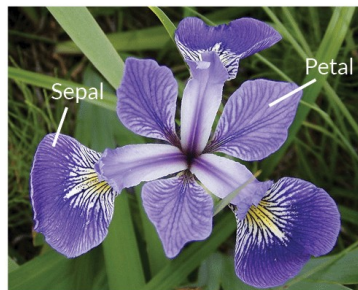


Búsqueda por Similitud

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)		
<ul style="list-style-type: none"> - Aplique la para de la (Q₁₅, Q₈₂, tres valores r₁ < r₂ < r₃). - El objeto de debe ser colección antes de aplicar la búsqueda. 	1. result = []	búsqueda elementos colección Q ₁₂₁) y para de radio (consulta retirado de la
	2. for all objects C _i in the collection	
	3. dist = ED (Q, C _i)	
	4. if dist < r	
	5. append(result, C _i)	
	6. endif	
	7. endfor	
	8. return result	

- 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?:

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

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

PR	Q_{15}	Q_{82}	Q_{121}
$r1 = i$			
$r2 = i$			
$r3 = i$			

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 C_i in the collection
3.	dist = ED (Q, C_i)
4.	append(result, { C_i , dist})
5.	endfor
6.	orderByDist(result)
7.	return result[1:k]

**** La mejor forma de implementación es gestionando la lista de resultado como una cola de prioridad máxima. Analice la complejidad.**

PR	Q_{15}	Q_{82}	Q_{121}
$k=2$			
$k=4$			
$k=8$			
$k=16$			
$k=32$			

Preguntas:

- 1- ¿Cuál es la complejidad computacional de ambos métodos de búsqueda en función de cálculos de la ED?
- 2- ¿Cuál de los dos métodos de búsqueda usted usaría en un ambiente real de recuperación de la información? Sustente su respuesta.