Empresas dirigidas por datos: Amazon.com

Amazon killed Borders and RadioShack

Empresas dirigidas por datos: Netflix.com

Netflix killed Blockbuster

Recomendaciones?

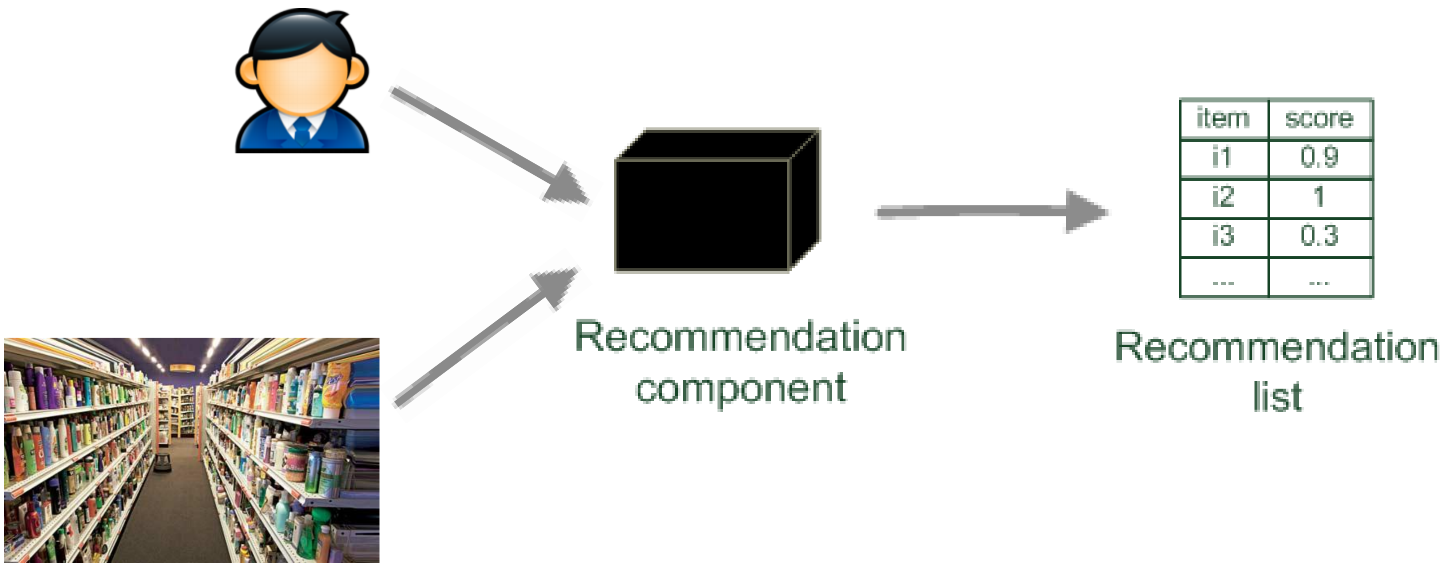
Recomendación Vs Búsqueda

Predicción de la toma de decisiones

sugerencia personalizada

El problema de la Recomendación

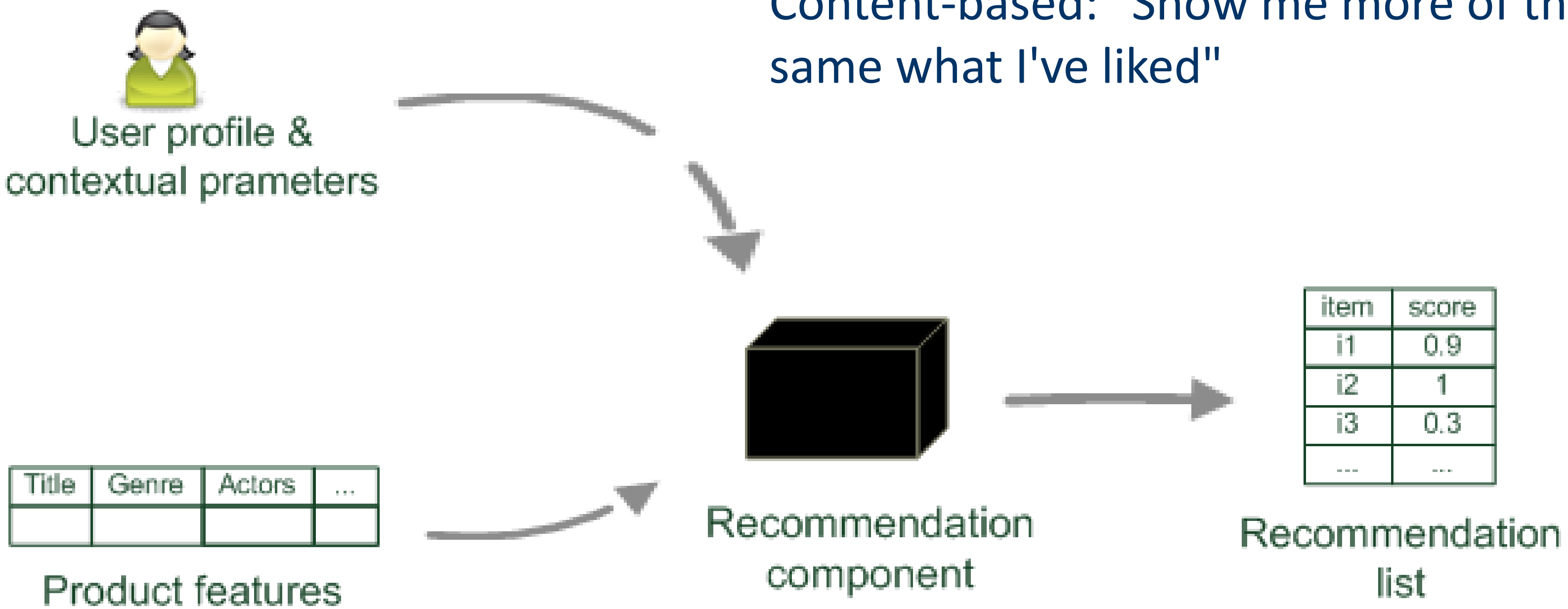
Problema: Predecir la utilidad de un item
Para un usuario!!!



Fuente: Recommender Systems: An introduction
(Cambridge University Press)

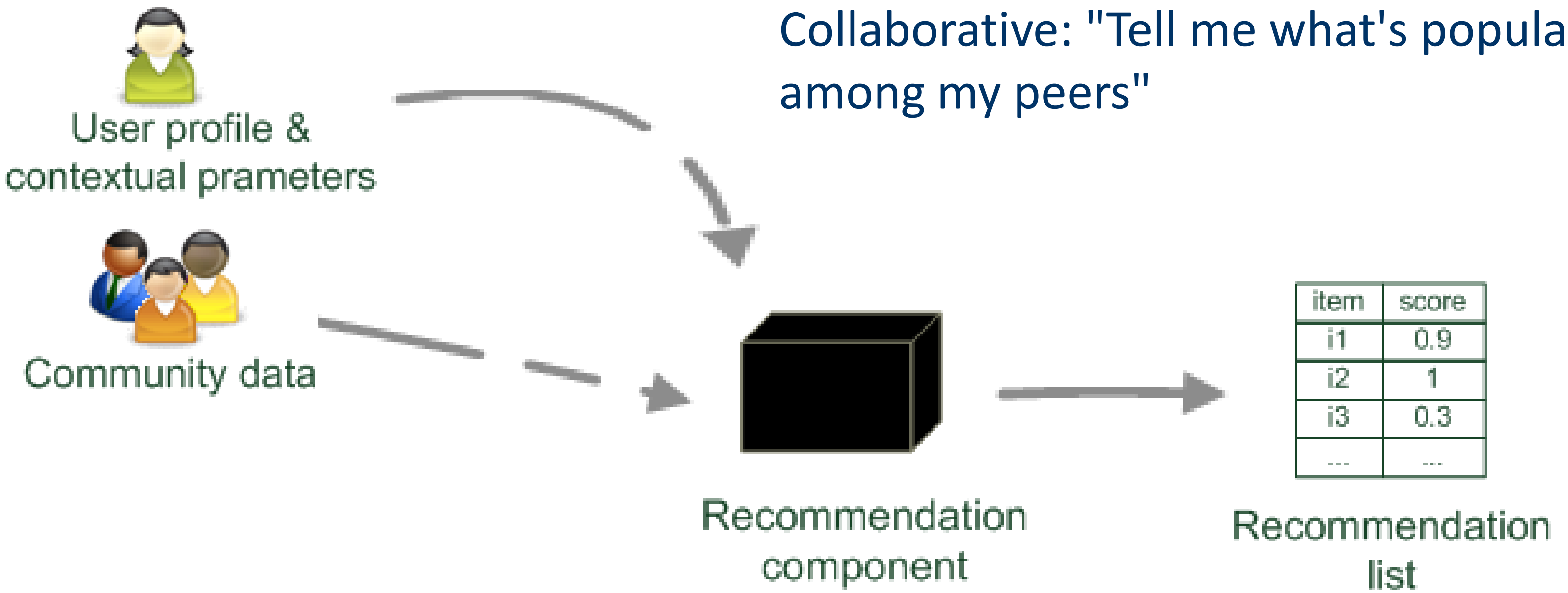
Estrategias de recomendación

Content-based: "Show me more of the same what I've liked"



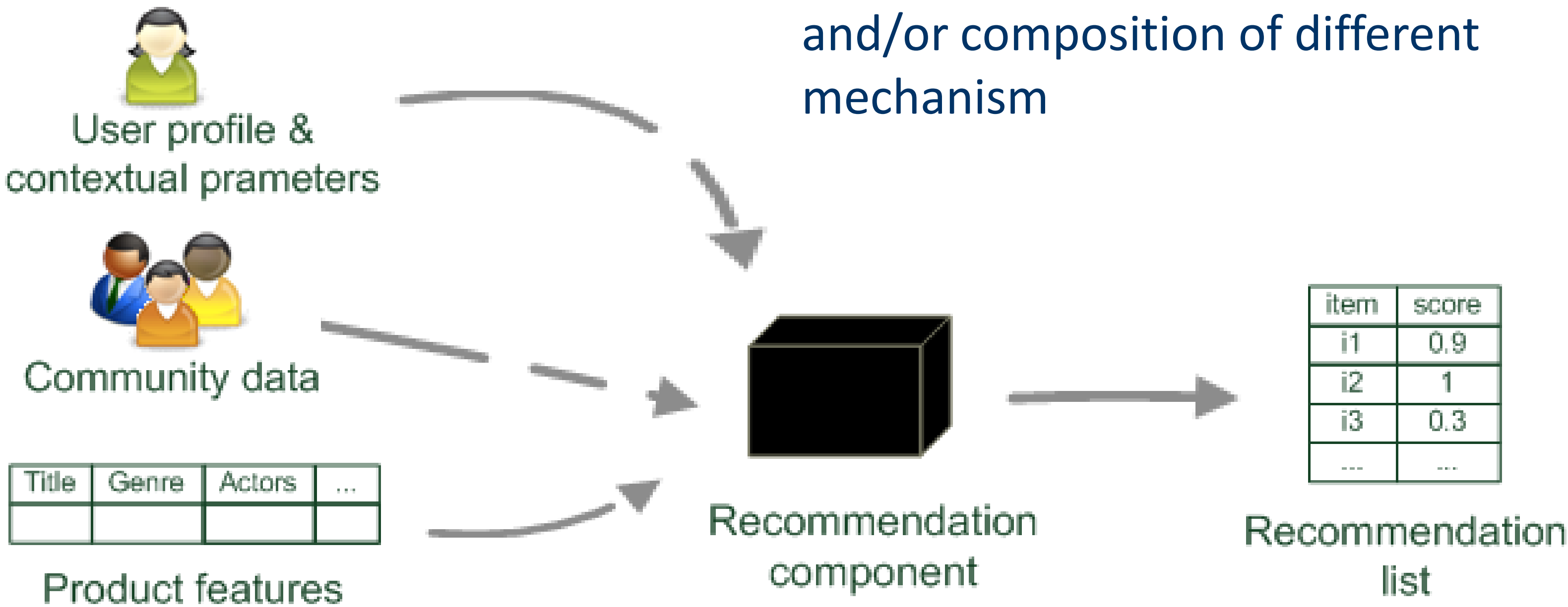
Estrategias de recomendación

Collaborative: "Tell me what's popular among my peers"



Estrategias de recomendación

Hybrid: combinations of various inputs and/or composition of different mechanism

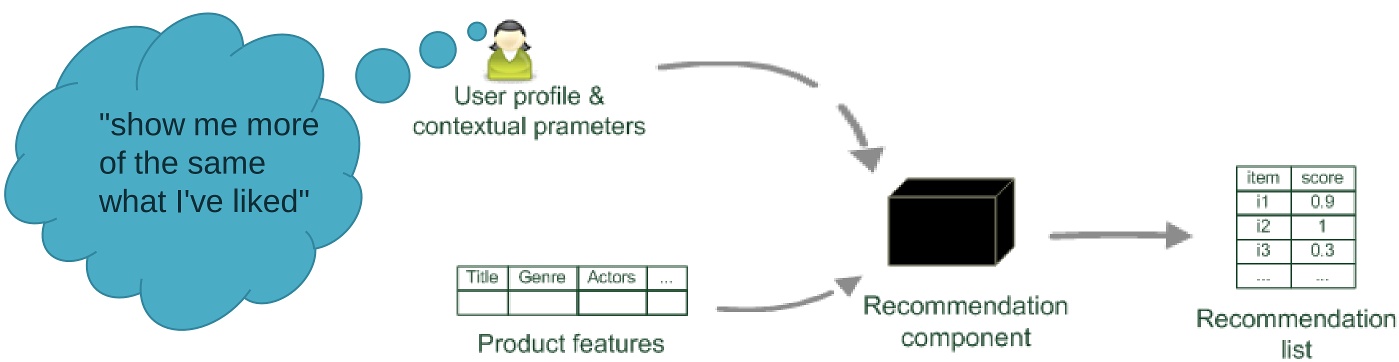


Recomendaciones en Amazon.com

Ejercicio: ¿Cómo genera amazon.com sus recomendaciones?

Recomendación basada en contenido

- What do we need:
  - some information about the available items such as the genre ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- The task:
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences



Recomendación y teoría de la Utilidad

¡ Score = Utilidad !

- ¿Cómo construimos la función de utilidad?:

1. Identificar los atributos relevantes de las alternativas
2. Identificar los valores de los atributos anteriores
3. Estimar las preferencias del individuo sobre los valores de los atributos.
4. Matemáticamente: Función lineal sobre los valores de los atributos. Dado individuo c y alternativa a:

$$u(c,a) = \sum_k \beta_{c,k}\, x_{a,k}$$

Con K indicando el cto de valores de todos los atributos de a.

Aprendizaje de preferencias (I)

Conjunto de alternativas

Instancias

Atributos

Producto	Autor	Fecha	Precio	Género	Utilidad
Libro1	Valor	Valor	Valor	Ficción	10
Libro2	Valor	Valor	Valor	Histórico	3
Libro3	Valor	Valor	Valor	Novela	7
LibroN	Valor	Valor	Valor	Novela	1

Juicio de utilidad

Técnica: Conjoint Analysis

Aprendizaje de preferencias (II)

Conjunto de datos (dataset)

Instancias | Atributos

Producto	Autor	Fecha	Precio	Género	Interacción
Libro1	Valor	Valor	Valor	Ficción	Comprado
Libro2	Valor	Valor	Valor	Histórico	Visitado
….	….	….	….	….	….
LibroN	Valor	Valor	Valor	Novela	Comentado

Inferencia de la Utilidad

Producto	Autor	Fecha	Precio	Género	Utilidad
Libro1	Valor	Valor	Valor	Ficción	10
Libro2	Valor	Valor	Valor	Histórico	5
LibroN	Valor	Valor	Valor	Novela	7

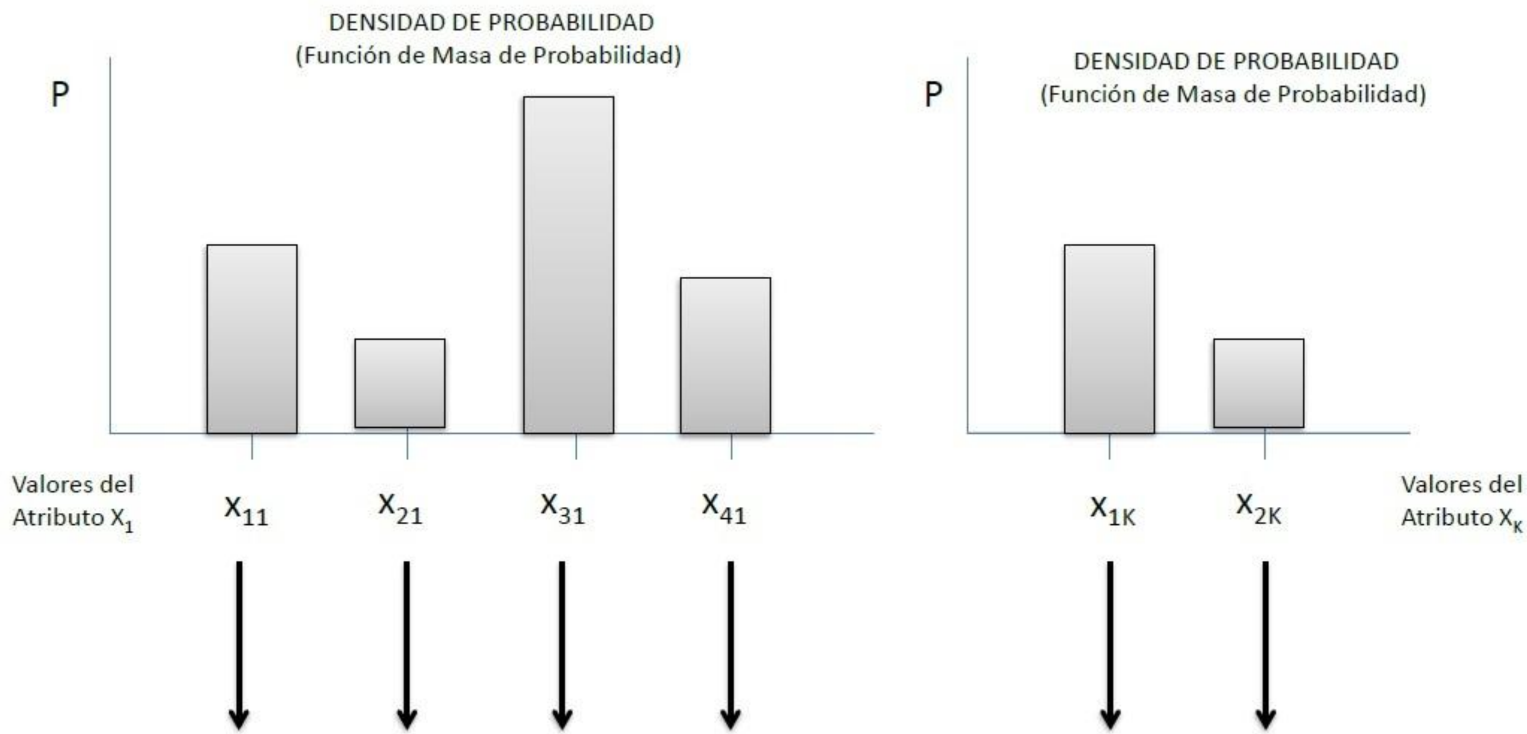
Aprendizaje de preferencias (III)

Conjunto de datos (dataset)

Instancias Atributos

Producto	Autor	Fecha	Precio	Género	Interacción
Libro1	Valor	Valor	Valor	Ficción	Comprado
Libro2	Valor	Valor	Valor	Histórico	No comprado
….	….	….	….	….	….
LibroN	Valor	Valor	Valor	Novela	Comprado

Aprendizaje de preferencias (III)

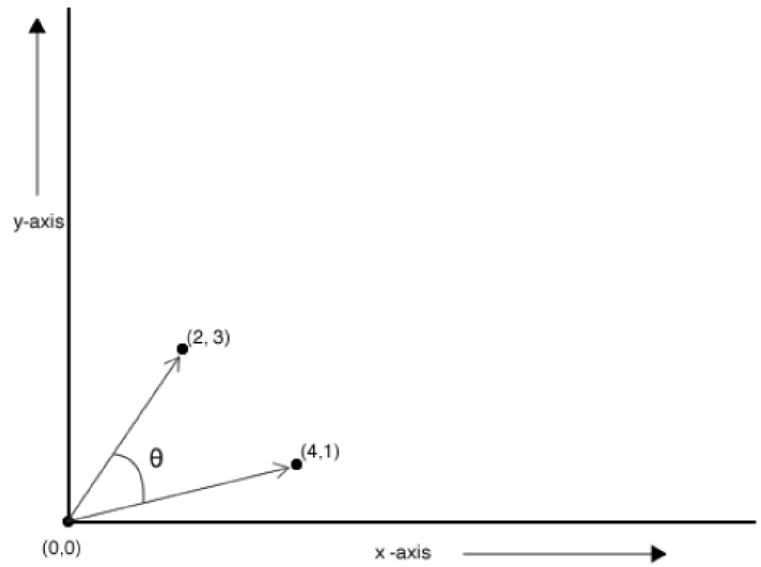


Vector de Utilidad = (u11 u21 u31 u41 u1K u2K)

||

Vector de Preferencias (B) = (β11 ……………………………………………………………… β2K)

Cálculo de utilidad en base a la similaridad

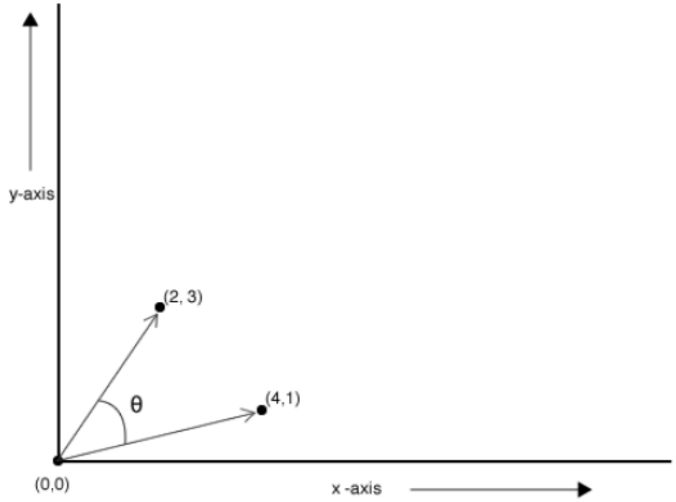


Medida del coseno:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Utilidad = Similaridad(B,X) = SimilaridadCoseno(B,X)

Cálculo de utilidad en base a la distancia



Distancia Euclidea entre dos vectores:

$$\delta(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\| = \sqrt{(\boldsymbol{a} - \boldsymbol{b})^T (\boldsymbol{a} - \boldsymbol{b})} = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2}$$

Utilidad = Similaridad(B,X) = 1 / (1 + d(B,X))

Recomendación de textos

- Most CB-recommendation techniques were applied to recommending text documents.
  - Like web pages or newsgroup messages for example.
- Content of items can also be represented as text documents.
  - With textual descriptions of their basic characteristics.
  - Structured: Each item is described by the same set of attributes

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

  - Unstructured: free-text description.

Recomendación de textos

- Item representation

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

- User profile

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

$keywords(b_j)$ describes Book b_j with a set of keywords

- Simple approach
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap (e.g. using the Dice coefficient)
  - Or use and combine multiple metrics

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

Recomendación de textos

Paso 1: Aprender el perfil del usuario (conjunto de términos clave preferentes)

Paso 2: Calcular la similaridad con el item (Tanimoto, Dice, etc.)

Paso 3: Clasificar el item en base a una función umbral:

Relevante? = Si, si Similaridad => R
Relevante? = No, en caso contrario

Term Frequency – Inverse Document Frequency

- Simple keyword representation has its problems
  - in particular when automatically extracted as
    - not every word has similar importance
    - longer documents have a higher chance to have an overlap with the user profile
- Standard measure: TF-IDF
  - Encodes text documents in multi-dimensional Euclidian space
    - weighted term vector
  - TF: Measures, how often a term appears (density in a document)
    - assuming that important terms appear more often
    - normalization has to be done in order to take document length into account
  - IDF: Aims to reduce the weight of terms that appear in all documents

Recomendación de textos

- **Given a keyword i and a document j**
- $TF(i,j)$
  - term frequency of keyword i in document j
- $IDF(i)$
  - inverse document frequency calculated as $\boldsymbol{IDF(i) = log\,\frac{N}{n(i)}}$
    - N : number of all recommendable documents
    - $n(i)$: number of documents from N in which keyword i appears
- $TF - IDF$
  - is calculated as: $\boldsymbol{TF\text{-}IDF(i,j) = TF(i,j) * IDF(i)}$

Recomendación de textos

- **Term frequency:**
  - Each document is a count vector in $\mathbb{N}^{|v|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	1.51	0	3	5	5	1
worser	1.37	0	1	1	1	0

Vector v with dimension $|v| = 7$

Example taken from http://informationretrieval.org

Recomendación de textos

- **Combined TF-IDF weights**
  - Each document is now represented by a real-valued vector of $TF\text{-}IDF$ weights $\in \mathbb{R}^{|v|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4					
Caesar	232					
Calpurnia	0					
Cleopatra	57					
mercy	1.51					
worser	1.37					

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Example taken from http://informationretrieval.org

Recomendación basada en contenido

Limitaciones:

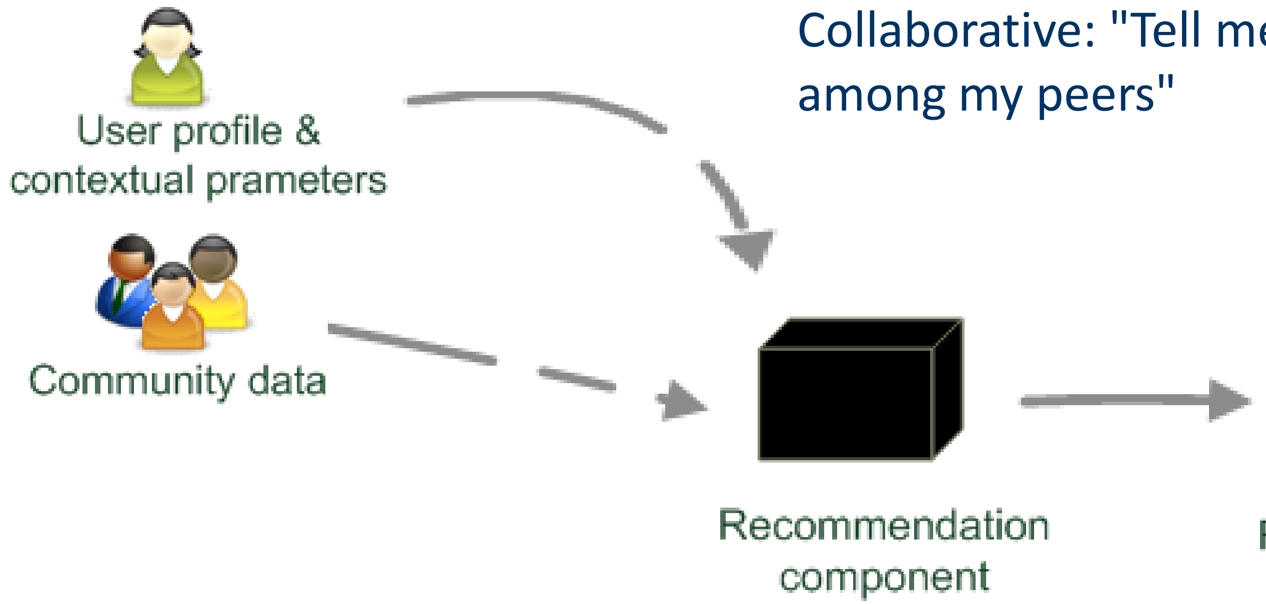
1. **Para el usuario:** Poca originalidad de las recomendaciones. Las recomendaciones suelen ser productos muy similares a los ya consumidos por el usuario.

2. **Para el ingeniero/científico:** Necesidad de conocer en detalle el dominio de la aplicación: productos, atributos y valores

3. **Para ambos:** Los valores de los atributos no aportan información acerca de la calidad del producto. Estimar la utilidad de un producto a través de sus atributos no garantiza la satisfacción o utilidad de la recomendación.

Recomendación colaborativa

Collaborative: "Tell me what's popular among my peers"



item	score
i1	0.9
i2	1
i3	0.3
...	...

Recomendación colaborativa

- The most prominent approach to generate recommendations
  - used by large, commercial e-commerce sites
  - well-understood, various algorithms and variations exist
  - applicable in many domains (book, movies, DVDs, ..)
- Approach
  - use the "wisdom of the crowd" to recommend items

- Basic assumption and idea
  - Users give ratings to catalog items (implicitly or explicitly)
  - Customers who had similar tastes in the past, will have similar tastes in the future
  - Users that rate items similarly have the same tastes

Recomendación colaborativa

- Input
  - Only a matrix of given user–item ratings
- Output types
  - A (numerical) prediction indicating to what degree the current user will like or dislike a certain item
  - A top-N list of recommended items

Items

Users	*1*	*2*	...	*i*	...	*m*
1	5	3		1	2	
2		2				4
:			5			
u	3	4		2	1	
:					4	
n			3	2		

a	3	5		?	1	

Recomendación colaborativa: user-based

- **The basic technique**
  - Given an "active user" (Alice) and an item i not yet seen by Alice
    - find a set of users (peers/nearest neighbors) who liked the same items as Alice in the past **and** who have rated item i
    - use, e.g. the average of their ratings to predict, if Alice will like item i
    - do this for all items Alice has not seen and recommend the best-rated
- **Basic assumption and idea**
  - If users had similar tastes in the past they will have similar tastes in the future
  - User preferences remain stable and consistent over time

Recomendación colaborativa: user-based

- Example
  - A database of ratings of the current user, Alice, and some other users is given:
  - Ejercicio: ¿Predicción del rating de Alice sobre el 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

Recomendación colaborativa: user-based

- Some first questions
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?

	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 colaborativa: user-based

- **A popular similarity measure in user-based CF: Pearson correlation**

  a, b : users

  $r_{a,p}$: rating of user a for item p

  P : set of items, rated both by a and b

  – Possible similarity values between -1 and 1

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

  – Ejercicio: Calcula la similaridad utilizando la correlación de Pearson y los datos anteriores

Recomendación colaborativa: user-based

- **A popular similarity measure in user-based CF: Pearson correlation**

  a, b : users

  $r_{a,p}$: rating of user a for item p

  P : set of items, rated both by a and b

  – Possible similarity values between -1 and 1

	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

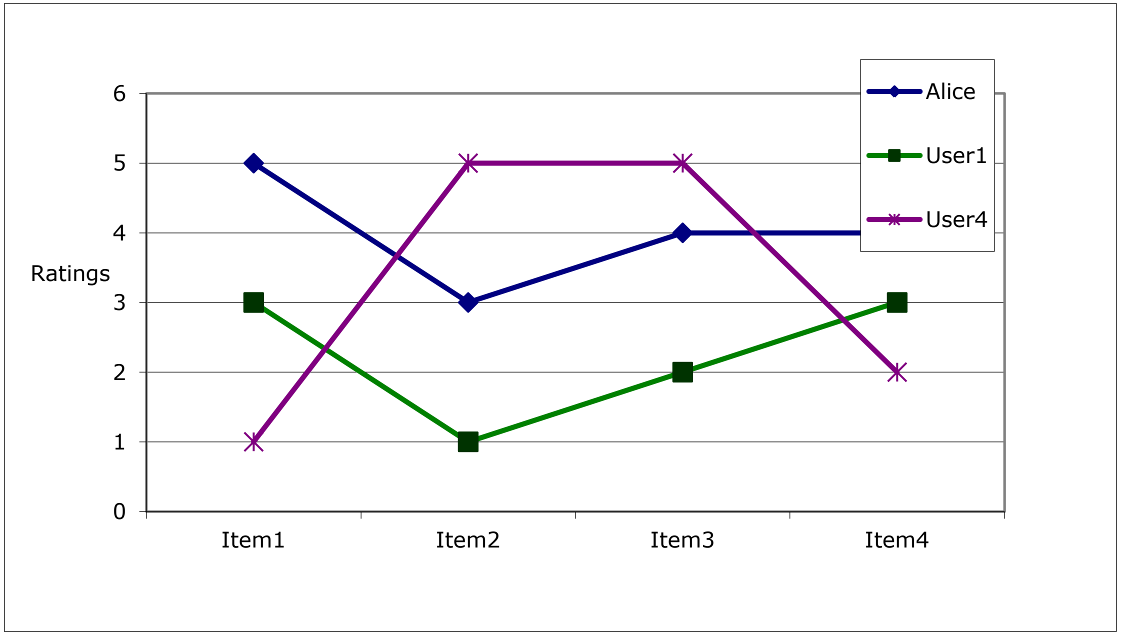
sim = 0,70

sim = 0

sim = -0,79

Recomendación colaborativa: user-based

- Ejercicio: ¿Otras formas de calcular la similaridad?
- Compara los resultados con la similaridad obtenida por correlación de Pearson



Recomendación colaborativa: user-based

- Media ponderada para predecir la valoración:

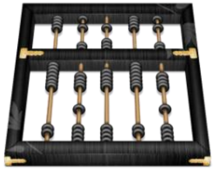
$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

- Wij = similaridad(i,j)

Recomendación colaborativa: user-based

- A common prediction function:

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



- Calculate, whether the neighbors' ratings for the unseen item i are higher or lower than their average
- Combine the rating differences – use the similarity with a as a weight
- Add/subtract the neighbors' bias from the active user's average and use this as a prediction

Recomendación colaborativa: user-based

- Not all neighbor ratings might be equally "valuable"
  - Agreement on commonly liked items is not so informative as agreement on controversial items
  - **Possible solution**: Give more weight to items that have a higher variance
- Value of number of co-rated items
  - Use "significance weighting", by e.g., linearly reducing the weight when the number of co-rated items is low
- Case amplification
  - Intuition: Give more weight to "very similar" neighbors, i.e., where the similarity value is close to 1.
- Neighborhood selection
  - Use similarity threshold or fixed number of neighbors

Recomendación colaborativa: item-based

- Basic idea:
  - Use the similarity between items (and not users) to make predictions
- Example:
  - Look for items that are similar to Item5
  - Take Alice's ratings for these items to predict the rating for 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

Recomendación colaborativa: item-based

- **Produces better results in item-to-item filtering**
- **Ratings are seen as vector in n-dimensional space**
- **Similarity is calculated based on the angle between the vectors**

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

- **Adjusted cosine similarity**
  - take average user ratings into account, transform the original ratings
  - U: set of users who have rated both items a and b

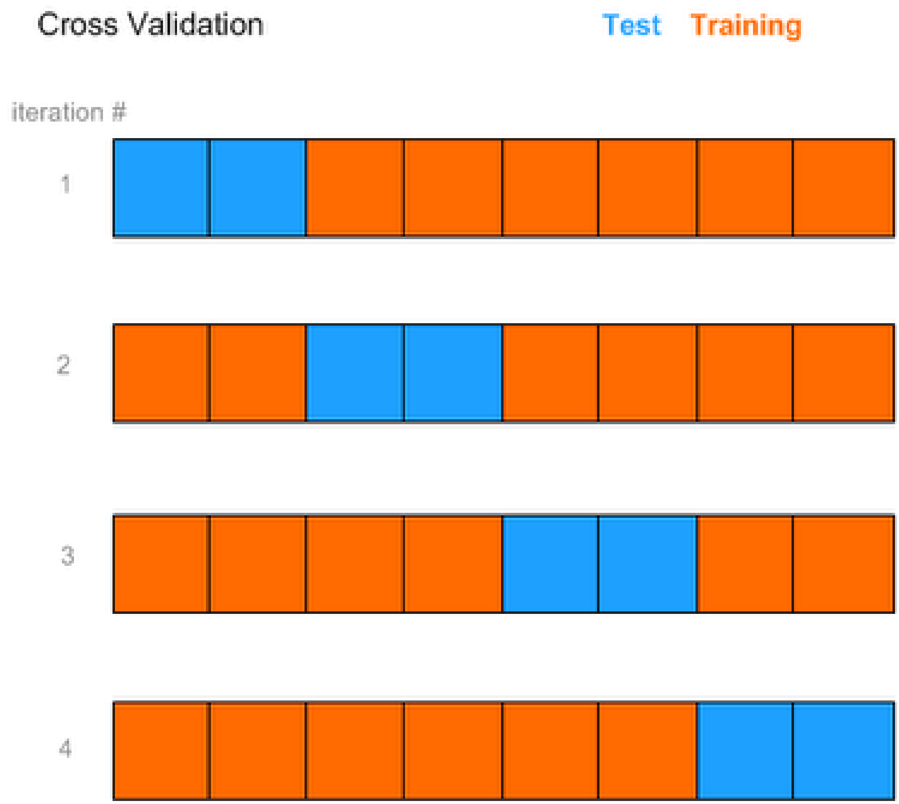
Recomendación colaborativa: item-based

- A common prediction function:

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

- Neighborhood size is typically also limited to a specific size
- Not all neighbors are taken into account for the prediction
- An analysis of the MovieLens dataset indicates that "in most real-world situations, a neighborhood of 20 to 50 neighbors seems reasonable" (Herlocker et al. 2002)

Validación de algoritmos



1. Validación cruzada con k subgrupos
2. Validación cruzada con 2 subgrupos
3. Validación dejando una instancia fuera (Leave one out)

Validación de algoritmos

MEAN SQUARED ERROR
(Error cuadrático medio)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - a \right)^2$$

MEAN ABSOLUTE ERROR
(Error absoluto medio)

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n} \sum_{i=1}^{n} |e_i|.$$

Problemas: la calidad de los datos

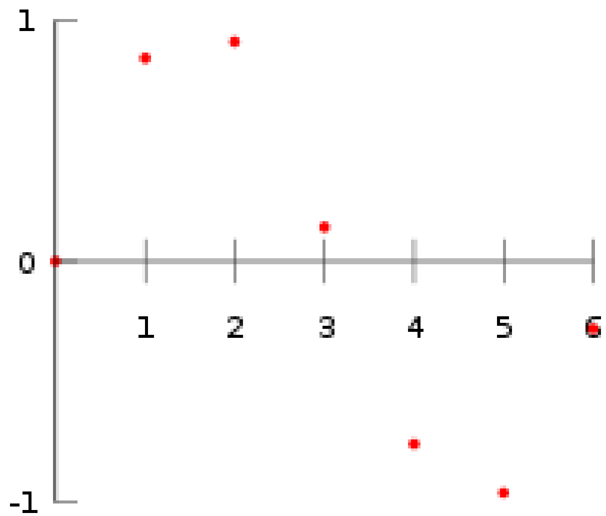
Id	Status	Age	Class
Training Set			
1	Single	20	Bad
2	Single	30	Good
3	Single	50	Bad
4	Single	60	Good
5	Married	20	Good
6	Married	30	Good
7	Married	40	Good
8	Married	50	Good
9	Divorced	40	Bad
10	Divorced	60	Good
Testing Set			
11	Single	40	(Bad)
12	Married	60	(Good)
13	Divorced	20	(Bad)
14	Divorced	30	(Bad)
15	Divorced	50	(Good)

1. Exactitud de los datos?

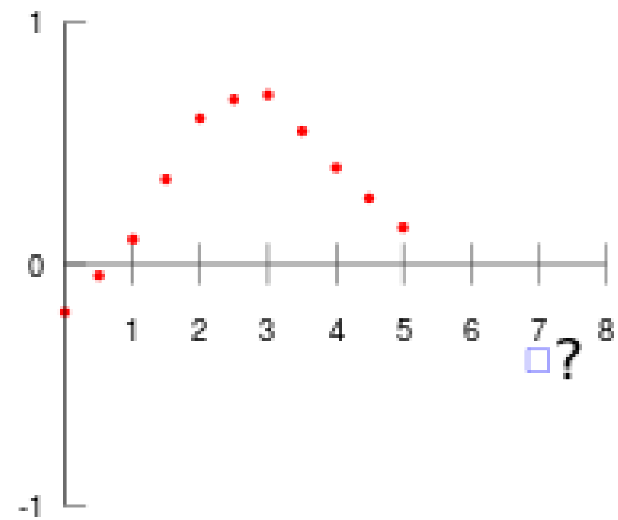
2. Imparcialidad del Testing Set Respecto al Training Set?

Problemas: La calidad de los datos

3. **Completitud de los datos**



Interpolación:
Situación deseable



Extrapolación:
Problema complicado

Sistemas predictivos:
Sistemas de Recomendación

Eduardo M. Sánchez Vila

eduardo.sanchez.vila@usc.es

CITIUS

Grupo de Sistemas Inteligentes

Universidad de Santiago de Compostela