Sistemas de recomendación

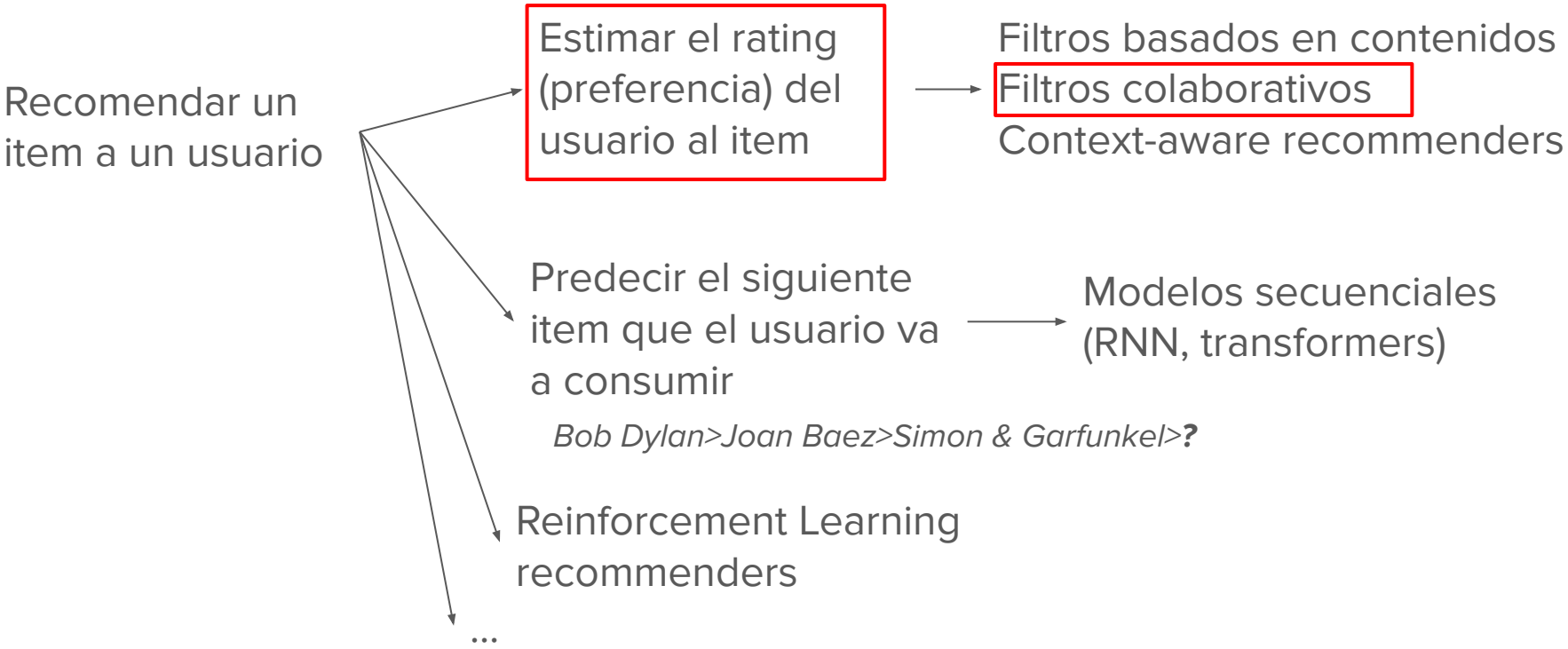
Resumen

Gestión de la Información
Grado en Ingeniería Informática
Universidad de Burgos
José Ignacio Santos

¿Por qué recomendar productos?

Muchas empresas e-commerce disponen de un catálogo de productos inmenso y necesitan proponer (filtrar) aquellos productos que a priori pueden interesar al cliente. Cuanto más se acerquen a sus preferencias mayores probabilidades de compra/?

¿Cómo recomendar?



Filtro basado en contenidos

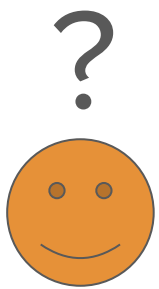
Content-Based Filtering (CBF)

Filtros basados en **contenidos** (CBF)

Supongamos un sistema de recomendación de películas. **¿Cómo recomendar?**

> Predecimos el rating o preferencia del usuario a un conjunto de películas y le proponemos las películas con mayor valoración

Un sistema de recomendación se preguntaría ¿qué rating pondría yo 😊 a la película IT?



Filtros basados en **contenidos** (CBF)

1) Supongo que existe una función que devuelve mi rating a la película

   $f_{😊}(IT) = y_{IT_{😊}}$ (rating)

2) ¿Cómo construyo esta función?

   Supongo que conozco un conjunto de características de las películas 📽️: romance, acción, ciencia ficción, humor, ...

   Asumo una función lineal

   $f_{😊}(📽️) = \theta_{😊_0} + \theta_{😊_{romance}}$ (romance$_{📽️}$) + $\theta_{😊_{acción}}$ (acción$_{📽️}$) +$\theta_{😊_{ciencia\ fiic.}}$ (ciencia fic.$_{📽️}$) + ...

   Los parámetros representan $\theta_{😊}$ mis preferencias hacia cada característica

   Recojo datos pasados de películas valoradas por mi y estimo los parámetros $\theta_{😊}$ (regresión)

Filtros basados en **contenidos** (CBF)

Datos pasados de mis valoraciones de otras películas

Película	Romance	Acción	Ciencia ficción	Humor	Duración (min)	Año	País:EEUU	Rating y_k😊
Love Actually	0.9	0	0	0.7	135	2003	1	3
Matrix	0	0.8	1	0.2	136	1999	1	4
Django Unchained	0.2	1	0	0.4	165	2012	1	4
Amélie	1	0	0	0.8	120	2001	0	2

Estimar θ😊 = encontrar los valores θ😊 que minimizan la función de costes

$$J(\theta_{😊}) = \sum(y_{k😊} - f_{😊}(x_k))^2$$

Filtros basados en **contenidos** (CBF)

1) Supongo que existe una función que devuelve mi rating a la película

   $f_{😊}(IT) = y_{IT_{😊}}$ (rating)

2) ¿Cómo construyo esta función?

   Supongo que conozco un conjunto de características de las películas 📽️: romance, acción, ciencia ficción, humor, ...

   Asumo una función lineal

   $f_{😊}(📽️) = \theta_{😊_0} + \theta_{😊_{romance}}$ (romance$_{📽️}$) + $\theta_{😊_{acción}}$ (acción$_{📽️}$) +$\theta_{😊_{ciencia\ fiic.}}$ (ciencia fic.$_{📽️}$) + ...

   Los parámetros representan $\theta_{😊}$ mis preferencias hacia cada característica

   Recojo datos pasados de películas valoradas por mi y estimo los parámetros $\theta_{😊}$ (regresión)

3) Calculo la predicción de rating para la película IT conocidas sus características

   $f_{😊}(IT) = \theta_{😊_0} + \theta_{😊_{romance}}$ (romance$_{IT}$) + $\theta_{😊_{acción}}$ (acción$_{IT}$) +$\theta_{😊_{ciencia\ fiic.}}$ (ciencia fic.$_{IT}$) + ...

Filtros basados en **contenidos** (CBF)

La principal **desventaja** del CBF es que se necesita definir y medir las características de todas las películas

¿Podemos hacer algo parecido pero sin tener que proponer y medir estas características?

Sí -> Filtro Colaborativo

Filtro colaborativos

Filtros Colaborativos

$Y = [\ y_{📽️😊}\ ]$

Modelos que utilizan **exclusivamente la matriz de ratings** de los usuarios

El adjetivo colaborativo se debe a que **aprendemos de los datos que los propios usuarios generan** al evaluar, puntuar o interactuar con los ítems, es decir, colaboran colectivamente en la construcción de la matriz de utilidad o matriz de ratings

No necesitamos conocer las características de los items (¡VENTAJA!)

Dos aproximaciones: basados en **modelos** y basados en **memoria**

Filtro colaborativos:

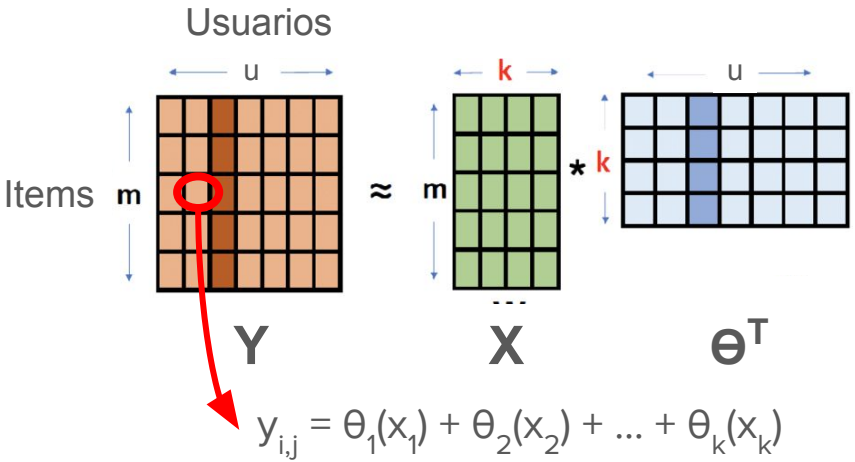
basados en **modelos**

Filtros Colaborativos basados en **modelos**

Descomponemos la matriz de ratings Y como **producto de dos matrices** X y Θ que capturan las características **latentes** de ítems (X) y de usuarios Θ

$$Y = X\Theta^t$$

Esto se llama **problema de factorización matricial de bajo rango**



k es un **parámetro** del modelo y representa el tamaño del espacio de características latentes

X y Θ son matrices que representan las películas (filas) y los usuarios (columnas) en dicho espacio. Podemos interpretarlas como matrices de características y de preferencias, aunque **sus valores no tienen el significado** que damos en el caso del CBF

Filtros Colaborativos basados en **modelos**

En los filtros colaborativos basados en modelos estimamos los parámetros (matrices X y Θ) minimizando una **función de costes** basada en MSE (enriquecida con los **términos de regularización** que evitan el overfitting)

$$J(\theta) = \sum\sum(y_{ij} - x_i\theta_j^T)^2 + \lambda\sum\sum(x_{ik})^2 + \lambda\sum\sum(\theta_{ik})^2$$

La función de costes es continua y derivable por lo que podemos utilizar técnicas de optimización basadas en el **descenso del gradiente**

Filtros Colaborativos basados en **modelos**

Existe 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 $X_i \cdot \theta_j^T$ sino que además sumamos un **sesgo global μ** (media de todos los ratings), un **sesgo por usuario b_j** y un **sesgo por ítem b_i**

$$y_{ij} = \mu + b_j + b_i + X_i \cdot \theta_j^T$$

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

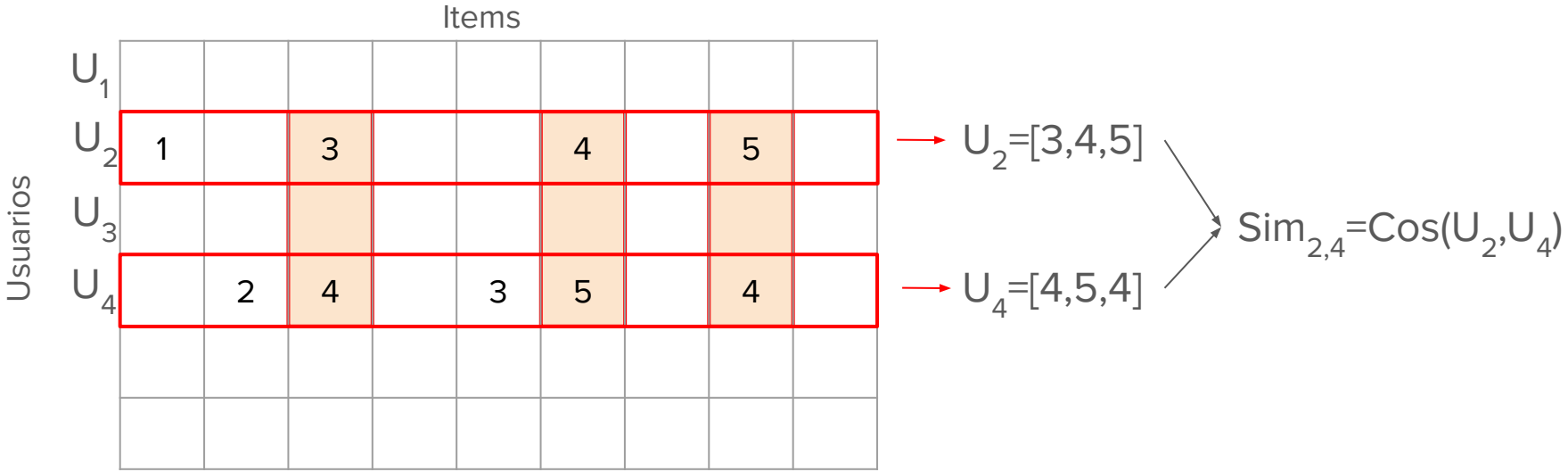
Filtro colaborativos:

basados en **memoria**

Filtros Colaborativos basados en **memoria**

No ajustamos ningún modelo, solo vamos a calcular **similitudes entre usuarios o productos** que utilizamos para estimar el rating

Por ejemplo, para la similitud entre usuarios:



Filtros Colaborativos basados en memoria