Búsqueda por Similitud

Profesor Heider Sanchez

El objetivo del laboratorio es aplicar la búsqueda por rango y la búsqueda de los k vecinos más cercano sobre un conjunto de vectores característicos.

Se toma como referencia la colección de imágenes de flores ***Iris*** *(https://archive.ics.uci.edu/ml/datasets/iris)*, en donde cada imagen es representada por un vector característico de 4 dimensiones que recoge información del ancho y largo del sépalo y del pétalo. Además, las imágenes están agrupadas en tres categorías: *versicolor, setosa y virginica*.

Iris Versicolor **Iris Setosa** **Iris Virginica**

P1. Búsqueda por Rango

Implementar en cualquier lenguaje de programación el algoritmo lineal de búsqueda por rango, el cual recibe como parámetro el objeto de consulta y un **radio de cobertura**. Luego usando la Distancia Euclidiana (ED) se retorna todos los elementos que son cubiertos por el radio.

```
Algorithm RangeSearch(Q, r)
1.  result = [ ]
2.  for all objects Ci in the collection
3.      dist = ED(Q, Ci)
4.      if dist < r
5.          append(result, Ci )
6.      endif
7.  endfor
8.  return result
```

- Aplique la búsqueda para 3 elementos de la colección (Q_{15}, Q_{82}, Q_{121}) y para tres valores de radio ($r1 < r2 < r3$).
- El objeto de consulta debe ser retirado de la colección antes de aplicar la búsqueda.
- Para saber que valores de radio seleccionar, **debe primero realizar un análisis de la distribución de las distancias computando N veces la distancia entre dos elementos aleatorios de la colección.**
- Para evaluar la efectividad del resultado se debe usar la medida de Precisión ¿Cuántos de los objetos recuperados pertenecen a la misma categoría de la consulta?:

$$PR = \frac{\#ObjetosRelevantesRecuperados}{\#ObjetosRecuperados}$$

A continuación, se proporciona el cuadro que debe ser llenado por el alumno.

PR	Q_{15}	Q_{82}	Q_{121}
$r1 =$ **0.8**	1.0	1.0	0.67
$r2 =$ **1.5**	1.0	0.79	0.45
$r3 =$ **2.6**	0.98	0.53	0.47

P2. Búsqueda KNN

Usando los mismos objetos de consulta del ejercicio anterior, implementar y aplicar el algoritmo lineal de búsqueda de los k vecinos más cercanos (KNN) variando el k entre {2, 4, 8, 16, 32}.

```
Algorithm KnnSearch(Q, k)
1.  result = [ ]
2.  for all objects Ci in the collection
3.      dist = ED(Q, Ci)
4.      append(result, {Ci , dist})
5.  endfor
6.  orderByDist(result)
7.  return result[1:k]
```

**** La forma más eficiente de implementar el KNN es gestionando la lista de resultado en una cola de prioridad máxima. ¿Analice la complejidad?**

PR	Q_{15}	Q_{82}	Q_{121}
$k = 2$	1.0	1.0	1.0
$k = 4$	1.0	1.0	1.0
$k = 8$	1.0	1.0	0.875
$k = 16$	1.0	1.0	0.625
$k = 32$	1.0	1.0	0.5

Preguntas:

1- ¿Cuál es la complejidad computacional de ambos métodos de búsqueda en función de cálculos de la ED?

La complejidad para la búsqueda por rango es de O(N), ya que se necesita recorrer todos los elementos de la colección. La complejidad para la búsqueda KNN es de O(N*D) + O(N log N), donde N es el número de elementos de la colección y D la dimensión del objeto, sin embargo, su complejidad puede mejorar al utilizar cola de prioridad, ya que mejora el costo de ordenación, la complejidad en este caso sería de O(N*D*log K) + O(K log K), donde K es el número de elementos del heap.

2- ¿Cuál de los dos métodos de búsqueda usted usaría en un ambiente real de recuperación de la información? Sustente su respuesta.

La elección entre la búsqueda por rango y la búsqueda de los K vecinos más cercanos (KNN) en un entorno real de recuperación de información depende en gran medida de la naturaleza del problema y del conjunto de datos. Respecto a la búsqueda por rango, esta es efectiva cuando se desea obtener todos los puntos de datos dentro de un rango específico, se puede aplicar por ejemplo en situaciones donde no se sabe de antemano cuántos resultados estarán dentro del rango definido. Por otro lado, respecto a KNN, esta es una opción efectiva cuando se desea un número específico de los puntos de datos más similares, sin importar qué tan cercanos o lejanos estén en realidad. Una ventaja de este tipo de búsqueda es que garantiza la obtención de resultados, a diferencia de la búsqueda por rango, donde en ocasiones no existen elementos similares al elemento de consulta en el rango seleccionado, por lo que el resultado final es vacío.