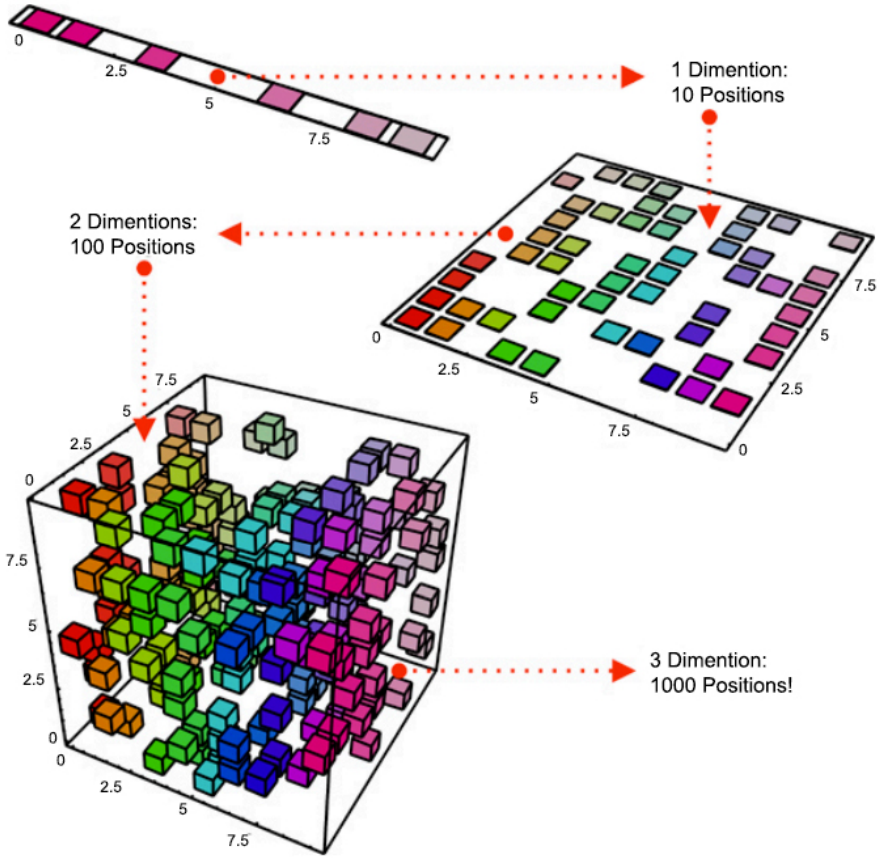
Informe de laboratorio 1: La maldición de la dimensionalidad

Laboratorio de introducción al curso de Estructuras de Datos Avanzadas



Fernando Jair Peralta Bustamante

31/08/2023

INTRODUCCIÓN

La maldición de la dimensionalidad es un término que hace referencia a las diferentes consecuencias que trae el aumentar las dimensiones de un objeto lo cual se ve reflejado directamente al momento de nosotros hacer uso de datos multidimensionales (cientos o miles de dimensiones) en los diferentes procesos computacionales que desarrollamos al momento de implementar algoritmos en los que necesitamos trabajar con datos multidimensionales ya sea una base de datos, características de un objeto, entre otros.

HIPÓTESIS

La maldición de la dimensionalidad o efecto Hughes señala que ha medida que aumentamos la dimensionalidad, el volumen del espacio aumenta exponencialmente haciendo que los datos disponibles se vuelvan dispersos.

Para entender mejor nuestra hipótesis imaginemos que tenemos 20 puntos para un espacio de una dimensión, supongamos que esos 20 puntos se pueden almacenar en un espacio de 10 casillas, debido a que tenemos 20 puntos, la tendencia sería a que cada una de nuestras casillas almacena desde 1 o 2 puntos haciendo además que la distancia entre ellas no sea lejana debido a la condición de su unidimensionalidad. Ahora imagina que esos mismos 20 puntos tienen 2 dimensiones, esto hace necesariamente que ahora su almacenamiento se de en una matriz de 10x10 es decir 100 posibles casillas para almacenar los mismos 20 puntos de antes, por consiguiente, tenemos que la distancia entre estos 20 puntos es mayor y además la cantidad de casillas disponibles a las que no se invoca en ningún momento pero son necesarias para el almacenamiento pasó de ser 0 o casi 0 a 80 o casi 80. Si los puntos ahora tienen 3 dimensiones sería un espacio de 1000 casillas, un cubo y si fuera de 4 dimensiones 10000 y así sucesivamente haciendo que cada vez la distancia entre puntos sea mayor.

MATERIAL

1. Código en C++.
2. Código en Python.

PROCEDIMIENTO

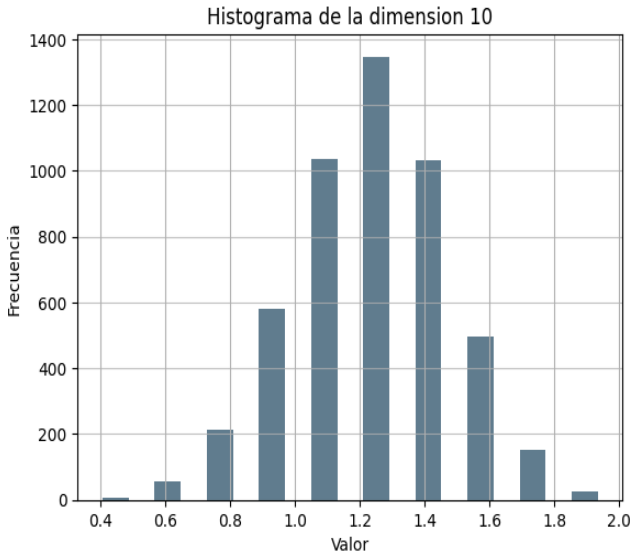
1. Diseñaremos un código que permita almacenar una cantidad de 100 puntos en c++ con una estructura que permita a esa cantidad de puntos indicar como parámetro la cantidad de dimensiones
2. Llenaremos cada una de las dimensiones de los puntos con valores aleatorios con un margen entre 0 y 1 con ayuda del código uniform_real_distribution
3. Con los valores obtenidos hallaremos la distancia entre todos los puntos haciendo uso de la fórmula de distancia euclidiana para n dimensiones
4. Exportamos los resultados obtenidos y realizaremos un histograma donde se debería denotar la diferencia entre un punto de dimensión 10 a un punto de dimensión 5000

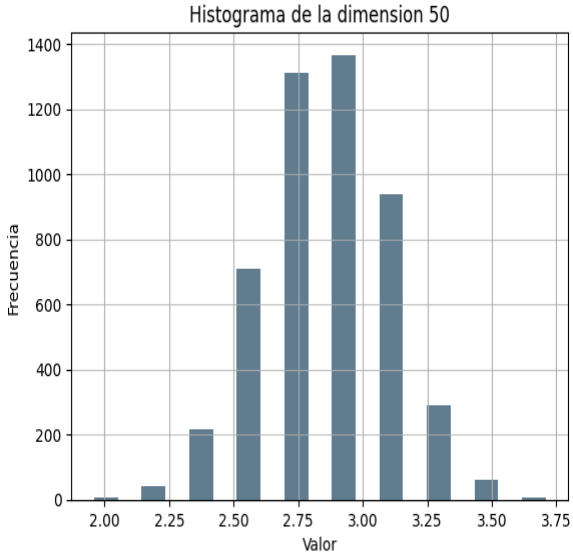
DATOS

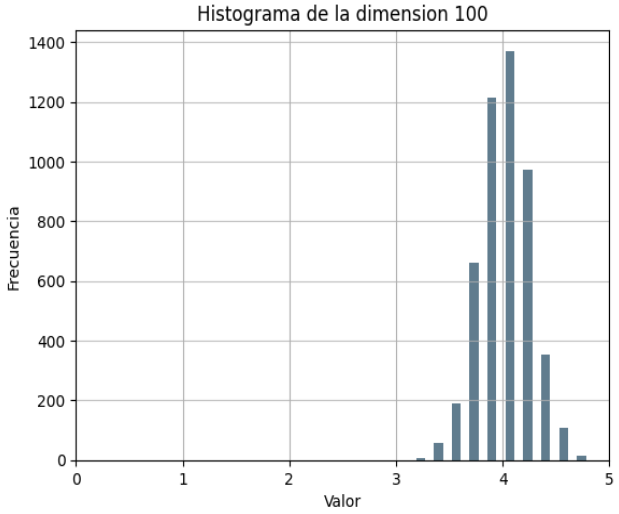
Luego de la implementación del código antes mencionado tenemos estos datos como promedio de dividir la sumatoria de todas las distancias entre la cantidad de distancias, para el caso en especifico de 100 puntos son 4950.

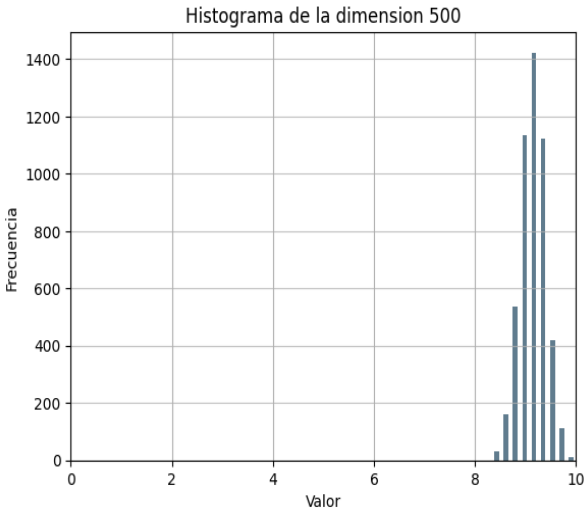
Dimensiones	Promedio de distancia entre puntos
10	1.23525747
50	2.85757522
100	4.02359723
500	9.14758377
1000	12.9091161
2000	18.2573258
5000	28.8658307

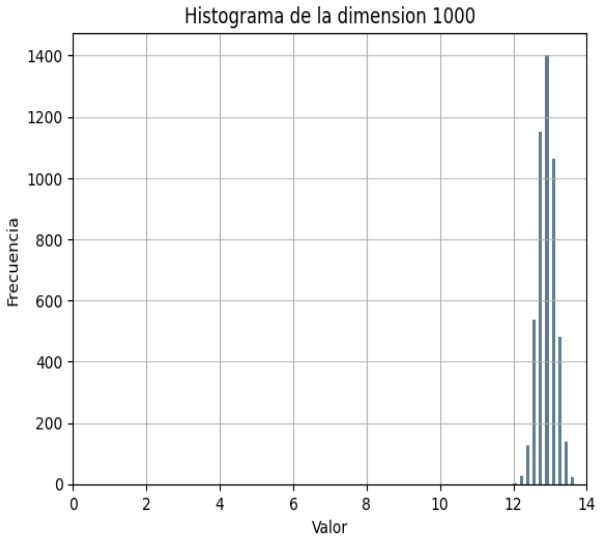
Además de la tabla antes citada tenemos los siguientes gráficos que nos ayudarán a entender mejor la distribución de los puntos y sus distancias a medida que aumentamos las dimensiones

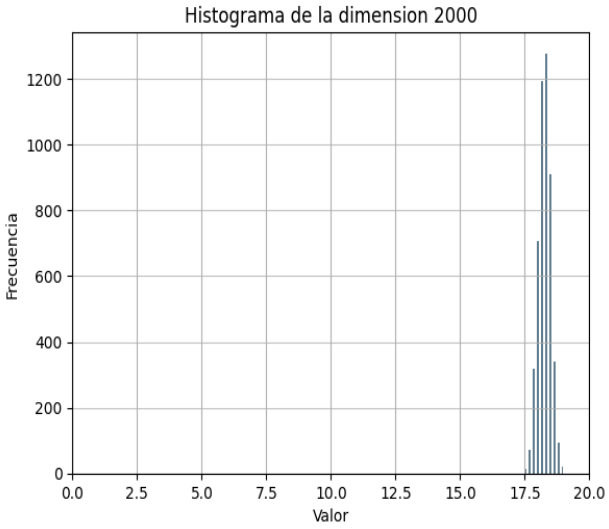


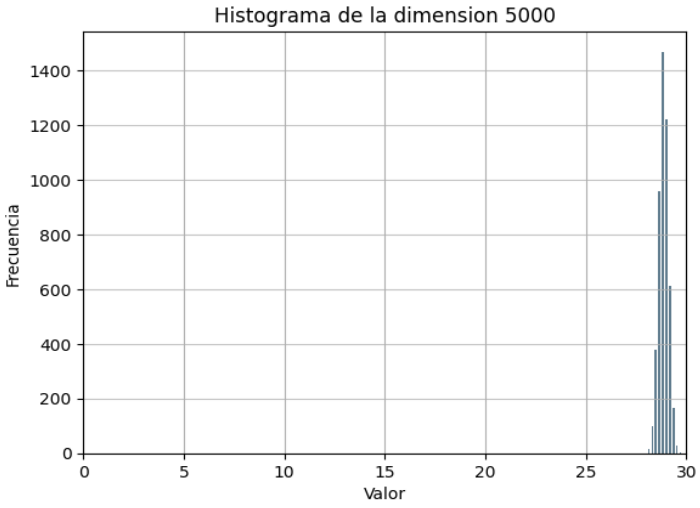












El eje x demarca el valor que toma las distancias entre los puntos mientras que el eje y es la frecuencia, es decir, la cantidad de veces que aparece uno de estos valores entre la distancia obtenida de analizar todos los puntos. Además el tiempo de obtención de las diferentes gráficas varía también a medida que aumenta la cantidad de dimensiones debido a la fórmula de distancia euclidiana que requiere si o si aumentar un total de 100 operaciones por cada dimensión que aumentemos.

RESULTADOS

Como se puede observar en las diferentes gráficas y en la tabla, la distancia tiende a aumentar de manera exponencial a medida que aumentamos la cantidad de dimensiones para cada uno de los puntos, además, esto va relacionado con la frecuencia de estas distancias entre los puntos, si bien es cierto que se sigue manteniendo una forma de campana en cada una de las gráficas, la tendencia hacia el valor promedio de las dimensiones es alta.

CONCLUSIÓN

Dado los resultados obtenidos y el análisis de toda la información recopilada se puede concluir que el aumento de las dimensiones si tiende a tener una mayor dispersión entre los puntos que se deseen evaluar siendo un problema siempre y cuando la cantidad de puntos se mantenga igual para una cantidad n de dimensiones, junto a ello tanto el costo de preprocesamiento como el costo de almacenamiento aumentan de igual manera debido a la cantidad de espacio necesario para almacenar cada una de las posibles coordenadas de los puntos.

Link del github:

https://github.com/fernando-peralta/EDA/tree/main/Laboratorios/La_maldicion_de_las_dimensiones

REFERENCIAS

Magdy, S. (n.d.). *IME Curse of dimensionality ML big data ML optimization PCA*. IME. Retrieved August 31, 2023, from http://www.infme.com/curse-of-dimensionality-ml-big-data-ml-optimization-pca/

Subramanian, N. B. (n.d.). *Curse of Dimensionality | Python | PCA - AI ASPIRANT*. ai aspirant. Retrieved August 31, 2023, from https://aiaspirant.com/curse-of-dimensionality/