Universitat de les Illes Balears

GRADO EN INGENIERÍA INFORMÁTICA

Aplicación práctica ACSI curso 2022-2023

Carlos Lozano Alemañy

PRÁCTICA 6

El objetivo de esta parte es la comprensión del concepto de caracterización de la carga. Para ello, se hará uso de la herramienta Weka.

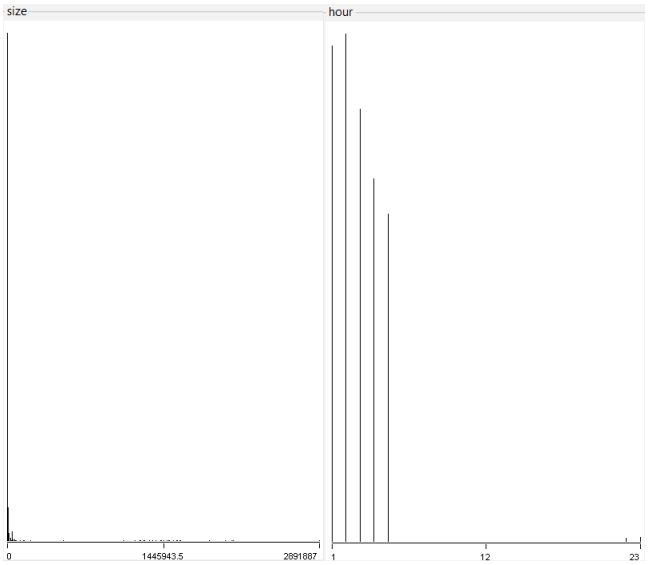
De la monitorización de un sistema de almacenamiento, se ha obtenido se proporciona un fichero de datos llamado "data.txt". En el fichero se almacenan tres columnas con la siguiente información:

- **El tamaño del fichero accedido (en MB). Los valores que correspondan con "-1" quieren decir que el acceso al fichero ha fallado.**
- **La hora a la que se hizo el acceso. El valor 22 representan las 22h, el valor 01 representan las 1h (a.m.), etc.**
- **El ancho de banda consumido (en MS/s). Los valores de esta columna están entre 453 y 1355, por lo tanto, los valores de esta columna deberán ser tratados. Es decir, el valor crudo de "1258.84,", corresponde con "1258,84".**

Con los datos proporcionados se pide caracterizar la carga haciendo uso del algoritmo de Kmeans y responder a las siguientes preguntas:

Antes de arrancar con las preguntas cabe aclarar que el fichero ha sido filtrado mediante un script de tal manera que los datos no den problemas a la hora de seleccionarlos con 'weka', además estos datos filtrados ahora serán con formato ' .csv'.

Tras la limpieza del fichero se ha decidido representar los datos en el 'preprocess' de Weka y se ve un comportamiento muy significativo en el 'Size' y en 'Hour':

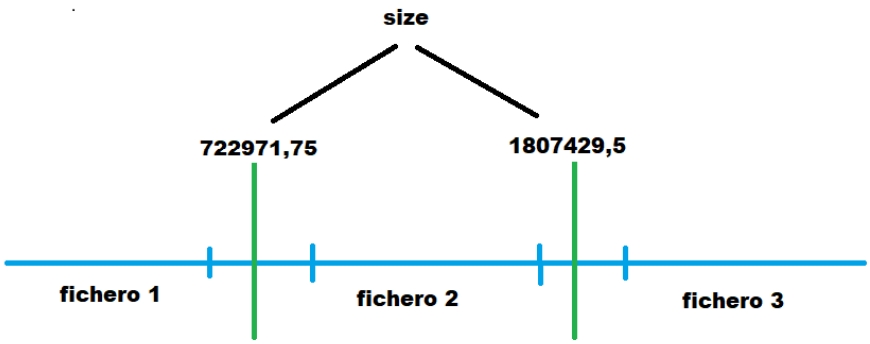


Por una parte, las horas no nos aportan nada de información útil respecto a un sistema de almacenamiento, pero además resulta que la variable 'hora' no aporta datos desde las 4am hasta las 22 pm, por estos motivos prescindiremos de este atributo durante el clustering.

Por otra parte, en el 'size' se diferencian claramente 3 grupos, es por eso que a través de un script se clasificarán previamente los datos en 3 grupos en función del tamaño, y después se usará los clusters para cada fichero.

Además, el conjunto de estos 3 ficheros será comparado con el resultado que se habría obtenido si no se hubiera separado en 3 el fichero origninal.

Para saber los puntos en los cuales el fichero será dividido se han ordenado los tamaños de menor a mayor para ver la serparación entre los diferentes grupos, luego se han codigo dos puntos arbitrarios dentro de estas separaciones,(las lineas verdes dentro de el espacio entre las franjas azules que corresponden a las separaciones), tal que:



Entonces el fichero 1 contiene todos los datos de menor tamaño que 722971 MB, el fichero 2 contiene aquellos datos mayores o igual que 722971 MB y menores que 1807429,5 MB y el fichero 3 contiene aquellos datos con tamaño superior a 1807429,5 MB.

Representantes correspondientes a los 3 ficheros resultantes de ejecutar el script:

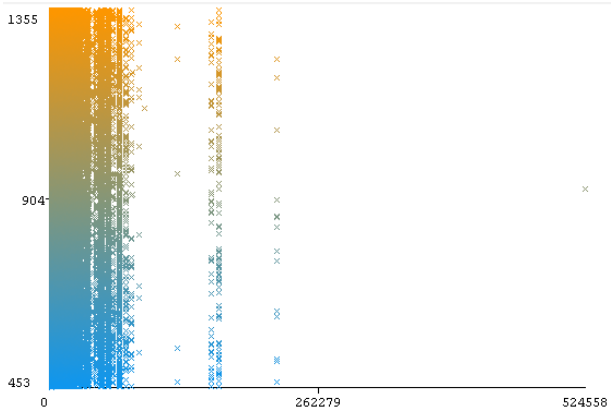
		FICHERO ORIGINAL	
	FICHERO 1	FICHERO 2	FICHERO 3
	Tamaño pequeño (MB)	Tamaño intermedio(MB)	Tamaño grande(MB)
REPRESENTANTE	3610,13	1504514,87	2077104,73

El valor de cada representante se corresponde con la media de los valores de cada grupo.

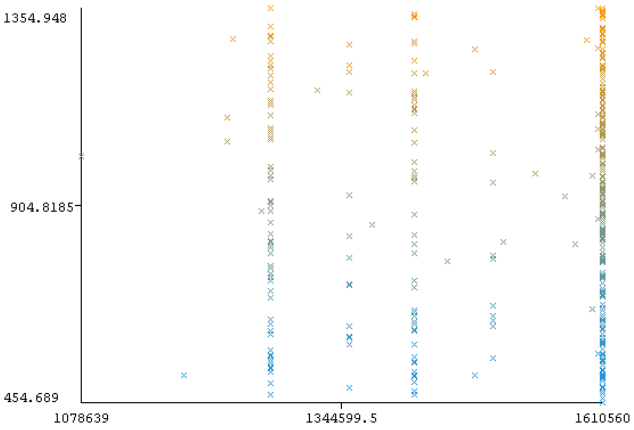
El script usado se mostrará al final del documento.

1. Aplicando el algoritmo con 100 iteraciones y agrupando los datos en 3 clases, ¿qué resultados se obtienen? Muéstralo gráficamente.

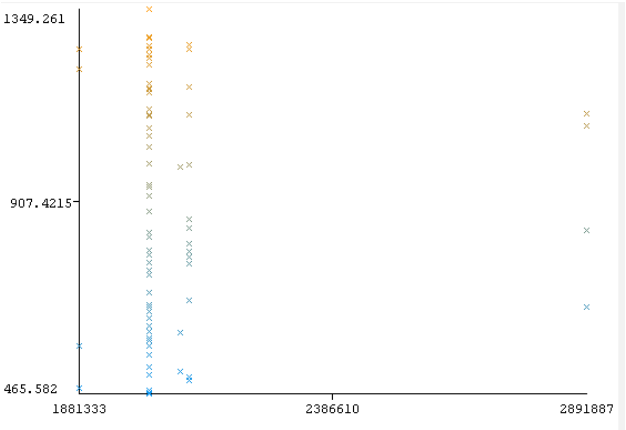
Fichero 1



Fichero 2



Fichero 3



El eje Y represente la velocidad(MB/s) y el eje X representa el tamaño(size)

Como muestran las gráficas, efectivamente los datos se han clasificado en 3 grupos que corresponden a los siguientes:

Velocidad Alta - Velocidad Media - Velocidad Baja

Si nos fijamos bien, en función de que fichero estemos tratando hay más o menos datos, siguiendo el siguiente orden de mayor o menor cantidad (Fichero1-Fichero2-Fichero3), lo que en definitiva viene a representar que los sistemas de almacenamiento tienen muchos más ficheros pequeños que grandes, esto es así porque se quiere obtener un sistema de almacenamiento más eficiente.

Los gráficos anteriores también muestran la relevancia del uso de la fragmentación de archivos dentro de un sistema de almacenamiento y la habitualidad de creación de ficheros pequeños por los usuarios, entre otros aspectos.

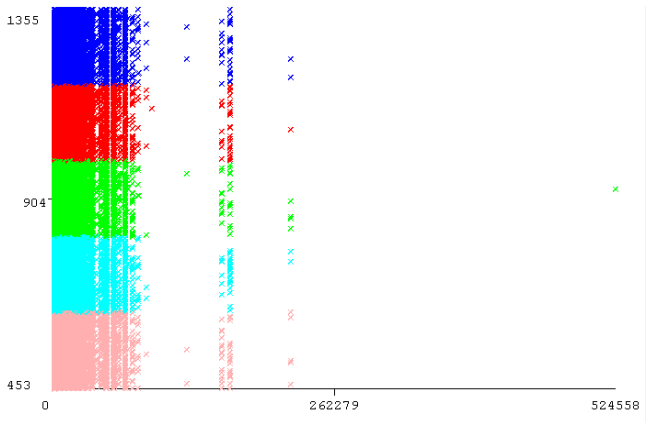
Los representantes de los 3 respectivos ficheros para el cluester de 3 son los siguientes:

		FICHERO 1	
	Velocidad Lenta (MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)
REPRESENTANTE	602,84	902,95	1204,2
		FICHERO 2	
	Velocidad Lenta (MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)
REPRESENTANTE	863,39	901,55	963,73
		FICHERO 3	
	Velocidad Lenta (MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)
REPRESENTANTE	628,88	872,68	1066,67

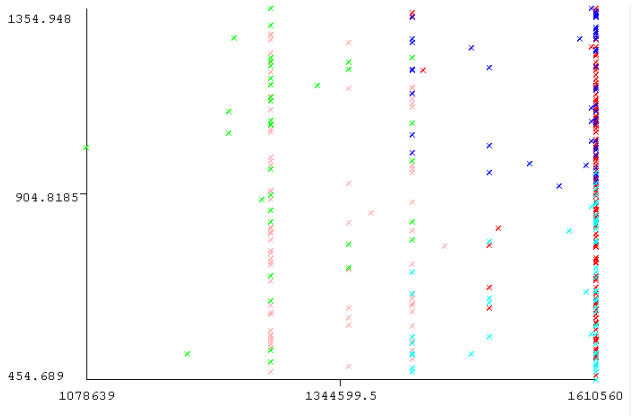
El valor de cada representante se corresponde con la media de los valores de cada grupo.

2. Con el mismo número de iteraciones y agrupando los datos en 5 clases, ¿qué resultados se obtienen? ¿Cómo difieren de los anteriormente obtenidos?

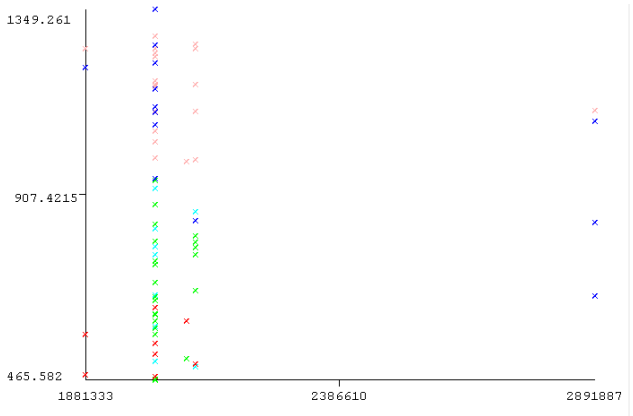
Fichero 1



Fichero 2



Fichero 3



El eje Y represente la velocidad(MB/s) y el eje X representa el tamaño(size)

Como muestra el gráfico, vemos que se obtienen los mismos resultados de antes pero en lugar de estar clasificados por 3 velocidades ahora lo estarán por 5, que serán las siguientes:

Velocidad muy baja- Velocidad baja- Velocidad media - Velocidad alta - Velocidad muy alta

Por lo demás el comportamiento es idéntico, aunque ahora se ve más fácilmente que ciertos datos de un grupo específico se encuentran en otros grupos, esto se debe a que hemos dividido los datos en más grupos que antes y es más fácil que se mezclen, puesto que hay más "fronteras", aunque realmente los datos estan distribuidos idénticamente a los gráficos anteriores.

Los representantes de los 3 respectivos ficheros para el cluester de 5 son los siguientes:

			FICHERO 1		
	Velocidad Muy Lenta (MB/S)	Velocidad Lenta(MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)	Velocidad muy Rápida(MB/S)
REPRESENTANTE	543,24	724,17	903,97	1084,89	1265,01

			FICHERO 2		
	Velocidad Muy Lenta (MB/S)	Velocidad Lenta(MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)	Velocidad muy Rápida(MB/S)
REPRESENTANTE	676,5	827	899,22	999,8	1158,1

			FICHERO 3		
	Velocidad Muy Lenta (MB/S)	Velocidad Lenta(MB/S)	Velocidad Intermedia(MB/S)	Velocidad Rápida(MB/S)	Velocidad muy Rápida(MB/S)
REPRESENTANTE	543,66	692,89	713,64	1066,67	1151,58

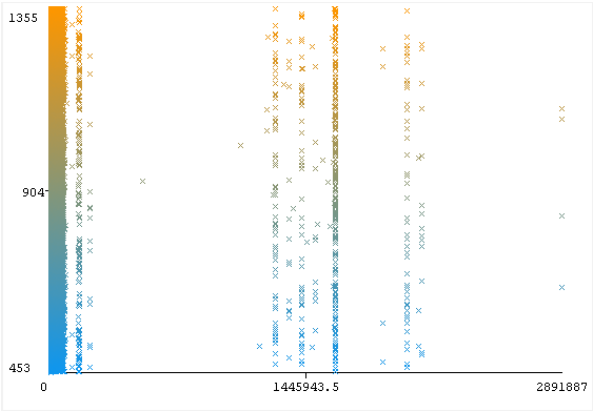
El valor de cada representante se corresponde con la media de los valores de cada grupo.

3. ¿Hay alguna característica especial en la carga proporcionada? Explícala con detalle.

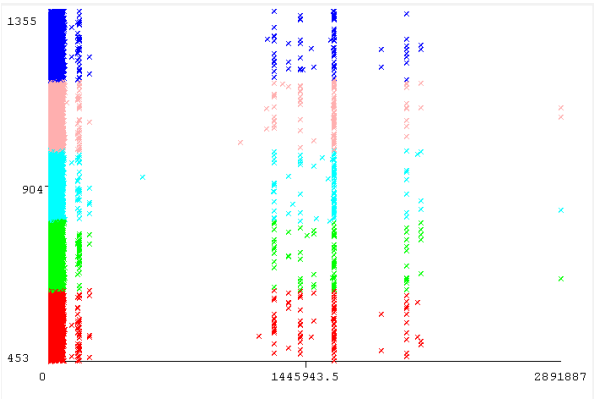
Hay varias características especiales, para empezar, los diferentes clústeres no han sido agrupados en grupos de nube de puntos si no que todos han tenido una agrupación muy similar, luego, como se

ha comentado anteriormente, es destacable que se concentran la mayoría de puntos al principio y como más lejos del eje central estés menos puntos hay, es decir, el gráfico sigue una distribución de cola pesada, otra característica interesante pero esperada es que la concatenación de los 3 ficheros en los que se han divido el fichero original se corresponden efectivamente al fichero original, sin ninguna diferencia, lo que implica que la separación se ha realizado correctamente y efectivamente era interesante de realizar.

Fichero Original sin separación previa cluster de 3



Fichero Original sin separación previa cluster de 5



El eje Y represente la velocidad(MB/s) y el eje X representa el tamaño(size)

Como hemos dicho antes, los datos del cluster de 3 y del cluster de 5 se comportan de la misma forma para cada fichero, y en consecuencia también si concatenásemos estos, como muestran los gráficos del modelo original.

SCRIPT

```
def main():
    #Abrimos fichero de lectura
    with open(args.input) as file:
        data = file.read().splitlines()

    #Eliminamos cabecera
    data.pop(0)
    #Creamos 3 ficheros de salida
    with open(f"data1.{'csv' if args.csv else 'arff'}", "w", newline='') as file1, \
        open(f"data2.{'csv' if args.csv else 'arff'}", "w", newline='') as file2, \
        open(f"data3.{'csv' if args.csv else 'arff'}", "w", newline='') as file3:
    #Escribimos nueva cabecera
        if args.csv:
            file1.write("size,hour,MB/s\n")
            file2.write("size,hour,MB/s\n")
            file3.write("size,hour,MB/s\n")
    #Tratamiento del fichero
        for line in data:
            line = line.split(",")
            if line[0] == "-1":
                continue
            try:
                line.pop(2)
            except:
                pass
            try:
                temp = line[1].split("\t")
            except:
                print(line)
            line[1] = temp[0]
            # Refactor
            line.append(fractor_number(temp[1]))
            #Distribuimos datos al fichero correspondiente
            if float(line[0]) <= 722971.75:
                file1.write(f"{line[0]},{line[1]},{line[2]}\n")
            elif 722971.75 < float(line[0]) <= 1807429.5:
                file2.write(f"{line[0]},{line[1]},{line[2]}\n")
            else:
                file3.write(f"{line[0]},{line[1]},{line[2]}\n")
```