Inteligencia Artificial *Juan Carlos Faz Leal*
Tarea 1: FUNDAMENTOS
23 de Febrero del 2023

1. Construyendo la intuición.

   a. Supongamos que corremos descenso de gradiente estocástico una vez por cada una de las cuatro muestras en el orden dado arriba, actualizando los pesos de acuerdo a

$$w \leftarrow w - \eta \nabla_w Loss_{hinge}(x, y, w),$$

   Despues de las actualizaciones, cuáles son los pesos de las seis palabras que aparecen en las reseñas de arriba?

$$\phi(x) = \text{'not good'}, y = -1, w = [0, 0, 0, 0, 0, 0], \eta = 0.1$$

$$\text{El orden es: "pretty, good, bad, plot, not, scenery"} \ \phi(x) = [0, 1, 0, 0, 1, 0]$$

$$Loss_{hinge}(x, y, w) = max(0, 1 - [0, 0, 0, 0, 0, 0] \cdot [0, 1, 0, 0, 1, 0] \cdot -1) = 1$$

$$\nabla_w Loss_{hinge} = -[0, 1, 0, 0, 1, 0] \cdot -1 = [0, 1, 0, 0, 1, 0]$$

$$w \leftarrow [0, 0, 0, 0, 0, 0] - 0.1 \cdot [0, 1, 0, 0, 1, 0] = [0, -0.1, 0, 0, -0.1, 0]$$

$$\phi(x) = \text{'pretty bad'}, y = -1, w = [0, -0.1, 0, 0, -0.1, 0], \eta = 0.1$$

$$\phi(x) = [1, 0, 1, 0, 0, 0]$$

$$Loss_{hinge}(x, y, w) = max(0, 1 - [0, -0.1, 0, 0, -0.1, 0] \cdot [1, 0, 1, 0, 0, 0] \cdot -1) = 1$$

$$\nabla_w Loss_{hinge} = -[1, 0, 1, 0, 0, 0] \cdot -1 = [1, 0, 1, 0, 0, 0]$$

$$w \leftarrow [0, -0.1, 0, 0, -0.1, 0] - 0.1 \cdot [1, 0, 1, 0, 0, 0] = [-0.1, -0.1, -0.1, 0, -0.1, 0]$$

$$\phi(x) = \text{'not bad'}, y = 1, w = [-0.1, -0.1, -0.1, 0, -0.1, 0], \eta = 0.1$$

$$\phi(x) = [0, 0, 1, 0, 1, 0]$$

$$Loss_{hinge}(x, y, w) = max(0, 1 - [0, -0.1, 0, 0, -0.1, 0] \cdot [0, 0, 1, 0, 1, 0] \cdot 1) = 1.1$$

$$\nabla_w Loss_{hinge} = -[0,0,1,0,1,0] \cdot 1 = [0,0,-1,0,-1,0]$$

$$w \leftarrow [-0.1,-0.1,-0.1,0,-0.1,0] - 0.1 \cdot [0,0,-1,0,-1,0] = [-0.1,0,-0.1,0.1,-0.1,0]$$

$$\phi(x) = \text{'pretty scenry'}, y = 1, w = [-0.1,0,-0.1,0.1,-0.1,0], \eta = 0.1$$

$$\phi(x) = [1,0,0,0,0,1]$$

$$Loss_{hinge}(x,y,w) = max(0, 1 - [-0.1,0,-0.1,0.1,-0.1,0] \cdot [1,0,0,0,0,1] \cdot 1) = 1.1$$

$$\nabla_w Loss_{hinge} = -[1,0,0,0,0,1] \cdot 1 = [-1,0,0,0,0,-1]$$

$$w \leftarrow [-0.1,0,-0.1,0.1,-0.1,0] - 0.1 \cdot [-1,0,0,0,0,-1] = [0,0,-0.1,0.1,-0.1,0.1]$$

El peso de las seis palabras fue: $[0,0,-0.1,0.1,-0.1,0.1]$

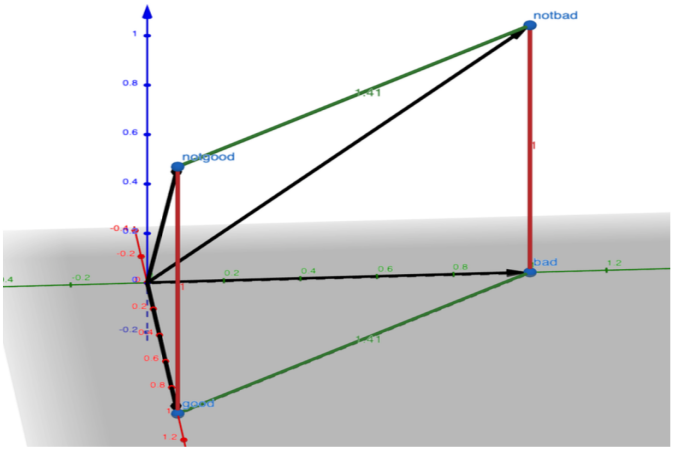
Pruebas: pruebas

b. Dado el siguiente conjunto de datos de reseñas:

i. (-1) bad
ii. (+1) good
iii. (+1) not bad
iv. (-1) not good

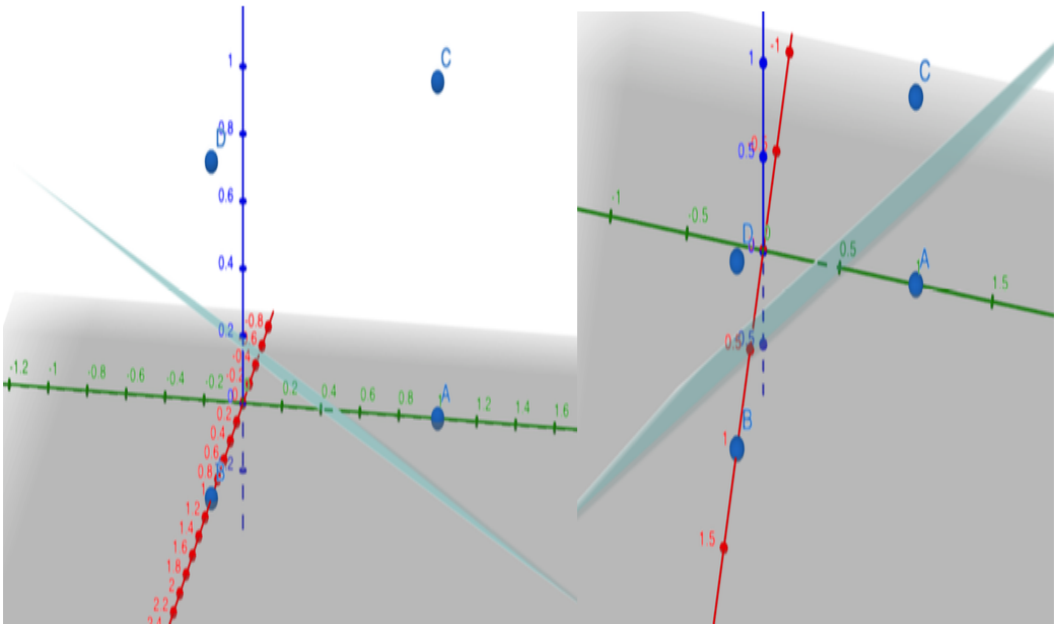
Muestra que no hay clasificador lineal que utilice características de palabras que tenga cero error sobre este conjunto de datos.

Si graficamos cualquier permutación donde el orden es: bad, good, not, $(P_3 = 3! = 6)$ de estos vectores podemos observar que existen ambiguedades en el etiquetado de las palabras como se muestra en la imagen:



donde en una de sus permutaciones quedó así: bad = $[0, 1, 0]$, good = $[1, 0, 0]$, not bad = $[0, 1, 1]$ y not good = $[1, 0, 1]$.

Semánticamente en el inglés 'not good' (No bueno) se parece más a 'bad' (malo) que a 'good' (bueno). Pero en la gráfica podemos ver que la distancia entre 'not good' y 'bad' ($\sqrt{1.41^2 + 1^2} = 1.73$) es mayor que la distancia entre 'not good' y 'good' que vale 1.



Por lo cual los datos no pueden ser clasificados correctamente utilizando unicamente un hiperplano.

Propón una sola característica adicional para tu conjunto de datos con la que pudieramos arreglar este problema.

Para solucionar este problema lo que podemos hacer es agregar una característica

binaria para saber diferenciar cuando se trata de una negación. Ej. Bad, good, not, not good.

Bad = [1, 0, 0, 0], good = [0, 1, 0, 0], not bad = [1, 0, 1, 0], not good = [0, 1, 1, 0]

Ya que la oración "not bad" significa que algo es bueno o al menos no malo, mientras que la oración "not good" significa que algo es malo o al menos no bueno, la negación en la oración puede indicar una opinión opuesta a lo que se podría esperar a partir del significado de las palabras individuales en la oración.

2. Prediciendo calificaciones de películas. Supongamos que ahora estamos interesados en predecir una calificación numérica para reseñas de películas. Vamos a usar un predictor no-lineal que toma una reseña de película x y regresa $\sigma(w \cdot \phi(x))$, donde $\sigma(z) = (1 + e^{-z})^{-}1$.

   a. Si quisiéramos usar la pérdida cuadrática, ¿Cuál sería la expresión para $Loss(x, y, w)$ para un solo punto (x, y).

   $$Loss(x, y, w) = (\sigma(w \cdot \phi(x)) - y)^2$$

   b. Considerando $Loss(x, y, w)$ de la anterior parte, calcula el gradiente de la pérdida con respecto a w, $\nabla_w Loss(x, y, w)$. Escribe tu respuesta en términos del valor predecido $p = \sigma(w \cdot \phi(x))$.

   $$\nabla_w Loss(x, y, w) = 2(\sigma(w \cdot \phi(x)) - y) \cdot \frac{d\sigma(w \cdot \phi(x))}{dw}$$

   Hacemos un cambio de variable, $z = w \cdot \phi(x) \rightarrow \frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))$

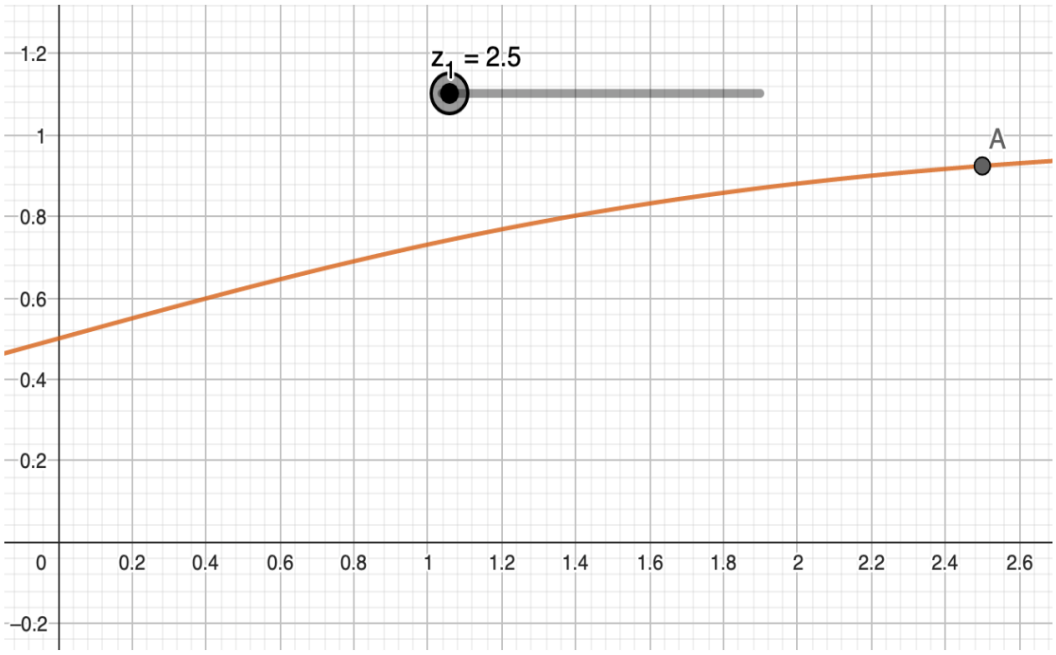
   $$\nabla_w Loss(x, y, w) = 2(\sigma(w \cdot \phi(x)) - y) \cdot \sigma(w \cdot \phi(x)) \cdot (1 - \sigma(w \cdot \phi(x))) \cdot \frac{d(w \cdot \phi(x))}{dw}$$

   $$\nabla_w Loss(x, y, w) = 2(p - y) \cdot p \cdot (1 - p) \cdot \phi(x)$$

   c. ¿Es posible que la magnitud del gradiente con respecto a w sea exactamente cero? Puedes hacer la magnituud de w arbitrariamente grande pero no infinita.

   Si observamos en nuestra función lógistica y sustituyendo por conveniencia $w = w \cdot \phi(x)$:

   $$\sigma(z) = \frac{1}{1+e^{-z}}$$



Al darle naturales grandes nos acercamos mas al 1, por lo que podemos verificar que $w \cdot \phi(x)$ sea positivo y grande, esto para que el gradiente de la pérdida sea pequeño.

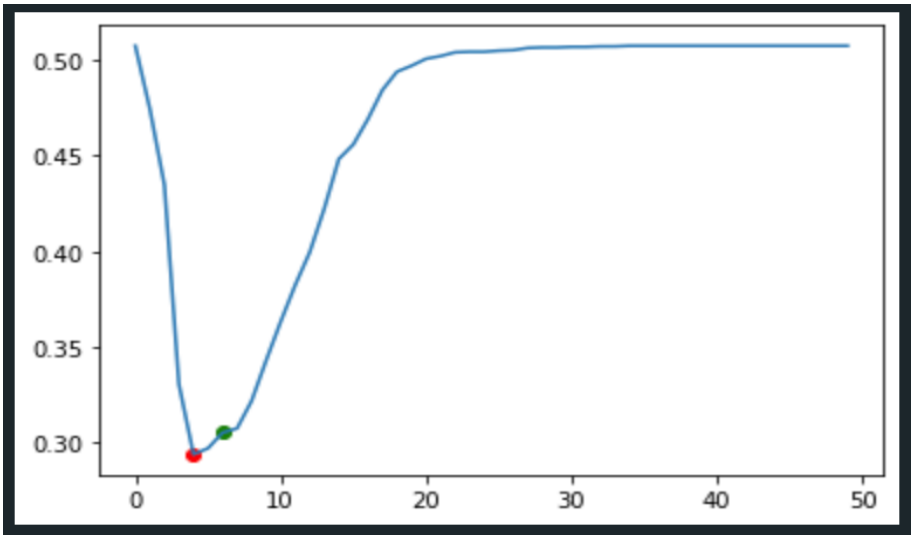
$$\lim_{x \to \infty} (\nabla_w Loss(x, y, w)) = 2(\sigma(w \cdot \phi(x)) - y) \cdot \sigma(w \cdot \phi(x)) \cdot (1 - \sigma(w \cdot \phi(x))) \cdot \phi(x)$$

Pero esto entra en conflicto con la inicial condición del problema. Asi que no es posible que la magnitud del gradiente con respecto a w sea exactamente cero, ya que para hacerlo, necesitamos que $w \cdot \phi(x))$ sea exactamente igual a y.

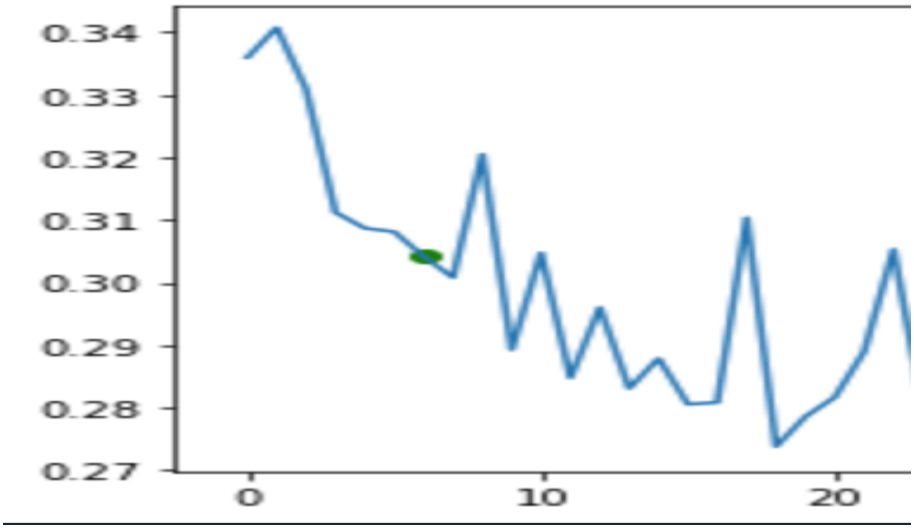
3. Clasificación de sentimientos. En este problema vamos a construir un clasificador lineal que lee reseñas de películas y adivina si son ”positivos” o ”negativos”.

   e. Corre tu predictor lineal con el extractor de características extractCharacterFeatures. Experimenta con distintos valores de n para ver cuál produce el menor error de validación. Debes observar que este error es casi tan pequeño como el que se produce con características de palabras. ¿Por qué este es el caso?

   Se pueden experimentar diferentes valores de n y elegir el que produzca el menor error. En mi experimento, encontré que el valor de n = 4. Con valores de ”n” más altos, se están considerando n-gramas más largos, lo que puede hacer que el modelo sea más específico a los datos de entrenamiento y, por lo tanto, menos generalizable a nuevos datos.



En mi gráfica n-gramas vs erro de validación, el punto mínimo fue $(4, 0.2940)$ el cual es el punto rojo y el punto verde es cuando el error es casi tan pequeño como el que se produce con características de palabras, la diferencia es que el extractWordsFeatures lo consiguió en menos epócas, como se muestra en mi gráfica epócas vs error:



La extracción de palabras es mejor en este caso ya que una de sus ventajas es que puede identificar palabras importantes en el texto y usarlas para hacer inferencias sobre el contenido del texto. Por otro lado los n-gramas nos sirven para la identificación de patrones en el otexto, capturar la estructura del lenguaje, etc.

Pruebas: pruebas

4. Clasificación de toxicidad y pérdida máxima de grupo.

Clasificador D: $w = [0.1, 1, 0]$, Clasificador T : $w = [0.1, 0, 1]$.

Comentario (x), Toxicidad (y), Presencia de menciones demográficas (d), Presencia de palabras tóxicas (t)

$\phi(x)$= [1, d, t]

Entonces tenemos cuatro grupos: $(y = 1, d = 1), (y = 1, d = 0), (y = 1, d = 1)$, y $(y = 1, d = 0)$

a. En palabras, describe el comportamiento del Claisificador D y el clasificador T.

$f_w = sign(w \cdot \phi(x)) = sign = ([-0.1, 1, 0] \cdot [1, d, t]) = sign([-0.1, d, 0])$
$+1$ si $d = 1$ o -1 si $d = 0$

El clasificador D asignará una salida de $+1$ a un comentario con $y = 1$ si y solo si el comentario menciona identidades demográficas independientemente si hay palabras toxicas o no. Y si no hay menciones demográficas lo clasifica como tóxico.

$Loss_{0,1}(x, y, w) = 1 \longleftrightarrow bool[f_w \neq 1]$

b. Calcula las siguientes tres cantidades sobre el Clasificador D usando el conjunto de datos de arriba:

$f_w = sign([-0.1, 1, 0] \cdot [1, 0, 0]) = -1 \rightarrow 1[-1 \neq -1] = 0$

$f_w = sign([-0.1, 1, 0] \cdot [1, 0, 1]) = -1 \rightarrow 1[-1 \neq -1] = 0$

$f_w = sign([-0.1, 1, 0] \cdot [1, 1, 0]) = +1 \rightarrow 1[+1 \neq -1] = 1$

$f_w = sign([-0.1, 1, 0] \cdot [1, 1, 1]) = +1 \rightarrow 1[+1 \neq -1] = 1$

$TrainLoss_{-1} = \frac{1}{4}(0 + 0 + 1 + 1) = 0.5$

$f_w = sign([-0.1, 1, 0] \cdot [1, 0, 0]) = -1 \rightarrow 1[-1 \neq 1] = 1$

$f_w = sign([-0.1, 1, 0] \cdot [1, 0, 1]) = -1 \rightarrow 1[-1 \neq 1] = 1$

$f_w = sign([-0.1, 1, 0] \cdot [1, 1, 0]) = +1 \rightarrow 1[+1 \neq 1] = 0$

$f_w = sign([-0.1, 1, 0] \cdot [1, 1, 1]) = +1 \rightarrow 1[+1 \neq 1] = 0$

$TrainLoss_{+1} = \frac{1}{4}(1 + 1 + 0 + 0) = 0.5$

c. Calcula las siguientes tres cantidades sobre el Clasificador T usando el conjunto de datos de arriba:

$f_w = sign([-0.1, 0, 1] \cdot [1, 0, 0]) = -1 \rightarrow 1[-1 \neq -1] = 0$

$f_w = sign([-0.1, 0, 1] \cdot [1, 0, 1]) = +1 \rightarrow 1[+1 \neq -1] = 1$

$f_w = sign([-0.1, 0, 1] \cdot [1, 1, 0]) = -1 \rightarrow 1[-1 \neq -1] = 0$

$f_w = sign([-0.1, 0, 1] \cdot [1, 1, 1]) = +1 \rightarrow 1[+1 \neq -1] = 1$

$TrainLoss_{-1} = \frac{1}{4}(0 + 1 + 0 + 1) = 0.5$

$f_w = sign([-0.1, 0, 1] \cdot [1, 0, 0]) = -1 \rightarrow 1[-1 \neq 1] = 1$

$f_w = sign([-0.1, 0, 1] \cdot [1, 0, 1]) = +1 \rightarrow 1[+1 \neq 1] = 0$

$f_w = sign([-0.1, 0, 1] \cdot [1, 1, 0]) = -1 \rightarrow 1[-1 \neq 1] = 1$

$f_w = sign([-0.1, 0, 1] \cdot [1, 1, 1]) = +1 \rightarrow 1[+1 \neq 1] = 0$

$TrainLoss_{+1} = \frac{1}{4}(1 + 0 + 1 + 0) = 0.5$

Según los calculos ambos clasificadores tienen menor pérdida promedio y máxima pérdida de grupo. Pero el clasificador T es más acertado.

i. Primero argumenta por el uso de de la pérdida promedio apelando a uno de los principios de arriba.

Al aplicar el clasificador D, algunos comentaristas no tóxicos son etiquetados incorrectamente como tóxicos, lo que puede resultar en consecuencias negativas para los usuarios. Un enfoque más justo y equitativo sería centrarse en las características del comentario en sí mismo, en lugar de las características del autor. Esto estaría en conectado con el segundo y tercer principio.

ii. ¿Cuál de los Clasificadores D o T implementarías en una plataforma web social real para identificar publicaciones etiquetadas como tóxicas para su revisión?

Tomaría el clasificador T ya que se centra si en verdad existe un comentario tóxico, identificandolo rápido, por lo que pienso que se conecta con el primer principio, porque beneficia a las personas que no clasifica como toxicas y penaliza a las que si lo son.

iii. Luego, haz lo mismo pero argumentando por el uso de la pérdida máxima de grupo como tu objetivo, igual apelando a uno de los tres princios.

Como el tercer principio lo comenta, el algoritmo debe ser justo. Por lo que la pérdida máxima de grupo limita la cantidad máxima de errores de clasificación permitidos en un grupo en particular. Esto asegura que ningún grupo sea perjudicado de manera desproporcionada por las etiquetas incorrectas.

e. ¿Qué métodos usarías para determinar la toxicidad de comentarios para usarlos como datos de entrenamiento con un clasificador de toxicidad? Explica por qué eligirás esos métodos sobre otros enlistados.

Yo haría una combinación entre contratar a científicos sociales y pedir a los usuarios que califiquen los comentarios. Mi justificación es aun que puede resultar costoso de dinero y tiempo las anotaciones son de calidad y consistentes, por otro lado a la comunidad no se le tiene que pagar por calificación de comentarios, pero puede ser que los usuarios no califiquen los comentarios de manera constante y que sus calificaciones sean confiables, pero almenos los estandares de toxicidad se alinearán a lo que los usurios piden.

5. Agrupación con K-medias.
$\phi(x_1) = [0, 0]$, $\phi(x_2) = [4, 0]$, $\phi(x_3) = [6, 0]$, $\phi(x_4) = [11, 0]$

a. Corre este algoritmo dos veces con los siguientes centros centros iniciales:

1..$\mu_1 = \phi(x_1) = [0, 0]$ y $\mu_2 = \phi(x_4) = [11, 0]$

Colocamos los datos en una recta para imaginarnos mejor el problema.

Ahora si conocemos esta información podemos calcular la distancia entre cada punto ϕ y el centroide de cada grupo, tomando el minimo para diferenciar entre grupos.

$z(\phi(x_1)) = min\{([0, 0] - [0, 0])^2, ([0, 0] - [11, 0])^2\} = min(0, 121)$ K: 1

$z(\phi(x_2)) = min\{([4, 0] - [0, 0])^2, ([4, 0] - [11, 0])^2\} = min(16, 49)$ K: 1

$z(\phi(x_3)) = min\{([6,0]-[0,0])^2, ([6,0]-[11,0])^2\} = min(36,25)$ K: 2

$z(\phi(x_4)) = min\{([11,0]-[0,0])^2, ([11,0]-[11,0])^2\} = min(121,0)$ K: 2

$\mu_1 = \frac{1}{2}([0,0]+[4,0]) = [2,0], mu_2 = \frac{1}{2}([6,0]+[11,0]) = [8.5,0]$

$z(\phi(x_1)) = min\{([0,0]-[2,0])^2, ([0,0]-[8.5,0])^2\} = min(0,121)$ K: 1

$z(\phi(x_2)) = min\{([4,0]-[2,0])^2, ([4,0]-[8.5,0])^2\} = min(16,49)$ K: 1

$z(\phi(x_3)) = min\{([6,0]-[2,0])^2, ([6,0]-[8.5,0])^2\} = min(36,25)$ K: 2

$z(\phi(x_4)) = min\{([11,0]-[2,0])^2, ([11,0]-[8.5,0])^2\} = min(121,0)$ K: 2

$\mu_1 = \frac{1}{2}([0,0]+[4,0]) = [2,0], mu_2 = \frac{1}{2}([6,0]+[11,0]) = [8.5,0]$

Dado que nuestros centroides no cambiaron, significa que nuestro modelo ha convergido. Por lo tanto:

$z(\phi(x_1))$ y $z(\phi(x_2)) \in$ K: 1

$z(\phi(x_3))$ y $z(\phi(x_4)) \in$ K: 2

2.$\mu_1 = \phi(x_1) = [0,0]$ y $\mu_2 = \phi(x_2) = [4,0]$

$z(\phi(x_1)) = min\{([0,0]-[0,0])^2, ([0,0]-[4,0])^2\} = min(0,16) = 0$ K: 1

$z(\phi(x_2)) = min\{([4,0]-[0,0])^2, ([4,0]-[4,0])^2\} = min(16,0) = 0$ K: 2

$z(\phi(x_3)) = min\{([6,0]-[0,0])^2, ([6,0]-[4,0])^2\} = min(36,4) = 4$ K: 2

$z(\phi(x_4)) = min\{([11,0]-[0,0])^2, ([11,0]-[4,0])^2\} = min(121,49) = 0$ K: 2

$\mu_1 = \frac{1}{1}([0,0]) = [0,0], mu_2 = \frac{1}{3}([4,0] + [6,0] + [11,0]) = [7,0]$

$z(\phi(x_1)) = min\{([0,0] - [0,0])^2, ([0,0] - [7,0])^2\} = min(0, 49) = 0$ K: 1

$z(\phi(x_2)) = min\{([4,0] - [0,0])^2, ([4,0] - [7,0])^2\} = min(16, 0) = 0$ K: 2

$z(\phi(x_3)) = min\{([6,0] - [0,0])^2, ([6,0] - [7,0])^2\} = min(36, 4) = 4$ K: 2

$z(\phi(x_4)) = min\{([11,0] - [0,0])^2, ([11,0] - [7,0])^2\} = min(121, 49) = 49$ K: 2

$\mu_1 = \frac{1}{1}([0,0]) = [0,0], mu_2 = \frac{1}{3}([4,0] + [6,0] + [11,0]) = [7,0]$

Dado que nuestros centroides no cambiaron, significa que nuestro modelo ha convergido. Por lo tanto:

$z(\phi(x_1)) \in$ K: 1

$z(\phi(x_2))$, $z(\phi(x_3))$ y $z(\phi(x_4)) \in$ K: 2

c. Si escalamos todas las dimensiones en nuestros centroides iniciales y los datos por un factor distinto a cero. ¿Garantizamos que recuperaremos los mismos agrupamientos después de correr k-medias (es decir, los mismos puntos de datos van a pertenecer al mismo grupo antes y después de escalar)?

No necesariamente garantizamos que recuperaremos los mismos agrupamientos después de escalar todas las dimensiones en nuestros centroides iniciales y los datos por un factor distinto de cero.

Ejemplo. Supongamos:
$\phi(x_1) = 1, \phi(x_2) = 2, \phi(x_3) = 3$

$\mu_1 = 1, \mu_2 = 2.5$

$z(\phi(x_1)) = min\{(1-1)^2, (1-2.5)^2\} = min(0, 0.25)$ K: 1

$z(\phi(x_2)) = min\{(2-1)^2, (2-2.5)^2\} = min(1, 0.25)$ K: 2

$z(\phi(x_3)) = min\{(3-1)^2, (3-2.5)^2\} = min(4, 0.25)$ K: 2

Si multiplicamos por un escalar que pasa:

$\phi(x_1) = 2, \phi(x_2) = 4, \phi(x_3) = 6$

$\mu_1 = 2, \mu_2 = 5$

$z(\phi(x_1)) = min\{(2-2)^2, (2-5)^2\} = min(0, 9)$ K: 1

$z(\phi(x_2)) = min\{(4-2)^2, (4-5)^2\} = min(4, 1)$ K: 2

$z(\phi(x_3)) = min\{(6-2)^2, (6-5)^2\} = min(16, 1)$ K: 2

En este caso, si se cumplió que recuperamos los mismos grupos. Como la distancia entre los puntos de datos se calcula en función de la distancia euclidiana . Pero si escalamos todas las $\phi(x_i)$ por diferentes factores, entonces las distancias euclidianas entre los puntos de datos pueden cambiar significativamente.

¿Qué pasa si escalamos solo ciertas dimensiones? Si tu respuesta es afirmativa, provee una explicaci ón corta, si no, presenta un contraejemplo.

Si escalamos solo ciertas dimensiones, los grupos no cambian porque la relación de proximidad sigue siendo la misma solo que aumentada por un escalar, aun que si este escalar es diferente en cada multiplicación entonces puede que nuestros grupos si cambien.

Ejemplo. Supongamos:
$\phi(x_1) = [1, 1], \phi(x_2) = [2, 2], \phi(x_3) = [3, 3]$

$\mu_1 = [1, 1], \mu_2 = [2.5, 2.5]$

$z(\phi(x_1)) = min\{([1, 1] - [1, 1])^2, ([1, 1] - [2.5, 2.5])^2\} = min(0, 4.5)$ K: 1

$z(\phi(x_2)) = min\{([2, 2] - [1, 1])^2, ([2, 2] - [2.5, 2.5])^2\} = min(2, 0.5)$ K: 2

$z(\phi(x_3)) = min\{([3, 3] - [1, 1])^2, ([3, 3] - [2.5, 2.5])^2\} = min(8, 0.5)$ K: 2

Si multiplicamos por un escalar una dimensión que pasa:

$\phi(x_1) = (1, 2), \phi(x_2) = (2, 4), \phi(x_3) = (3, 6)$

$\mu_1 = (1, 2), \mu_2 = (2.5, 5)$

$z(\phi(x_1)) = min\{([1, 2] - [1, 2])^2, ([1, 2] - [2.5, 5])^2\} = min(0, 11.25)$ K: 1

$z(\phi(x_2)) = min\{([2, 4] - [1, 2])^2, ([2, 4] - [2.5, 5])^2\} = min(5, 1.25)$ K: 2

$z(\phi(x_3)) = min\{([3, 6) - [1, 2])^2, ([3, 6] - [2.5, 5])^2\} = min(20, 1.25)$ K: 2

Ya que este algoritmo de agrupamiento es equivalente a una transformación lineal de escalamiento, esta nos dice que si se escalan los vectores y no cambian su dirección significa que la relación de proximidad entre los puntos y los centroides se mantiene igual.