Sistemas de recomendación: filtro colaborativo basado en modelos

Gestión de la Información
Grado en Ingeniería Informática
Universidad de Burgos

UNIVERSIDAD
DE BURGOS

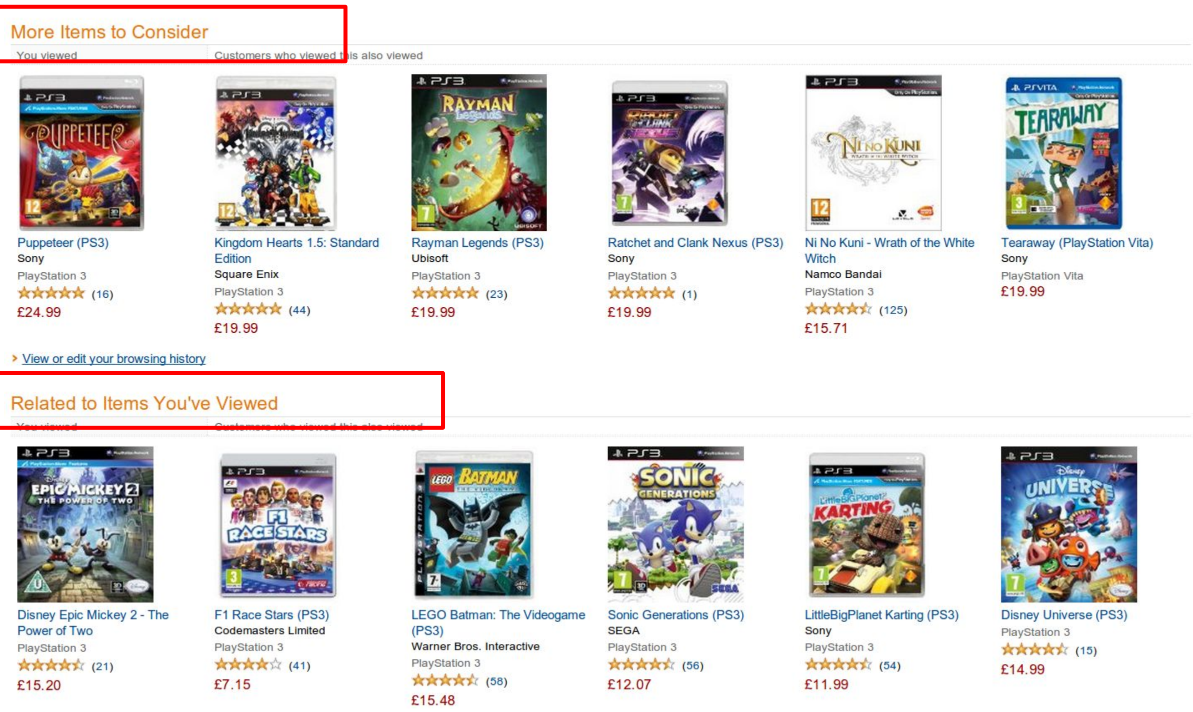
José Ignacio Santos, José Manuel Galán
jisantos@ ubu.es, jmgalan@ ubu.es

Contenidos

Problema

Ejemplo: Amazon: ¿Cómo aumentar las ventas?

Sistema de recomendación



Problema

Muchas empresas e-commerce disponen de un catálogo de productos inmenso.

Las empresas necesitan facilitar al cliente encontrar lo que busca. Existen diferentes estrategias (complementarias):

- ➔ Estrategia de **búsqueda**: facilitar al cliente la navegación y búsqueda.
  - ◆ Proceso lento, y puede no terminar con éxito
- ➔ Estrategia de **recomendación**: proponer una muestra de los productos que más pueden interesar al cliente
  - ◆ Cuanto más acertemos en sus intereses mayores probabilidades de compra

Un **sistema de recomendación filtra*** la información y muestra aquellos datos que más pueden interesar al usuario.

En general, los algoritmos de recomendación **aprenden*** de las elecciones de cada cliente para construir la lista más adecuada a sus preferencias

* Por eso se les llama filtros
* Aprender=estimar parámetros de un modelo en base a un conjunto de datos

Clases de sistemas de recomendación

Sistemas de recomendación

Un **sistema de recomendación** filtra la información y muestra aquellos datos que potencialmente pueden interesar al usuario

Por ejemplo, muchas empresas e-commerce disponen de un **catálogo de productos** inmenso, y necesitan proponer aquellos productos que **a priori pueden interesar** al cliente

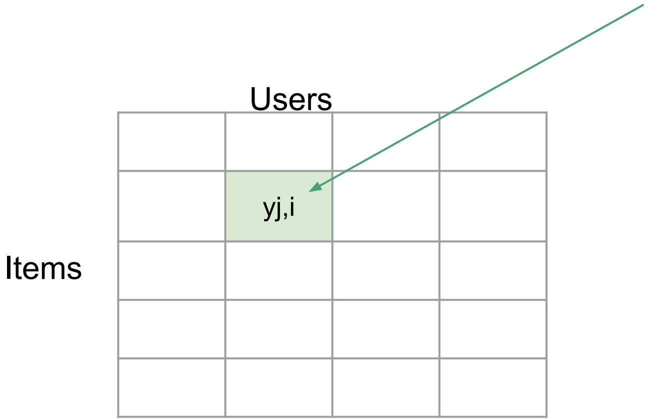
Cuanto más se acerquen a sus intereses mayores probabilidades de compra

Clases de sistemas de recomendación

Sistemas de recomendación
➔ Filtros basados en contenidos
➔ Filtros colaborativos

Filtros basados en contenidos: analizan las características del contenido de los productos para encontrar patrones que permitan hacer recomendaciones de nuevos productos

Filtros colaborativos: utilizan una matriz de utilidad, e.g. valoraciones (ratings=y_{ji}), de los usuarios sobre un conjunto de productos para recomendar nuevos productos.



Clases de sistemas de recomendación

Sistemas de recomendación
➔ Filtros basados en contenidos
➔ Filtros colaborativos

➔ Basados en modelos
➔ Basados en memoria

FC basados en modelos: utilizan los datos para ajustar modelos que después pueden ser utilizados para proponer recomendaciones (e.g. modelos de regresión)

FC basados en memoria: utilizan los datos para definir similitudes entre usuarios y productos que serán utilizados para construir las recomendaciones (e.g. Amazon)

Clases de sistemas de recomendación

Sistemas de recomendación
➔ Filtros basados en contenidos
➔ Filtros colaborativos

➔ Basados en modelos
➔ Basados en memoria
➔ Basados en usuarios
➔ Basados en productos

FC basados en usuarios:
La estimación del rating de un usuario **u** al producto **i** se basa en los ratings al producto **i** de otros **usuarios similares** a **u**

FC basados en productos: (e.g. Amazon)
La estimación del rating de un usuario **u** al producto **i** se basa en los ratings del usuario **u** a **productos similares** a **i**

Introducción a los filtros colaborativos basados en modelos

Filtros colaborativos basados en modelos que utilizan los datos para ajustar (=aprender) un modelo predefinido (generalmente modelos de regresión lineal)

Formalización del problema

Problema: **predecir el rating de películas**

➔ Supondremos que los usuarios de una tienda de alquiler de películas pueden valorarlas (rating) de 0 a 5 las películas que han visto. Se define una **matriz de utilidad** (ratings)

Película	**Sara**	**Raul**	**Pedro**	**Jorge**
Love Actually	5	5	0	0
Shakespeare enamorado	5	**?**	**?**	0
Tienes un email	**?**	4	0	**?**
Los juegos del hambre	0	0	5	4
Django desencadenado	0	0	5	**?**

Formalización del problema

Problema: **predecir el rating de películas**

➔ Supondremos que los usuarios de una tienda de alquiler de películas pueden valorarlas (rating) de 0 a 5 las películas que han visto. Se define una **matriz de utilidad** (ratings)

Película	**Sara**	**Raul**	**Pedro**	**Jorge**
Love Actually	5	5	0	0
Shakespeare enamorado	5	**?**	**?**	0
Tienes un email	**?**	4	0	**?**
Los juegos del hambre	0	0	5	4
Django desencadenado	0	0	5	**?**

Problema:
¿Qué rating ponemos en las incógnitas?

Si pudiéramos estimar todos los ratings, podríamos proponer a un usuario aquellas películas con mayor rating que todavía no ha visto

Problema de regresión: (1) datos

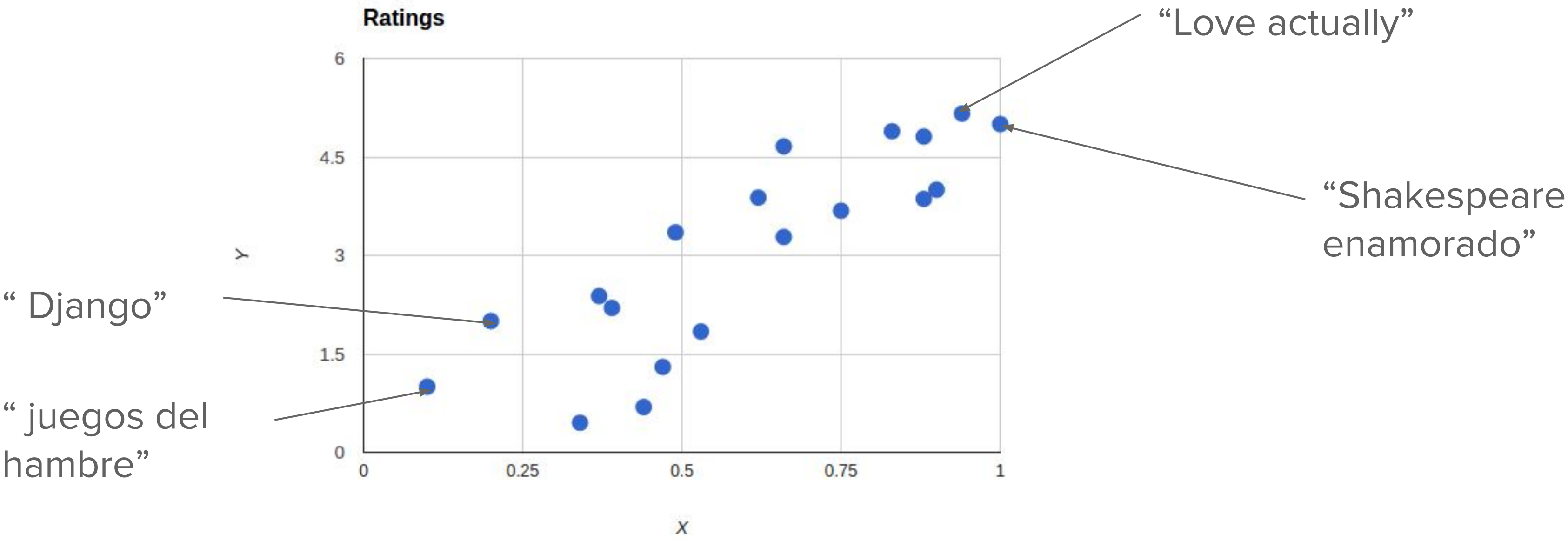
Centrémonos en **un usuario** y **todas las películas que ha visto**.
Supongamos que existe **una característica** de cada película, e.g. grado de romanticismo (X).
Es un contenido intrínseco a cada película

Películas (vistas)	**Usuario (Y)**	**Característica (X)**
(1) Love Actually	4	0.9
(2) Shakespeare enamorado	5	1.0
(4) Los juegos del hambre	1	0.1
(5) Django desencadenado	2	0.2
...	...	...

m

Problema de regresión: (1) datos

La matriz de utilidad de las películas vistas por el usuario puede representarse gráficamente (**diagrama de dispersión** Y~X)



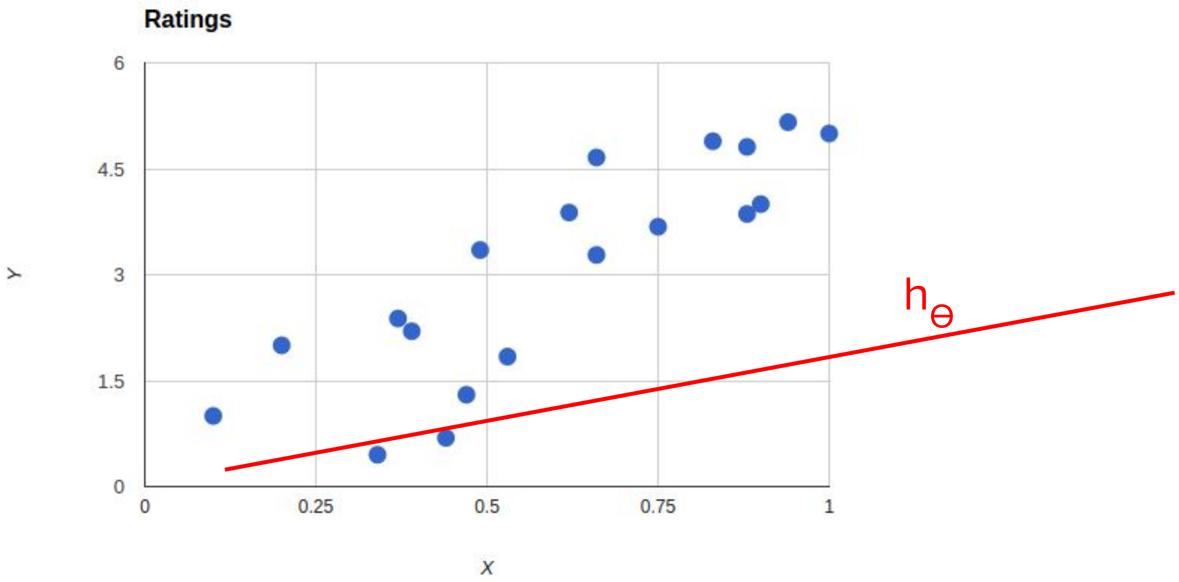
Problema de regresión: (2) modelo

Se propone (hipótesis) un **modelo lineal** que explica el rating esperado h_Θ de una película i como una función lineal de su característica $x^{(i)}$:

$$h_\Theta(x(i)) = \Theta_0 + \Theta_1 x^{(i)}$$

Este modelo lineal es un **conjunto infinito de "rectas"** definidas por los valores particulares de los **parámetros del usuario** Θ_0 y Θ_1

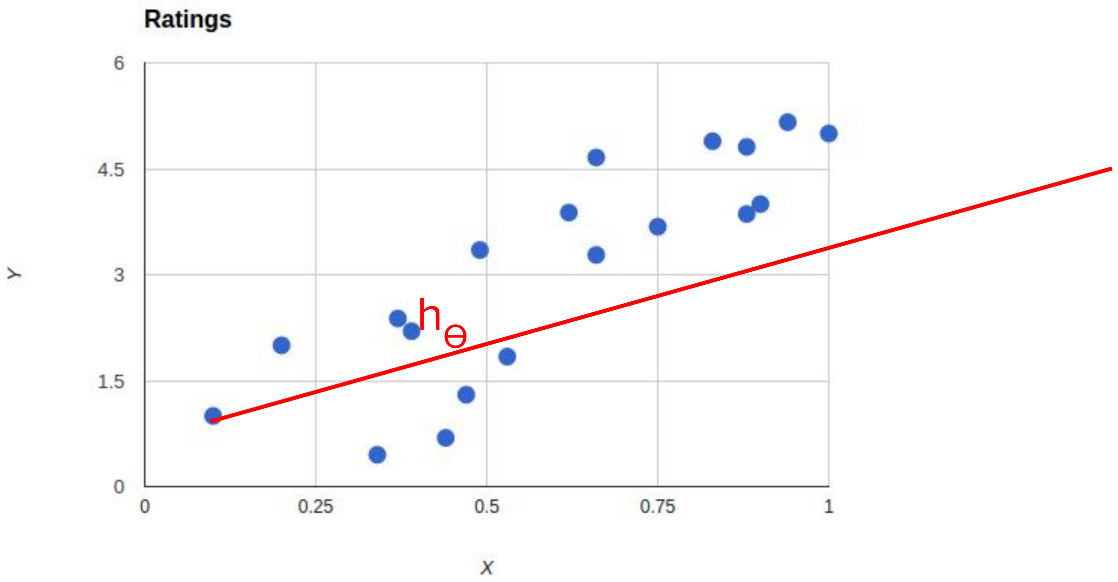
Problema de regresión: (2) modelo



$\theta_0 = 0$

$\theta_1 = 2$

Problema de regresión: (2) modelo



$\theta_0 = 1$

$\theta_1 = 4.5$

Problema de regresión: (3) aprendizaje

El problema de regresión puede ser explicado como un **problema de aprendizaje supervisado**. Se dispone de un conjunto de **datos de entrenamiento** (rating de películas vista,característica de la película) que puede ser utilizado para que el modelo aprenda (ajuste los parámetros)

m: número de muestras de entrenamiento (películas)

$x^{(i)}$: variable de entrada para la película i-ésima: característica de la película

$y^{(i)}$: variable de salida (objetivo) para la película i-ésima: rating del usuario

Problema de regresión: (3) aprendizaje

Conjunto de entrenamiento $\{x^{(i)},y^{(i)}\}$

Modelo $h_\Theta(x)$

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

Algoritmo de aprendizaje

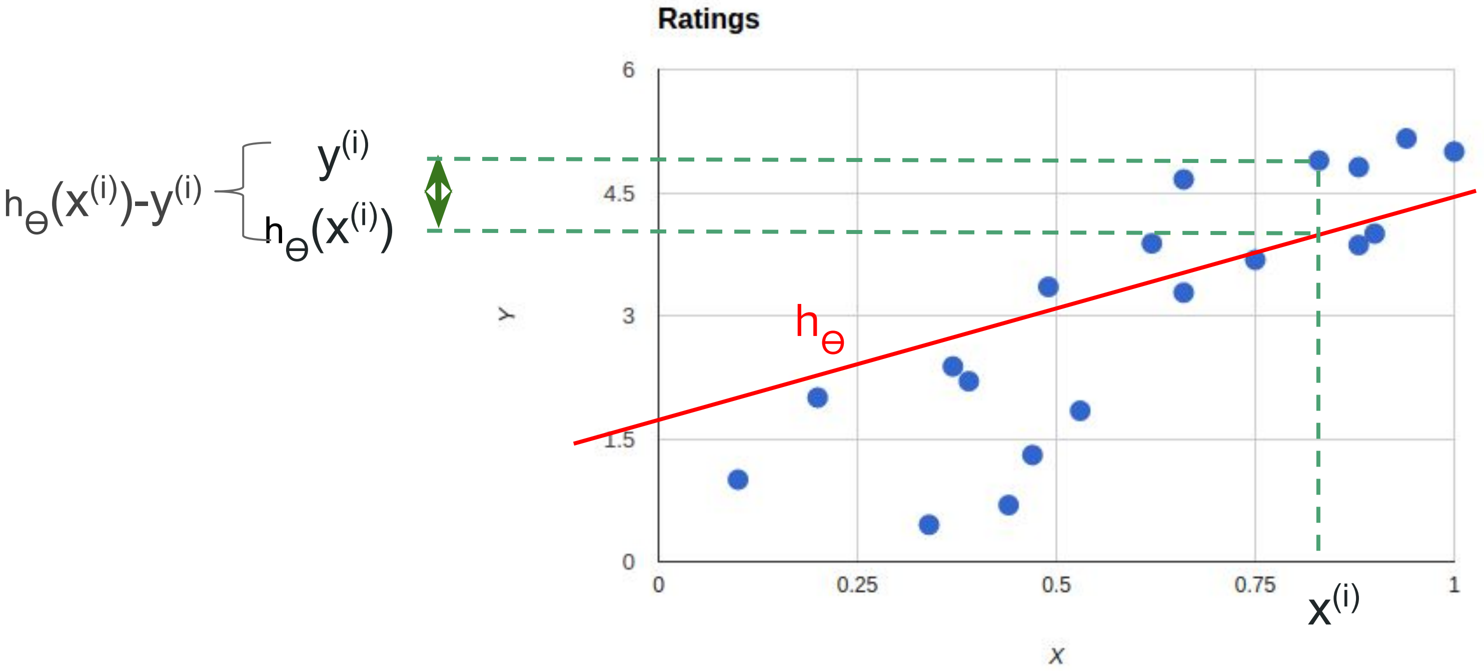
Escoger $\{\Theta_0,\Theta_1\}$ para que $\mathbf{h_\Theta(x)}$ esté lo más **próximo** posible a la variable objetivo **y** para todos los valores de entrenamiento $\mathbf{\{x^{(i)},y^{(i)}\}}$

Próximo:
función de coste
$J(\Theta_0,\Theta_1)$

Escoger:
$\text{Min}_{\Theta_0\Theta_1} J(\Theta_0,\Theta_1)$

Problema de regresión: (3) aprendizaje

Definimos una **función de coste**: $$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$



Problema de regresión: (3) aprendizaje

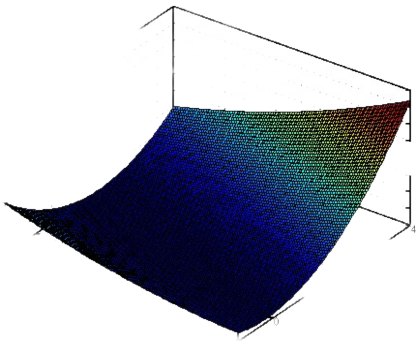
Algoritmo de aprendizaje: **descenso del gradiente**

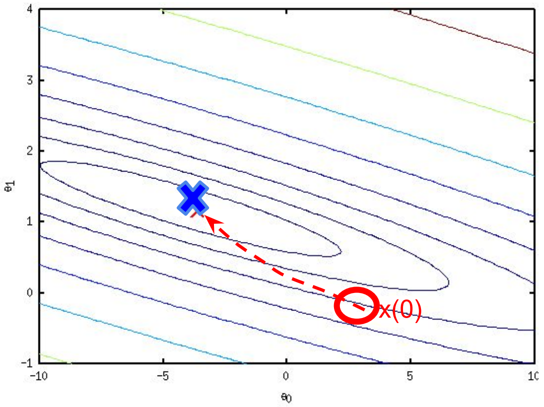
$\mathrm{Min}_{\Theta_0\Theta_1}\ J(\Theta_0,\Theta_1)$

Algoritmo:

(1) Inicializar (Θ_0,Θ_1)

(2) Actualizar (Θ_0,Θ_1) tal que se produzca un descenso en $J(\Theta_0,\Theta_1)$



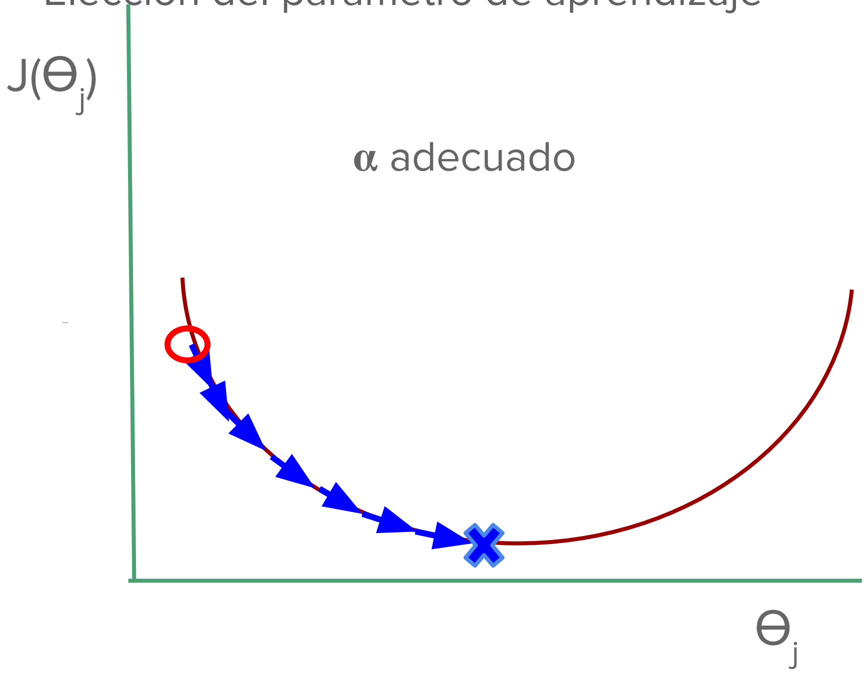


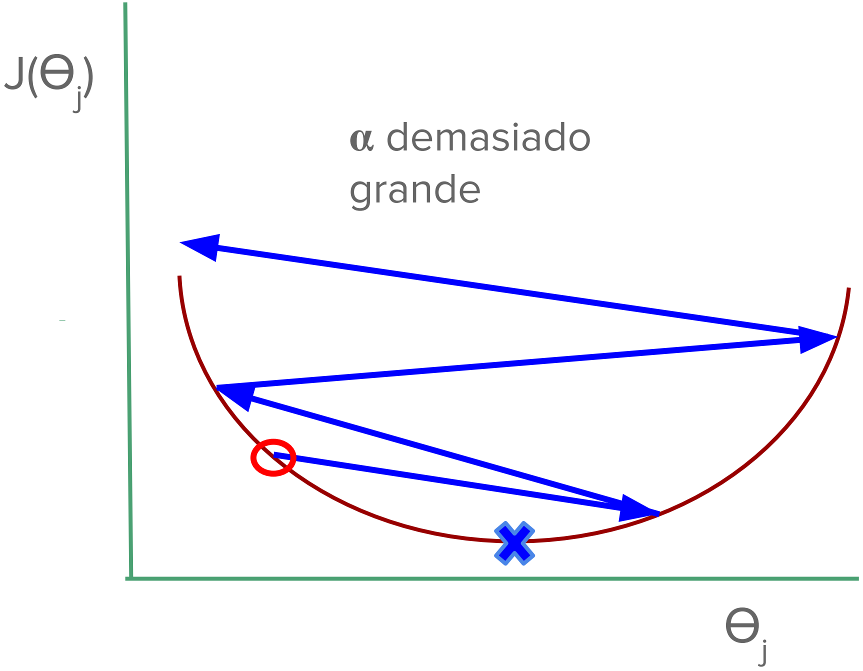
$$\theta_j := \theta_j - \alpha \frac{\partial}{\theta_j} J(\theta)$$

parámetro de aprendizaje

Problema de regresión: (3) aprendizaje

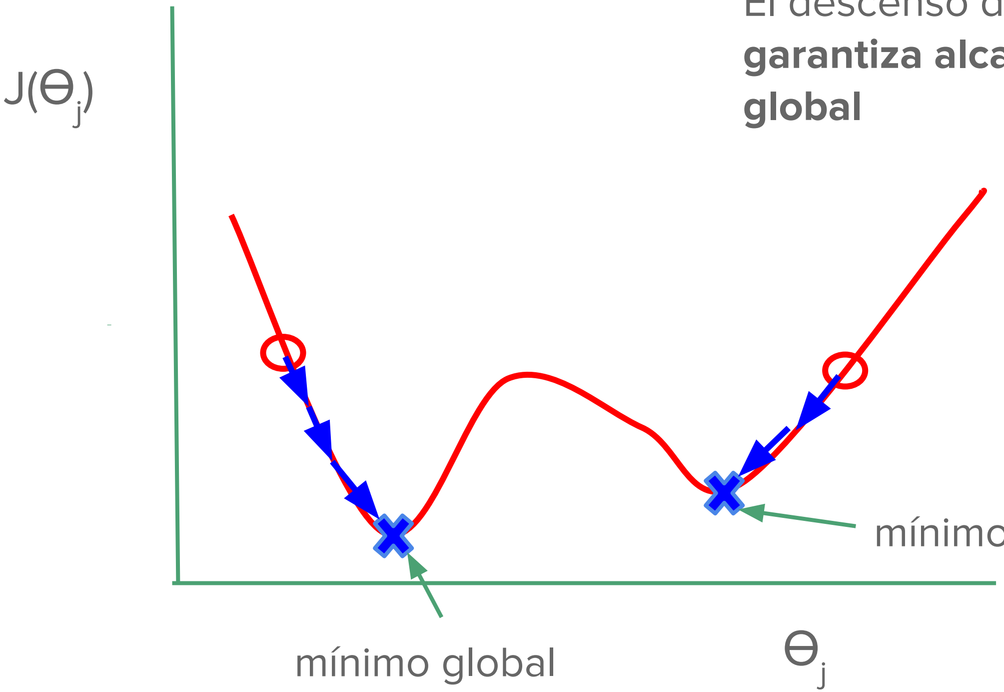
Algoritmo de aprendizaje: **descenso del gradiente.**
Elección del parámetro de aprendizaje





Problema de regresión: (3) aprendizaje

Optimización local



El descenso del gradiente **no garantiza alcanzar un mínimo global**

Filtro colaborativo basado en un modelo lineal

Filtro colaborativo basado en un modelo lineal que explica las valoraciones de los usuarios como función lineal de preferencias de usuario y características de los productos

Formalización del problema

- ➔ Volviendo al problema inicial donde tenemos más de un usuario.
- ➔ Además de la **matriz de utilidad** (ratings) que llamaremos Y, definiremos una **matriz R** de valores binarios {1,0} y que identifica las películas vistas por cada usuario
- ➔ Utilizaremos las siguientes variables:

	u	número de usuarios
	m	número de películas
Matriz R ->	$r^{(i,j)}$	= 1 si el Usuario j vió la Película i
Matriz Y ->	$y^{(i,j)}$	= rating Usuario j a la Película i (si r(i,j)=1)

FC conocidas las características

- Ahora supondremos que existen dos **características** (n=2) de las películas: grado de romanticismo (x1) y grado de acción (x2)
- Las características se suponen que son contenidos intrínsecos a cada película: cada película tiene un valor en cada característica

u

n=2

Película	Sara(1)	Raúl(2)	Pedro(3)	Jorge(4)	x1	x2
(1) Love Actually	5	5	0	0	0.9	0
(2) Shakespeare enamorado	5	?	?	0	1.0	0.01
(3) Tienes un email	?	4	0	?	0.99	0
(4) Los juegos del hambre	0	0	5	4	0.1	1.0
(5) Django desencadenado	0	0	5	?	0	0.9

m

FC conocidas las características

Cada película se representa por un **vector de (n+1) características (teniendo en cuenta el intercepto)***, para nuestro ejemplo n=2

$$x^{(i)} = [\ x_0^{(i)} = 1\ x_1^{(i)}\ x_2^{(i)}\ ] \in \Re^{n+1}$$

Para hacer una predicción necesitamos estimar (aprender) para cada usuario j un **vector de (n+1) parámetros** (preferencias)

$$\theta^{(j)} \in \Re^{n+1}$$

Por ejemplo, **si el vector de preferencias** de Sara (usuario 1) **es conocido** (0,5,0), entonces la predicción del rating de la película 3 (1,0.99,0) para Sara será:

$$\hat{y}^{(3,1)} = x^{(3)} (\theta^{(1)})^T = [\ 1\ 0.99\ 0\ ] \boxed{\begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}} = 0.99 * 5 = 4.95$$

Función objetivo

Para estimar el vector de parámetros **del usuario j** se hace una **regresión lineal** que minimiza la siguiente función objetivo:

$$\min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right)^2 + \underbrace{\frac{\lambda}{2} \sum_{k=0}^{n} (\theta_k^{(j)})^2}$$

donde se introduce un término de **regularización** para evitar el overfitting

Para estimar los vectores de parámetros **de todos los usuarios** se generaliza la **función objetivo**:

$$\min_{\theta^{(1)},\dots,\theta^{(u)}} \frac{1}{2} \sum_{j=1}^{u} \sum_{i:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{u} \sum_{k=0}^{n} (\theta_k^{(j)})^2$$

Ajuste mediante descenso del gradiente

Función objetivo:

$$\min_{\theta^{(1)},\dots,\theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=0}^{n} (\theta_k^{(j)})^2$$

Ajuste iterativo mediante **descenso del gradiente** (con un parámetro alfa de aprendizaje)

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right) x_k^{(i)} \quad para \;\; k=0$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right) x_k^{(i)} + \lambda \theta_k^{(j)} \right) \quad para \;\; k \neq 0$$

Limitaciones de esta primera aproximación

Este filtro es lo que se denomina un Filtro Basado en Contenidos (**Content-Based Filtering**)

El algoritmo del sistema de recomendación basado en contenidos **requiere que exista un vector de características** para todas las películas que describa adecuadamente los contenidos

Sin embargo, en general:

- No se dispone de estos valores para todas las películas
- Resulta difícil y lento encontrar a “alguien” que haga la evaluación completa de las características* de todas las películas

* Hablamos de muchas más de 2 características

FC conocidas las preferencias

Suponemos que ahora **se desconocen las características** de cada película, pero **sí se conocen los vectores de parámetros** de cada usuario (sus **preferencias**), e.g. cuánto valora que una película sea romántica y de acción. ¿Es posible estimar las características?

$$\theta^{(1)}=[\ 0\ 5\ 0\ ],\theta^{(2)}=[\ 0\ 5\ 0\ ],\theta^{(3)}=[\ 0\ 0\ 5\ ],\theta^{(4)}=[\ 0\ 0\ 5\ ]$$

Película	**Sara(1)**	**Raúl(2)**	**Pedro(3)**	**Jorge(4)**	**x1**	**x2**
(1) Love Actually	5	5	0	0	?	?
(2) Shakespeare enamorado	5	**?**	**?**	0	?	?
(3) Tienes un email	**?**	4	0	**?**	?	?
(4) Los juegos del hambre	0	0	5	4	?	?
(5) Django desencadenado	0	0	5	**?**	?	?

FC conocidas las preferencias

Suponer por ejemplo que **desconocemos** las características de la película 1:

$$x^{(i)}=[\ x_0^{(i)}=1\ x_1^{(i)}\ x_2^{(i)}\ ]\in\Re^{n+1} \quad \text{i=1}$$

pero **conocemos** las preferencias de los usuarios y sus ratings:

$$\theta^{(1)}=[\ 0\ 5\ 0\ ],\theta^{(2)}=[\ 0\ 5\ 0\ ],\theta^{(3)}=[\ 0\ 0\ 5\ ],\theta^{(4)}=[\ 0\ 0\ 5\ ]$$

En este ejemplo podemos aproximar los parámetros desconocidos fácilmente:

$$x^{(1)}(\theta^{(1)})^T=5,\ \ x^{(1)}(\theta^{(2)})^T=5,\ \ x^{(1)}(\theta^{(3)})^T=0,\ \ x^{(1)}(\theta^{(4)})^T=0$$

$$x_1^{(1)}=1,\ \ x_2^{(1)}=0$$

FC conocidas las preferencias: estimar $x^{(i)}$

Conocidos los vectores de parámetros de cada usuario, podemos estimar el vector de características de la película i mediante una **regresión lineal**:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} (x^{(i)}(\theta^{(j)})^T - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{k=0}^{n} (x_k^{(i)})^2$$

Ahora los parámetros de la regresión están en el vector $x^{(i)}$

Para estimar los vectores de características de todas las películas se generaliza la **función objetivo***:

$$\min_{x^{(1)},\dots,x^{(m)}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j:r(i,j)=1} (x^{(i)}(\theta^{(j)})^T - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{m} \sum_{k=0}^{n} (x_k^{(i)})^2$$

* Es un problema de regresiones lineales separadas idéntico al propuesto para el sistema basado en contenidos

FC conocidas las preferencias: estimar $x^{(i)}$

Conocidos los vectores de parámetros de cada usuario, podemos estimar el vector de características de la película i mediante una **regresión lineal**:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{k=0}^{n} (x_k^{(i)})^2$$

Ahora los parámetros de la regresión están en el vector $x^{(i)}$

Para estimar los vectores de características de todas las películas se generaliza la **función objetivo***:

$$\min_{x^{(1)},\dots,x^{(m)}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j:r(i,j)=1} \left(x^{(i)}(\theta^{(j)})^T - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{i=1}^{m} \sum_{k=0}^{n} (x_k^{(i)})^2$$

* Es un problema de regresiones lineales separadas idéntico al propuesto para el sistema basado en contenidos

FC completo: estimaciones recursivas

El filtro cooperativo requiere dos etapas:

$$Dado\ \{x^{(1)},\ldots,x^{(m)}\},\ y\ los\ ratings\ \{y^{(i,j)}\}\ estimamos\ \{\theta^{(1)},\ldots,\theta^{u)}\}$$

$$Dado\ \{\theta^{(1)},\ldots,\theta^{(u)}\},\ y\ los\ ratings\ \{y^{(i,j)}\}\ estimamos\ \{x^{(1)},\ldots,x^{(m)}\}$$

que se realizan iterativamente desde una solución inicial aleatoria

$$aleatorio\ \{\theta^{(j)}\}\rightarrow\{x^{(i)}\}\rightarrow\{\theta^{(j)}\}\rightarrow\{x^{(i)}\}\rightarrow\{\theta^{(j)}\}\rightarrow\{x^{(i)}\}\rightarrow\ldots$$

FC función objetivo integrada

Ahora no se requiere intercepto, $\theta^{(j)}, x^{(i)} \in \Re^n$
y la **Función objetivo integrada**:

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

Parámetros de los dos conjuntos
de regresiones iterativas

Función de pérdida	=	Suma errores de predicción	+	Penalización (regularización Ridge) de parámetros X	+	Penalización (regularización Ridge) de parámetros Ɵ

ANEXO

FC algoritmo

(1) Inicializar con valores pequeños aleatorios $\{\theta^{(1)},\ldots,\theta^{u)}\},\{x^{(1)},\ldots,x^{(m)}\}$

(2) Minimizar la función objetivo aplicando el descenso del gradiente:

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

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

(3) Calcular la predicción del rating de la película i para el usuario j como:

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

FC: normalización de media

¿Qué ocurre si aparece un nuevo usuario que no ha valorado **ninguna** película?

Película	Sara(1)	Raúl(2)	Pedro(3)	Jorge(4)	Bea(5)
(1) Love Actually	5	5	0	0	?
(2) Shakespeare enamorado	5	**?**	**?**	0	?
(3) Tienes un email	**?**	4	0	**?**	?
(4) Los juegos del hambre	0	0	5	4	?
(5) Django desencadenado	0	0	5	**?**	?

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

El usuario 5 no tiene ratings

$$\min_{\theta_1^{(5)},\theta_2^{(5)}} \frac{\lambda}{2}\left((\theta_1^{(5)})^2 + (\theta_2^{(5)})^2\right)$$

$$\theta_1^{(5)} = \theta_2^{(5)} = 0$$

$$\hat{y}^{(i,5)} = (\theta^{(5)})^T x^{(i)} = 0 \quad \forall i$$

FC: normalización de media

Para solucionar esta indefinición, antes de minimizar **normalizamos la matriz de ratings** restando a cada rating la media de rating de la correspondiente película

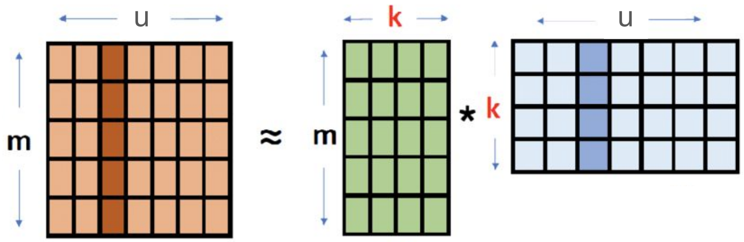
$$\acute{Y} = Y - \mu = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} - \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2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

Ahora la predicción para el usuario 5 coincide con el rating promedio de las películas (aproximación razonable)

$$\hat{y}^{(i,5)} = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix}$$

FC como problema de factorización

El problema de **factorización matricial de bajo rango** permite descomponer una matriz como producto de dos matrices de menor rango (máxima dimensión de subespacios linealmente independientes)



A U V^T

Antes lo llamábamos Y X ϴ

$$A_{[mxu]} = UV^T = \begin{bmatrix} U_1^{(1)} & \dots & U_n^{(1)} \\ . & \dots & . \\ U_1^{(m)} & \dots & U_n^{(m)} \end{bmatrix} \begin{bmatrix} V_1^{(1)} & \dots & V_n^{(1)} \\ . & \dots & . \\ V_1^{(u)} & \dots & V_n^{(u)} \end{bmatrix}^T$$

k = $n << (m, u)$

Es útil con sistemas de recomendación. La matriz A a descomponer es la matriz de ratings de los usuarios al conjunto de ítems. Las dos **matrices U y V capturan las características latentes de ítems y de usuarios**. Si somos capaces de construir esta dos matrices, podemos reconstruir la matriz original y consecuentemente estimar las celdas vacías con las que hacer recomendaciones

Algoritmo factorización en Surprise

La librería Surprise recoge una serie de algoritmos para sistemas de recomendación. Entre ellos, propone una variante del problema de factorización de bajo rango propuesta por Koren, Bell y Volinsky (2009) en el contexto del Netflix Prize (2008–2009)

Incluye sesgos (biases). Cada predicción no es solo $Ui{\cdot}Vj$ sino que además suma un **sesgo global μ** (media de todos los ratings), un **sesgo por usuario b_u** y un **sesgo por ítem b_i**

$$r_{ui} = \mu + b_u + b_i + U_u \cdot V_i$$

Los sesgos capturan que algunos usuarios tienden a puntuar más alto/bajo en general, y que algunos ítems suelen recibir mejores/peores valoraciones (son más o menos populares), independientemente de las características latentes

Como en otros modelos impone regularización en la función de coste para evitar overfitting. Además, utiliza técnicas de **descenso de gradiente estocástico** para estimar las matrices y sesgos, y está implementado teniendo en cuenta la dispersión habitual de las matrices de ratings

¡En este algoritmo no es necesario normalizar con usuarios sin ratings puesto que estima el sesgo general y el de los items!

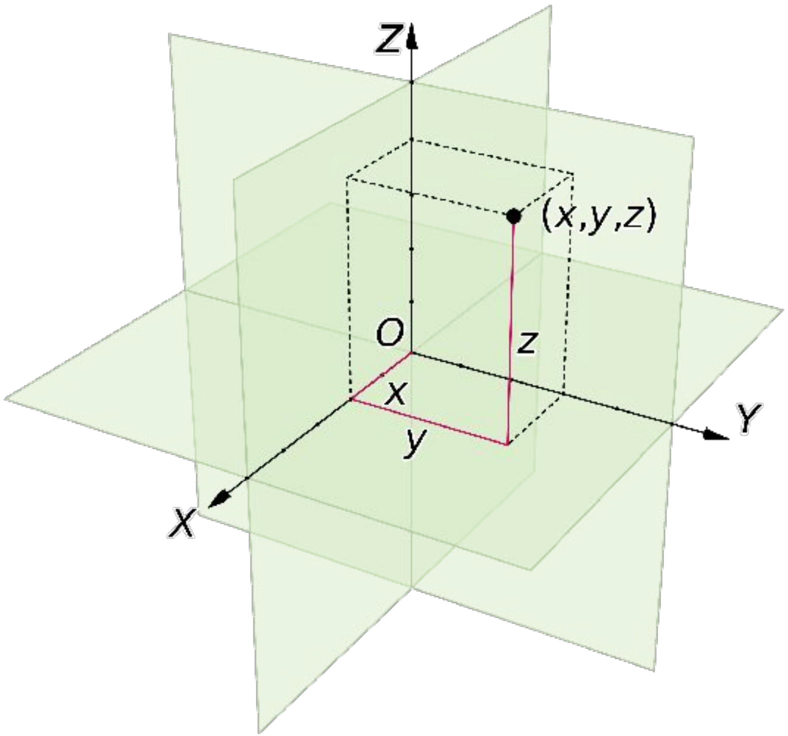
FC como problema de factorización

Cada **item** {1,2,...,m} se representa en un espacio de n características latentes mediante el vector

$$\begin{bmatrix} U_1^{(1)} & \cdots & U_n^{(1)} \end{bmatrix}$$

Análogamente, cada **usuario** {1,2,...,u} se representa en un espacio de n características latentes mediante el vector

$$\begin{bmatrix} V_1^{(1)} & \cdots & V_n^{(1)} \end{bmatrix}$$



Espacio de características latentes de n=3

Distancia entre películas, y entre usuarios

Una vez resuelto el problema del filtro cooperativo podemos estimar la **relación entre dos películas** i1 e i2 como su **distancia** (e.g. distancia euclídea):

$$||x^{(i1)}-x^{(i2)}||$$

Si se quiere proponer las 10 películas más relacionadas con una dada i, podemos calcular la distancia de i a todas las demás y filtrar las 10 **distancias más pequeñas**

De forma análoga se puede definir la **distancia entre usuarios** y calcular por ejemplo los 10 usuarios más próximos a uno dado (aquellos con distancia más pequeña)

Problemas del Filtro Colaborativo

- 💻**Cold Start**: la introducción de nuevos usuarios o nuevos productos genera problemas por la falta de datos para poder proponer recomendaciones
- 💻**Escalabilidad**: al aumentar el número de usuarios y/o productos aumentan el número de regresiones y regresores
  - E.g. ¿Cuántos regresores se requiere en el filtro cooperativo del ejemplo?
- 💻**Sinónimos**: dificultad de identificar productos idénticos
- 🙋**Sesgo** hacia la **popularidad**: aquellos productos con pocos datos pierden peso frente a los más populares (por razones históricas), impidiendo hacer buenas recomendaciones de productos novedosos

Problemas del Filtro Colaborativo

- 🙋 **Ovejas grises:** aquellos usuarios cuyas preferencias no suelen coincidir con la de la mayoría de los grupos aprendidos por el filtro cooperativo reciben malas recomendaciones, i.e. el sistema no tiene éxito haciendo coincidir los contenidos con las preferencias de un usuario.
  - El caso extremo es el usuario con preferencias únicas (**oveja negra**)
- 🙋 **Robustez**: cuando los usuarios tienen incentivos a modificar los rating para conseguir mejorar/empeorar productos la calidad del sistema se ve comprometida

Aplicaciones de filtros colaborativos

- http://www.last.fm/
- http://www.seethisnext.com/
- http://www.netflixprize.com/ (concurso entre 2006 y 2009 patrocinado por Netflix con premios de 1M$)