UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

CI-0117 Programación paralela y concurrente

Grupos 2 y 3

Enunciado tarea programada II

Fecha de entrega: 2020/Nov/06

Modalidad: **individual**

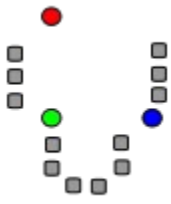
Agrupamiento de un bloque de ***N*** observaciones en ***R*** conjuntos

Tenemos un bloque de datos con ***N*** observaciones (o_1, o_2, …, o_N) y queremos agruparlas en ***R*** conjuntos (S_1, S_2, …, S_R), con ***R*** <= ***N***. Cada observación es un vector de números reales de ***d*** dimensiones (o_i={x_1, x_2, …, x_d}). La idea es encontrar el centro o media C_i de cada conjunto S_i de manera que la suma de las distancias al cuadrado (varianza) de cada observación o_i al centro C_i de su conjunto sea mínima. La idea es agrupar los datos por similaridad y se trata de un problema computacionalmente complejo (NP-hard).

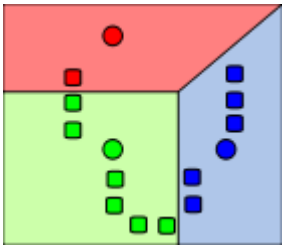
Descripción del algoritmo:

- Escoger de manera aleatoria los ***R*** centros C_i (opcional)
- Colocar a cada punto en una clase de manera aleatoria
- Mientras (no haya convergencia)
  - Construir los nuevos centros C_i utilizando un promedio de los elementos de S_i
  - Crear los ***R*** conjuntos S_i asignando cada muestra al centro más cercano
  - Si los centros C_i no cambian, tenemos convergencia

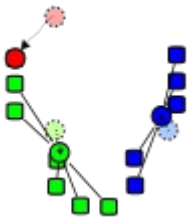
Las siguientes imágenes demuestran este algoritmo para construir los agrupamientos, en este caso R=3, los centros (puntos de color) fueron generados de manera aleatoria, los datos están representados por los cuadrados:



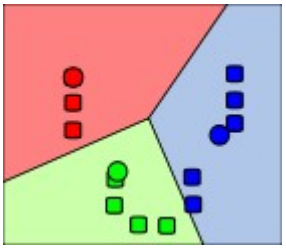
1. Construimos los centros como el promedio de los elementos asociados (R=3)



2. Construimos los grupos asociando cada punto con el centro más cercano



3. Calculamos los nuevos R centros a partir de la media de los puntos en el grupo



4. Repetimos los pasos 2 y 3 hasta lograr el punto de convergencia

Pretendemos que la suma de todas las distancias hasta sus centros sea mínima (disimilaridad). El algoritmo descrito no garantiza la convergencia al máximo global, pues depende de la selección inicial de los centros. A partir de distintos valores para los centros, haga que su programa obtenga el "mejor" mínimo de tres intentos.

Construir en C++ una versión serial para resolver este problema, creando un vector para almacenar las muestras (**N**), utilizando el constructor indicado más abajo y otro vector para representar los centros (**R**). Proveemos un ejemplo base que construye los elementos indicados. Además, el programa debe construir un vector de enteros para almacenar la clase $\mathbf{C_i}$ a la que pertenece cada punto.

La nota de esta tarea está basada en 100 puntos, por 10 puntos adicionales, modifique su programa para que realice los cálculos empleando puntos en **tres** dimensiones, tanto la versión serial como la de OpenMP.

Clases C++ provistas

- **Punto**
  - Representa las observaciones, en este proyecto puntos de dos dimensiones (x, y)
  - Posee un método "dist2" para calcular la distancia al cuadrado entre dos puntos (varianza) de dos dimensiones
  - Hay operaciones básicas para los puntos: sume, divida, ponga, ver

- **VectorPuntos**
  - Representa el bloque de observaciones como un vector
  - Posee un constructor para crear puntos al azar, circunscritos a un círculo del radio indicado
  - Posee un constructor para crear todos los puntos en cero
  - Método "variabilidad" para sumar las distancias entre un punto central y todos las observaciones
  - Método "disimilaridad" para calcular la suma de todas las variabilidades de un grupo de centros
  - Posee un método visualizar los conjuntos generados como una imagen EPS

Ejemplo provisto (medios.cc)

- Construye un bloque de observaciones aleatorias y un conjunto de centros inicializados en el origen
- Makefile para compilar los códigos provistos

Imagen ejemplo (ci0117.eps)

- Imagen con 100000 puntos y 23 conjuntos

Restricciones adicionales

- Su solución **debe** emplear las clases provistas, no puede representar las muestras (puntos) por medio de otras estructuras de datos. Puede modificar las clases agregando nuevos métodos, pero no puede agregar variables de instancia, excepto si están autorizadas por el profesor. Puede agregar clases adicionales
- Debe desplegar la cantidad total de cambios que se realizaron para encontrar el mínimo
- Debe crear dos maneras distintas para construir los centros inicialmente, primer paso en el algoritmo descrito. Es válido utilizar como primer método “aleatorio”
- El programa puede recibir por parámetros la cantidad de muestras a crear, la cantidad de conjuntos para dividir las muestras y el nombre del archivo para visualizar el resultado
- Debe construir una versión utilizando **OpenMP** para mejorar el rendimiento, las muestras deben ser procesadas por varios hilos. También puede generar hilos para cubrir otras tareas

Referencias

Imagen ejemplo animada: https://os.ecci.ucr.ac.cr/ci0117/proyectos/Centros.gif