

Métodos Numéricos

**Sistema de Recomendación
Proyecto Parcial**

Marcos Jared Castro Pérez

Resumen

En este trabajo presentamos un método de recomendación. Dada una base de datos de usuarios que han calificado ciertos objetos (pueden ser películas, cosas compradas, etc.), damos un método para saber que artículos que no ha calificado un usuario se le podrían recomendar. Lo hacemos mediante una función de costo que presentamos, y al optimizarla nos da los datos necesarios para decidir las recomendaciones.

Introducción

Cada vez que vamos entramos a una página de compras por internet como Amazon o sitios como Netflix, siempre hay objetos (artículos para comprar, películas, series) que nos recomiendan estos sitios. Estas recomendaciones no son aleatorias y son únicas para cada usuario. Éstas recomendaciones son cosas que creen que nos gustaran. Cosas que compraríamos, películas o series que calificaríamos con 4-5 estrellas, etc. Para muchas empresas, gran parte de sus ventas vienen por este tipo de sistemas de recomendación.

Dada la situación anterior, exponemos la función de costo

$$J(w, b, x) = \frac{1}{2} \sum_{(i,j):r(i,j)} (w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{1}{2} \left(\sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^j)^2 + \sum_{j=1}^{n_i} \sum_{k=1}^n (x_k^j)^2 \right)$$

y explicamos cómo al optimizarla, podemos dar predicciones de qué podríamos recomendar a los usuarios.

Método

Tomemos una tabla como la siguiente, con usuarios, películas y calificadas por los usuarios. Las x 's son porque el usuario aún no calificado dicha película. Además, para este primer ejemplo supondremos que sabemos ciertas características sobre las películas, tales como que tanto son acción o romance.

	John Wick	Tu nombre	Romance Para siempre	Karate Kid	La La land
Adrián (1)	0	5	5	x	x
Bere (2)	5	0	x	4	0
Carolina (3)	x	1	0	5	1
Diego (4)	0	x	4	1	5
Romance (x_1)	0	0.9	1	0.3	.95
Acción (x_2)	1	0	0	0.8	0

Los datos de cuanto es una película de acción o romance vienen dados. La predicción para el usuario (j) de la película (j) estará dado por regresión lineal: $w^{(j)} \cdot x^{(i)} + b^{(j)}$, donde \cdot es el producto interno.

Por ejemplo, si tomamos $w^{(1)} = (5, 0)$ y $b^{(1)} = 0$, estos valores son apropiados viendo las películas que Adrián ha calificado y $w^{(1)} \cdot x^{(5)} + b^{(1)} = 4,75$. Así que bajo estos valores podríamos recomendarle

la película de La La Land a Adrián.

En orden para definir la función de costo, primero definiremos algo de notación.

- $r(i, j) = 1$ si el usuario j ha calificado la película (i) . 0 en otro caso
- $y^{(i,j)}$ es la calificación dada por el usuario j a la película i . Definida sólo si $r(i, j) = 1$.
- $w^{(j)}, b^{(j)}$, los parámetros para el usuario j .
- n es la cantidad de características que para las películas. En el caso del ejemplo con la tabla anterior $n = 2$
- $x^{(i)}$ las características de la película i .

El objetivo es aprender los parámetros $w^{(j)}$ y $b^{(j)}$.

Así, queremos minimizar la función

$$J(w^{(i)}, b^{(i)}) = \frac{1}{2} \sum_{i:r(i,j)=1} (w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{1}{2} \sum_{k=1}^n (w_k^{(i)})^2$$

Ésta suma es de los errores, sobre todas las películas que el usuario i ha calificado, y lo elevamos al cuadrado para que se mantenga positivo. El último término es para prevenir el sobre-ajuste. Al minimizar esta función obtendremos buenas opciones para los valores $w^{(i)}$ y $b^{(i)}$ con los cuales ya podríamos empezar a recomendar películas.

La función anterior sólo nos sirve para aprender los parámetros individuales de cada usuario i . Lo ideal sería aprender todos los parámetros $w^{(i)}$ y $b^{(i)}$. Si n_u es la cantidad de usuarios, nuestra nueva función de costo nos resulta cómo sigue

$$J_0(w, b) = \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} (w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{1}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^{(j)})^2$$

Y así, al minimizar esta función, obtendremos los parámetros para todos los usuarios y con ello hacer las recomendaciones.

Ahora, que pasa si de alguna manera lo que ya tenemos son los parámetros $w^{(i)}$ y $b^{(i)}$. En este caso, también podríamos aprender las características de las películas de la misma forma, es decir, optimizando la función de costo siguiente donde n_m es la cantidad de películas que hay

$$J_1(x) = \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} (w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{1}{2} \sum_{j=1}^{n_m} \sum_{k=1}^n (x_k^{(j)})^2$$

Cómo se dijo, este método sólo funciona si ya conocemos los los parámetros. Aún así, podemos juntar ambas funciones y al optimizar la nueva función resultante, obtendremos buenos valores para los parámetros y para las características. La función resultante es

$$J(w, b, x) = \frac{1}{2} \sum_{(i,j):r(i,j)} (w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{1}{2} \left(\sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^j)^2 + \sum_{j=1}^{n_i} \sum_{k=1}^n (x_k^j)^2 \right)$$

La función anterior la optimizamos con el Gradiente descendente. No planeo entrar mucho en los detalles de éste método, pero la forma de hacerlo es tomando una sucesión definida cómo sigue:

$$x_{n+1} = x_n - \gamma \nabla F(x_n),$$

el γ que utilizamos es $\gamma = 0,001$

Mas allá de las recomendaciones, aprender las características $x^{(i)}$ de los objetos, sirve para encontrar similitudes entre ellos. Es cómo al entrar a un Netflix, por poner un ejemplo, este además de recomendar películas, también clasifica por categorías. Esto también es útil, pues a veces los usuarios buscan por categorías y es mejor tenerlas.

Lo que hicimos fue implementar la función de costo así cómo el método de gradiente descendente. Generamos un tabla de números aleatorios entre 1 y 5 y -1 . Cada fila representa las calificaciones que de un usuario y cada columna una película o cualquier objeto que se esté calificando. Los -1 representan películas no calificadas por los usuarios.

Al final el programa escribe en documentos las recomendaciones para los usuarios.

Hice dos ejemplos, uno donde las recomendaciones se dan si la predicción es mayor o igual que 3.5 y otro donde la predicción tiene que ser mayor o igual que 3. Subimos estos documentos, son Recomendaciones1.txt y ecomendaciones2.txt respectivamente.

Referencias

- [1] Burden, R., & Faires, J. D. (2001). *Analisis Numerico* (7a ed.). Cengage Learning Editores S.A. de C.V.
- [2] C, M. A., & Guido, S. (2016). *Introduction to machine learning with python: A guide for data scientists*. O'Reilly Media.
- [3] Coursera. Recuperado el 15 de noviembre de 2022, de <https://www.coursera.org/learn/unsupervised-learning-recommenders-reinforcement-learning/home/info>