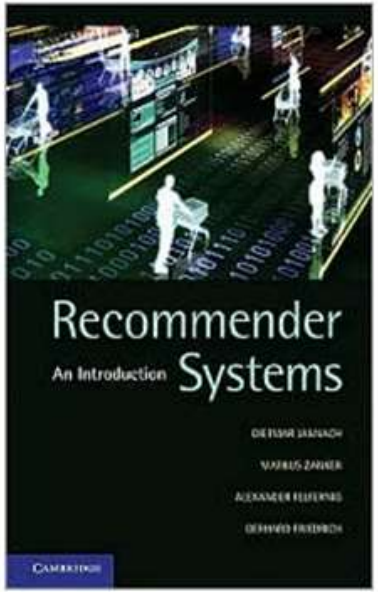
Universidad Popular del Cesar

Ciencia de Datos y BigData

Sistemas de Recomendación

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar



Recommender Systems: An Introduction

by Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich

AVERAGE CUSTOMER RATING:

(Be the first to review)

Gefällt mir Registrieren, um sehen zu können, was deinen Freunden gefällt.

FORMAT:
Hardcover

NOOKbook (eBook) - not available
Tell the publisher you want this in NOOKbook format

NEW FROM BN.COM

~~$65.00~~ List Price
$52.00 Online Price
(You Save 20%)

Add to Cart

NEW & USED FROM OUR

New starting at **$56.46** (You S
Used starting at **$51.98** (You S

See All Prices

Table of Contents

Customers who bought this also bought

Basado en el tutorial de Dietmar Jannach, Markus Zanker and Gerhard Friedrich en IJCAI 2013

http://ijcai13.org/files/tutorial_slides/td3.pdf

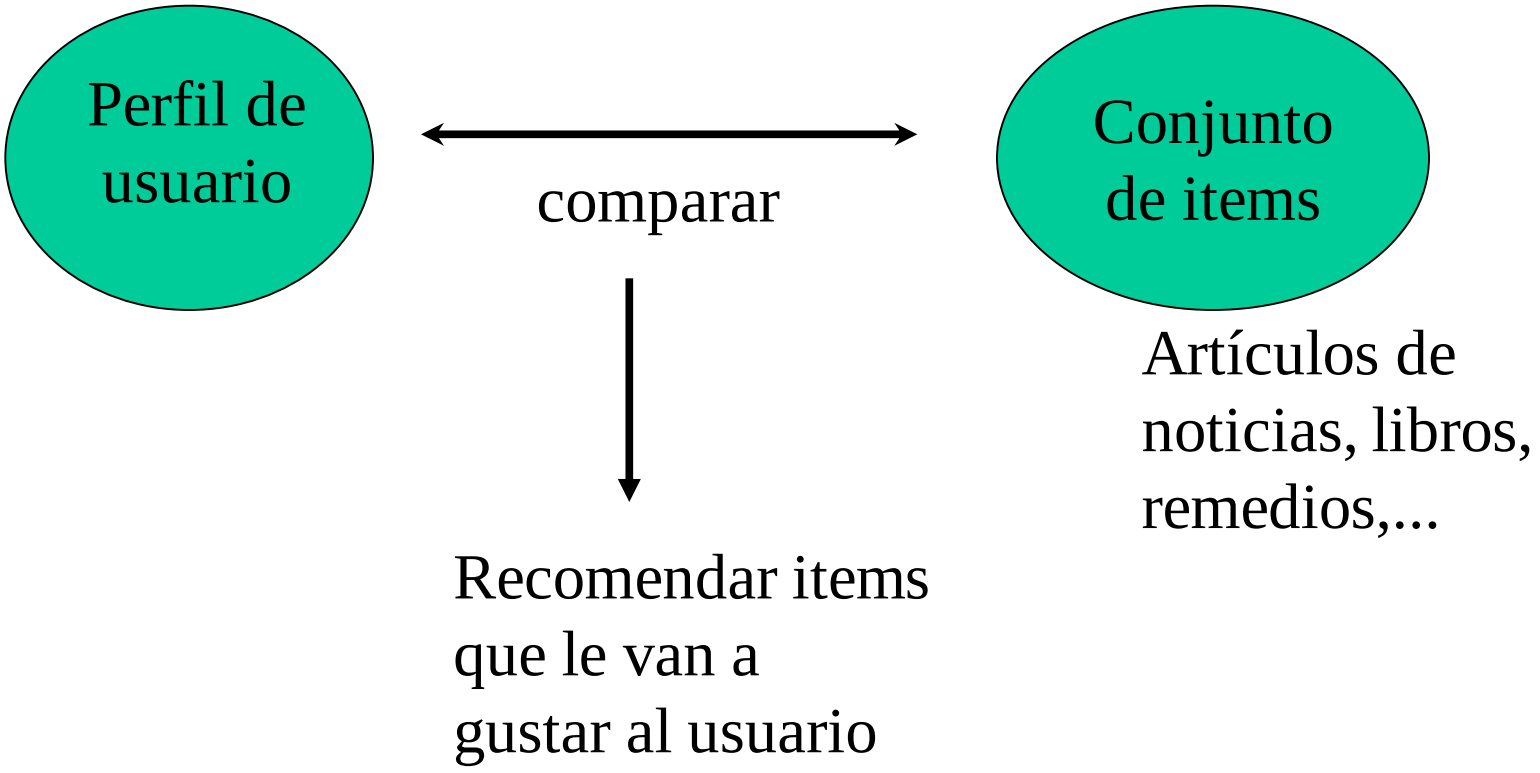
El excelente notebook de James Le

https://github.com/khanhnamle1994/movielens/blob/master/README.md

También para ver, el artículo de la Wikipedia sobre sistemas de recomendación

http://en.wikipedia.org/wiki/Recommender_system

La solución


Perfil de usuario
comparar
Conjunto de items
Artículos de noticias, libros, remedios,...
Recomendar items que le van a gustar al usuario

La solución

Dado un usuario, recomendarle productos que pueden interesarle

- Recomendar tratamientos de salud preventivos
- Personalizar cursos
- Recomendar películas, libros en la biblioteca, restaurantes, amigos, amores

Qué tenemos

- Muchos datos de comportamientos de las personas
- “Etiquetas” asociadas a esos comportamientos:
  - Likes
  - Buenos resultados al final de una secuencia (reinforcement learning!)
- Contenido (palabras, etiquetas)

Qué no tenemos

- Una buena abstracción sobre los datos
- Datos suficientes para items nuevos, raros, usuarios nuevos, perfiles únicos

Qué no tenemos

- Una buena abstracción sobre los datos
- Datos suficientes para items nuevos, raros, usuarios nuevos, perfiles únicos

Cómo lo solucionamos?

Qué no tenemos

- Una buena abstracción sobre los datos
- Datos suficientes para items nuevos, raros, usuarios nuevos, perfiles únicos

Cómo lo solucionamos?

Embeddings!

Qué no tenemos

- Una buena abstracción sobre los datos
- Datos suficientes para items nuevos, raros, usuarios nuevos, perfiles únicos

Cómo lo solucionamos?

Embeddings!

En este área se llama... matrix factorization

Qué no tenemos

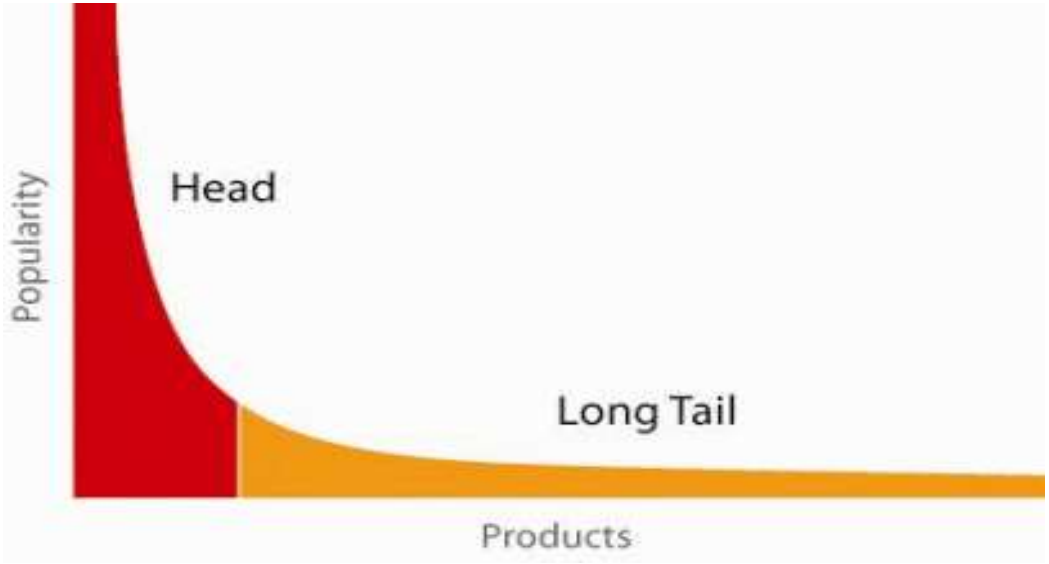
- Una buena abstracción sobre los datos
- Datos suficientes para items nuevos, raros, usuarios nuevos, perfiles únicos

Cómo lo solucionamos?

Embeddings!

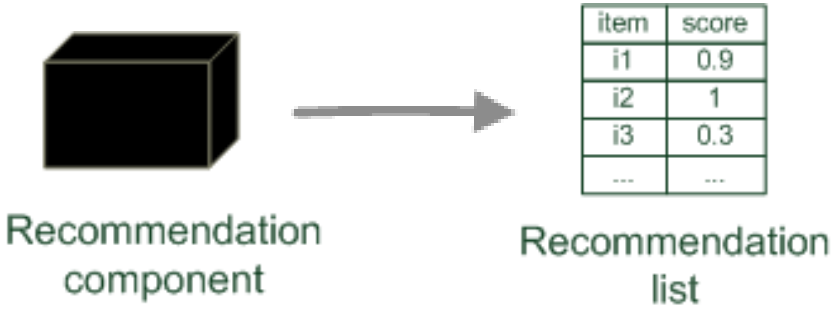
En este área se llama... matrix factorization

Problema del túnel: hay que explorar o buscar serendipia: recomendar items fuera del espacio de interés conocido

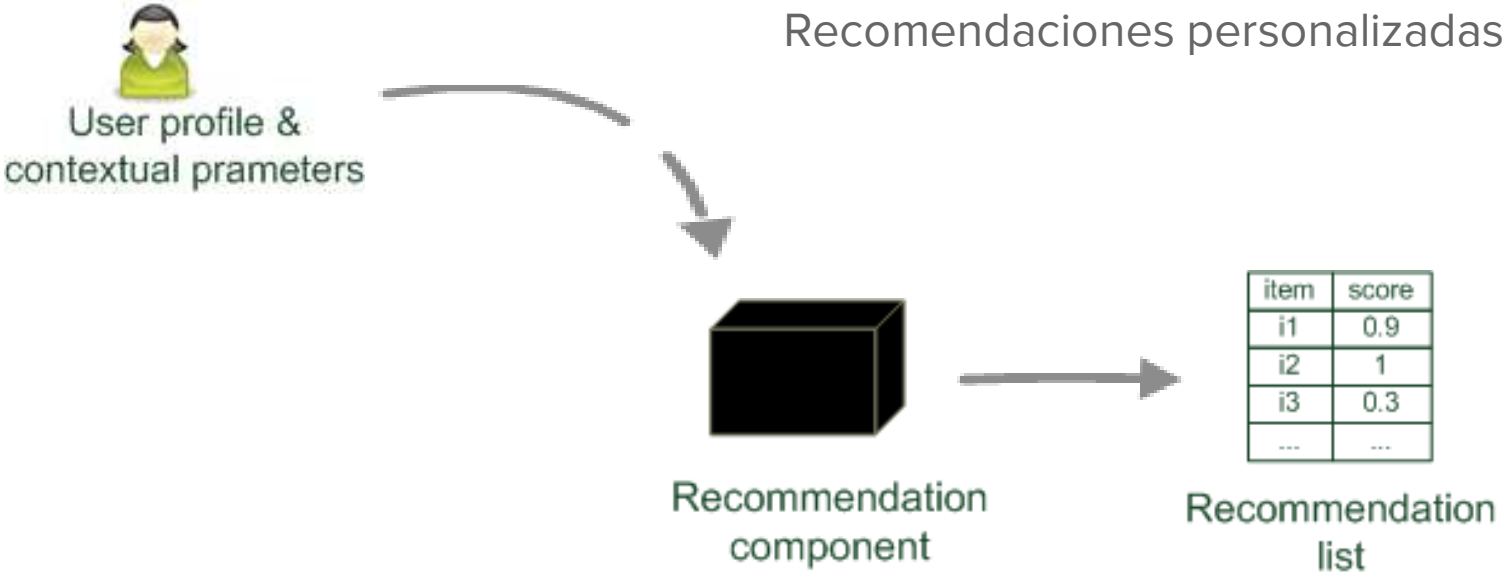

Popularity
Head
Long Tail
Products

Aproximaciones a recomendación

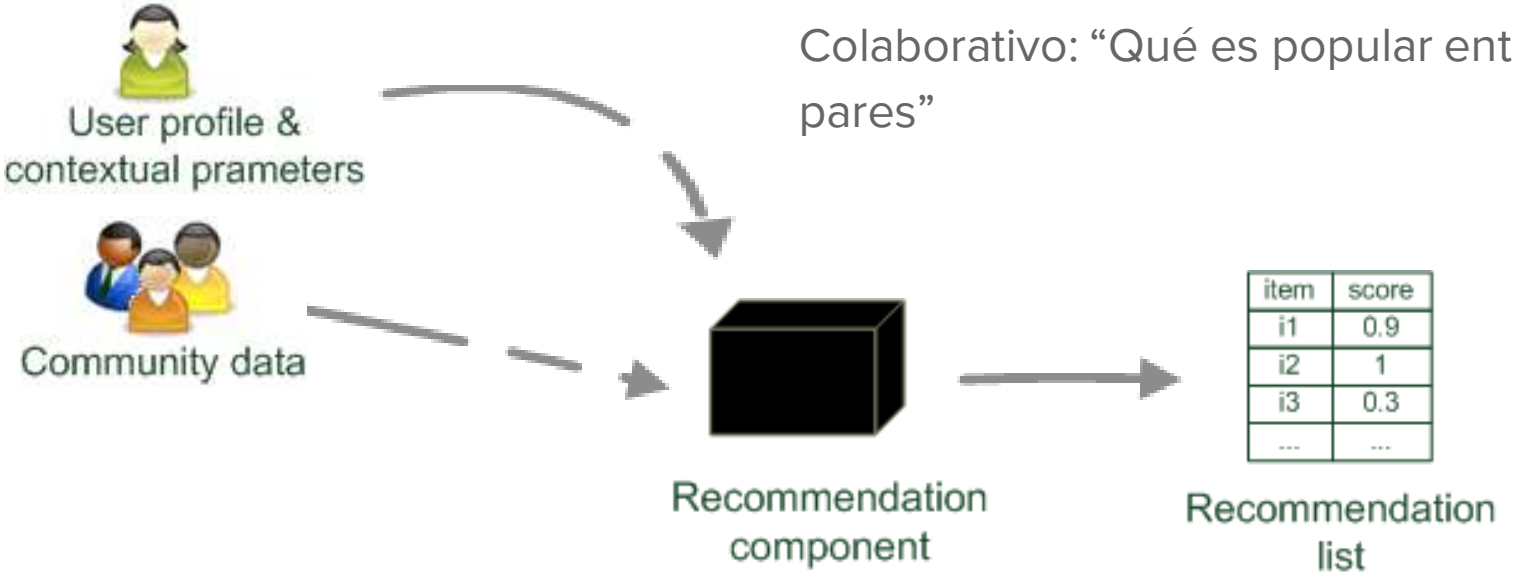
Reducir la información estimando relevancia



Aproximaciones a recomendación

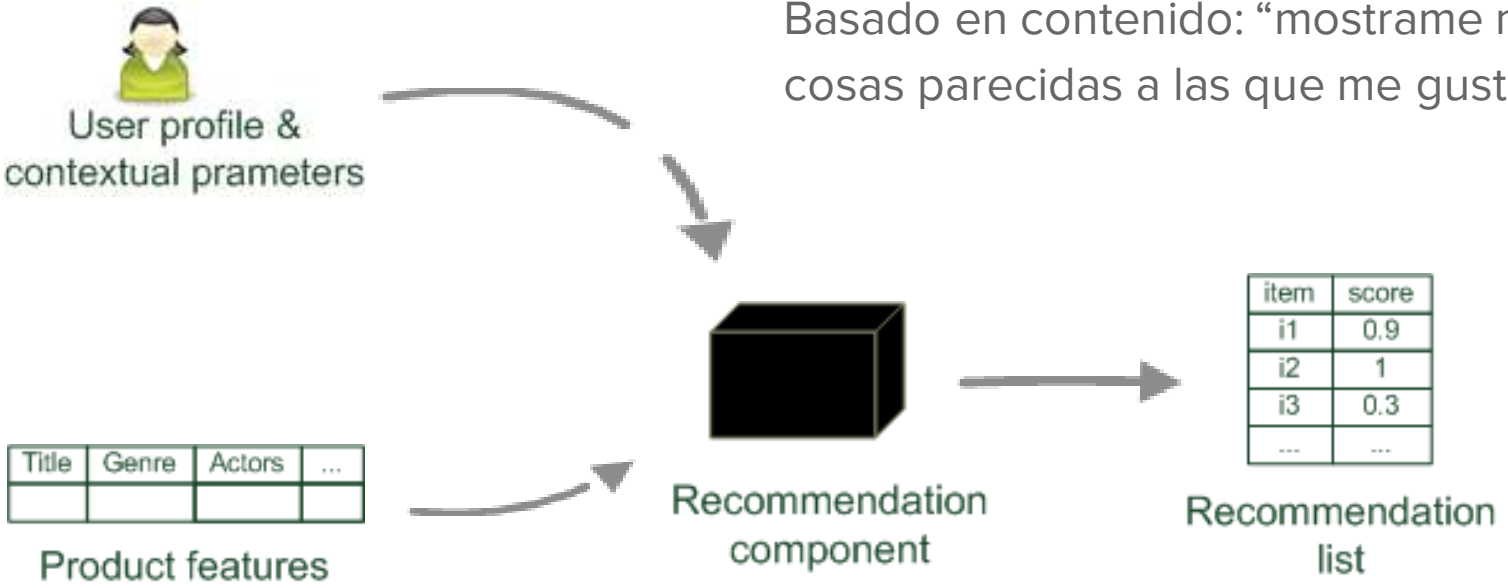


Filtrado colaborativo *(collaborative filtering)*



Colaborativo: “Qué es popular entre mis pares”

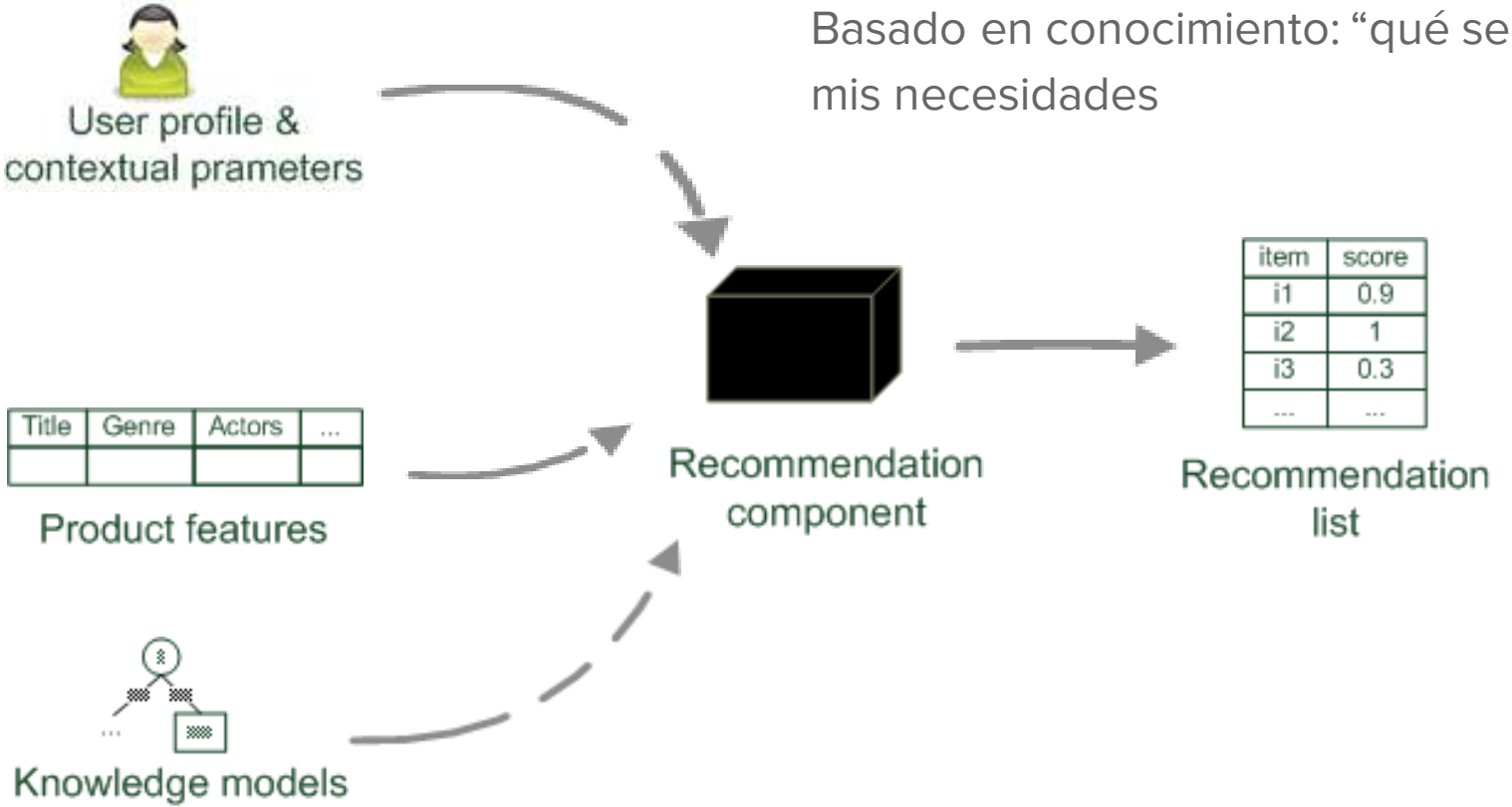
Basado en contenido



Basado en contenido: “mostrame más cosas parecidas a las que me gustan”

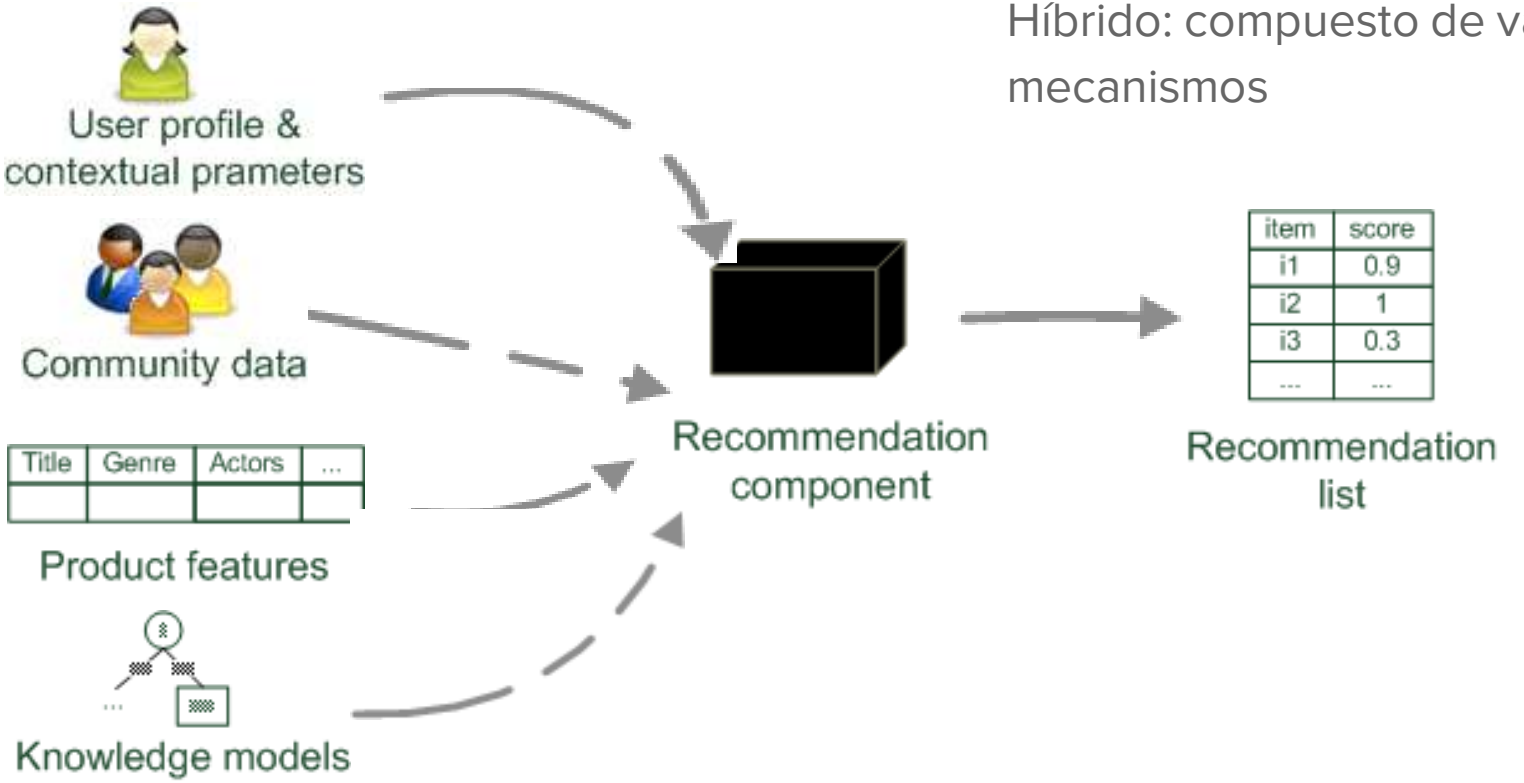
Basado en conocimiento

Basado en conocimiento: “qué se ajusta a mis necesidades



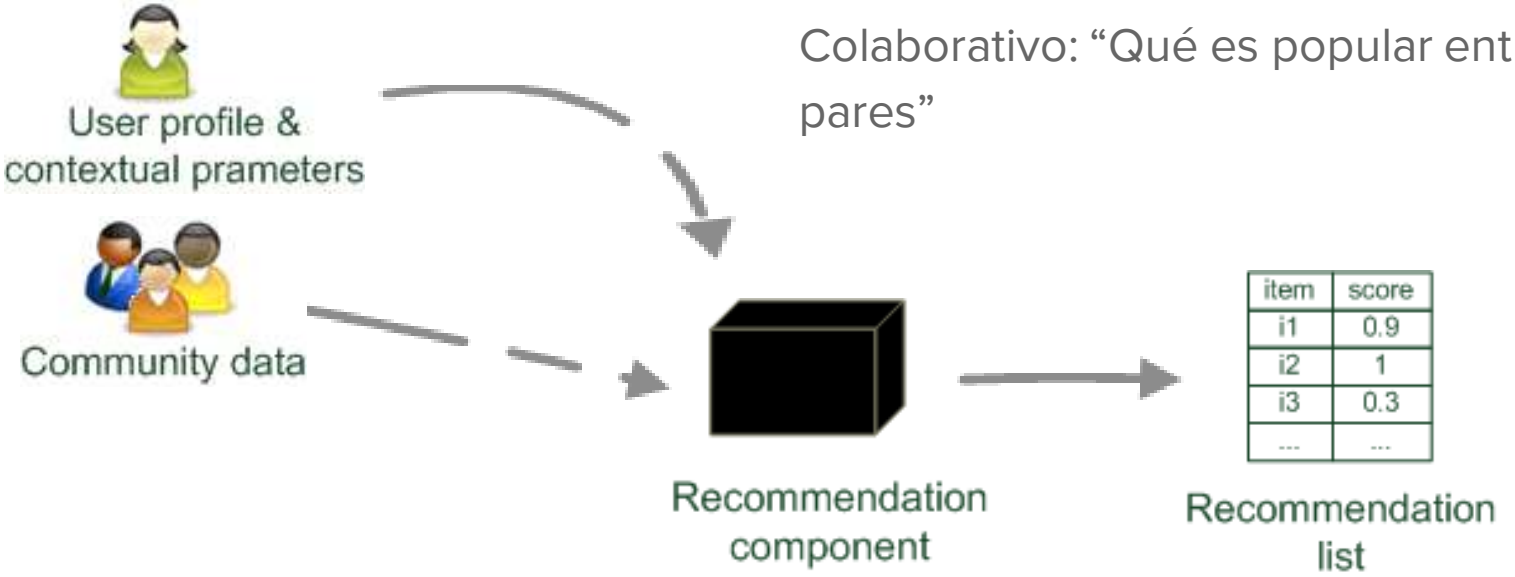
Híbrido

Híbrido: compuesto de varios mecanismos



Filtrado Colaborativo

Filtrado colaborativo *(collaborative filtering)*



Colaborativo: “Qué es popular entre mis pares”

Filtrado colaborativo

- La aproximación más usada para generar recomendaciones
- Bien conocida, con muchos algoritmos implementados y variantes
- Aplicable a muchos dominios

- Se usa la “sabiduría de las masas” (*"wisdom of the crowd"*)
- Se asume que los usuarios con gustos parecidos en el pasado tendrán usos parecidos en el futuro

Entrada - Salida

Input

Una matriz de relaciones usuario - item (valoraciones, compras, usos...)

Output

- Una predicción de cuánto le gustará un determinado item a un usuario
- Una lista de los items más recomendados

Entrada - Salida: ejemplo

Le gustará el item 5 a Alice?

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Recomendación por vecinos más cercanos

Dada una matriz usuarios - items, un usuario U y un item I que U no ha visto todavía

1. Encontrar los k usuarios más semejantes a U (*k-nearest-neighbors*)
2. Encontrar el valor que esos k usuarios han asignado a I
3. Darle a I un valor obtenido del valor asignado por los k usuarios

Parámetros:

- ¿Cómo determinar semejanza?
- ¿Qué número k de usuarios?
- ¿Cómo determinar el valor asignado a I? ¿Promedio? ¿Promedio pesado por distancia? ¿Heurística?

Semejanza es correlación

Correlación de Pearson para *collaborative filtering*

a, b: usuarios

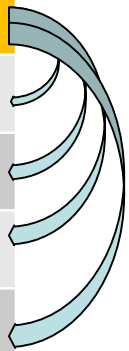
$r_{a,p}$: valoración del usuario *a* para el item *p*

P : conjunto de items valorados por *a* y por *b*

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

Semejanza es correlación

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1



sim = 0,85

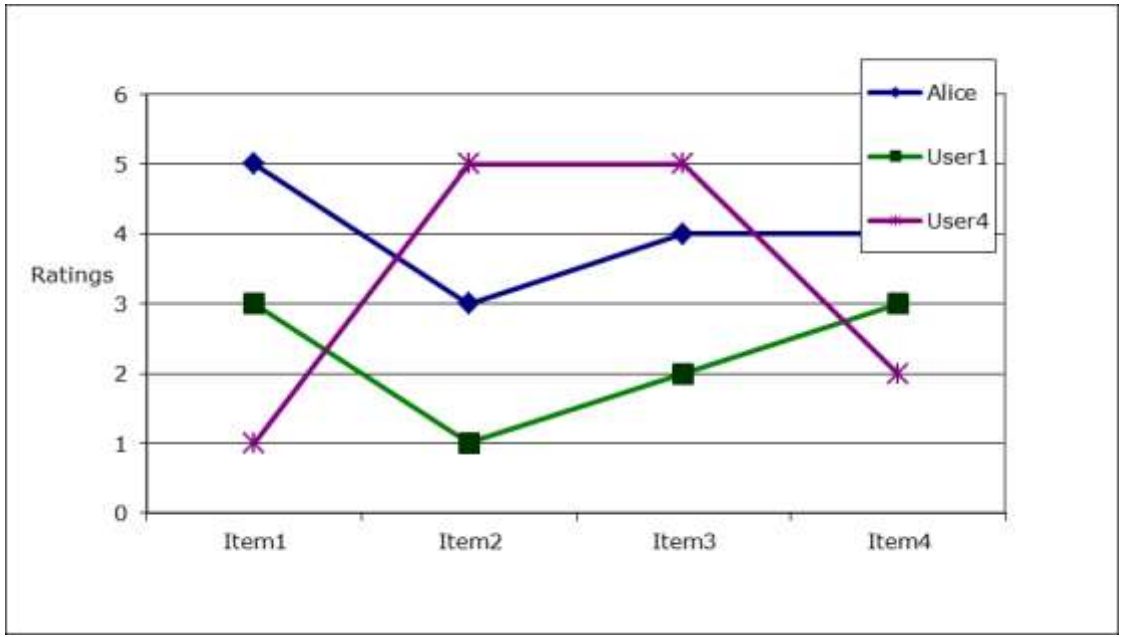
sim = 0,00

sim = 0,70

sim = -0,79

Semejanza es correlación

- Funciona mejor que el coseno
- Registra las diferencias en comportamiento de valoración



Recomendar a partir de la semejanza

- Una función de predicción común:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

1. Calcula si la valoración de los vecinos para el item *i* son más altas o más bajas que su promedio
2. Combina las diferencias de valoración con el promedio, pesándolas por su semejanza *a*
3. Sumar o restar el sesgo del vecino del promedio del usuario objetivo, y usar eso como una predicción

Mejorar la recomendación

- No todas las valoraciones valen lo mismo: estar de acuerdo en items controvertidos es más informativo que estar de acuerdo en items con acuerdo general

  Solución: dar más peso a items con mayor varianza en sus valoraciones

- Valorar el número de items co-valorados, usando "*significance weighting*", por ejemplo, reduciendo el peso cuando el número de items co-valorados es bajo

- Aumentar el peso de los vecinos más cercanos (*case amplification*)

- Seleccionar vecinos no por número sino por umbral de semejanza

Memoria vs. Modelo

Collaborative filtering basado en usuarios es "*memory-based*"

- Se usa la matriz de valoraciones directamente
  - Entrenamiento poco costoso
  - Predicción muy costosa
- No escala para la mayor parte de escenarios del mundo real

Aproximaciones basadas en modelos

- Se aprende un modelo off-line, se reentrena periódicamente
- En tiempo de predicción se usa el modelo
- Cómo podría entrar clustering aquí?

Modelo basado en items

Una aproximación basada en modelos, con la idea básica de usar la semejanza entre items (y no entre usuarios)

1. Buscar items semejantes a Item5
2. Extrapolar la valoración de Alice para esos items para predecir la valoración para Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Semejanza para items: coseno

- Mejores resultados para filtrado item-a-item

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

- El coseno normalizado (*adjusted cosine similarity*)
  - Usa el promedio de valoraciones del usuario y transforma con eso las valoraciones originales
  - U: conjunto de usuarios que han valorado los items *a* y *b*

$$sim(\vec{a}, \vec{b}) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2} \sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Recomendar de semejanza entre items

- Una función de predicción común:

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$

Cómo lo hacemos escalar

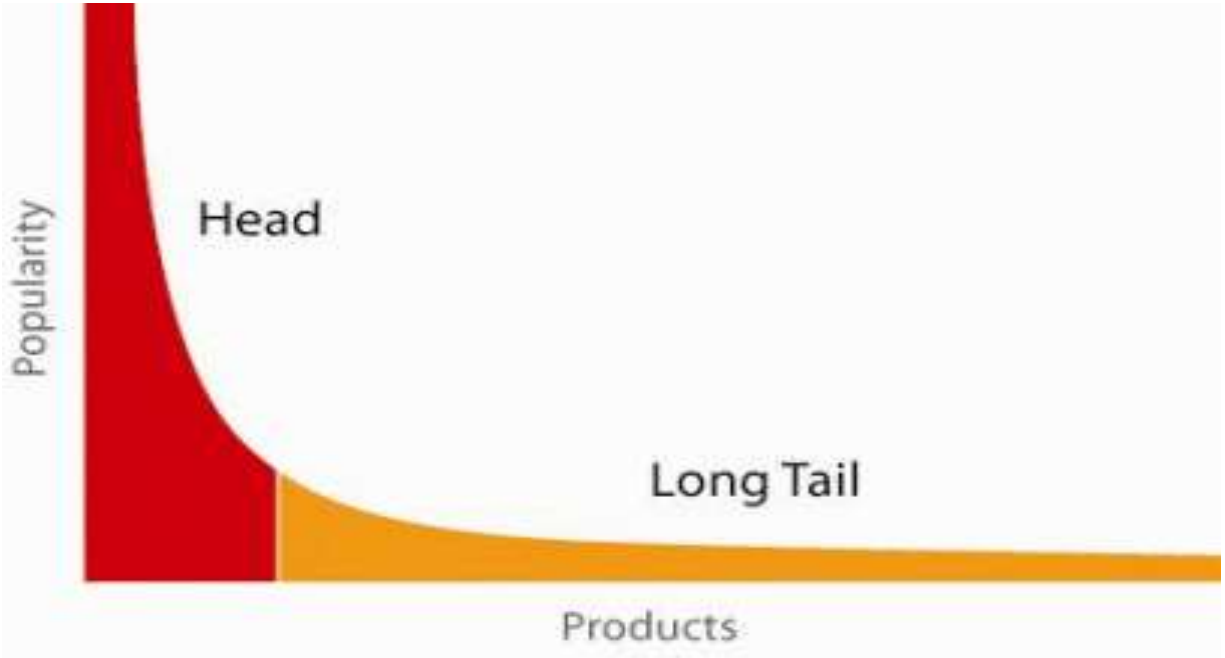
Basarse en items solamente no garantiza escalabilidad

- Pre-procesar todas las semejanzas entre items off-line
- Más estables que las semejanzas entre usuarios
- Más densas, aunque sigue habiendo muchos items sin co-valoraciones
- La vecindad activa en tiempo de predicción es bastante chica, porque sólo se usan items que el usuario haya valorado
- Se puede poner un umbral de número mínimo de co-valoraciones
- Usar clustering?

Valoraciones

- Muchos usuarios no quieren dejar valoraciones: cómo estimularlos
- Obtener valoraciones implícitas
  - Compras
  - Clicks, vistas de página, tiempo de permanencia en la pantalla, descargas
  - Visitas al médico, siniestros, pulsaciones por minuto, interacciones en redes sociales...
  - Pero ¿estamos interpretando correctamente el comportamiento del usuario?
    - Compra no es lo mismo que satisfacción
    - Podemos comprar para otra persona (un regalo, un encargo)

Escasez de datos (data sparsity)



Escasez de datos (data sparsity)

Cold start

¿Cómo recomendar items nuevos? ¿Cómo recomendar a usuarios nuevos?

- Forzar a los usuarios a valorar un conjunto de items (con aprendizaje activo!)
- Combinar con otro método en el principio: basado en contenido, demográfico, no personalizado
- Aumentar la potencia de vecinos más cercanos, porque el conjunto de vecinos más cercanos puede ser demasiado chico: asumir transitividad de vecindades

Otros métodos basados en modelos

- Factorización de matrices (PCA)
- Reglas de Asociación
- Métodos probabilísticos (bayesianos)
- Clustering

- Se reduce el espacio
- Se hace más denso ➜ menos escasez de datos (*cold start, long tail*)
- Emergen las causas latentes
- Se pierde información ➜ podemos sacrificar más información cuanto más redundante sea la información, depende del dominio

Factorización de matrices

SVD: $M_k = U_k \times \Sigma_k \times V_k^T$

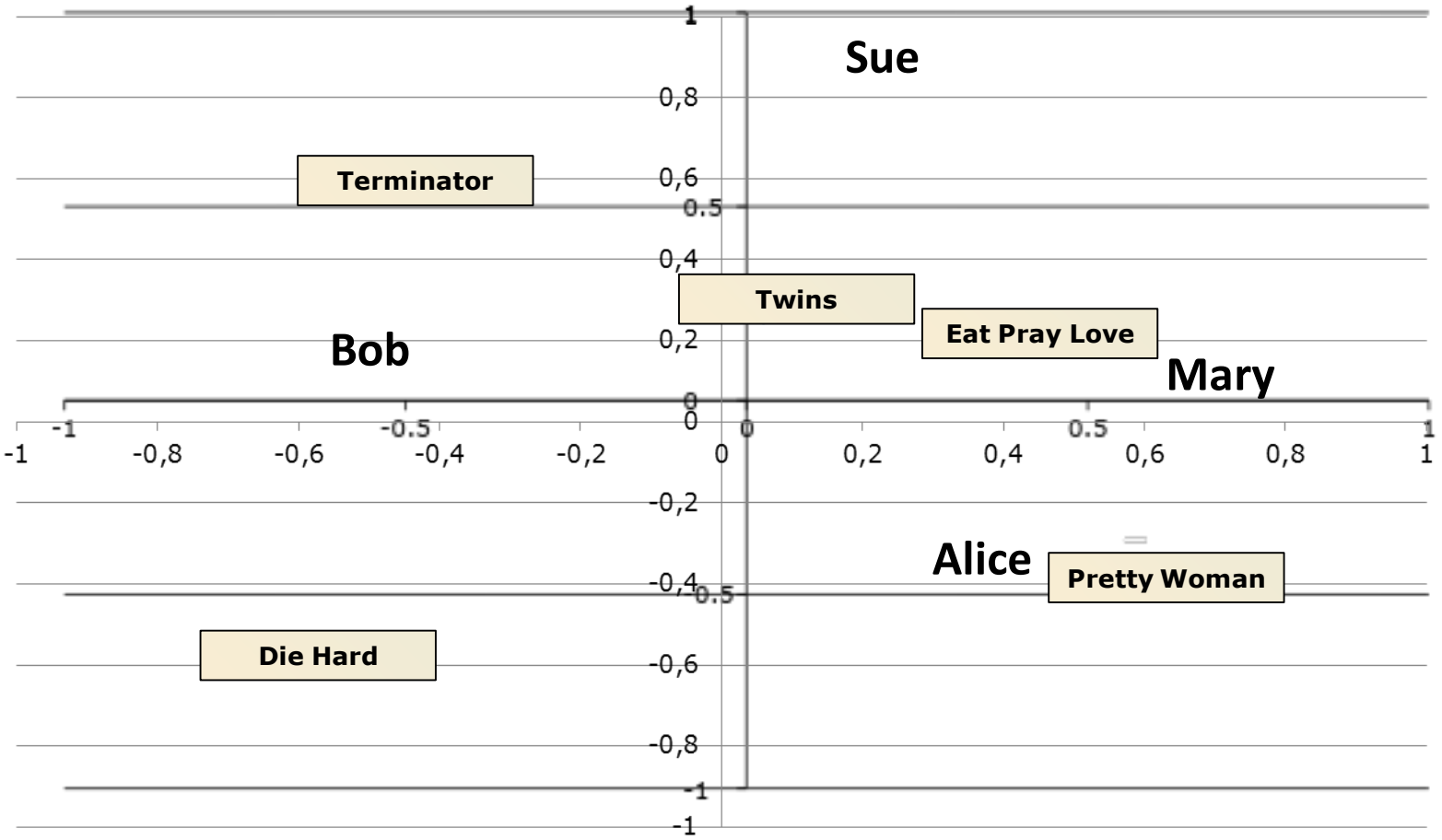
U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

V_k^T	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

Predicción: $\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$

= 3 + 0.84 = 3.84


Sue
Terminator
Twins
Eat Pray Love
Bob
Mary
Alice
Pretty Woman
Die Hard
1
0,8
0,6
0.5
0,4
0,2
0
0
-0,2
-0,4
-0.5
-0,6
-0,8
-1
-1
-0.5
0
0.5
1
-1
-0,8
-0,6
-0,4
-0,2
0
0,2
0,4
0,6
0,8
1

Reglas de asociación

1. Transformar valoraciones a binario (1 = por encima del promedio)

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	1	0	0	0	?
User 1	1	0	1	0	1
User 2	1	0	1	0	1
User 3	0	0	0	1	1
User 4	0	1	1	0	0

1. Minar reglas de
2. Reglas relevant edente pero no el consecuente
3. Ordenar por co

Métodos probabilísticos

A partir de la matriz de valoraciones, determinar la probabilidad de que un usuario valore positivamente un item, usando el Teorema de Bayes

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)} \qquad P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

Naïve Bayes: asumimos que las valoraciones son independientes!

Métodos probabilísticos

	Item1	Item2	Item3	Item4	Item5
Alice	1	3	3	2	?
User1	2	4	2	2	4
User2	1	3	3	5	1
User3	4	5	2	3	3
User4	1	1	5	2	1

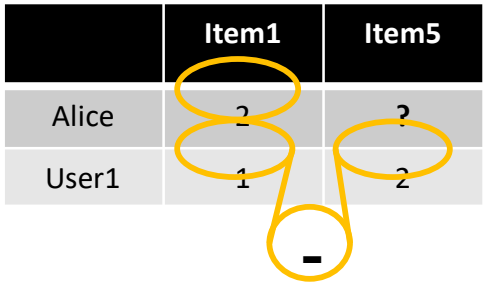
X = (Item1 =1, Item2=3, Item3= …)

$$P(X|Item5 = 1)$$
$$= P(Item1 = 1|Item5 = 1) \times P(Item2 = 3|Item5 = 1)$$
$$\times P(Item3 = 3|Item5 = 1) \times P(Item4 = 2|Item5 = 1)$$
$$= \frac{2}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \approx 0.125$$

$$P(X|Item5 = 2)$$
$$= P(Item1 = 1|Item5 = 2) \times P(Item2 = 3|Item5 = 2)$$
$$\times P(Item3 = 3|Item5 = 2) \times P(Item4 = 2|Item5 = 2)$$
$$= \frac{0}{0} \times \cdots \times \cdots \times \cdots = 0$$

Slope One

Diferencial de popularidad entre items



	Item1	Item5
Alice	2	?
User1	1	2

p(Alice, Item5) = 2 + (2-1) = 3

Obtener el promedio de estas diferencias en co-valoraciones para hacer la predicción

Evaluación

Error cuadrado (RMSE)

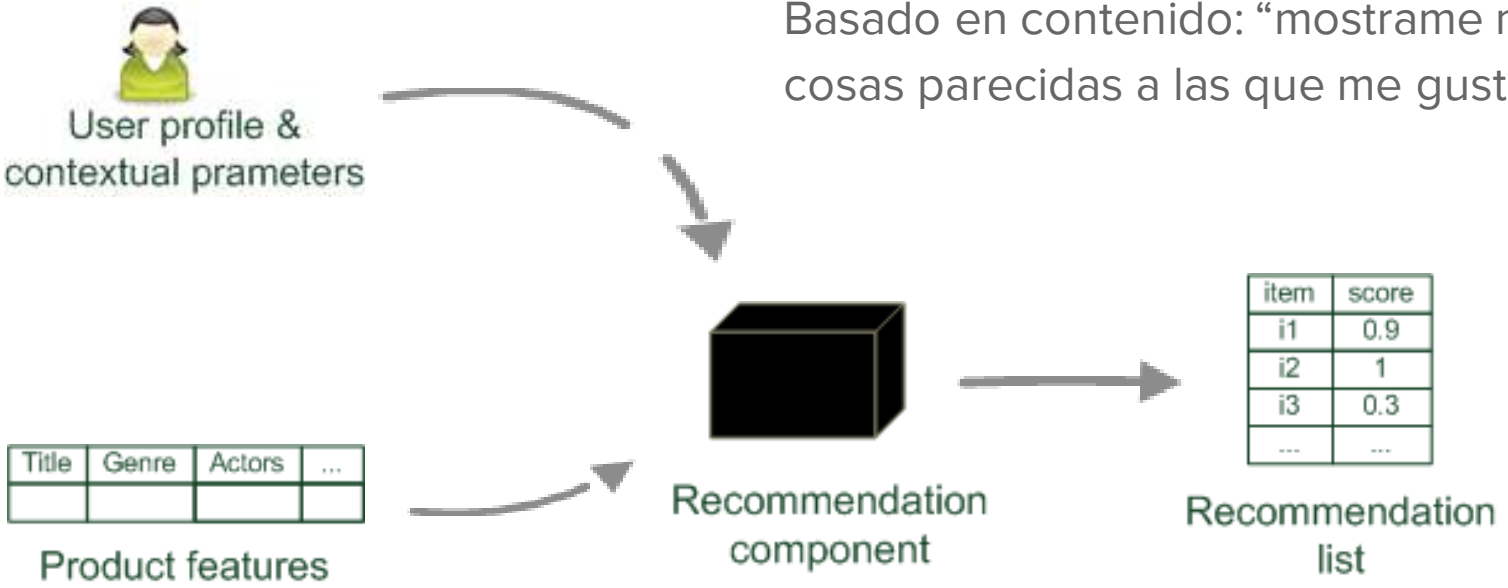
Fortalezas y limitaciones de collaborative filtering

- No supervisado, poco costoso: no requiere introducir información manualmente
- Bien conocido, con buenos resultados en muchos dominios

pero...

- Requiere valoraciones de los usuarios
- Sufre en los puntos de escasez de datos ➔ aplicar generalización
- No integrado con otras fuentes de conocimiento, sin explicación de resultados
- Es un fenómeno complejo que estamos reduciendo, hay que ver cómo incorporar otros factores como tiempo, costo, sorpresa, etc.

Basado en contenido



Basado en contenido: “mostrame más cosas parecidas a las que me gustan”

Fortalezas de basado en contenido

Items con pocas valoraciones, usuarios con pocas valoraciones

> Se buscan items / usuarios semejantes basándose en características, no en comportamiento (contenido, género, características demográficas)

- Necesitamos información sobre los items
  - Categoría
  - Contenido
- Proyectar el perfil del usuario en esa nueva información

Contenido de los items

Los documentos textuales son una buena metáfora

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

Representación de contenido, semejanza

Representación de los items

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

Perfil del usuario

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

Representación de contenido, semejanza

Representación de los items

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective,
Into the Fire	Romance Suspens				n, murder,

Semejanza item - usuario: Solapamiento de palabras clave (Dice coefficient, coseno)

$$\frac{2 \times |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i)| + |keywords(b_j)|}$$

Perfil del usuario

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

Recomendar items por semejanza entre items

Para un item *i* no valorado por el usuario *U*

1. Encontrar items semejantes a *i* (por contenido, por categoría) ya valorados por el usuario
2. Usar las valoraciones de esos items para valorar *i*

Recomendación explicativa

Tomar los items valorados como ejemplos etiquetados y aplicarles un algoritmo de aprendizaje automático interpretable (árbol de decisión, reglas de decisión (RIPPER))

- Funciona bien para pocas características, bien generales
- Las reglas que se infieren se pueden integrar con conocimiento del dominio

Fortalezas y limitaciones de basado en contenido

- Útil para escasez de datos de comportamiento
- No se requiere comunidad, sino solamente al propio usuario ➜ privacidad

- Necesita algunos datos de partida (no sirve en *cold start* absoluto)
- Sobreajuste
- Las características pueden no representar bien el problema

Combinar estrategias

- Atacar el *cold start* usando elicitación de restricciones ➔ basado en active learning?
- Usar recomendación basada en contenido cuando tenemos poca información de comportamiento (item nuevo, usuario nuevo), o cuando no se quiere compartir la información de comportamiento
  - ¿en qué momento cambiar de estrategia?
- Usar PCA para atacar la escasez de datos y evitar overfitting
- Usar clustering para escalabilidad y alcanzar serendipia

Veamos algunos ejemplos

https://github.com/nakulcr7/recommender-system-instacart