Inteligencia Artificial - Sentimientos

Rodolfo Armando Jaramillo Ruiz
25 de Febrero de 2023

1. Construyendo la intuición

Se tiene el siguiente conjunto de mini-reseñas:

1. (−1) not good
2. (−1) pretty bad
3. (+1) good plot
4. (+1) pretty scenery

A cada reseña se le asocia un vector de características $\phi(x)$ donde se asocia a cada palabra la cantidad de veces que aparece en la reseña. Por ejemplo:

$$\phi(x) = \{\text{not} : 1, \text{good} : 1\}$$

Se usa la definición de pérdida de articulación

$$\text{Loss}_{\text{hinge}} = \max\{0, 1 - \boldsymbol{w}\phi(x)y\}$$

Todo donde x es el texto de la reseña, y es la etiqueta correcta y $\boldsymbol{w}$ es el vector de pesos.
a. Supongamos que corremos el descenso de gradiente estocastico una vez por cada una de las cuatro muestras en el orden dado, actualizando los pesos de acuerdo a:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_w \text{Loss}_{\text{hinge}}$$

Despues de las actualizaciones, ¿cuales son los pesos de las seis palabras (”pretty”, ”good”, ”bad”, ”plot”, ”not”, ”scenery”)que aparecen en las reseñas de arriba?

- $\eta = 0.1$
- $\mathbf{w}$ inicial: $\mathbf{w} = [0, 0, 0, 0, 0, 0]$
- $\nabla_w \text{Loss}_{\text{hinge}}$ es cero cuando el margen es exactamente 1

Se empieza calculando el $\nabla_w \text{Loss}_{\text{hinge}}$ que sería la siguiente expresión:

$$\nabla_w \text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \begin{cases} -\phi(x)y & si \quad 1 - \mathbf{w} \cdot \phi > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Para la primera reseña tenemos lo siguiente:

$$\phi(x) = \{\text{not} : 1, \text{good} : 1\} = [0, 1, 0, 0, 1, 0]$$

Por lo que

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

Entonces $\mathbf{w}$ queda

$$\mathbf{w} = [0, 0, 0, 0, 0, 0] - 0.1[0, 1, 0, 0, 1, 0]$$

$$\mathbf{w} = [0, -0.1, 0, 0, -0.1, 0]$$

Para la segunda reseña tenemos lo siguiente:

$$\phi(x) = \{\text{pretty} : 1, \text{bad} : 1\} = [1, 0, 1, 0, 0, 0]$$

Por lo que

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

Entonces **w** queda

$$\mathbf{w} = [0, -0.1, 0, 0, -0.1, 0] - 0.1[1, 0, 1, 0, 0, 0]$$
$$\mathbf{w} = [-0.1, -0.1, -0.1, 0, -0.1, 0]$$

Para la tercera reseña tenemos lo siguiente:

$$\phi(x) = \{\text{good} : 1, \text{plot} : 1\} = [0, 1, 0, 1, 0, 0]$$

Por lo que

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

Entonces **w** queda

$$\mathbf{w} = [0, -0.1, 0, 0, -0.1, 0] - 0.1[0, -1, 0, -1, 0, 0]$$
$$\mathbf{w} = [-0.1, 0, -0.1, 0.1, -0.1, 0]$$

Para la cuarta reseña tenemos lo siguiente:

$$\phi(x) = \{\text{pretty} : 1, \text{scenary} : 1\} = [1, 0, 0, 0, 0, 1]$$

Por lo que

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

Entonces **w** queda

$$\mathbf{w} = [-0.1, 0, -0.1, 0.1, -0.1, 0] - 0.1[-1, 0, 0, 0, 0, -1]$$

Tenemos finalmente:

$$\mathbf{w} = [0, 0, -0.1, 0.1, -0.1, -0.1]$$

b. Dado el siguiente conjunto de reseñas

1. (-1) bad
2. (1) good
3. $(+1)$ not bad
4. (-1) not good

Muestra que no hay clasificador lineal que utilice características de palabras que tenga cero error sobre este conjunto de datos. Recuerda que esta es una pregunta sobre clasificadores, no sobre algoritmos de optimización; tu demostración debe ser correcta para cualquier clasificador lineal, sin importar en cómo se aprenden los pesos.
El vector de pesos para este conjunto de reseñas es:

$$\mathbf{w} = [w_{bad}, w_{good}, w_{not}]$$

Los vectores de caracteristicas correspondientes son

$$\phi(x) = \{\text{bad} : 1\} \quad \phi(x) = \{\text{not} : 1, \text{bad} : 1\}$$
$$\phi(x) = \{\text{good} : 1\} \quad \phi(x) = \{\text{not} : 1, \text{bad} : 1\}$$

De aquí se entiende el siguiente sistema de ecuaciones:

$$w_{bad} = -1$$
$$w_{good} = 1$$
$$w_{bad} + w_{not} = 1$$
$$w_{good} + w_{not} = -1$$

Que no tiene solución, es inconsistente, por lo tanto, no existe un clasificador lineal que tenga cero error. Se puede añadir la siguiente característica

$$cos(\mathbf{w} \cdot \phi(x))$$

Para que el sistema tenga solución, y en consecuencia haya un clasificador lineal que tenga cero error.

2. Prediciendo calificadores de películas

Supongamos que estamos interesados en predecir una calificación numérica para reseñas de películas. Vamos a usar un predictor no-lineal que toma una reseña de películas x y regresa $\sigma(\mathbf{w} \cdot \phi(x))$, donde $\sigma(z) = (1 + e^z)^{-1}$ es la función logística que aplasta un numero real al rango (0, 1). Para este problema, supón que la calificación de películas y es una variable con valor real en el rango [0, 1].
a. La pérdida cuadratica tiene la siguiente forma:

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

b. El gradiente de la perdida cuadratica se calcula como sigue:

$$\text{Loss}(x, y, \mathbf{w}) = ((1 + e^{-\mathbf{w}\cdot\phi(x)})^{-1} - y)^2$$

$$\nabla_w \text{Loss}(x, y, \mathbf{w}) = 2((1 + e^{-\mathbf{w}\cdot\phi(x)})^{-1} - y) \cdot (-1)(1 + e^{-\mathbf{w}\cdot\phi(x)})^{-2} \cdot (e^{-\mathbf{w}\cdot\phi(x)})\phi(x)$$

$$\nabla_w \text{Loss}(x, y, \mathbf{w}) = 2((1 + e^{-\mathbf{w}\cdot\phi(x)})^{-1} - y) \cdot (1 + e^{-\mathbf{w}\cdot\phi(x)})^{-2} \cdot (e^{-\mathbf{w}\cdot\phi(x)})\phi(x)$$

$$\nabla_w \text{Loss}(x, y, \mathbf{w}) = 2(\sigma(\mathbf{w} \cdot \phi(x)) - y) \cdot (\sigma(\mathbf{w} \cdot \phi(x)))^2 \cdot (\sigma(\mathbf{w} \cdot \phi(x))^{-1} - 1)\phi(x)$$

Considerando $p = \sigma(\mathbf{w} \cdot \phi(x))$:

$$\nabla_w \text{Loss}(x, y, \mathbf{w}) = 2(p - y) \cdot (p)^2 \cdot (p^{-1} - 1)\phi(x)$$

c. Si consideramos $\mathbf{w}$ pequeño el primer termino con $\sigma(\mathbf{w}\cdot\phi(x))$ se acerca solamente a $1/2$, en consecuencia el segundo termino se acerta a $1/4$, y el ultimo termino se acerca a la unidad, por lo tanto hacer $\mathbf{w}$ cada vez más pequeño converge a un valor diferente de cero. Ahora, con $\mathbf{w}$ muy grande, el $\sigma(\mathbf{w}\cdot\phi(x))$ contenido en el primer termino se acerca a la unidad, pero como $\mathbf{w}$ es finito nunca llega a cero, solo se hace cada vez más pequeño, al igual que en el tercer termino. En sintesis, con $\mathbf{w}$ grande el gradiente se acerca arbitrariamente a cero, pero nunca puede ser completamente cero.

3. Clasificación de sentimientos

Use la función *testValuesOfN()* en un bucle que recorría un rango de valores enteros en $[1, 10]$ para ver como cambiaba el *validation error*, viendo que entre los valores de $n = 4$ y $n = 8$ esta cantidad era parecida a cuando se usaba el extractor con características de palabras, siendo minima con $n = 7$. Asumo que esto la mayoría de la palabras de uso común tienen esta misma cantidad de caracteres. Se puede pensar en que capturar caracteres en vez de palabras es mejor si las reseñas con las que se va a entrenar y se va a validar son reseñas de palabras cortas, de menos de 4 palabras aproximadamente. Por ejemplo: "muy bien todo en la peli".

4. Clasificación de toxicidad y pérdida máxima de grupo

a. Se tiene al clasificador $D : \mathbf{w} = [-0.1, 1, 0]$ y al clasificador $T : \mathbf{w} = [-0.1, 0, 1]$, con $\phi(x) = [1, d, t]$ y $f_{\mathbf{w}}(x) = sign(\mathbf{w} \cdot \phi(x)$. Se procede a analizar el clasificador D:

$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, d, t]) = sign(-0.1 + d)$$

Tomando la definición de sign(z) entonces vemos que $f_{\mathbf{w}}(x) = +1$ cuando $d = 1$, y $f_{\mathbf{w}}(x) = -1$ cuando $d = -1$. Vemos que se clasifica a un comentario donde se mencionan identidades demograficas como un comentario tóxico, aunque no haya palabras toxicas.
Se procede a analizar el clasificador T:

$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, d, t]) = sign(-0.1 + t)$$

Vemos que $f_{\mathbf{w}}(x) = +1$ cuando $t = 1$, y $f_{\mathbf{w}}(x) = -1$ cuando $t = -1$. Por lo que este clasificador encontrara como comentarios toxicos a los que tengan palabras toxicas, pero no identidades demograficas. **b.** Para el clasificador D Tenemos que la perdida cero-uno es de la siguiente forma:

$$\text{Loss}_{0-1} = 1[f_{\mathbf{w}}(x) \neq y] \tag{1}$$

Con y=-1

$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 0, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 0, 1]) = -1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 1, 0]) = +1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 1, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 1$$
$$\text{TrainLoss}_{-1} = \frac{1}{4}(0 + 0 + 1 + 1) = 0.5$$

Con y=1

$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 0, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 0, 1]) = -1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 1, 0]) = +1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 1, 0] \cdot [1, 1, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 0$$
$$\text{TrainLoss}_{+1} = \frac{1}{4}(1 + 1 + 0 + 0) = 0.5$$

c. Para el clasificador T Tenemos que la perdida cero-uno es de la siguiente forma: Con y=-1

$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 0, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 0, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 1, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 1, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 1$$
$$\text{TrainLoss}_{-1} = \frac{1}{4}(0 + 1 + 0 + 1) = 0.5$$

Con y=1

$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 0, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 0, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 0$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 1, 0]) = -1 \rightarrow \text{Loss}_{0-1} = 1$$
$$f_{\mathbf{w}}(x) = sign([-0.1, 0, 1] \cdot [1, 1, 1]) = +1 \rightarrow \text{Loss}_{0-1} = 0$$
$$\text{TrainLoss}_{+1} = \frac{1}{4}(1 + 0 + 1 + 0) = 0.5$$

d. Como se vió más arriba, el clasificador D etiqueta incorrectamente a comentarios como toxico al contener menciones de identidades demográficas, lo cual contribuye a la invisibilización de estos grupos. El clasificador T también tiene errores, pero sus errores no contienen la consecuencia anterior mencionada, y eligiendo el mal menor, este sería el mejor clasificador a mi parecer. Para que el algoritmo sea justo se debe limitar a la cantidad de errores de clasificación de cada grupo.

e. En principio, creo yo, tendría que haber una discusión sobre los criterios que seguirá la red social para determinar que actitudes o comportamientos son tóxicos. Todo esto acompañado de personas con que sepan sobre el tema, tener una perspectiva amplia. Una vez tenidos los criterios, a mano se clasificarán datos de entrenamiento siguiendo estos criterios. A su vez que estos criterios se comparten con la comunidad, y posteriormente se entrena a la IA con ayuda de los usuarios.