고소실 5주차 보고서

20191571 김세영

1. 프로그램의 구동 방법 및 간략한 소개

본 실습에서 제공하는 GUI 소프트웨어를 사용하여 임의의 확률 밀도 함수 pX (x)를 설계하고, pX (x)를 따르는 확률 변수 X를 시뮬레이션 해주는 난수 수열 X0, X1, X2, X3,···를 확률적으로 올바른 방법으로 발생시킨후, 자신이 올바르게 난수를 생성하였는지 통계적으로 확인할 수 있는 프로그램이다.

구동방법: (실습 2-1 )sampling table.txt 를 읽어 확률밀도 함수 p(x)로 만든다. 이 샘플 데이터는 pdf table.txt에 저장된다. 또한 강의자료에 나와있는 p(x)의 구간의 적분값은 콘솔 윈도우에 출력된다.

(실습2-2) pdf table.txt을 읽어 난수수열을 만들고, 이를 random event table.txt에 저장한다. 이때 생성할 난수의 개수는 콘솔 윈도우에서 입력받는다. (input nr: )

(숙제 2-1)함수 program2\_1\_a()에서 생성할 난수 n을 콘솔 윈도우에서 입력받고, (input n: ) hw-2-1.txt에는

지수분포로 구한 평균과 분산, hw2-1-x.txt에서는 지수분포로 구한 난수들을 저장한다. 지수 분포의 인자 λ는 0.5 ,1.0 ,5.0이다.

(숙제 2-2) (i)함수 program2\_2\_a()에는 함수 program 2\_2()에서 불필요한 부분이 삭제되었다. 이 난수 생성 결과는 random\_event\_table\_bisection.txt 에 저장된다.

(ii) 함수 program2\_3() 은 random\_event\_table.txt와 pdf\_table.txt를 읽어 histogram.txt에 pdf의 x,y값 , 난수의 히스토그램에 해당 구간과 해당하는 개수를 저장한다.

(iii) 함수 program2\_2\_b()에서는 secant 방법으로 대치하여 난수를 생성한다. 생성된 난수는 ramdom\_event\_table\_secant.txt 에 저장된다. 함수 program2\_3\_test()에서는 2-3과 기능은 동일하지만 입력과 출력 파일명을 지정할 수 있는데, 이를 이용해서 (i)과 (iii)에서 구한 난수에 대한 정보(pdf,histogram) 가 histogram\_bisection.txt와 histogram\_secant.txt에 저장된다.

(iv) 프로그램 2-2(a), (b)의 수행결과는 콘솔 윈도우에 출력된다. 난수의 개수는 1000개이다.

1. 실습

1에 기재함

1. 과제

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 평균 | 분산 | 평균 | 분산 | 평균 | 분산 |
|  | λ= 0.500000000000000 | | λ= 1.000000000000000 | | λ= 5.000000000000000 | |
| 기댓값 | 2.0 | 4.0 | 1.0 | 1.0 | 0.2 | 0.04 |
| N=50 | 1.643967830489618 | 1.953112548372713 | 0.821983915244809 | 0.488278137093178 | 0.164396783048962 | 0.019531125483727 |
| N=100 | 2.003382743485420 | 3.041788097331587 | 1.001691371742710 | 0.760447024332897 | 0.200338274348542 | 0.030417880973316 |
| N=500 | 1.908506680049237 | 3.138555067488644 | 0.954253340024618 | 0.784638766872161 | 0.190850668004924 | 0.031385550674886 |
| N=1000 | 2.077852513416468 | 4.257767668311924 | 1.038926256708234 | 1.064441917077981 | 0.207785251341647 | 0.042577676683119 |

숙제 2-1

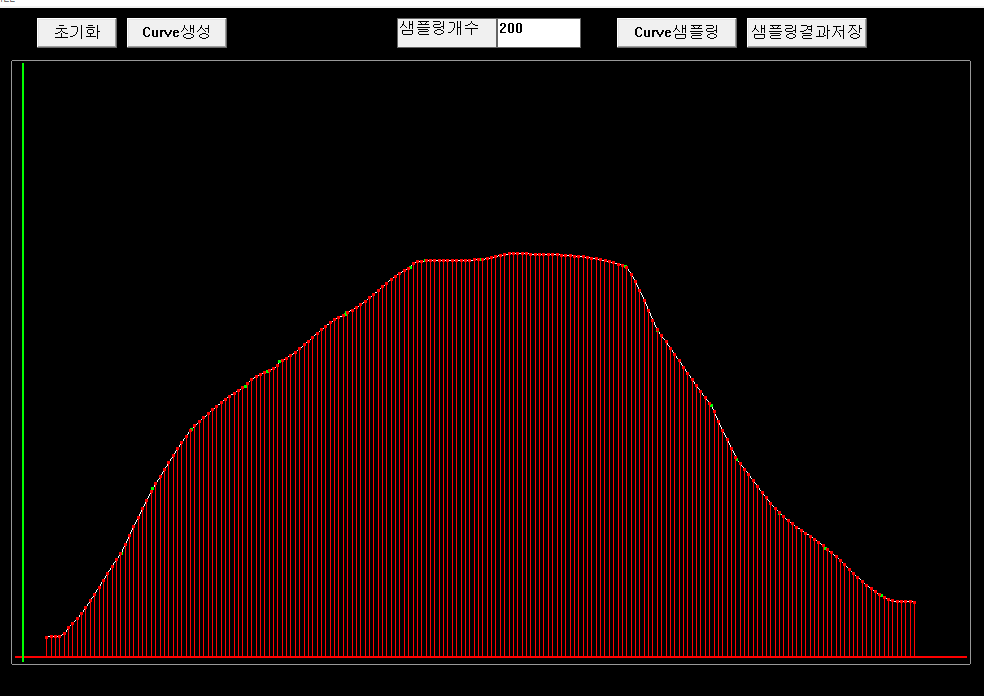
오차율의 절대값 (단위 : %)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 평균 | 분산 | 평균 | 분산 | 평균 | 분산 |
|  | λ= 0.500000000000000 | | λ= 1.000000000000000 | | λ= 5.000000000000000 | |
| 기댓값 | 2.0 | 4.0 | 1.0 | 1.0 | 0.2 | 0.04 |
| N=50 | 17.80161 | 51.17219 | 17.80161 | 51.17219 | 17.80161 | 51.17219 |
| N=100 | 0.169137 | 23.9553 | 0.169137 | 23.9553 | 0.169137 | 23.9553 |
| N=500 | 4.574666 | 21.53612 | 4.574666 | 21.53612 | 4.574666 | 21.53612 |
| N=1000 | 3.892626 | 6.444192 | 3.892626 | 6.444192 | 3.892626 | 6.444192 |

위의 4가지 경우 중에 N=1000인 경우가 오차율의 절대값이 제일 낮고, 10퍼센트 미만이기 때문에

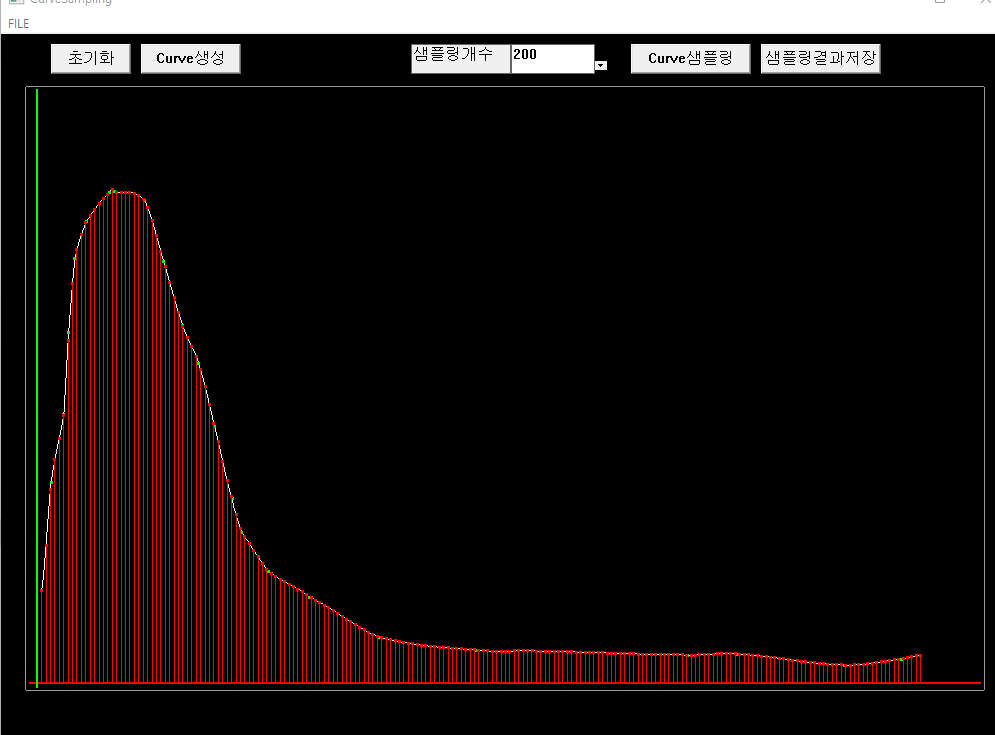
위의 실험 중에 이론적인 값과 가장 일치하고 통계적인 값이 이론값과 충분히 일치하게 된다고 할 수 있다.

숙제 2-2

1. 시간 측정을 위해 nr값을 scanf로 입력받는 것에서 nr=200으로 직접 코드 내에서 지정하는 것으로 바꾸었다. 또한 코드에서 선언한 후 사용하지 않는 변수도 삭제하였다.
2. 

다음과 같은 curve로 sampling\_table.txt를 생성하였다. 오른쪽 그래프는 pdf 생성 결과를 2차원 그래프로 나타낸 것이다.

난수 생성 결과를 히스토그램으로 나타낸 것이다. 대략적으로 위의 curve 모양을 난수 생성 결과가 따른다고 할 수 있다.



다음과 같이 앞쪽의 값이 큰 curve로 sampling했을 때는 아래와 같이 난수의 값이 작은 결과가 많이 나타났다. 따라서 curve의 경향성을 난수 생성에서도 따른다고 할 수 있다.

(두 가지 경우 모두 난수를 100개 생성한 것)

1. Secant method를 사용하였다. 처음에는 구간을 [pdf\_x[0],pdf[num-1]]으로 잡아 값이 될 수 있는 양 끝점을 사용하였다. (program2\_2\_b가 secant 수행 시간이다)

계산 시간을 줄이기 위해 bisection 방법을 몇 회 반복하여 구간의 폭을 줄이도록 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1회

텍스트이(가) 표시된 사진

자동 생성된 설명

2회

텍스트이(가) 표시된 사진

자동 생성된 설명

bisection 3회

텍스트이(가) 표시된 사진

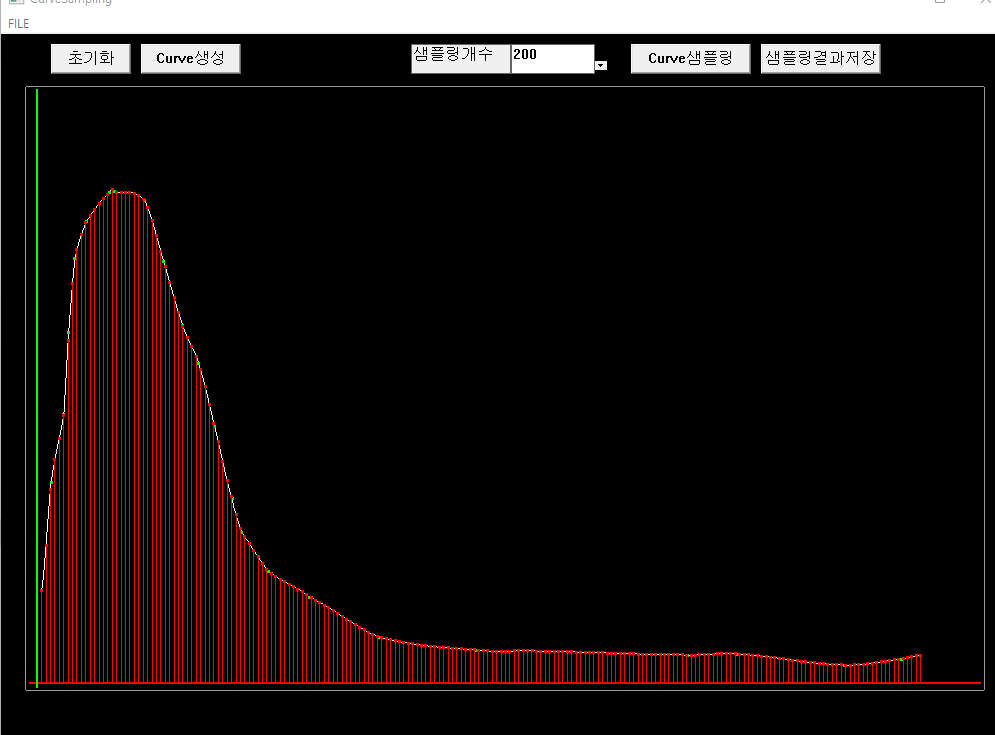
자동 생성된 설명

4회

bisection을 2회 실시한 후 나온 a0 b0를 x0, x1(구간의 양 끝점)로 하는 방법이 수행시간이 제일 적게 걸렸기 때문에 secant 방법에서는 bisection을 2회 실시하기로 하였다. (난수 생성 200개)

이 수행결과가 올바른지 판단하기 위해 숙제 2-2(ii)에서 작성한 프로그램 2-3을 이용해 secant 와 bisection결과 두개 다 테스트 한 후 그래프로 나타내 보도록 한다.

결과를 직관적으로 확인하기 위해 앞쪽의 확률이 높은 curve를 사용하도록 한다.



두 방법 모두 curve와 같은 경향을 띄고 있기 때문에 제대로 작동한다. ( 두 방법 다 난수를 200개 생성하였다)

1. 충분히 큰 난수의 개수 nr=1000으로 하고, 10번 수행하여 걸린 시간을 비교하도록 한다.

(단위: ms)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | 12.305500 | 12.621100 | 16.947700 | 17.115699 | 17.007601 | 12.236700 | 12.414300 | 12.291500 | 12.264100 | 12.434800 | 12.270300 |
| b | 3.071600 | 3.124200 | 4.206200 | 4.179000 | 3.700100 | 2.943000 | 3.039200 | 3.021300 | 3.067700 | 3.045500 | 3.022100 |

프로그램 2-2(a)의 수행시간 평균은 13.62811818 ms, (b)의 평균은 3.3109ms 이다. 따라서 (b)가 (a)보다 4.116137057배 더 빠르다.