Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-2693 Laboratorio de Algoritmos III

Proyecto 2: Clustering

1. Introducción

La tarea de *clustering* o agrupamiento consiste en distribuir en grupos un conjunto de objetos, de forma tal que los objetos similares entre sí queden en el mismo grupo, y los objetos distintos entre sí queden en grupos separados. Formalmente, se tienen un conjunto de objetos C y una función de distancia $d(p, q)$, $p, q \in C$, que representa el grado de disimilitud entre dos objetos. Esta función de distancia debe ser simétrica, es decir, $d(p, q) = d(q, p)$.

2. Descripción del proyecto

En este proyecto, se requiere que usted implemente un método de clustering para agrupar una serie de puntos en el plano 2D, donde cada punto $p = (x, y)$ se define como un par de coordenadas x e y donde x es la abscisa e y es la ordenada.

La distancia entre dos puntos $p = (x_1, y_1)$ y $q = (x_2, y_2)$ se define como:

$$d(p, q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

En otras palabras, se utiliza la distancia Euclidiana como función de distancia.

El método de clustering que usted debe implementar es el método conocido como *single-linkage clustering*. Éste método agrupa un conjunto de objetos en K clusters, es decir, el número de clusters K es un valor que se debe determinar a priori.

El método de single-linkage tiene como función objetivo una función conocida como *spacing function* o función de espaciamiento. Esta función representa la distancia más pequeña entre dos objetos separados, es decir, pertenecientes a clusters distintos. Dado que el objetivo de la clusterización es que objetos cercanos estén en la misma clase y objetos distantes en clases

separadas, el objetivo es maximizar la función de espaciamiento. En otras palabras, mientras mayor sea el valor de la función de espaciamiento, mejor.

La función de espaciamiento se define formalmente como:

$$\min_{p,q \in C,\, p,q \text{ separados}} d(p,q)$$

En el método de single-linkage, inicialmente todos los objetos están separados. Es decir, existe un cluster diferente para cada objeto. Luego, el método procede de manera voraz. Es otras palabras, en cada iteración se reduce la función de espaciamiento. Para esto, se unen los dos clusters que contienen los dos objetos separados con menor distancia entre ellos. Éste ciclo se repite hasta que el número de clusters sea igual a K. A continuación se presenta el pseudocódigo del algoritmo de single-linkage:

Entrada: conjunto de puntos C, número de clusters K
agrupar(C, K)
    Inicialmente, cada punto en un cluster separado
    Repetir hasta que haya sólo K clusters:
        $(p, q) \leftarrow$ par de puntos separados más cercanos
        unir los clusters que contienen a p y q fusionándolos en un solo cluster

3. Formato de entrada

Cada instancia estará guardada en un archivo de texto donde la primera línea contendrá el número de clusters K que se desean obtener. De la segunda línea en adelante, se listarán los puntos uno por línea. Cada punto se representará simplemente como x e y separados por un espacio. Por ejemplo:

```
3
3.4727 3.6628
1.8200 -7.8574
3.7897 -7.8014
-8.1517 3.8678
-7.9008 -7.6990
-7.9611 1.2429
1.2769 6.0866
0.1041 -7.7261
```

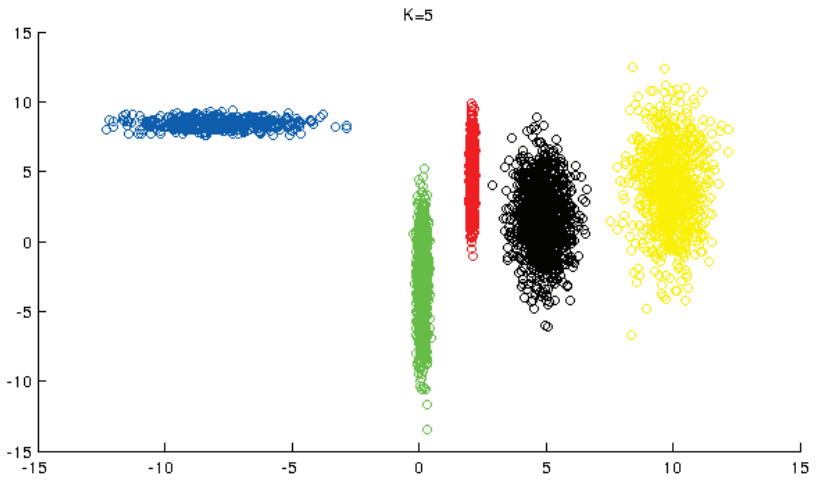

Figura 1: Visualización de cinco clusters de puntos en el plano.

4. Interfaz

El formato de llamada del programa es:

```
agrupar [-q] nombre_de_archivo
```

Donde:

- `-q` es un parámetro opcional. Si está presente, el programa no tendrá salida gráfica.
- `nombre_de_archivo` es el nombre del archivo a procesar.

Si no se especifica el parámetro `-q`, el programa debe demostrar como resultado una gráfica de puntos en el plano cartesiano donde se muestre el resultado de la clusterización, en otras palabras, los puntos de cada cluster se dibujan de un color distinto. El título de la gráfica debe indicar el valor de K. La figura 1 muestra un ejemplo de dicha visualización.

El programa debe además imprimir por consola el número de puntos procesados y el tiempo tomado por el algoritmo, en milisegundos. Ambos valores se deben imprimir en una línea, separados por un espacio.

5. Informe

Usted debe entregar un informe que incluya las siguientes secciones:

1. Introducción
2. Detalles de implementación
3. Análisis de tiempo de ejecución: En esta sección, usted debe determinar analíticamente la complejidad del algoritmo de *single-linkage clustering*. Luego, debe analizar experimentalmente el tiempo de ejecución del algoritmo. Para ello, fueron publicados 100 archivos de datos con un número aleatorio de puntos entre 1 y 10000 en `ldc.usb.ve/~cgomez/algoritmos3/p2`. Usted debe graficar el tiempo de corrida del algoritmo en función del número de puntos. Debe además incluir una discusión de los resultados donde discuta los siguientes aspectos:
   - ¿Cuál es la complejidad del algoritmo?
   - ¿Se corresponde la complejidad con el comportamiento del tiempo de corrida del algoritmo? Si este comportamiento es peor del esperado, explique por qué y cómo se podría mejorar.
   - ¿Es similar el algoritmo de single-linkage a alguno de los algoritmos de grafos vistos en clase? En caso afirmativo, ¿cuál algoritmo y en qué se diferencia de este?
4. Conclusiones
5. Bibliografía

6. Indicaciones adicionales

- El formato de llamada de su programa debe ser el indicado en este enunciado, y no `java nombreDelPrograma`. Para lograr esto, usted debe realizar un *shell script*.
- Para la graficación se sugiere que utilice la biblioteca *JFreeChart*.

7. Condiciones para la entrega

Todo código y escrito que entregue debe ser obra de su equipo. En caso de utilizar código o texto tomado de alguna otra fuente, esta debe ser citada adecuadamente. Usted deberá acompañar ambas entregas con la declaración de autenticidad para entregas que se adjunta a este enunciado, firmada por todos los integrantes de su equipo.

8. Entrega

Usted debe entregar el código fuente de su proyecto en un archivo llamado `proyecto2Apellido1Apellido2.zip`, donde *Apellido1* y *Apellido2* son el primer apellido de los integrantes de su equipo. Incluya en este archivo un *Makefile* que permita compilar su proyecto. El informe debe ser entregado en formato PDF en un archivo llamado `proyecto2Apellido1Apellido2.pdf`, siguiendo la misma convención.

Las entregas deben ser enviadas a ambas direcciones de correo: `armarpc@gmail.com` y `saulhidalgoaular@gmail.com` con el título *Proyecto 2*.

El código de su proyecto puede ser entregado hasta el **miércoles 4 de marzo de 2015 hasta las 11:59 pm**.

El informe de su proyecto puede ser entregado hasta el **viernes 13 de marzo de 2015 hasta las 11:59 pm**.

Declaramos que entendemos que la honestidad es uno de los valores fundamentales de la Universidad Simón Bolívar y que el plagio o la copia en cualquier evaluación constituye una falta de probidad en el ejercicio de nuestras obligaciones como alumnos. Sabemos que esta falta puede ser severamente sancionada según el *Reglamento de Sanciones y Procedimientos Disciplinarios*, disponible en sistema.cenda.usb.ve/reglamentos/ver/428. Asimismo declaramos que el trabajo contenido en esta entrega ha sido realizado solamente por los suscritos miembros del grupo.

Carnet	Nombre y apellido	Firma

(Este modelo es una adaptación del presentado en el documento *Prevención del Fraude Académico en los Cursos de Laboratorio* Elaborado por la Coordinación de Ingeniería de Materiales).