

CLUSTERING DE CLIENTES O PRODUCTOS
SISTEMAS DE RECOMENDACIÓN.
FILTRO COLABORATIVO

Sistemas de recomendación

- Dado un cliente, queremos recomendarle productos (similares a los que ha consumido) que han sido consumidos por otros clientes similares a él.
- Una de las aplicaciones más importante del aprendizaje automático.
 - Amazon: recomendar productos,
 - Youtube: videos,
 - Facebook: noticias recomendadas,
 - Google play: aplicaciones descargadas,
 - Netflix: series, películas
 - Eroski: ofertas, descuentos, etc..
- Porcentaje de las ganancias depende de las compras/vistas que hacen los usuarios de los productos recomendados.

Sistemas de recomendación

- ❑ Caso real: muchos productos y muchos clientes.
- ❑ Cada cliente está representado por los productos que ha comprado (valorado, etc..)
- ❑ Vectores “vacíos”: la mayoría de los valores están vacíos, por lo tanto no podemos calcular las distancias.
- ❑ Tampoco podemos construir una matriz de similitud entre clientes (no los conocemos).
- ❑ Tampoco podemos construir una matriz de similitud entre productos (son muchos).

Ejemplo: recomendar películas en base a las valoraciones

Película	Ander	Sergio	Adriana	Uxue
Transformers: La era de la extinción	5	5	0	1
Capitán América: El soldado de invierno	5	?	?	0
X-Men: Días del futuro pasado	?	4	2	?
El amor está en el aire	0	0	4	5
El lado bueno de las cosas	1	0	5	?

$n_u = \text{número usuarios}$

$n_m = \text{número de películas}$

$r(i, j) = 1$ si el usuario j ha valorado la película i .

$y(i, j) = \text{valoración del usuario } j \text{ de la película } i$.

Sistemas de recomendación. Recomendaciones basadas en el contenido

Película	Ander	Sergio	Adriana	Uxue	x_1 (Acción)	x_2 (Romance)
Transformers: La era de la extinción	5	5	0	1	0,9	0,05
Capitán América: El soldado de invierno	5	?	?	0	1	0
X-Men: Días del futuro pasado	?	4	2	?	0,8	0,1
El amor está en el aire	0	0	4	5	0	1
El lado bueno de las cosas	1	0	5	?	0,1	0,95

- Para cada usuario j podemos aprender los parámetros $\theta^{(j)}$ de un modelo de regresión lineal.
- Predecir las valoraciones del usuario j para la película i :
- $(\theta^{(j)})^T x^{(i)}$

Recomendaciones basadas en el contenido

□ Formulación del problema:

$r(i, j) = 1$ si el usuario j ha valorado la película i , 0 en otro caso.

$y(i, j)$ = valoración del usuario j de la película i .

$\theta^{(j)}$ = vector del parámetros del usuario j .

$x^{(i)}$ = vector de características de la película i .

$m^{(j)}$ = no. de películas que ha valorado el usuario j .

Para encontrar $\theta^{(j)}$:

$$\min_{\theta^{(j)}} \quad \frac{1}{2\cancel{m^{(j)}}} \left[\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \lambda \sum_{k=1}^n (\theta_k^{(j)})^2 \right]$$

Podemos eliminar $m^{(j)}$, ya que es una constante y es diferente para cada usuario.

Recomendaciones basadas en el contenido

- Función a minimizar. Para un único usuario j , $\theta^{(j)}$:

$$\min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^n (\theta_k^{(j)})^2$$

- Para todos los usuarios a la vez

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

Recomendaciones basadas en el contenido

□ Algoritmo de optimización

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

□ Descenso por gradiente:

$$\theta_0^{(j)} := \theta_0^{(j)} - \alpha \left[\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) \cdot x_0^{(i)} \right]$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left[\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) \cdot x_k^{(i)} + \lambda \theta_k^{(j)} \right] \quad \text{Para } k = 1, \dots, n$$

Recomendaciones basadas en el contenido

- ❑ Necesitamos que un experto valore las características que hemos creado para cada película.
- ❑ Obtenemos un vector de parámetros que define a cada usuario.
- ❑ Podemos realizar un clustering utilizando para usuario el vector de características que hemos obtenido.
- ❑ Obtendremos grupos basados en las valoraciones que han realizado sobre las películas.

Aprendiendo las características de las películas

Película	Ander	Sergio	Adriana	Uxue		x_1 (Acción)	x_2 (Romance)
Transformers: La era de la extinción	5	5	0	1		?	?
Capitán América: El soldado de invierno	5	?	?	0		?	?
X-Men: Días del futuro pasado	?	4	2	?		?	?
El amor está en el aire	0	0	4	5		?	?
El lado bueno de las cosas	1	0	5	?		?	?

- No conocemos las características de las películas pero si conocemos (porque se lo hemos preguntado) el perfil de los usuarios.

Aprendiendo las características de las películas

Película	Ander $\theta^{(1)}$	Sergio $\theta^{(2)}$	Adriana $\theta^{(3)}$	Uxue $\theta^{(4)}$		x_1 (Acción)	x_2 (Romance)
Transformers: La era de la extinción	5	5	0	1		≈ 1	≈ 0
Capitán América: El soldado de invierno	5	?	?	0		?	?
X-Men: Días del futuro pasado	?	4	2	?		?	?
El amor está en el aire	0	0	4	5		?	?
El lado bueno de las cosas	1	0	5	?		?	?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \begin{array}{l} \leftarrow \theta_0 \\ \leftarrow \text{Gusto por las películas de acción} \\ \leftarrow \text{Gusto por las películas de romance} \end{array}$$

$$\theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \quad \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

$$\left\{ \begin{array}{l} (\theta^{(1)})^T x^{(1)} \approx 5 \\ (\theta^{(2)})^T x^{(1)} \approx 5 \\ (\theta^{(3)})^T x^{(1)} \approx 0 \\ (\theta^{(4)})^T x^{(1)} \approx 1 \end{array} \right.$$

Aprendiendo las características de las películas

- Algoritmo de optimización (suponiendo que tenemos los perfiles de los usuarios $\theta^{(j)}$ para aprender las características de la película $x^{(i)}$:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^n (x_k^{(i)})^2$$

- Para todas las películas:

$$\min_{x^{(1)}, \dots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$

Aprendiendo las características de las películas

- Necesitamos que TODOS los clientes completen un perfil con sus gustos.
- Obtenemos un vector de parámetros que define a cada película.
- Podemos realizar un clustering utilizando para película el vector de características que hemos obtenido.
- Obtendremos grupos de películas basados en las valoraciones que han realizado los usuarios sobre las películas.

Algoritmo filtro colaborativo básico

- Dadas unas características de las películas $x^{(1)}, \dots, x^{(n_m)}$, y unas valoraciones, podemos estimar el perfil del usuario $\theta^{(1)}, \dots, \theta^{(n_u)}$.
- Dados los perfiles de usuarios $\theta^{(1)}, \dots, \theta^{(n_u)}$, y unas valoraciones, podemos estimar las características de las películas $x^{(1)}, \dots, x^{(n_m)}$.
- ¿el huevo o la gallina?
- Comenzar con θ aleatorios $\Rightarrow x \Rightarrow \theta \Rightarrow x \Rightarrow \theta \Rightarrow x \Rightarrow$
- ¿Podemos hacerlo mejor?

Algoritmo filtro colaborativo

- Dadas unas características de las películas $x^{(1)}, \dots, x^{(n_m)}$, y unas valoraciones, podemos estimar el perfil del usuario $\theta^{(1)}, \dots, \theta^{(n_u)}$.

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

- Dados los perfiles de usuarios $\theta^{(1)}, \dots, \theta^{(n_u)}$, y unas valoraciones, podemos estimar las características de las películas $x^{(1)}, \dots, x^{(n_m)}$.

$$\min_{x^{(1)}, \dots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$

Algoritmo filtro colaborativo

- Juntamos todo en una misma función objetivo:

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) = \frac{1}{2} \sum_{i,j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

- Vamos a minimizar la función objetivo y los parámetros serán:
 $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$
- No incluimos el término $x_0^{(i)}$, ni el término $\theta_0^{(j)}$

Algoritmo filtro colaborativo

1. Inicializar con números aleatorios pequeños,
 $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$
2. Minimizar $J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$ utilizando el descenso por gradiente. Para cada $i = 1, \dots, n_m$ y $j = 1, \dots, n_u$:

$$x_k^{(i)} := x_k^{(i)} - \alpha \left[\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) \cdot \theta_k^{(j)} + \lambda x_k^{(i)} \right]$$
$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left[\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) \cdot x_k^{(i)} + \lambda \theta_k^{(j)} \right]$$

Filtro colaborativo. Normalización

Película	Ander $\theta^{(1)}$	Sergio $\theta^{(2)}$	Adriana $\theta^{(3)}$	Uxue $\theta^{(4)}$	Edson
Transformers: La era de la extinción	5	5	0	1	?
Capitán América: El soldado de invierno	5	?	?	0	?
X-Men: Días del futuro pasado	?	4	2	?	?
El amor está en el aire	0	0	4	5	?
El lado bueno de las cosas	1	0	5	?	?

$$Y = \begin{bmatrix} 5 & \dots & ? \\ \vdots & \ddots & \vdots \\ 1 & \dots & ? \end{bmatrix}$$

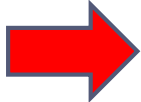
Añadimos un nuevo usuario que no ha visto ninguna película.

En la función de coste solo se tendrá en cuenta el termino de regularización. Por lo tanto $\theta^{(nuevo)} = 0$.

El sistema predecirá todas las valoraciones del nuevo = 0.
 $(\theta^{(nuevo)})^T x^{(i)} = 0$

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) = \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i,j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

Filtro colaborativo. Normalización por la media

Y=	5	5	0	1	?
	5	?	?	0	?
	?	4	2	?	?
	0	0	4	5	?
	1	0	5	?	?
$\mu=$					
2,75					
2,5					
3					
2,25					
2					
					
Y=	2,25	2,25	0	-1,75	?
	2,5	?	?	-2,5	?
	?	1	-1	?	?
	-2,25	-2,25	1,75	2,75	?
	-1	-2	3	?	?

- Utilizar la nueva matriz Y para aprender $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$
- Al predecir la valoración del usuario j de la película i: $(\theta^{(j)})^T x^{(i)} + \mu_i$
- Por tanto al usuario que no ha visto ninguna película le recomendaré según la valoración media de todos los usuarios.

Clustering de clientes o productos

- Una vez que hemos entrenado el filtro colaborativo cada película estará representada por un vector $x^{(i)}$.
- Cada usuario será un vector $\theta^{(j)}$
- Podemos aplicar el algoritmo k-means o el EM para encontrar grupos de clientes parecidos en sus gustos o grupos de películas.