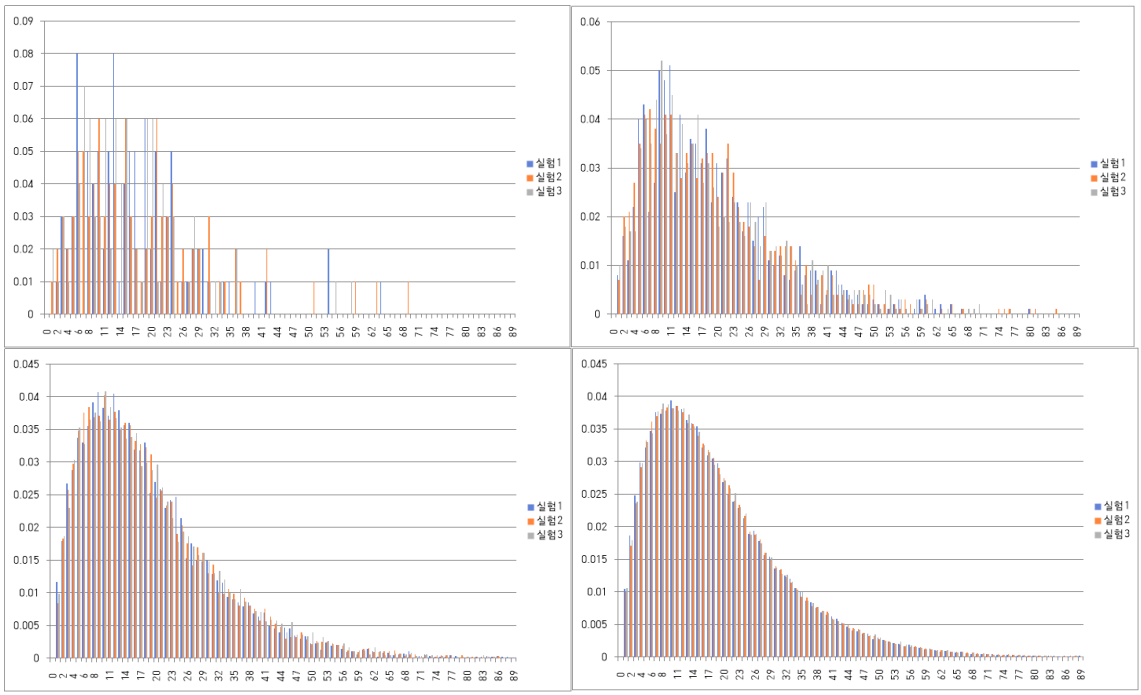


3.3 Erlang-2 분포

α가 2, β가 0.1인 감마분포이다. (= 얼랑-2 분포)



시간 간격 측정을 각각 100, 1000, 10000, 100000회 진행했을 때 히스토그램이다.
0이 세 번 나올때 간격이 0이 될 수 없으므로 p(X=0) = 0이다.
또한 p(X = 1)은 000이 이어서 나와야 하므로 연속으로 0이 나올 확률
0.1 * 0.1 = 0.01에 근접할 것임을 예상할 수 있다.



3.4 결과 분석

공통적으로 실험횟수가 늘어나면서 점점 각 분포 그래프와 비슷해져 간다. 특히 n=100000일 때는 매끄러운 그래프가 보이며 각 분포 그래프와 값이 거의 일치한다. 수학적 확률분포와 비교해봤을 때 가장 큰 차이점은 수학적 확률분포는 연속적이고 어떤 값이 나타날 확률을 나타낸 것이기 때문에 예를 들어 100이 나타날 확률보다 101이 나타날 확률이 작다고 하면 수학적 확률분포에서는 연속적으로 나타내기 위해 매끄럽게 작아지지만 난수 발생 히스토그램에서는 101이 100보다 많이 일어날 수도 있고 99보다 많이 일어날 수도 있으므로 그래프가 매끄럽지 않다. 즉, 수학적 확률분포는 말 그대로 각각이 일어날 확률을 나타낸 것이므로 확률적으로 어떤거는 작고, 어떤거는 크다를 나타내지만 실제 실험에서는 실제 일어난 일이므로 꼭 확률대로 100번 시행하면 30번 나오는 것이 아닌 29번이 나올 수도 있고 10번도 나올 수 있다. 이는 n = 100일 때 실험결과로 확인할 수 있다.

현재 난수를 발생하는 실험은 이항분포를 따른다. 이항분포는 n이 커질수록 푸아송분포에 가까워지므로 실험횟수가 증가할수록 푸아송분포에 가까워지는 것을 확인할 수 있다. 마찬가지로 실험횟수가 푸아송분포에 가까워지므로 실험의 시간 간격도 지수분포와 얼랭분포에 가까워지는 것을 확인할 수 있다.

4. Discussion

4.1 Poisson 분포의 시간 간격

위 실험에서는 시간 간격을 1000으로 잡았다. 이는 곧 λ값에 영향을 주는 것으로 실제 실험 결과에는 큰 영향을 주지 않으므로 어떤 값으로 설정해도 상관없다. 시간 간격이 달라지면 그에 따른 람다 값의 푸아송분포에 가깝게 나타날 것이다.

4.2 오차 보완 방법

먼저 실험을 많이 진행할수록 n값이 커져 오차가 작아지는 것을 알았다. 이 외에도 코드에서 몇 개의 값이 범위 내에서 발생했는지에 대한 변수를 구했으므로 신뢰도나 얼마나 구간 안에 나타나는지에 대한 지표도 포함하여 계산하면 더 정확한 결과 값이라 할 수 있다.

5. Reference

[1] https://ko.wikipedia.org/
[2] https://homepage.stat.uiowa.edu/~mbognar/applets/