Sugerencia Tarea 3

Recocido simulado para IRIS

Clustering IRIS

- Paso 0.
- Lectura de datos. Guarden las medidas de cada observación en una matriz $Obs[150][4]$
- Ahora normalicen los datos mediante la siguiente fórmula

$$Obs[i][j] = \frac{Obs[i][j] - C_{min}(j)}{C_{max}(j) - C_{min}(j)}$$

Donde

$Obs[i][j]$, es el valor de la observación *i* en la columna *j*

$C_{min}(j)$, es el mínimo valor de la columna *j*

- $C_{max}(j)$, es el máximo valor de la columna *j*

Clustering IRIS

Paso 1.

Representación de las soluciones:

$$Sol\_act[150] = (x_0, x_1, x_2, x_3, \dots, x_{148}, x_{149})$$

Donde $x_i \in \{0,1,2\}$.

Por ejemplo

$$Sol\_act[150] = (2,2,0,1, \dots, 1,1\ )$$

Clustering IRIS

- Paso 2.
- Generación de solución inicial.
- $Sol\_Inicial()$
  - $int\ Solucion[150]$
  - For i = 0 to 149
  - $\quad Solucion[i] \leftarrow rnd(0,1,2)$
  - Revisar validez de la solución, en caso contrario corregir
  - return($Solucion$)

Al final debe revisar que cada uno de estos valores aparece al menos una vez.

Clustering IRIS

- Paso 3.
- Generar la función objetivo.
- $Func\_Obj(Solucion[150])$
  - Calculamos los centroides de cada grupo
  - $float\ C[3][4] = \bar{0}$ //Son 3 centroides, cada uno de 4 dimensiones
  - $int\ contador[3] = \bar{0}$
  - For i = 0 to 2
  - For j = 0 to 149
  - if $Solucion[j] == i$
  - $C[i] \mathrel{+}= Obs[j]$
  - $contador[i] \mathrel{+}= 1$
  - For i = 0 to 2
  - $C[i] = C[i]/contador[i]$ //Ya calculamos los centroides

Clustering IRIS

- Paso 3. (continuación…)
  - $float\ distancia = 0$
  - For i = 0 to 2
  - For j = 1 to 150
  - if $Solucion[j] == i$
  - $distancia \mathrel{+}= \sum_{k=0}^{3} abs(C[i][k] - Obs[j][k])$
  - $return\ (distancia)$

Clustering IRIS

- Paso 4.
- Main()
  - $float\ Obs[150][4]$
  - Leer, cargar y normalizar los datos
  - $int\ Mejor\_sol[150], Sol\_act[150]$, $Mov1$
  - $float\ Mejor\_costo, Costo\_act, Costo\_Vecino$
  - $float\ TempI = 10, TemF = 0.001, alfa = 0.95, Temp$
  - $int\ MaxIt = 200, It$
  - $Mejor\_Sol = Sol\_act = Sol\_Inicial()$ //Generar solución inicial
  - $Mejor\_costo = Costo\_act = Func\_Obj(Sol\_act)$

- // Inicia recocido simulado
- $Temp = TempI$
- $While(Temp > TempF)$
- for It = 0 to MaxIt
- a = $rnd\_int(0,150)$
- $Mov1 = Sol\_act[a]$
- $b = rnd(0,1)$
- $if\ b < 0.5$
- $Sol\_act[a] = (Sol\_act[a] + 1)\ \%3$
- else
- $Sol\_act[a] = (Sol\_act[a] - 1)\ \%3$
- $Costo\_Vecino = Func\_Obj(Sol\_act[150])$
- $v = Costo\_Vecino - Costo\_act$
- $b = rnd(0,1)$
- $if\left(b < exp\left(-{}^{v}/_{Temp}\right)\right)$ // Criterio de Metrópolis
- $Costo\_act = Costo\_Vecino$
- $else$
- $Sol\_act[a]$ = $Mov1$

- $if(Mejor\_costo > Costo\_Vecino)$
- $Mejor\_costo \leftarrow Costo\_Vecino$
- $Mejor\_Sol \leftarrow Sol\_act$
- end for
- $Temp = Temp * alfa$
- end While
- $print(Mejor\_Sol)$
- $return(Mejor\_costo)$

Recocido adaptativo

Clustering IRIS

- Paso 4.
- Main()
  - $float\ Obs[150][4]$
  - Leer, cargar y normalizar los datos
  - $int\ Mejor\_sol[150], Sol\_act[150]\ , Mov1$
  - $float\ Mejor\_costo, Costo\_act, Costo\_Vecino$
  - $float\ alfa3 = 0.90, alfa2 = 0.95, alfa1 = 0.97,$ temperatura
  - $int\ MaxIt3 = 200, MaxIt2 = 300, MaxIt1 = 400, It$
  - $float\ Aceptada, Visitada, Nivel\_Aceptacion$
  - $int\ Precalentado = 1$
  - $Mejor\_Sol = Sol\_act = Sol\_Inicial()$ //Generar solución inicial
  - $Mejor\_costo = Costo\_act = Func\_Obj(Sol\_act)$

- // Inicia recocido simulado
- $temperatura = 5$
- $Nivel\_Aceptacion = 0.5$
- $While(Nivel\_Aceptacion > 0.01)$ //Cuidado podemos caer en ciclos infinitos
- for It = 0 to $Iteraciones$
- a $= rnd\_int(0,150)$
- $Mov1 = Sol\_act[a]$
- $b = rnd(0,1)$
- $if\ b < 0.5$
- $Sol\_act[a] = (Sol\_act[150] + 1)\ \%3$
- else
- $Sol\_act[a] = (Sol\_act[150] - 1)\ \%3$
- $Costo\_Vecino = Func\_Obj(Sol\_act[150])$
- $v = Costo\_Vecino - Costo\_act$
- $b = rnd(0,1)$
- $if(Costo\_Vecino > Costo\_act)$
- $Visitada \mathrel{+}= 1.0$
- $if\left(b < exp\left(-{}^{v}/_{Temp}\right)\right)$ // Criterio de Metrópolis
- $Costo\_act = Costo\_Vecino$
- $if(Costo\_Vecino > Costo\_act)$
- $Aceptada \mathrel{+}= 1.0$
- $else$
- $Sol\_act[a] = Mov1$

- $if(Mejor\_costo > Costo\_Vecino)$
- $Mejor\_costo \leftarrow Costo\_Vecino$
- $Mejor\_Sol \leftarrow Sol\_act$
- end for
- $Nivel\_Aceptacion$ = Aceptada / Visitada
- if ($Nivel\_Aceptacion$ < 0.8 && Precalentado == 1)
- Temperatura = Temperatura * 1.1
- if ($Nivel\_Aceptacion$ > 0.8 && Precalentado == 1)
- Precalentado = 0
- If ($Nivel\_Aceptacion$ > 0.5)
- Alfa = alfa3
- Iteraciones = $MaxIt3$
- if (0.2 < $Nivel\_Aceptacion$ < 0.5)
- Alfa = alfa2
- Iteraciones = $MaxIt2$
- If ($Nivel\_Aceptacion$ < 0.2)
- Alfa = alfa1
- Iteraciones = $MaxIt1$
- If (precalentado == 0)
- temperatura = temperatura * alfa
- Aceptada = Visitada = 1.0
- End while
- $print(Mejor\_Sol)$
- $return(Mejor\_costo)$