Sistemas de Recomendación

- Netflix (Peliculas)
- Amazon (Productos)
- Youtube (Contenido)
- Ebay, Ali express, Banggood, etc. (Productos)
-

Ej: Recomendar Peliculas

Usuario califica una pelicula con un rating de 0 a 5 estrellas.

Pelicula	Alicia (1)	Juan (2)	Pedro (3)	Carolina (4)
Interestelar	5	5	0	1
Blade runner	5	? = 4	?	1
Yo Robot	?	4	1	?
La Monja	0	0	4	4
Chuckie	0	0	5	?

n_u = # usuarios
n_m = # peliculas
$r(i,j)$ = 1 si el usuario calificó
$y(i,j)$ = rating del usuario j para la pelicula i (solo si $r(i,j) = 1$)

Enfoques (Tradicional)

1: Sistemas Recomendadores basados en Contenido
Content-based R.S. (Content-based filtering)

2: S.R. basados en filtros colaborativos
Collaborative Filtering

3*: Sistema híbrido

Content-based Recommender Systems:

Pelicula		Alicia (1) $\Theta^{(1)}$	Juan (2) $\Theta^{(2)}$	Pedro (3) $\Theta^{(3)}$	Carolina (4) $\Theta^{(4)}$	x_1 Ciencia ficción	x_2 Terror
Interestelar	$x^{(1)}$	5	5	0	1	1,0	0
Blade runner	$x^{(2)}$	5	? 4	?	1	0,99	0,02
Yo Robot	$x^{(3)}$	? 4.95	4	1	?	0,99	0
La Monja	$x^{(4)}$	0	0	4	4	0	1
Chuckie	$x^{(5)}$	0	0	5	?	0,1	0,98

x_0
$x_1 \rightarrow 5$
x_2

$n_u = 4$
$n_m = 5$

$x^{(1)} = 1 \cdot x_0 + 1 \cdot x_1 + 0 \cdot x_2 = x^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

Para cada usuario j, aprender $\Theta^{(j)} \in \mathbb{R}^3$
Para predecir el rating de i para j: $(\Theta^{(j)})^T x^{(i)}$

$$x^{(3)} = \begin{bmatrix} 1 \\ 0,99 \\ 0 \end{bmatrix} \quad \Theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad (\Theta^{(1)})^T x^{(3)} = [0 \; 5 \; 0] \begin{bmatrix} 1 \\ 0,99 \\ 0 \end{bmatrix} = 4.95$$

Objetivo de optimización

Para un usuario j, aprender $\Theta^{(j)}$:

Penaliza Θ_K grandes

$$\min_{\Theta^{(j)}} \underbrace{\frac{1}{2} \sum_{i:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)})^2}_{\text{regresión lineal}} + \underbrace{\frac{\lambda}{2} \sum_{K=1}^{n} (\Theta_K^{(j)})^2}_{\text{regularización}}$$

Para todos los usuarios: $\Theta^{(1)}, \Theta^{(2)}, \ldots \Theta^{(n_u)}$

$$\min_{\Theta^{(1)}, \Theta^{(2)} \ldots \Theta^{(n_u)}} \underbrace{\frac{1}{2} \underbrace{\sum_{j=1}^{n_u}}_{\Theta^{(j)}} \sum_{i:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{K=1}^{n} (\Theta_K^{(j)})^2}_{J(\Theta^{(1)}, \Theta^{(2)}, \ldots \Theta^{(n_u)})}$$

Descenso de gradiente:

$$\Theta_K^{(j)} = \Theta_K^{(j)} - \alpha \frac{\partial}{\partial \Theta_K^{(j)}} J(\Theta^{(1)}, \Theta^{(2)}, \ldots \Theta^{(n_u)})$$

$$\Theta_K^{(j)} = \Theta_K^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_K^{(i)} + \lambda \Theta_K^{(j)} \right)$$

Content-based

$x^{(i)}$ = caracteristicas de cada pelicula i
$\Theta^{(j)}$ = gustos del usuario j de acuerdo a caracteristicas

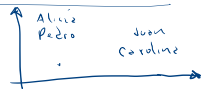
$x \in \mathbb{R}^n$ n=#features.
$\Theta \in \mathbb{R}^n$

	Acción	Romance,	Duración,	Subt,	Hollywood
Alicia	5	0,3	0,2	0,01	6

$(\Theta^{(j)})^T x^{(i)}$ → rating predicho del user j para peli i

Carolina
Juan
Pedro

↳ PCA → Clustering



$\begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 4,9 \\ 0,01 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1,5 \\ 4.8 \end{bmatrix}$

$x \in \mathbb{R}^3$
$n = 3$

Pelicula		Alicia (1) $\Theta^{(1)}$	Juan (2) $\Theta^{(2)}$	Pedro (3) $\Theta^{(3)}$	Carolina (4) $\Theta^{(4)}$	x_1 Ciencia ficción	x_2 Terror
Interestelar	$x^{(1)}$	5	5	0	1	?	?
Blade runner	$x^{(2)}$	5	? = 4	?	1	?	?
Yo Robot	$x^{(3)}$	? 4.95	4	1	?	?	?
La Monja	$x^{(4)}$	0	0	4	4	?	?
Chuckie	$x^{(5)}$	0	0	5	?	?	?

Dados $x^{(1)}, x^{(2)}, \ldots x^{(n_m)}$, estimar $\Theta^{(1)}, \Theta^{(2)}, \ldots \Theta^{(n_u)}$

$$\min_{\Theta^{(1)} \ldots \Theta^{(n_u)}} \frac{1}{2} \sum_{j} \sum_{i:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j} \sum_{K=1}^{n} (\Theta_K^{(j)})^2$$

Dados $\Theta^{(1)}, \Theta^{(2)}, \ldots \Theta^{(n_u)}$, estimar $x^{(1)}, x^{(2)}, \ldots x^{(n_m)}$

$$\min_{x^{(1)} \ldots x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} ((\Theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i} \sum_{K=1}^{n} (x_K^{(i)})^2$$

Filtro colaborativo

Pelicula		Alicia (1) $\Theta^{(1)}$	Juan (2) $\Theta^{(2)}$	Pedro (3) $\Theta^{(3)}$	Carolina (4) $\Theta^{(4)}$	Jorge $\Theta^{(5)}$		
Interestelar	$x^{(1)}$	5	5	0	1	?	0	2,6
Blade runner	$x^{(2)}$	5	? = 4	?	1	?	0	1,5
Yo Robot	$x^{(3)}$	? 4.95	4	1	?	?	0	1,3
La Monja	$x^{(4)}$	0	0	4	4	?	0	2
Chuckie	$x^{(5)}$	0	0	5	?	?	0	1,3

random

$X \rightarrow \Theta \rightarrow X \rightarrow \Theta \rightarrow X \rightarrow \Theta \rightarrow X$

Algoritmo:

$x \in \mathbb{R}^n$ $n = ???$ → hiperparámetro

$n = 10$, X X

1: Inicializar $x^{(1)}, x^{(2)}, \ldots x^{(n_m)}$ y $\Theta^{(1)}, \ldots \Theta^{(n_u)}$ de forma aleatoria (en valores pequeños)

2: Minimizar $J(x^{(1)} \ldots x^{(n_m)}, \Theta^{(1)} \ldots \Theta^{(n_u)})$ (hasta la convergencia)

$$x_K^{(i)} = x_K^{(i)} - \alpha \frac{\partial}{\partial x_K^{(i)}} J(\;)$$

$$\Theta_K^{(j)} = \Theta_K^{(j)} - \alpha \frac{\partial}{\partial \Theta_K^{(j)}} J(\;)$$

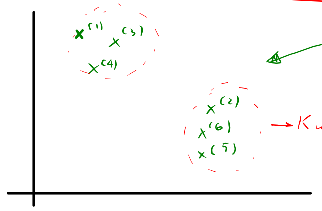
3: Para un usuario con parametros Θ y una peli con caracteristicas x, rating predicho: $\Theta^T x$

Analisis de similaridad

Para cada producto hemo aprendido un vector de caracteristicas $x \in \mathbb{R}^n$.

~~x_1 = C. Ficc, x_2 = terror, x_3 = rom.~~

n=10
↳ PCA
↳ 2



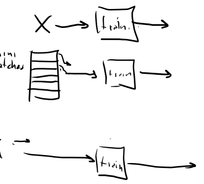
$\text{dist}(x^{(K)}, x^{(1)})$ = similares

→ K means

$x^{(2)}$ = la novia de chuckie
$x^{(6)}$ = herbie
$x^{(5)}$ =

A medida que se generan compras, el algoritmo mejora su precisión.

- Batch gradient descent
- Stochastic gradient descent
- Online learning



B2C

B2B