Taller de Matemática Computacional 2017

Facultad de Ciencias Exactas, UNCPBA

Informe de entrega

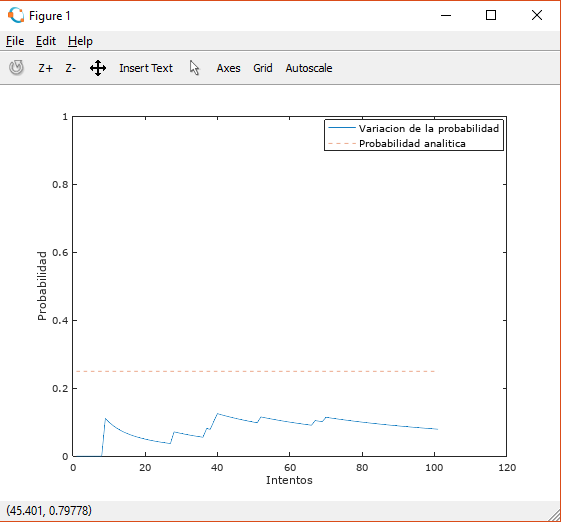
Bacigaluppi Nahuel DNI 41.675.923

**Introducción al tema:**   
Se planteó que la nave con la que veníamos trabajando en el laboratorio ahora tenía un sistema de autentificación que nos permite saber si podíamos disparar o no, para esto debíamos usar una función que como parámetro utiliza el número de documento, esta devuelve tanto un 0(en caso que no se autorice) como un 1 (en caso de que se autorice).   
Mediante lo aprendido había que crear las funciones necesarias para poder calcular la probabilidad de que el sistema de autentificación nos devuelva dos veces seguidas un 0 (no se autoriza a disparar dos veces seguidas). Una vez creado todo lo necesario se debía seguir una serie de ítems planteados en la consigna

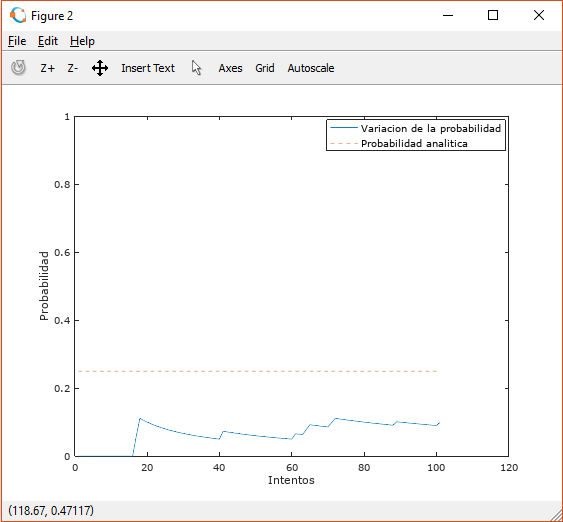
**Desarrollo:**  
Para poder resolver el problema partí desde la creación del script, donde se inicializaba un valor de épsilon, se llamaría a una función que haría la probabilidad y después se imprimiría por pantalla los resultados y además se crearía la gráfica que demuestra la variación de la probabilidad mediante sucesivas interacciones.   
Una vez ya diseñado el script pase a crear una función que calcula la probabilidad de que salgan dos ceros seguidos mediante el método de un motor de Montecarlo, la cual tiene como parámetro el valor de épsilon y nos devuelve datos como la probabilidad y la cantidad de probabilidades.  
La función dentro es controlada mediante un while, el cual toma la condición de que como mínimo debe haber 100 tiradas/interacciones o que la diferencia entre las condiciones que van saliendo tiene que estar lo más cercano a épsilon posible. Mientras que cumpla con las condiciones la función hace una interacción en la cual se llama al sistema de autentificación dos veces con parámetro DNI y los dos resultados se guardas en dos variables distintas, una vez ya obtenidos los dos resultados se ejecuta una condición donde se evalúa si ambos números son ceros, si lo son, un contador de casos favorables se va sumando y a su vez la cantidad de tiradas, y sino solo se suma la cantidad de tiradas, además también se van guardando las probabilidades parciales en cada interacción en un arreglo.  
Una vez hecho esto, la función divide la cantidad de casos favorables por la cantidad de tiradas totales y esto devuelve el número de probabilidad hacia el script, en el script se toman estos resultados y con ellos se hace el grafico que demuestra la variación. Una vez que el programa ya funcionaba con lo básico, en el script le agregue un while para que con una sola llamada al script se pueda hacer los tres ítems de la consigna número 3, para este el while adentro tiene una condición que contrala que se haga mientras que épsilon sea mayor o igual a 0.001. Una vez que el programa entraba en el while, cuando está por llegar al final se divide a épsilon que en este caso esta iniciado en 0.1 por diez, esta división lo que hace es agregar un cero después de la coma y antes del 1, esto se hace la cantidad de veces hasta que épsilon ya no cumpla con la condición.  
Luego de que se crea el while dentro del script, se pasa a hacer la parte de probabilidad, donde en el arreglo que obtuvimos para cada valor distinto de épsilon, se toman los primero 20 números de este arreglo y se saca el desvió estándar de estos, y lo mismo se hace con los últimos 20 números del mismo arreglo, este proceso se repetirá tantas veces como se ejecute la función de épsilon (para cada valor de épsilon se hace el proceso). Y también se le agrega la función Tic-Toc para tomar el tiempo que tarda en ejecutarse la función para cada valor de épsilon.

**Resultado:**

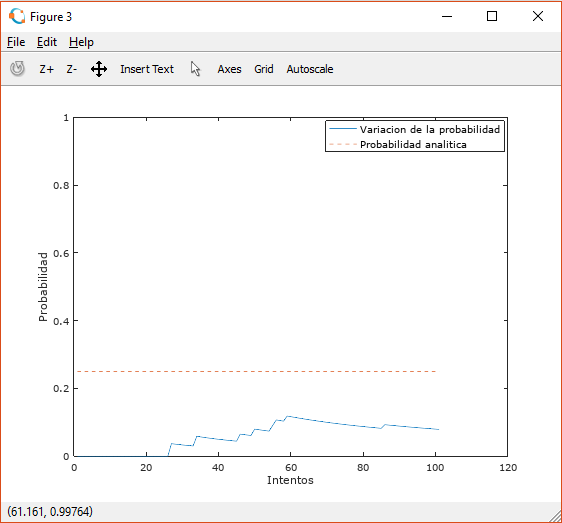
1. Épsilon= 0,1  
   Resultado de probabilidad= 0.079208; Tiempo que tomo el cálculo= 0.74608



1. Épsilon= 0.01  
   Resultado de probabilidad= 0.099010; Tiempo que tomo el cálculo= 0.78326



1. Épsilon= 0.001  
   Resultado de probabilidad= 0.079208; Tiempo que tomo el cálculo= 0.82985



**Conclusiones:**  
Depende el valor de épsilon los cálculos de la probabilidad y el tiempo que toma van cambiando, mientras más grande es épsilon, más certero va a ser el resultado y menor el margen de diferencia entre las variaciones, además que mientras más grande es épsilon más va a tardar el programa en hacer la función, ya que tiene que tener en cuenta una mayor cantidad de interacciones y resultados.  
El desvió estándar en las primeras 20 interacciones normalmente es mayor que en los últimos 20 ya que en las primeras 20 interacciones los resultados están más disipados, por lo tanto hay una mayor distancia de dispersión respecto al valor promedio, en cambio, en las ultimas 20 interacciones el desvió estándar es menor ya que en el final las probabilidades de las interacciones están mucho más cercas por lo tanto hay una menor medida de dispersión respecto al valor promedio