Reporte tarea 12 - Programación y algoritmos

Erick Salvador Alvarez Valencia, Centro de Investigación en Matemáticas, A.C.

***Index Terms*—Programación orientada a objetos, Big data, Clustering, K-Means.**

I. Introducción

En el presente reporte se expondrá un algoritmo de aprendizaje no supervisado que funciona para realizar *clustering* con grandes cantidades de información, el K-Means, dicho algoritmo funciona de forma iterativa organizando los datos en objetos llamados clusters, esto en base a unas estructuras llamadas centroides que se determinan aleatoriamente. Se analizará más a detalle el algoritmo, así como unos ejemplos de su funcionamiento mostrando las gráficas generadas, y por último se mencionarán algunas conclusiones sobre el mismo algoritmo.

mds
Noviembre 21, 2017

II. Triage

II-A. Descripción

El triage es un protocolo de intervención que en este caso se ejecuta en cierto hospital a los pacientes que ingresan al mismo, los pasos a ejecutar son los siguientes:

El K-Means es un algoritmo de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano. Es un método utilizado en minería de datos.

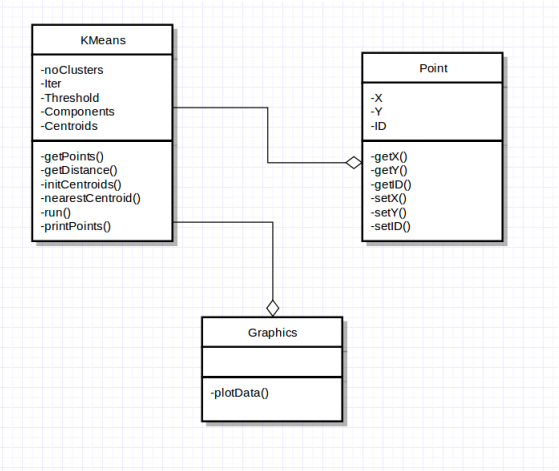
Para comenzar se determina arbitrariamente el número de clusters con los que se trabajará y a continuación por cada centroide se eligen sus coordenadas iniciales. Aquí podemos hacer una elección arbitraria o completamente aleatoria, para el presente se utilizó esta técnica, se eligieron las coordenadas mediante un $rand()$ siempre y cuendo estuvieran dentro del rango de valores que tomaban los datos. Una vez elegidos los centroides iniciales se procedió a iniciar la parte iterativa, la idea es que por cada punto de los datos, se determina el centroide más cercano usando la distancia euclideana, de esta forma podemos asociar los datos a los centroides disponibles. EL siguiente paso es calcular las nuevas posiciones de los centroides, esto se hace tomando la media de todos los puntos en su componente X y Y por cada centroide.

Estos dos pasos se realizan hasta que se presente convergencia, la cual podemos medir con el movimiento que realizaron los centroides con respecto a su posición anterior, se calcula la distancia y si esta es menor que un cierto límite podemos asumir que el algoritmo ha convergido y lo detenemos. Otro criterio de paro es el cumplir un cierto número de iteraciones especificado por el usuario, de esta forma nos aseguramos que el algoritmo no tarde demasiado con un costo de exactitud en sus datos de salida. Para ilustrar lo anterior se mostrará el pseudocódigo principal del algoritmo:

Algorithm 1 KMeans.

```
1:  procedure KMEANS(points, noClusters)
2:      components ← Vector de vectores de puntos.
3:      cAct ← Vector de puntos.
4:      cAnt ← Vector de puntos.
5:      for i ← 0 to noClusters do
6:          cAct ← rand(min, max).
7:      while True do
8:          for i ← 0 to noCLusters do
9:              Asociar el punto con su centroide más cercano.
10:         for i ← 0 to points.length do
11:             cAct ← (1/n) Σ_{i=1}^{n} points_i.
12:         if Cumplió convergencia o número de iteraciones
    realizado then
13:             Terminar.
14:     return components
```

Hay que mencionar que la presente implementación se realizó utilizando el paradigma de programación orientada a objetos para facilitar su desarrollo, primeramente se generó una clase de almacenamiento a los puntos donde se guarda su componente x, su componente y y un Id el cual sirvió para asociar el punto a un cierto centroide. La otra clase más importante fue la **KMeans** la cual contenía los métodos principales para ejecutar el algoritmo. A continuación se muestra el diagrama de clases mediante el cual se basó la estructura del programa.



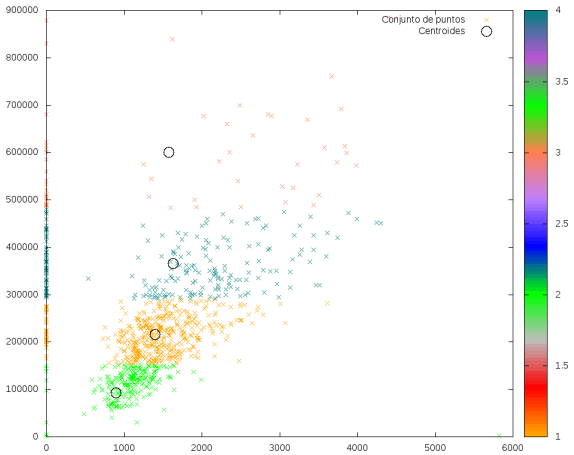
(a) Figura 1. UML del programa.

Podemos ver en la Figura 1. que hay una clase de gráficos, la cual se utilizó para generar la gráfica de los puntos ya clasificados y sus centroides, para esto se usó la librería de GNUPlot la cual contiene varios métodos que facilitan la generación de gráficas. La manera de ejecutar los comandos de GNUPlot fue con el uso de pipes de que usan el estándar del sistema operativo para comunicar a varios programas mediante una conexión bidireccional.
En la siguiente sección se mostrarán algunas gráficas hechas en GNUPlot como resultado de la ejecución del K-Means.

III. Ejemplo de ejecución

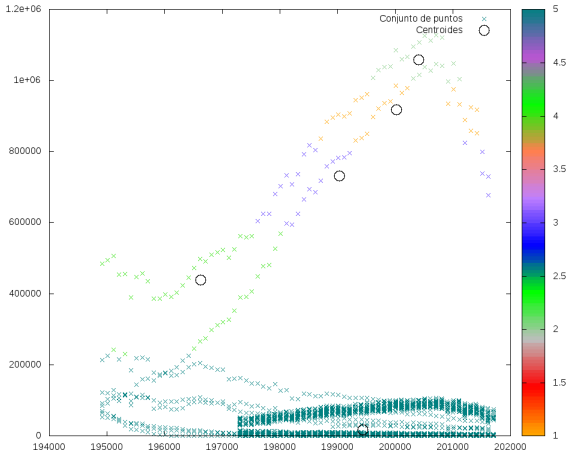
A continuación se muestra el resultado de la ejecución del programa usando dos diferentes conjuntos de datos:

(b) Figura 2. Salida del programa.



(c) Figura 3. Gráfica del primer conjunto de datos.

(d) Figura 4. Salida del programa.



(e) Figura 5. Gráfica del segundo conjunto de datos.

En las Figuras anteriores se puede ver la división de los puntos por medio de distintos colores y, a su vez se marcaron de manera diferente a los centroides para ver que en efecto se asocian a su cluster de datos.

IV. Conclusiones

En el presente reporte se analizó el desarrollo y los resultados de la implementación del algoritmo de K-Means en el lenguaje C++ utilizando el paradigma POO, como una posible mejora el algoritmo se puede adaptar para trabajar utilizando con bases de datos SQL ya que la versión actual sólo cuenta con un parser sobre archivos CSV. De la misma forma también se podrían utilizar otras métricas para clasificación en diferentes conjuntos de datos en lugar de la norma euclideana.