Optimización del cálculo de la correlación de la distancia con estructuras de datos

Andrés Rubiano, Jorge Camacho, Santiago Mariño.

No. de Equipo Trabajo: 8

INTRODUCCIÓN

En este proyecto se busca optimizar el cálculo de la *correlación de la distancia* a través del lenguaje de programación *Python* usando como herramienta las diferentes estructuras de datos con las que contamos hoy en día. En este programa dichas herramientas serán utilizadas para crear una interfaz gráfica con la cual el usuario tendrá una mayor facilidad al representar las muestras que el usuario necesita ingresar para calcular el coeficiente *de correlación*.

Calcular este valor es algo sencillo temiendo en cuenta las herramientas computacionales con las que contamos hoy en día, pero debido a la gran cantidad de datos que se pueden llegar a tener, se presentaran complicaciones a la hora de ejecutar el programa, más específicamente, en el tiempo de ejecución de este, de ahí ña importancia de optimizar este proceso.

PLANTEAMIENTO DEL PROBLEMA

El cálculo de la *correlación de la distancia* se puede realizar a través del paquete *energy* escrito en el lenguaje R. Lo que se busca en este proyecto es implementar el cálculo de este en el lenguaje de programación *Python*, cambiando la forma en las cuales se almacenan los datos experimentales que se tienen, esto se realizara con la ayuda de las estructuras de datos que conocemos(*Listas, Colas.*) y así, utilizando distintos datos experimentales de problema reales, se obtendrá el tiempo que se tarda en calcular la *correlación de la distancia* y se podrá escoger la forma más óptima de guardar los datos para realizar el cálculo.

USUARIOS DEL PRODUCTO DE SOFTWARE

El público al que está dirigido el software es en su mayoría estadísticos, matemáticos, ingenieros y economistas, entre otros, puesto que estos en el desarrollo de sus carreras o en el campo laboral utilizan aplicaciones y software de análisis, bases y estructuras de datos.

REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- Los datos que se ingresan al software, son tomados de bases de datos, los cuales se encuentran en dependencia lineal, por ejemplo, datos como años de estudio vs sueldo o peso vs altura.
- El sistema realiza la asignación de datos en arreglos de listas enlazadas, donde estas conforman matrices, las cuales se les calcula la matriz de distancias.
- Los reportes que arroja el sistema son el coeficiente correlación de distancia y la gráfica interactiva de datos que se analizó para calcular dicho coeficiente de correlación.
- En este prototipo el usuario podrá ingresar datos aleatorios a través de un archivo CSV en donde el nombre de las columnas son las referencias X y Y.

INTERACCIONES CON EL USUARIO:

- Ingreso de base de datos por el usuario: por medio de un algoritmo y la interfaz gráfica con el botón (Figura 1), el usuario puede ingresar su base de datos desde el directorio en donde esté guardado.

Figura 1

- Calcular el coeficiente de correlación: después de haber ingresado la base de datos el usuario puede calcular el coeficiente con el botón (Figura 2) que está en la interfaz donde este resultado se refleja en un cuadro de texto.

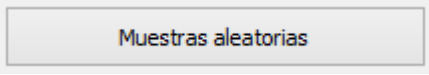


Figura 2

- Para calcular el coeficiente de correlación con una base de datos en particular, con el botón (Figura 3), el usuario tiene la opción de ingresar su base de datos personal.

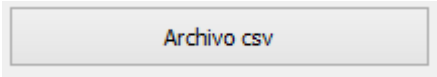


Figura 3

- También se podrá visualizar el comportamiento de los datos. Con en el boton (Figura 4), el usuario podrá graficar su conjunto de puntos para el respectivo análisis.

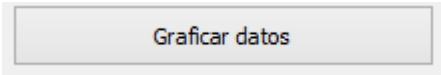


Figura 4

- Con los modos de ingreso, el usuario podrá seleccionar si ingresa los datos por listas o por arreglos.

Figura 5

- El usuario debe ingresar los datos adecuadamente, por ejemplo, los datos deben estar completos, no deben tener fallas de escritura, el nombre del archivo debe ser acorde al nombre que se ingresa en el sistema.

REQUISITOS FUNCIONALES MINIMOS DEL SISTEMA:

- Mostrar la ubicación del archivo donde se encuentra la base de datos.
- Agregar nuevas bases de datos para el cálculo.
- Graficar los datos en diagramas de dispersión.
- Mostrar el cálculo de la correlación de la distancia
- Permitir almacenar los datos en listas o en arreglos.
- Permitir el ingreso de datos en formato CSV a través de la librería pandas.

INTERFAZ GRAFICA DEL USUARIO

La interfaz gráfica está diseñada y realizada en Python con la librería Pyqt5, cuenta con tres botones que cumplen la función de calcular el coeficiente de correlación de distancia, importar el archivo CSV de base de datos a analizar y el botón graficar los datos, por su parte, cuenta con dos cuadros de texto que se usan para ingresar el tamaño de la muestra que se va a usar y otro cuadro que imprime el coeficiente de correlación calculado. Además, la gráfica de pyqt5 ofrece otro tipo de funcionalidades como zoom y configuración de los ejes de coordenadas, entre otras funcionalidades (Figura 6).

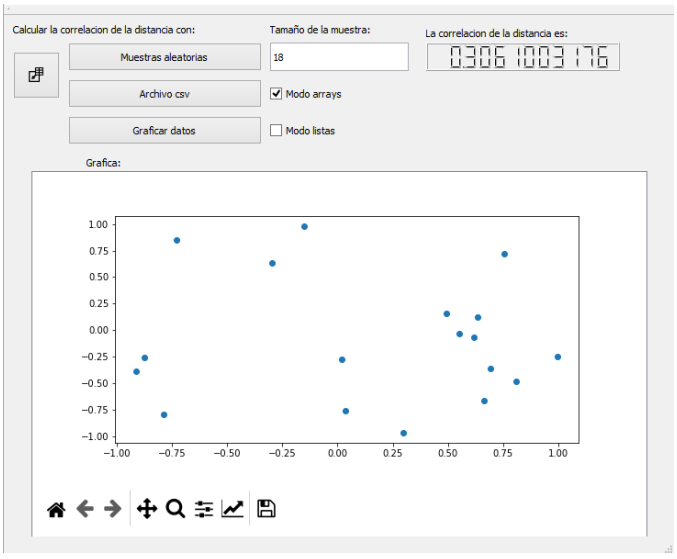


Figura 6

Por otra parte, la estructura del proyecto en GitHub (Figura 7) se presenta a continuación, donde se puede ver como se organizó en carpetas las respectivas clases, librerías y datos de ingreso CSV.

Figura 7

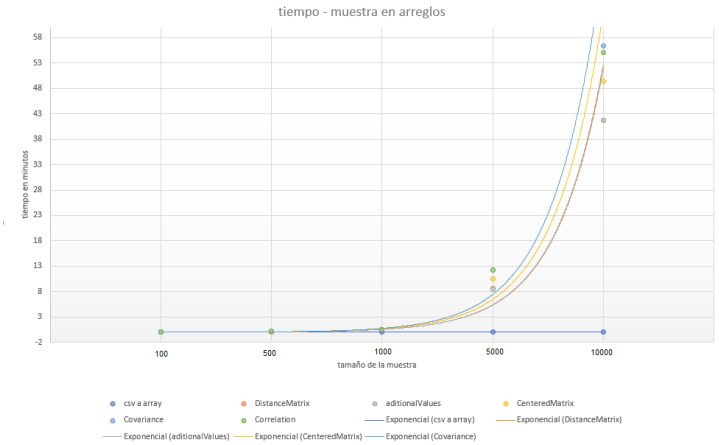
ENTORNO DE DESARROLLO DEL SISTEMA

El sistema está soportado en computador con sistema operativo Windows 8.1 pro, desarrollado en Python con el entorno de desarrollo de Anaconda – Spyder - Jupyter, con librerías como Pandas, Numpy, Pyqt5 y Timeit. Por su parte los requisitos del computador en donde se ejecuta el programa son un procesador AMD FX(tm) – 6300 Six –Core 3.5GHz, memoria RAM 8GB ddr3, disco duro SanDisk SDSSDA120G ATA Device y GPU AMD Radeon R9 200 series.

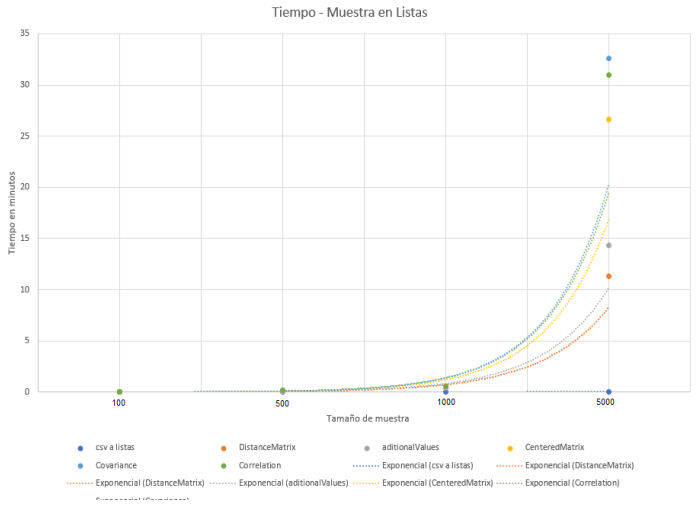
PRUEBAS DEL PROTOTIPO

Las pruebas realizadas al sistema se hicieron con las funcionalidades ingresar datos por medio de la librería Pandas, llenar matrices distantes, generar valores adicionales, llenar matrices centrales, calcular la distancia de covarianza, calcular la distancia de covariancia, con muestras de 100, 500, 1000, 5000 y 10000 datos arrojando sus respectivos tiempos.

En los siguientes gráficos se puede apreciar el comportamiento que tiene el tiempo respecto al tamaño de muestra con las diferentes estructuras de datos (Grafica 1, Grafica 2).



Grafica 1



Grafica 2

ANÁLISIS COMPARATIVO

Por su parte en las siguientes tablas (Tabla 1, Tabla 2) y gráficos comparativos (Grafica 3, Grafica 4) se puede ver, cuáles fueron los datos obtenidos para las diferentes funcionalidades en ambos métodos de ingreso (listas y arrays).
Para los gráficos se tomó cada una de las barras de frecuencia como una funcionalidad distinta, en el eje x al tamaño de muestra y en el eje y el tiempo de ejecución.

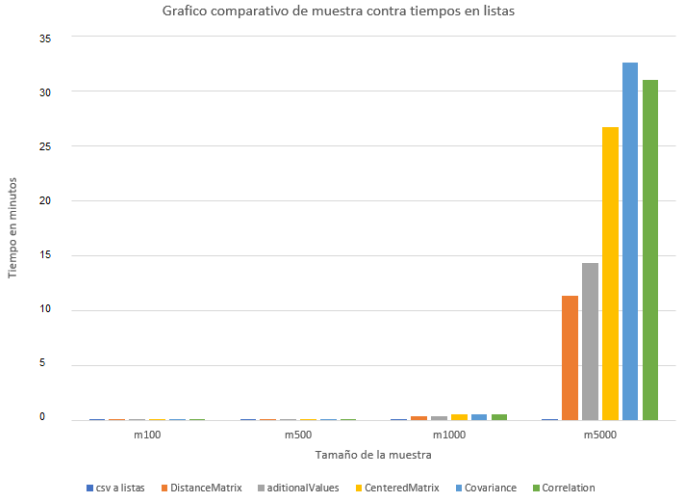
Tiempos en minutos para las funcionalidades con arreglos					
Tamaño muestra	100	500	1000	5000	10000
csv a array	0,0019	0,0048	0,0093	0,0461	0,0923
DistanceMatrix	0,0046	0,0874	0,3456	8,5594	41,7352
aditionalValues	0,0047	0,0886	0,3421	8,4788	41,6590
CenteredMatrix	0,0056	0,1060	0,4119	10,5402	49,4396
Covariance	0,0061	0,1218	0,4782	12,1982	56,3443
Correlation	0,0063	0,1219	0,4768	12,3111	55,1157

Tabla 1

Grafica 3

Tiempo en minutos para las funcionalidades				
Tamaño muestra	100	500	1000	5000
csv a listas	0,0016	0,0052	0,0098	0,0467
DistanceMatrix	0,0056	0,0980	0,3783	11,3474
aditionalValues	0,0054	0,1045	0,4084	14,3383
CenteredMatrix	0,0065	0,1303	0,5091	26,6442
Covariance	0,0070	0,1438	0,5721	32,6033
Correlation	0,0074	0,1414	0,5811	30,9743

Tabla 2



Grafica 4

Como se puede notar en las tablas y gráficos anteriores el crecimiento de los tiempos tiene un comportamiento O(n^2) puesto que al aumentar el tamaño de la muestra su tiempo respectivo se aproxima a una función cuadrática exceptuando la funcionalidad de ingreso de datos y esta por su parte tiene un comportamiento O(n).

ROLES Y ACTIVIDADES

Para la realización de este proyecto, Andrés Rubiano tomó los roles de líder y coordinador con su compromiso de programar y agendar reuniones por GoogleMeet y la mayoría del aporte teórico-práctico al proyecto. Por otra parte, Santiago Mariño tomo los roles de investigador y coordinador por su aporte a la consulta e investigación de las funciones y algoritmos para tomar el tiempo de las pruebas y Jorge Camacho tomo los roles de secretario y técnico al crear los informes y documentos del proyecto, además hacer la ejecución de los tiempos y toma de datos de las pruebas.

DIFICULTADES Y LECCIONES APRENDIDAS

- Implementación del cálculo del coeficiente de correlación con listas y arreglos.

- Creación de la interfaz gráfica.
- Creación del GitHub.
- No poder reunirse con los demás integrantes.
- Implementación de las técnicas de medición del tiempo para los distintos fragmentos del programa.
- Realizar las pruebas de los tiempos por falta de capacidad de computo.
- Adaptación al lenguaje de programación.

Las principales lecciones aprendidas son la capacidad de trabajo autónomo que se puede adquirir con herramientas como Google Meet, la diferencia notoria que se puede obtener al implementar estructuras de datos distintas, ver la utilidad que tiene un código organizado, la facilidad de utilizar GitHub para compartir código y archivos, el aprendizaje de nuevos algoritmos de búsqueda y ordenamiento que se deben tener para un código.