1) Al leer el enunciado entiendo que el resultado no es satisfactorio ya que hay puntos aislados que no son ni muy poderosos, ni muy débiles por lo tanto lo que se puede cambiar es la distribución de dichos ~~algoritmos~~ puntos. Para esto se me ocurren dos posibles soluciones, una podría ser aumentar la cantidad de clusters y en lugar de solo tomar dos clusters tomar 3 o 4 para que haya uno o dos clusters para estos casos borde.

Otra solución podría ser aplicar soft K-means el cual es una generalización de K-means que permite a cada punto pertenecer al mismo tiempo a más de un cluster. Se busca estimar la probabilidad de que el punto pertenezca a cada cluster. Con esta solución, los pesos que fueron muy poderosos serían de un rojo más claro y los muy débiles de un azul claro. De esta forma podemos ver el grado de pertenencia de los puntos a cada cluster, y los que estén en el medio no serían categóricos ni a uno ni a otro necesariamente.

Otra posible alternativa sería utilizar DBScan, ya que este algoritmo permite tener puntos que no son parte de ningún cluster, por lo tanto si solo queremos estimar los poderosos y los débiles como posibles grupos los puntos que no pertenezcan a estas regiones densas pueden ser descartados.

NOTA

De las soluciones propuestas, las primeras las descarto ya que entiendo que se buscan obtener solo los grupos: fuertes y débiles.

* Propongo utilizar soft K means:

Ahora en lugar de asignarle a cada punto un centroide mas cercano como hace K means, se le asigna a cada punto dos probabilidades, siendo cada una de estas la probabilidad que el punto pertenezca a uno de los dos clusters, de tal manera que cuanto mas cerca esta el punto del centroide mayor es su probabilidad.

Para esto calculo la distancia del punto a cada centroide y la probabilidad de pertenecer a un cluster sera: $1 - \frac{\text{dist al cluster}}{\text{suma de todas distancias}}$

Utilizando por ejemplo la distancia euclidea como metrica

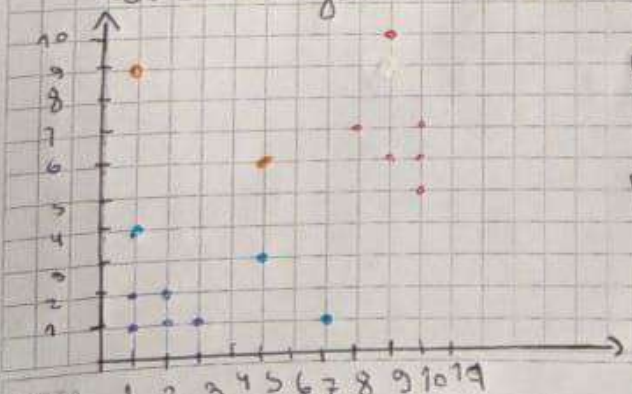
El algoritmo es el siguiente:

Calculo los centroides, asignados al azar como K means

Calculo la distancia de cada punto a los centroides

Calculo la probabilidad de pertenecer a C/cluster

Los clusters quedan determinados con este metodo serán:



$$P(\text{Rojo}) = 1 - \frac{\text{dist rojo}}{\text{dist rojo} + \text{dist azul}}$$

$$P(\text{Azul}) = 1 - \frac{\text{dist Azul}}{\text{dist rojo} + \text{dist azul}}$$

NOTA

Consideramos que este resultado es más apropiado ya que los países que antes eran considerados como poderosos o débiles pero en realidad no eran ninguno de estos, ahora son considerados por su grado de pertenencia a cada cluster, obteniendo una clasificación más interesante y una visualización más rica.
Por ejemplo el punto (2,3) antes era considerado como poderoso, aunque hubiera mandado solo 2 delegaciones, los países que mandan pocos atletas no suelen ser muy poderosos, pero con oro ganaron 9 medallas, por lo que lo vemos mirando, es un país más poderoso que otros pero no es de los más poderosos.
Por esta razón consideramos que la solución propuesta es mejor que la de DBS con ya que en lugar de eliminar los datos en el medio de los dos grupos, los mantenemos y obtenemos información interesante sobre estos.

NOTA

2) Utilizar CF User-User con K=2 para estimar la calificación del item 3 para el usuario a.

En primer lugar, busco los dos usuarios mas similares al usuario a.

a: [1 ? ? 5 1 ? 2 3] → $\bar{a}$ = 2,4
b: [3 2 ? 1 1 ? 3 2] → $\bar{b}$ = 2
c: [2 ? 4 ? 2 ? 4 2] → $\bar{c}$ = 2,8
d: [1 2 ? ? 2 ? ? 3] → $\bar{d}$ = 1,75
e: [? 3 3 ? 1 4 ? 2] → $\bar{e}$ = 2,6
F: [4 ? 5 1 ? ? 4 1] → $\bar{f}$ = 3

Los datos faltantes no los voy a tener en cuenta a la hora de realizar los calculos

Calculo el promedio de cada usuario y se lo resto al vector para centrar las calificaciones

a [-1,4 ? ? 2,6 -1,4 ? -0,4 0.6]
b [1 0 ? -1 -1 ? 1 0]
c [-0.8 ? 1.2 ? -0.8 ? 1.2 -0.8]
d [-0.75 0.25 ? ? -0.75 ? ? 1.25]
e [? 0.4 0.4 ? -1.6 1.4 ? -0.6]
f [1 ? 2 -2 ? ? 1 -2]

Calculo similaridad con correlacion de pearson

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_{xs})(r_{ys} - \bar{r}_{ys})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_{xs})^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_{ys})^2}}$$

NOTA

* Semejanza entre a y b: solo tengo en cuenta puntos en común

A: (-1,4 2,6 -1,4 -0.4 0.6) ||A|| = 3.35

B: (1 -1 1 1 0) ||B|| = 2

Normalizo

A: (-0,418, 0,776, -0.418, -0,119, 0,179)

B: (0.5, -0,5, 0,5, 0,5, 0)

Sum (A,B) = A · B = -0,8655

* Semejanza entre a y c:

A: (-1,4, -1.4, -0.4, 0.6) ||A|| = 2,11

C: (-0.8, -0.8, 1.2, -0.8) ||B|| = 1,83

Normalizo:

A: (-0,66, -0,66, -0,19, 0,284)

C: (-0.437, -0,437, 0,656, -0,437)

sem (A,C) = A · C = 0,328

* semejanza entre a y d:

A: (-1.4 -1.4 0.6) ||A|| = 2,07

D: (-0.75 -0.75 1.25) ||D|| = 1,64

Normalizo:

A: (-0,676, -0,676, 0,29)

D: (-0,457, -0,457, 0,762)

sem (A,D) = 0,839

NOTA

Dante Renardo
102848

Promocional

HOJA 6
FECHA

* semejanza entre a y B:

A: (-1.4 0.6) ||A|| = 1,52

E: (-1.6 -0.6) ||E|| = 1,71

Normalizo:

A: (-0.92 , 0.395)

E: (-0,936 , -0,35)

sim (A,E) = 0,723

* semejanza entre a y F:

A: (-1.4 2.6 -0.4 0.6) ||A|| = 3.04

F: (1 -2 1 -2) ||F|| = 3.16

Normalizo:

A: (-0.46 , 0.855 , -0.132 , 0.197)

F: (0.316 , -0.633, 0.316 , -0.633)

sim (A,F) = -0,85

- Los dos usuarios mas similares a A son D y E
- Sus ~~puntos~~ calificaciones del usuario a para el item 3 son:

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

$$r_{a,i_3} = \frac{0,727}{0,723 + 0,839} \quad \frac{0.839 \cdot 3}{0.839} = 3$$

- Como el usuario D no calificó el item 3, no lo tengo en cuenta para el calculo
- El usuario a calificaría con 3 estrellas el item 3.

NOTA

3)

$PR(0) = 0,2 \cdot 1/8$

$PR(1) = 0.8\left(\frac{1}{2} P_{R0} + P_{R2}\right) + 0.2 \cdot 1/8$

$P_R(2) = 0.8\left(\frac{1}{2} P_{R0} + \frac{1}{4} P_{R3}\right) + 0.2 \cdot 1/8$

$P_R(3) = 0.8\left(P_{R8}\right) + 0.2\ 1/8$

$P_R(4) = 0.8\left(P_{R1} + \frac{1}{4} P_{R3}\right) + 0.2\ 1/8$

$P_R(5) = 0.8\left(\frac{1}{2} P_{R4} + \frac{1}{4} P_{R3}\right) + 0.2 \cdot 1/8$

$P_R(6) = 0.8\left(\frac{1}{4} P_{R3}\right) + 0.2 \cdot 1/8$

$P_R(8) = 0.8\left(\frac{1}{2} P_{R4} + P_{R5} + P_{R6}\right) + 0.2 \cdot 1/8$

a) Con estas fórmulas para calcular el Page Rank con teletransportación basándonos en el grafo, cree un algoritmo y obtenga los siguientes tras 10 iteraciones

	0	1	2	3	4	5	6	8	
0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8	Redondeo
...									
10	0,025	0,~~35~~	0,08	0,22	0,14	0,13	0,07	0,24	→ Para que sume 1

b) sabiendo que la page 0 no es confiable, vamos a aplicar Trust Rank y favorecemos en la teletransportación a las páginas confiables:

$P_R(0) = 0$ → No tiene tanta de entrada y no favorecemos teletransportación

$P_R(1) = 0.8\left(\frac{1}{2} P_{r0} + P_{r2}\right) + 0,2 \cdot 1/7$ → Cantidad de Páginas confiables

$P_R(2) = 0.8\left(\frac{1}{2} P_{r0} + \frac{1}{4} P_{r3}\right) + 0,2 \cdot 1/7$

$P_R(3) = 0.8\left(P_{r8}\right) + 0.2\ 1/7$

$P_R(4) = 0.8\left(P_{r1} + \frac{1}{4} P_{r3}\right) + 0.2 \cdot 1/7$

$P_R(5) = 0.8\left(\frac{1}{2} P_{r4} + \frac{1}{4} P_{r3}\right) + 0.2\ 1/7$

$P_R(6) = 0.8\left(\frac{1}{4} P_{r3}\right) + 0.2\ 1/7$

$P_R(8) = 0.8\left(\frac{1}{2} P_{R4} + P_{r5} + P_{r6}\right) + 0.2 \cdot 1/7$

Con estas nuevas modificaciones, el peso unitario obtenido tras diez iteraciones es:

	0	1	2	3	4	5	6	8
0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
...								
10	0	0,09	0,07	0,23	0,15	0,13	0,07	0,26

→ Rama 1

4) Stream: {3, 6, 6, 3, 3, 6, 5, 6}

$h(x) = x \bmod 32$

Flajolet Martin: único estimador

↳ 5 bits, se consideran bits a derecha

Primero calculo momento de orden 0 del stream

$M^0(S) = 3$ → Hay tres elementos diferentes en el stream: 3, 6 y 5

$h(3) = 3 \bmod 32 = 3 \rightarrow 00011$

$h(5) = 5 \bmod 32 = 5 \rightarrow 00101$

$h(6) = 6 \bmod 32 = 6 \rightarrow 00110 \rightarrow r = 1$

Por lo tanto el momento de orden 0 para el stream se calcula como $2^r = 2^1 = 2$.

El algoritmo de Flajolet Martin estima con esta función que el momento de orden 0 para el stream dado, es decir, la cantidad de elementos diferentes es 2

Considero que la aproximación es muy cercana a 3, por lo que esta función parecería andar bien al menos con la poca cantidad de datos que tenemos. A su vez utilizando un único estimador solo podremos obtener resultados que sean potencias de 2 y ya vimos obtendríamos 3 como resultado.

B) Consideramos que plantear una función de hashing alternativa no mejoraría sustancialmente lo obtenido en el punto a ya que con un único estimador podríamos obtener un único bit 0 a la derecha o a lo sumo dos (en caso de haber más los estimaríamos cero por) con estos resultados podríamos obtener 2 o 4 como momento de orden 0 del stream, y ambos resultados distorsionan en una unidad el momento de orden 0 real que es 3. Además, con un único estimador solo podemos aproximar con valores que sean potencias de 2,
En lugar de una función alternativa propondría utilizar varias funciones del estilo $(a \times + b) \bmod 32$ de manera de obtener varios estimadores y reportarlos en grupo. Luego obtengo la mediana de los momentos de cada grupo y le aplico el promedio. De esta manera podré aproximar a momentos que no sean potencias de 2 ~~eventualmente pudiendo ser impares~~.

C) Plantear función del tipo $h(x) = (a \times + b) \bmod 32$ que empeore peligrosamente

$h(x) = (0 \cdot x + 16) \bmod 32$

$h(3) = 16 \rightarrow 10000$ $\quad M_0(3) = 2^4 = 16$
$h(5) = 16 \rightarrow 10000$
$h(6) = 16 \rightarrow 10000 \Rightarrow r = 4$

NOTA

Dante Renovato
102848

Promocional

10

Esta función es muy inútil ya que genera muchas colisiones al tender valores constantes y además genera muchos bits en 0 a lo largo, por lo que el algoritmo de Floyd-Steinberg produce un valor muy errado para el momento de orden 0 del stream.

NOTA

5) Proponer datasets de train y/o test de al menos 4 puntos cada uno, y un modelo tal que se cumpla

a) Una regresion cuyo MSE en test sea 0 pero en train sea 0,5

Propongo una regresion lineal: $\overset{h(x)}{\tilde{y}} = mx + b$

MSE: $(y_{test} - h(x))^2$

Set de Train:

X	Y	h(x)
0	0	$\sqrt{2}/2$
0	0	$\sqrt{2}/2$
0	0	$\sqrt{2}/2$
0	0	$\sqrt{2}/2$

$\frac{1}{N} \Sigma (y - h(x))^2 = 0,5$

$\hookrightarrow N = 4$, 4 puntos

$\Sigma (y - h(x))^2 = 2$

$(y - h(x))^2 = \frac{\sqrt{2}}{2}$

Set de Test:

X	Y	h(x)
$\sqrt{2}/2$	$\sqrt{2}/2$	$\sqrt{2}/2$
$\sqrt{2}/2$	$\sqrt{2}/2$	$\sqrt{2}/2$
$\sqrt{2}/2$	$\sqrt{2}/2$	$\sqrt{2}/2$
$\sqrt{2}/2$	$\sqrt{2}/2$	$\sqrt{2}/2$

$\frac{1}{N} \Sigma (y - h(x))^2 = 0$

De esta manera $\overset{h(x)}{\tilde{y}} = \overset{0}{\tilde{a}} \cdot x + \overset{\sqrt{2}/2}{\tilde{b}}$

$h(x) = \sqrt{2}/2$

EMS Train: $\frac{1}{4} \cdot 4 (0 - \frac{\sqrt{2}}{2})^2 = 0,5$

EMS Test: $\frac{1}{4} \cdot 4 (\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2})^2 = 0$

NOTA

Promocional

c) Arbol de decision entrenado sobre datset de train cuyas variables son el binary encoding de una categorica de 3 clases y tiene 0,75 de accuracy entonces

Accuracy: $\frac{\text{bien clasificados}}{\text{elementos clasificados}}$

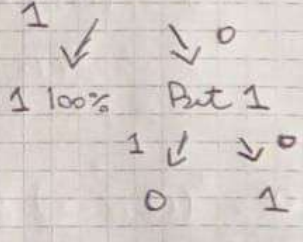
Modelo: ID3

Clase A: 0 0
Clase B: 0 1
Clase C: 1 0
} Binary encoding

Clase D: 1 1
↳ Label 0

Data train:

Bit 0	Bit 1	Label
0	0	1
0	1	0
1	0	1
1	0	1



Con este modelo el tutor de predecir el target de la clase D diremos que es 1 ya que su bit 0 es 1, por estar cargan de las 4 clases, siempre estimaremos bien 3 de ellos y mal 1, obteniendo un accuracy del 75%.

NOTA