Aplicaciones en Internet

Sistemas de Recomendación

Motivación

Sistemas de recomendación

Ejemplos:

amazon.com, Pandora, StumbleUpon, Netflix, del.icio.us, movielens (helping you find the right movies), last.fm, Google News, YouTube, Xbox Live

---

De la escasez a la abundancia

El espacio de exposición es un recurso escaso en los comercios tradicionales

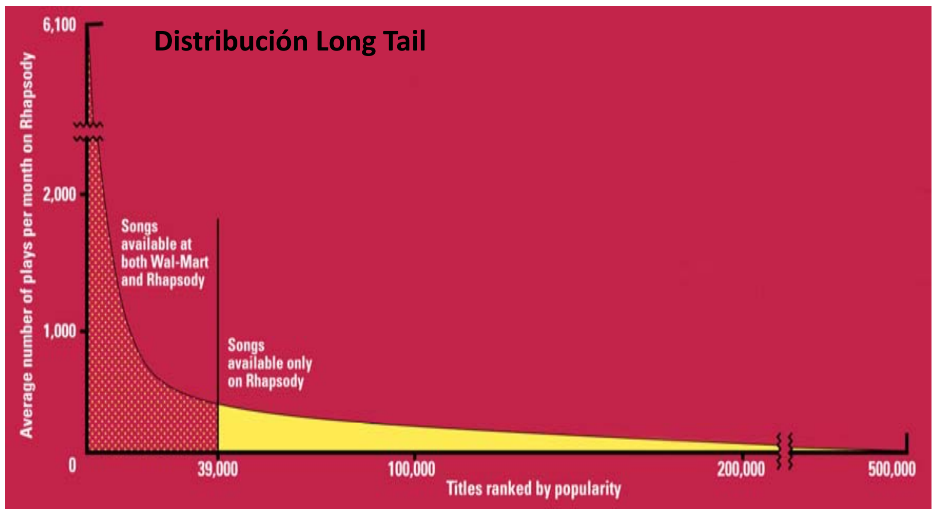
- También lo es el tiempo en TV, los trailers en el cine, etc

La web permite diseminar la información sobre productos a un coste casi nulo

- De la escasez a la abundancia

Más opciones requieren mejores filtros

- Motores de recomendación (Recommendation engines)



Tipos de recomendación

Manual
- Listas de favoritos
- Elaborados por criticos y expertos

Agregadores simples
- Top 10, más populares, más recientes

Ajustados a los intereses de cada usuario
- Amazon, Netflix,…

The 10 Algorithms That Dominate Our World

http://io9.com/the-10-algorithms-that-dominate-our-world-1580110464
The importance of algorithms in our lives today cannot be overstated. They are used virtually everywhere, from financial institutions to dating sites. But some algorithms shape and control our world more than others — and these ten are the most significant.

Goodfilms: descubre, califica y comparte películas con tus amigos

http://bitelia.com/2014/12/goodfilms-descubre-peliculas?utm_content=buffercdc5f&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
Goodfilms: descubre buenas películas en tus sitios de streaming favoritos Compartir Twittear Pinear Comentar Goodfilms es un sitio web social que te ayuda es descubrir películas que te gusten, encontrarlas en Netflix o Itunes , y además escribir reseñas para compartir tus opiniones con amigos o la comunidad. ¡No te pierdas nuestro contenido! hipertextual

Forgotify: canciones para descubrir. O para olvidar. O para preguntarse que hacen ahí | Microsiervos (Música)

http://www.microsiervos.com/archivo/musica/canciones-para-descubrir.html
19 Dic 2014 Forgotify: canciones para descubrir. O para olvidar. O para preguntarse que hacen ahí Forgotify parte de un dato impactante y una idea curiosa. El dato es que existen más de 4 millones de canciones en Spotify que nunca, jamás, nadie ha escuchado . Y eso que hay 40 millones de usuarios dale que te pego al play . Pero 4 millones dentro del catálogo de tal vez 20 o 25 millones de canciones de Spotify es un porcentaje enorme. ¿Tal vez la gente «se olvidó de ellas»? Así que la idea tras ...

The Shazam Effect - The Atlantic

http://www.theatlantic.com/magazine/archive/2014/12/the-shazam-effect/382237/?single_page=true
December 2014 The Shazam Effect Record companies are tracking download and search data to predict which new songs will be hits. This has been good for business—but is it bad for music? Derek Thompson Nov 17 2014, 7:58 PM ET Share on Facebook Twitter LinkedIn Email Print Comments Gluekit In 2000, a Stanford Ph.D. named Avery Wang co-founded, with a couple of business-school graduates, a tech start-up called Shazam. Their idea was to develop a service that could identify any song within a few sec...

Ejemplo: Predicción de puntuaciones de películas

Usuarios puntúan las películas de 1 a 5 estrellas

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)
Pretty Woman	5	5	0	0
Titanic	5	?	?	0
Algo para Recordar	?	4	0	?
La Jungla de Cristal	0	0	5	4
Skyfall	0	0	5	2

$n_u = 4$ $n_m = 5$ $y^{(2,1)} = 5$ $r(2,1) = 1$ $r(2,2) = ?$ $r(i,j) = 0$ en caso contrario

n_u = no. usuarios
n_m = no. películas
$r(i,j)$ = 1 si user j ha puntuado peli. i
$y^{(i,j)}$ = puntuación dada por user j a peli. i (definida solo si $r(i,j)=1$)

En nuestra notación, $r(i,j) = 1$ si el usuario j ha puntuado la película i, y $y^{(i,j)}$ es su puntuación para esa película. Considera el siguiente ejemplo (número de películas $n_m = 2$, numero de usuarios $n_u = 3$):

	Usuario 1	Usuario 2	Usuario 3
Película 1	0	1	?
Película 2	?	5	5

Qué valores tienen $r(2,1)$ y $y^{(2,1)}$?

- ❑ $r(2,1)=0$, $y^{(2,1)}=1$
- ❑ $r(2,1)=1$, $y^{(2,1)}=1$
- ☒ $r(2,1)=0$, $y^{(2,1)}$=undefined
- ❑ $r(2,1)=1$, $y^{(2,1)}$=undefined

Aplicaciones en Internet

Sistemas de Recomendación

Recomendaciones basadas en contenido

Sistemas de recomendación basados en contenidos → caracteristicas

vector de parametros

Película	Alicia (1) $\theta^{(1)}$	Paco (2) $\theta^{(2)}$	Elena (3) $\theta^{(3)}$	Pepe (4) $\theta^{(4)}$	x_1 (romantica)	x_2 (acción)	
Pretty Woman	5	5	0	0	0.9	0	$\rightarrow x^{(1)}$
Titanic	5	?	?	0	1.0	0.2	
Algo para Recordar	?	4	0	?	1.0	0	⋮
La Jungla de Cristal	0	0	5	4	0	1.0	
Skyfall	0	0	5	?	0.1	0.9	$\rightarrow x^{(5)}$

Para cada usuario, j aprender parámetro $\theta^{(j)} \in \mathbb{R}^3$.

Predecir para el usuario j la puntuación de la película i: $(\theta^{(j)})^T x^{(i)}$

$$x^{(3)} = \begin{bmatrix}1\\1\\0\end{bmatrix} \quad (1 \rightarrow x_0^{(3)})$$

$$(\theta^{(1)})^T x^{(3)} = \left(\theta_0^{(1)}\ \theta_1^{(1)}\ \theta_2^{(1)}\right)\begin{pmatrix}1\\1\\0\end{pmatrix} = 5$$

$$(\theta^{(1)})^T x^{(3)} = \theta_0^{(1)} + \theta_1^{(1)} x_1^{(3)} + \theta_2^{(1)} x_2^{(3)} = \hat{y}^{(3,1)}$$

Para las siguientes puntuaciones

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)	(romantica)	(acción)
Pretty Woman	5	5	0	0	0.9	0
Titanic	5	?	?	0	1.0	0.2
Algo para Recordar	?	4	0	?	1.0	0
La Jungla de Cristal	0	0	5	4	0	1.0
Skyfall	0	0	5	?	0.1	0.9

¿Cuál de los siguientes vectores es un valor razonable para $\theta^{(3)}$? Recuerda que $x_0 = 1$

- ❑ $\theta^{(3)} = [0; 5; 0]$
- ❑ $\theta^{(3)} = [1; 0; 4]$
- ❑ $\theta^{(3)} = [0; 0; 1]$
- ☒ $\theta^{(3)} = [0; 0; 5]$

Formulación del problema

$r(i,j) = 1$ si el usuario j ha puntuado la película i (0 si no)

$y^{(i,j)} =$ puntuación del usuario j a la película i (si la ha puntuado)

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

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

Para el usuario j, película i, predecir puntución $(\theta^{(j)})^T(x^{(i)})$

$m^{(j)}$ = número de películas puntuadas por el usuario j

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

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

Objetivo de optimización:

Para aprender $\theta^{(j)}$ (parámetro del usuario 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 aprender $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(n_u)}$

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

Algoritmo de optimización:

$J(\theta^{(1)}, \theta^{(2)} \ldots \theta^{(n_u)}) = J(\Theta)$

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

Actualización del descenso del gradiente: $\frac{\partial J(\Theta)}{\partial \theta_0}$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} \text{ (for } k = 0)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \underbrace{\left(\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right)}_{\frac{\partial J(\Theta)}{\partial \theta_k^{(j)}}} \text{ (for } k \neq 0)$$

Aplicaciones en Internet

Sistemas de Recomendación

Filtrado Colaborativo

Motivación del problema

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)	x_1 (romantica)	x_2 (acción)
Pretty Woman	5	5	0	0	0.9	0
Titanic	5	?	?	0	1.0	0.2
Algo para Recordar	?	4	0	?	1.0	0
La Jungla de Cristal	0	0	5	4	0	1.0
Skyfall	0	0	5	?	0.1	0.9

Motivación del problema

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)	x_1 (romantica)	x_2 (acción)
Pretty Woman	5	5	0	0	?	?
Titanic	5	?	?	0	?	?
Algo para Recordar	?	4	0	?	?	?
La Jungla de Cristal	0	0	5	4	?	?
Skyfall	0	0	5	?	?	?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

Algoritmo de optimización

Dado $\theta^{(1)}, \dots, \theta^{(n_u)}$, para aprender $x^{(i)}$:

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

Dado $\theta^{(1)}, \dots, \theta^{(n_u)}$, para aprender $x^{(1)}, \dots, x^{(n_m)}$:

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

Filtrado Colaborativo

Dados $x^{(1)}, \dots, x^{(n_m)}$ (y las puntuaciones de las películas), podemos estimar $\theta^{(1)}, \dots, \theta^{(n_u)}$

Dados $\theta^{(1)}, \dots, \theta^{(n_u)}$, podemos estimar $x^{(1)}, \dots, x^{(n_m)}$

Inicializar $\theta^{(1)} \dots \theta^{(n_u)}$ a valores aleatorios $\xrightarrow{\text{R.L.}}$ obtenemos $x^{(1)} \dots x^{(n_m)}$ $\xrightarrow{\text{R.L.}}$ $\theta^{(1)} \dots \theta^{(n_u)}$ $\xrightarrow{\text{R.L.}}$ $x^{(1)} \dots x^{(n_m)}$ ... hasta que converge

MÉTODO DE LAS PROYECCIONES ALTERNADAS

Supongamos que usamos descenso del gradiente para minimizar

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

¿Cuál de las siguientes es una regla correcta de actualización del gradiente para i≠0?

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

Aplicaciones en Internet

Sistemas de Recomendación

Algoritmo de filtrado colaborativo

Objetivo de optimización del filtrado colaborativo

Dados $x^{(1)},\ldots,x^{(n_m)}$, estimar $\theta^{(1)},\ldots,\theta^{(n_u)}$:

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

Dados $\theta^{(1)},\ldots,\theta^{(n_u)}$, estimar $x^{(1)},\ldots,x^{(n_m)}$:

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

Minimizar $x^{(1)},\ldots,x^{(n_m)}$ y $\theta^{(1)},\ldots,\theta^{(n_u)}$ simultaneaménte:

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

$$\min_{\substack{x^{(1)},\ldots,x^{(n_m)}\\ \theta^{(1)},\ldots,\theta^{(n_u)}}} J(x^{(1)},\ldots,x^{(n_m)},\theta^{(1)},\ldots,\theta^{(n_u)})$$

Algoritmo de filtrado colaborativo

1. Inicializar $x^{(1)},\ldots,x^{(n_m)},\theta^{(1)},\ldots,\theta^{(n_u)}$ a pequeños valores aleatorios.
2. Minimizar $J(x^{(1)},\ldots,x^{(n_m)},\theta^{(1)},\ldots,\theta^{(n_u)})$ usando descenso del gradiente (u otro algoritmo de optimización más avanzado). Es decir, para todo $j = 1,\ldots,n_u, i = 1,\ldots,n_m$

$$x_k^{(i)} := x_k^{(i)} - \alpha\left(\sum_{j:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})\theta_k^{(j)} + \lambda x_k^{(i)}\right) \quad \frac{\partial J(X,\Theta)}{\partial x_k^{(i)}}$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha\left(\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})x_k^{(i)} + \lambda\theta_k^{(j)}\right) \quad \frac{\partial J(X,\Theta)}{\partial \theta_k^{(j)}}$$

3. Para un usuario con parámetros θ y una película con características (aprendidas) x , predecimos una puntuación de $\theta^T x$ estrellas.

usuario j, película i $\rightarrow \hat{y}^{(i,j)} = (\theta^{(j)})^T x^{(i)}$

Aplicaciones en Internet

Sistemas de Recomendación

Vectorización: Factorización de bajo rango.

Filtrado colaborativo

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)
Pretty Woman	5	5	0	0
Titanic	5	?	?	0
Algo para Recordar	?	4	0	?
La Jungla de Cristal	0	0	5	4
Skyfall	0	0	5	?

$$\boldsymbol{Y} = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Filtrado colaborativo

$$\boldsymbol{Y} = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Puntuaciones pronosticadas:

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

$$X = \underbrace{\begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(n_m)})^T \end{bmatrix}}_{n} \; n_m \qquad \Theta = \underbrace{\begin{bmatrix} (\theta^{(1)})^T \\ (\theta^{(2)})^T \\ \vdots \\ (\theta^{(n_u)})^T \end{bmatrix}}_{n} \; n_u$$

Definamos

$$X = \begin{bmatrix} - & (x^{(1)})^T & - \\ & \vdots & \\ - & (x^{(n_m)}) & - \end{bmatrix}, \quad \Theta = \begin{bmatrix} - & (\theta^{(1)})^T & - \\ & \vdots & \\ - & (\theta^{(n_u)}) & - \end{bmatrix}$$

¿De que otra forma podemos expresar

$$\hat{Y} = n_m \underbrace{\begin{bmatrix} (x^{(1)})^T(\theta^{(1)}) & \dots & (x^{(1)})^T(\theta^{(n_u)}) \\ \vdots & \ddots & \vdots \\ (x^{(n_m)})^T(\theta^{(1)}) & \dots & (x^{(n_m)})^T(\theta^{(n_u)}) \end{bmatrix}}_{n_u}$$

$X = \underbrace{[\;]}_{n}\; n_m \qquad \Theta = \underbrace{[\;]}_{n}\; n_u$

- ❑ $X\Theta$
- ☒ $X\Theta^T$
- ❑ $X^T\Theta$
- ❑ $\Theta^T X^T$

$X * \Theta^T$

Encontrar películas relacionadas

Para cada producto i encontramos un vector de características

. $x^{(i)} \in \mathbb{R}^n$

Cómo encontrar películas j relacionadas con la película i?

Encontrando vectores $x^{(j)}$ en X que tengan una distancia pequeña con $x^{(i)}$. Si $\underline{\|x^{(j)} - x^{(i)}\| \text{ es pequeño}}$, entonces $x^{(j)}$ y $x^{(i)}$ son "similares". $\|x\| = \sqrt{x_1^2 + x_2^2 + \dots x_n^2}$ $\quad \|x\|^2 = x^T \cdot x$

Las 5 películas más parecidas a i:

Encontrar las 5 películas j con menor $\|x^{(i)} - x^{(j)}\|$.

Aplicaciones en Internet

Sistemas de Recomendación

Detalle de implementación: Normalización en media

Usuarios que no han puntuado ninguna película

Película	Alicia (1)	Paco (2)	Elena (3)	Pepe (4)	Eva (5)
Pretty Woman	5	5	0	0	?
Titanic	5	?	?	0	?
Algo para Recordar	?	4	0	?	?
La Jungla de Cristal	0	0	5	4	?
Skyfall	0	0	5	?	?

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

$$\min_{\substack{x^{(1)},\dots,x^{(n_m)} \\ \theta^{(1)},\dots,\theta^{(n_u)}}} \frac{1}{2} \sum_{(i,j):\underline{r(i,j)=1}} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2 + \underbrace{\frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2}_{\theta^{(5)} \text{ sólo contribuye en este término}}$$

¿En que contribuye $\theta^{(5)}$ en este coste? $\quad r(i,5) = 0$

consecuencia $\quad \theta^{(5)} \approx \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ cualquier predicción $\underline{\hat{y}^{(i,5)} = 0}$

NO UTIL

Normalización en media:

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \qquad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

$y^{(i,j)} = y^{(i,j)} - \mu_i$

Redefinimos Y como la matriz de desviaciones con respecto a la media

Para el usuario j, predecir la siguiente puntuación de la película i:

$(\theta^{(j)})^T x^{(i)} + \mu_i$

Usuario 5 (Eva):

$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \underbrace{(\theta^{(j)})^T x^{(i)}}_{\emptyset} + \mu^i = \mu^i$

Hemos hablado de normalización en media. Sin embargo no aplicamos escalado de características, es decir, no hemos escalado las puntaciones dividiéndolas por el rango (max – min). Esto es porque

- ❑ Este tipo de escalado no es útil cuando el valor a predecir es un número real.
- ☒ Las puntuaciones de las películas ya son comparables (por ejemplo de 0 a 5 estrellas), por tanto ya están en la misma escala.
- ❑ Restar la media es matemáticamente equivalente a dividir por el rango.
- ❑ De esta forma el algoritmo es más eficiente computacionalmente.

Incorporación de sesgos (bias):

Motivación:

- Algunos usarios tienden a dar puntuaciones más altas y otros tienden a dar puntuaciones más bajas: sesgo usuario.
- También algunas películas tienden a recibir mejor puntuación que otras: sesgo películas.

Podemos introducir estos sesgos en nuestra formulación:

- Sesgo usuario j: b_j
- Sesgo película i: b_i

Incorporación de sesgos (bias):

Predicción sin incluir sesgos: $\hat{y}^{(i,j)} = \mu_i + \left(\theta^{(j)}\right)^T x^{(i)}$

Predicción incluyendo sesgos: $\hat{y}^{(i,j)} = \mu_i + b_j + b_i + \left(\theta^{(j)}\right)^T x^{(i)}$

Función de coste sin incluir sesgos:

$$J(\Theta, X) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mu_i + \left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \left\|\theta^{(j)}\right\|^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \left\|x^{(i)}\right\|^2$$

Función de coste incluyendo sesgos:

$$J\left(\Theta, X, \{b_j\}, \{b_i\}\right) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mu_i + b_j + b_i + \left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \left\|\theta^{(j)}\right\|^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \left\|x^{(i)}\right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} b_j^{\,2} + \frac{\lambda}{2} \sum_{i=1}^{n_m} b_i^{\,2}$$