

Sistemas de Recomendación

FIUBA - 75.06 - Noviembre 2016

Martín Ramos Mejía

Objetivo del sistema

Descubrir contenidos que sean interesantes para el usuario.

Cambio de Paradigma / Modelo:

- Búsqueda (Basada en el conocimiento del usuario)
- Recomendación (Descubrimiento, cosas que el usuario desconoce)

¿Cual es su relevancia?

- **“We are leaving the age of information and entering the age of recommendation”**
Chris Anderson, The Long Tail.
- **Netflix:** 2/3 de las películas vistas, son elegidas a través de una recomendaciones.
- **Amazon:** 35% de las ventas proviene de recomendaciones.

Características: Precisión

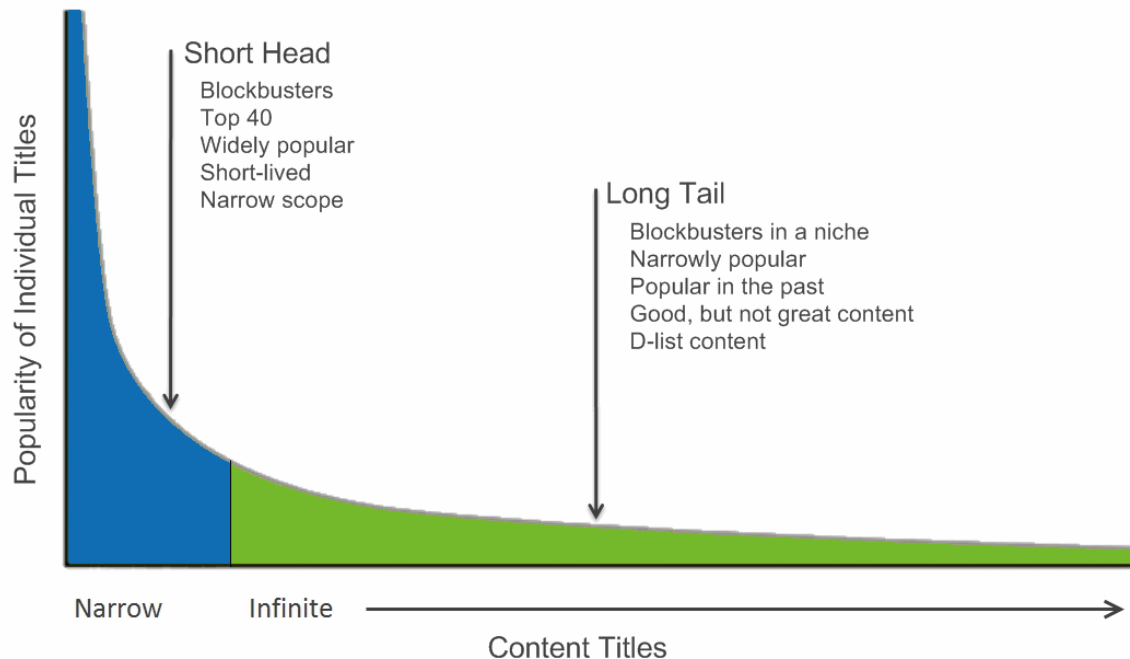
- Recomendarle contenidos al usuario que sean de su interés.
- De no ser de su interés el usuario ignorará el sistema.

Características: Serendipity

- No mostrarle al usuario contenidos que ya conoce o que puede descubrir por sus propios medios.
- El sistema debe ser capaz de recomendar contenidos que no son muy populares en general.

Serendipity: The Long Tail

The Long Tail



Características: Diversidad

No mostrar contenidos que son exclusivamente de un mismo tipo, a su vez dándole conciencia al usuario de la personalización.



Aún más desafíos

- “Los gustos de los usuarios pueden cambiar”
 - Diversidad/Serendipity al rescate
- “Influencia del tiempo”
 - Las calificaciones van perdiendo relevancia.
 - Novedad del contenido como parte de la calificación.
- “El Usuario quiere ver cosas que no le gustan.”
- “Lo que el usuario califica vs. lo que el usuario realmente ve”

Collaborative Filtering (CF)

CF: Pasos Básicos

- Identificar un conjunto de rating del **usuario objetivo**.
- Identificar el conjunto de usuarios similares al usuario objetivo según una **función de similitud**. (**vecinos**)
- Identificar productos que les gustaron a esos usuarios similares.
- **Generar una predicción** - que rating se les daría a esos productos por el usuario objetivo.
- **Basándonos** en las predicciones de rating **recomendarle N productos**.

CF: Ingredientes

- Lista de **n usuarios** con una lista de **m items**
- Cada usuario tiene una **lista de items** con una **opinion asociada**
 - **Explicita (rating numérico).**
 - **Implicita (cantidad de veces que se escucho una canción)**
- **Usuario activo** para quien estamos realizando la predicción.
- Métrica para medir la **semejanza entre usuarios/items**.
- Método para seleccionar un subconjunto de **vecinos**
- Método para **predecir un rating** para items que **no fueron calificados por el usuario**.

CF: Matriz de Utilidad

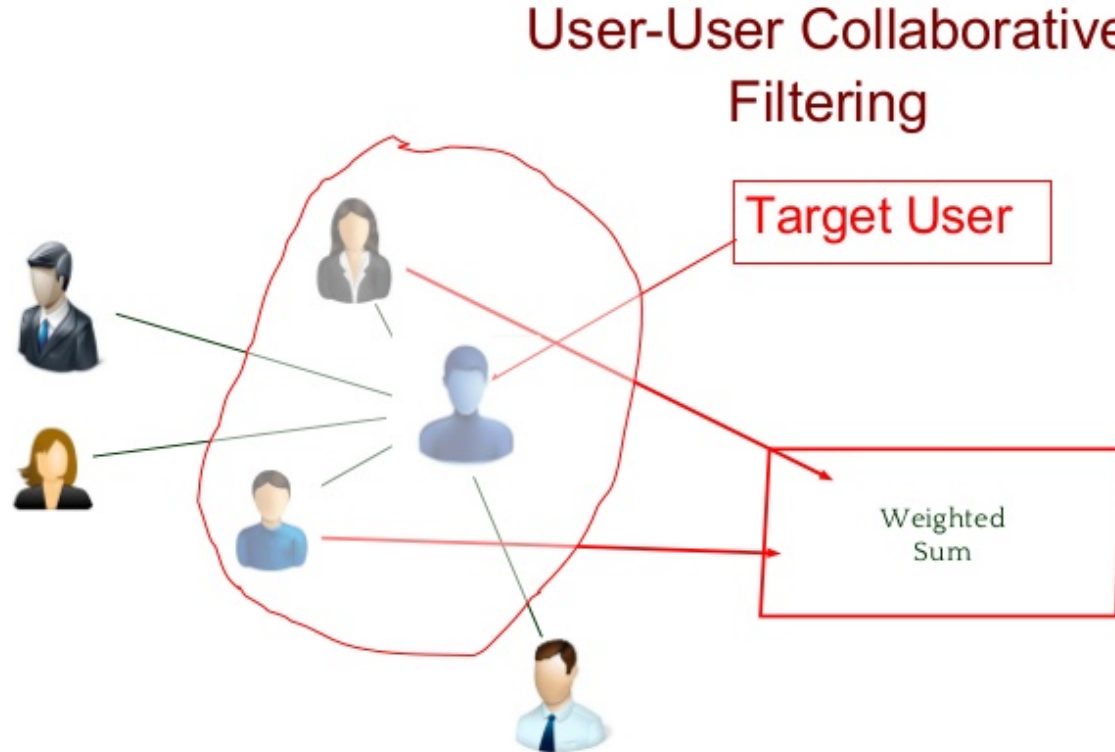
N usuarios con una lista de **M items** puede representarse como una **matriz de NxN**.

Similar a una matriz dispersa, pero con **números desconocidos** (posibles calificaciones)

Objetivo CF: estimar las calificaciones que nos faltan ($V_{ij} = ?$).

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Semejanza User-User (User-based CF)



Semejanza User-User (User-based CF)

Para estimar las calificaciones que le faltan al usuario i .

- Buscamos los usuarios más similares al **usuario i** .
- Estimamos las calificaciones que faltan en base a un promedio ponderado de las calificaciones de los demás usuarios, ponderadas de acuerdo a la semejanza **$\text{sim}(\mathbf{x}, \mathbf{y})$** que tengan con el **usuario i** .

Semejanza User-User (User-based CF)

La semejanza puede calcularse usando la correlación de Pearson.

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

Donde \bar{r}_x es el promedio de las calificaciones del usuario x.

Nota: Si los promedios de calificaciones fueran cero la correlación sería el cos(x,y), por lo que usualmente se centran las calificaciones de los usuarios en cero (se le resta el promedio a cada calificación del usuario)

Semejanza User-User (User-based CF)

Si tenemos como usuario destino al 5to:

Por ejemplo para el primer usuario y 5to usuario:

$x = (2, NA, 2, 4, 5, NA)$

$y = (NA, NA, 4, NA, 2)$

tomando el Promedio de calificación para el usuario

$(2 + 2 + 4 + 5) / 4 = 3.25$ y $(4 + 2) / 2 = 3$ centramos:

$(-1.25, NA, -1.25, 0.75, 1.75, NA)$

$(NA, NA, 1, NA, NA, -1)$

Semejanza User-User (User-based CF)

Una vez centrados, podemos hacer el cálculo de similitud con un usuario específico para el cual haremos las estimaciones usando el **coseno**

$$\text{sim}(x,y) = \cos(x,y) = \cos(u_i, u_j) = \frac{\sum_{k=1}^m v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^m v_{ik}^2 \sum_{k=1}^m v_{jk}^2}}$$

Semejanza User-User (User-based CF)

Por ejemplo con el usuario 5, la correlación de Pearson queda como

$$\text{sim}(5,1) = -0.27$$

$$\text{sim}(5,2) = 0.72$$

$$\text{sim}(5,3) = 0.5$$

$$\text{sim}(5,4) = -0.13$$

$$\text{sim}(5,6) = \text{NA}$$

Semejanza User-User, Estimación calificación

La estimación de las calificaciones faltantes se hacen con la siguiente fórmula

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

donde N es el conjunto de los usuarios más similares al que queremos estimar, s_{xy} es la similitud entre x e y, r_{yi} es el rating del usuario y para la película i.

Semejanza User-User, Estimación calificación

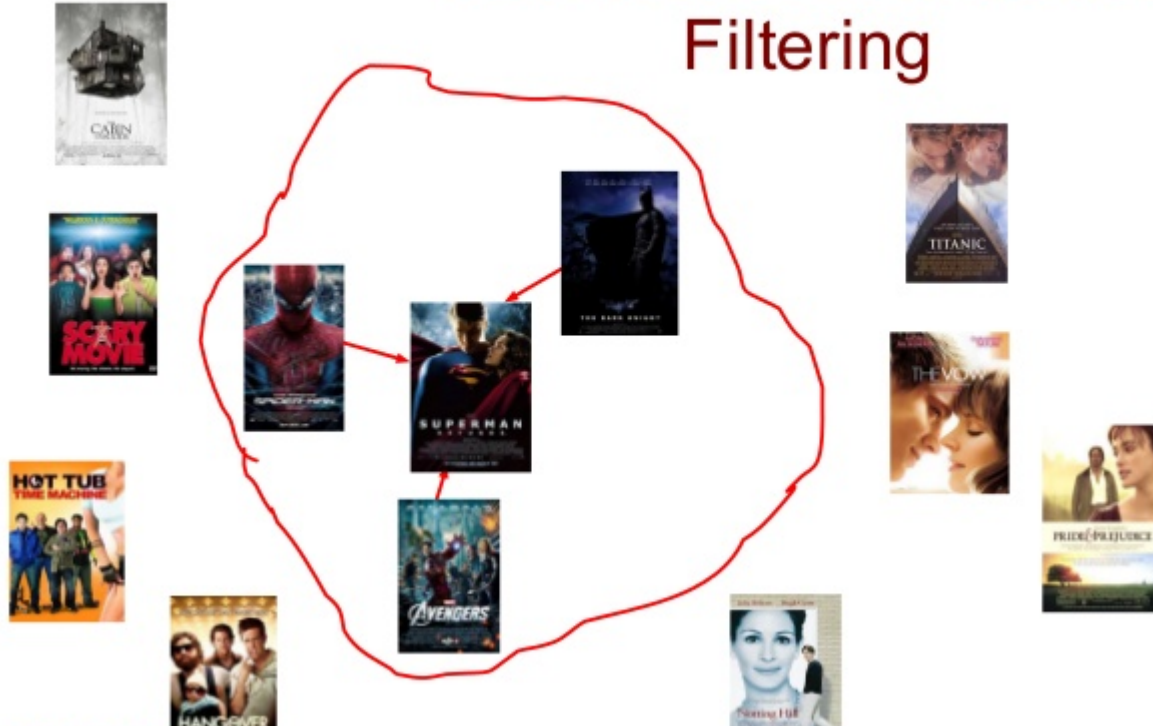
Con $N = 2$ podemos usar los dos usuarios mas similares al usuario 5. (Consideramos el usuario 2 y el 3)

Si queremos estimar la calificación para el usuario 5 de sherlock (1)

$= (5 * 0.72) / 0.72 = 5$, siendo 5 la calificación estimada para Sherlock.

Semejanza Item-Item (Item-based CF)

Item-Item Collaborative Filtering



Semejanza Item-Item (Item-based CF)

Para estimar las calificaciones de un **usuario i** para un item

- Buscamos **los items más parecidos al que queremos estimar y que haya calificado el usuario.**
- Estimamos la calificación haciendo un **promedio ponderado entre las calificaciones del usuario para esos items y la semejanza que tiene con el item a estimar.**

Semejanza Item-Item (Item-based CF)

Usando la siguiente
Matriz de utilidad,
queremos estimar la
calificación del
usuario 5 para la
primer película

		users											
		1	2	3	4	5	6	7	8	9	10	11	12
movies	1	1		3			5			5		4	
	2			5	4			4			2	1	3
	3	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	6	1		3		3			2			4	
		- unknown rating						- rating between 1 to 5					

Semejanza Item-Item (Item-based CF)

- Calculamos la semejanza de todas las películas con respecto a la uno.
 - Restamos a cada calificación **el promedio de calificación de la película.**
 - **Usamos el coseno** entre la primera película y cada una de las otras.

Semejanza Item-Item

Estimación de calificación

Siendo N el conjunto de películas más similares a la objetivo.

Con $N=2$ tomamos las dos películas más similares a 1.

La 6 (con semejanza 0.59) y **la 3** (con semejanza 0.41).

		users												sim(1,m)
		1	2	3	4	5	6	7	8	9	10	11	12	
movies	1	1		3		?	5			5		4		1.00
	2			5	4			4			2	1	3	-0.18
	<u>3</u>	2	4		1	2		3		4	3	5		<u>0.41</u>
	4		2	4		5			4			2		-0.10
	5			4	3	4	2					2	5	-0.31
	<u>6</u>	1		3		3			2			4		<u>0.59</u>

Semejanza Item-Item, Estimación calificación

Aplicando la misma estimación que anteriormente obtenemos

$$= (2 * 0.41 + 3 * 0.59) / (0.41 + 0.59) = 2.6$$

		users											
		1	2	3	4	5	6	7	8	9	10	11	12
movies	1	1		3		2.6	5			5		4	
	2			5	4			4			2	1	3
	<u>3</u>	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	<u>6</u>	1		3		3			2			4	

CF en Base a Desviaciones

- Base de implementaciones reales.

Estimación de la calificación del usuario i a la película j como:

$$r_{ij} = \mu + \delta_i + \delta_j + \delta_{ij}$$

μ Promedio de todas las calificaciones

δ_i Desviación del usuario i

δ_j Desviación de la película j

δ_{ij} Desviación del usuario i para la película j

¿Cómo evaluar Sistemas de Recomendación?

- Necesidad: Calificaciones para construir la matriz de utilidad y completar las faltantes via estimación.
- ¿Cómo medimos la calidad de las estimaciones?
- ¿Podemos hacerlo sin Feedback del Usuario?

¿Cómo evaluar Sistemas de Recomendación?

- Separamos un porcentaje de calificaciones al azar (20%)
- Construimos nuestro sistema de recomendaciones con el 80% restante y luego lo probamos con las calificaciones que hemos separado.

¿Cómo evaluar Sistemas de Recomendación?

- Al conocer el valor real de las calificaciones podemos calcular la efectividad del sistema (20%)

RMSE

(Root Mean Squared Root)

$$J = \frac{1}{N} \sqrt{\sum_{ij} (\hat{r}_{ij} - r_{ij})^2}$$

¡Tenemos una metrica para poder probar distintos algoritmos y parametros con distintos muestreos!

Modelos Latentes

Modelos Latentes

- Limitaciones de Collaborative Filtering
 - **Solo capaz de calcular semejanzas entre items.**
 - **No es capaz de darse cuenta que un conjunto de items similares expresa la actitud del usuario a ese tipo de items.**

Modelos Latentes

- **Modelización a partir de la cual un conjunto de items similares expresa la actitud del usuario en relación a su tipo.**
- **Nos permite inferir preferencias de los usuarios que no están explícitas en la matriz de utilidad.**
- **Dos Opciones:**
 - **Descomposición de la matriz de utilidad (SGD, SVD++)**
 - **Restricted Boltzmann Machines**

Descomposición de la Matriz de Utilidad

Descomponer la matriz de utilidad en 2 matrices.

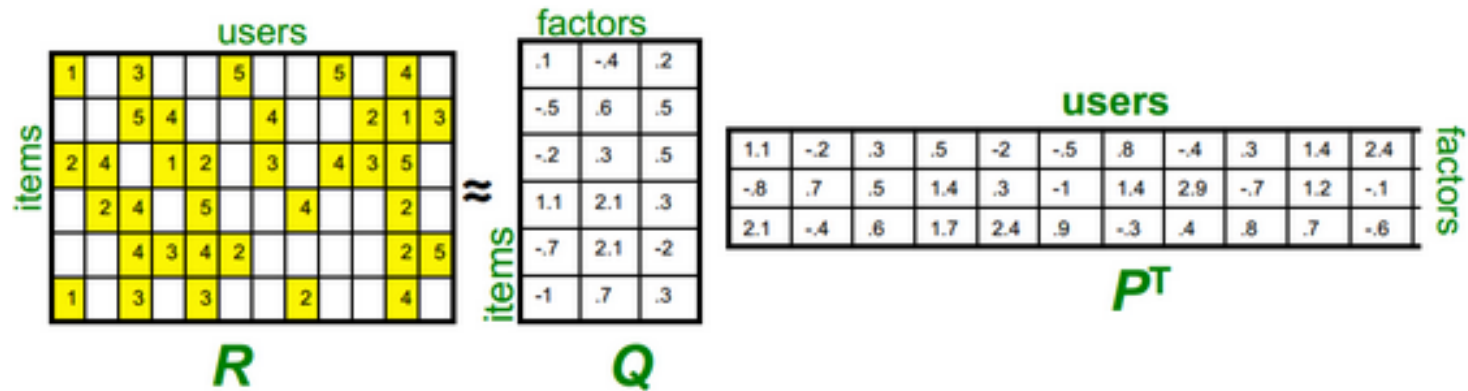
$$U_{m \times n} = Q_{m \times r} P^t_{n \times r}$$

Q: Tantas filas como items y columnas como factores latentes.

P: Tantas filas como usuarios y tantas columnas como factores latentes.

Descomposición de la Matriz de Utilidad

Objetivo: Hallar **Q** y **P** tales que minimicen el error con la matriz de utilidad original



Descomposición de la Matriz de Utilidad

Hallando la descomposición podemos estimar la calificación para el usuario i y la item j haciendo: **$Q_i * P_j$**

Ejemplo: usuario 5 y el item 2

$$Q_1 = (-0.5, 0.6, 0.5)$$

$$Q_1 * P_{5t} = 2.38$$

$$P_{5t} = (-2, 0.3, 2.4)$$

Descomposición de la Matriz de Utilidad

- Problema similar a Indexación Semántica latente. (reducción de dimensiones y obtención del modelo latente)
- No podemos usar **SVD**. (**Necesitamos la matriz de utilidad completa, sin “huecos”**).
- Algo similar a la **SVD**, **SVD Parcial...**

Descomposición de la Matriz de Utilidad

Problema de optimización

$$\min_{P,Q} \sum_{(i,x) \in R} (r_{xi} - q_i \cdot p_x)^2$$

donde el R que usamos es nuestro set de entrenamiento (80% datos del sistema).

Potencial problema de overfitting.

Descomposición de la Matriz de Utilidad

una forma de solucionar esto es intentar minimizar las normas de las matrices **Q** y **P** (regularización), por lo que reformulamos:

$$\min_{P, Q} \underbrace{\sum_{\text{training}} (r_{xi} - q_i p_x)^2}_{\text{"error"}} + \underbrace{\left[\lambda_1 \sum_x \|p_x\|^2 + \lambda_2 \sum_i \|q_i\|^2 \right]}_{\text{"length"}}$$

$\lambda_1, \lambda_2 \dots$ user set regularization parameters

con lambda 1,2 como parametros a definir.

Resolviendo el problema de optimización: Gradient Stochastic Descent (SGD)

- Inicializar Q y P con los valores de la SVD suponiendo que las calificaciones que faltan son cero.
- Por cada calificación r_{ij} hasta convergencia
 - Actualizar $Q_{ij} = Q_{ij} - \alpha \nabla Q_{ij}$
 - Actualizar $P_{ij} = P_{ij} - \alpha \nabla P_{ij}$

Gradient Stochastic Descent (SGD)

Cálculo del Gradiente

Derivada del error: $\epsilon_{xi} = r_{xi} - Q_i P_x$

Actualizar Q: $Q_i = Q_i + \alpha(\epsilon_{xi} P_x - \lambda_1 Q_i)$

Actualizar P: $P_x = P_x + \alpha(\epsilon_{xi} Q_i - \lambda_2 P_x)$

La implementación puede hacerse con un doble for sobre número de iteraciones y luego por cada r_{xi} , actualizando Q_i y P_x

SGD: Agregando desviaciones de usuario y el item

Podemos modelar la recomendacion en base a desviaciones a partir del promedio de calificación global.

$$r_{xi} = \underbrace{\mu}_{\text{Overall mean rating}} + \underbrace{b_x}_{\text{Bias for user } x} + \underbrace{b_i}_{\text{Bias for movie } i} + \underbrace{q_i \cdot p_x}_{\text{User-Movie interaction}}$$

siendo μ el promedio global de todas las calificaciones. y b_x (desviacion usuario) y b_i (desvio item) son los que buscaremos optimizar

SGD: Agregando desviaciones de usuario y el item

Problema de optimización

$$\min_{Q,P} \sum_{(x,i) \in R} \left(r_{xi} - (\mu + b_x + b_i + q_i p_x) \right)^2$$

goodness of fit

$$+ \left(\lambda_1 \sum_i \|q_i\|^2 + \lambda_2 \sum_x \|p_x\|^2 + \lambda_3 \sum_x \|b_x\|^2 + \lambda_4 \sum_i \|b_i\|^2 \right)$$

regularization

λ is selected via grid-search on a validation set

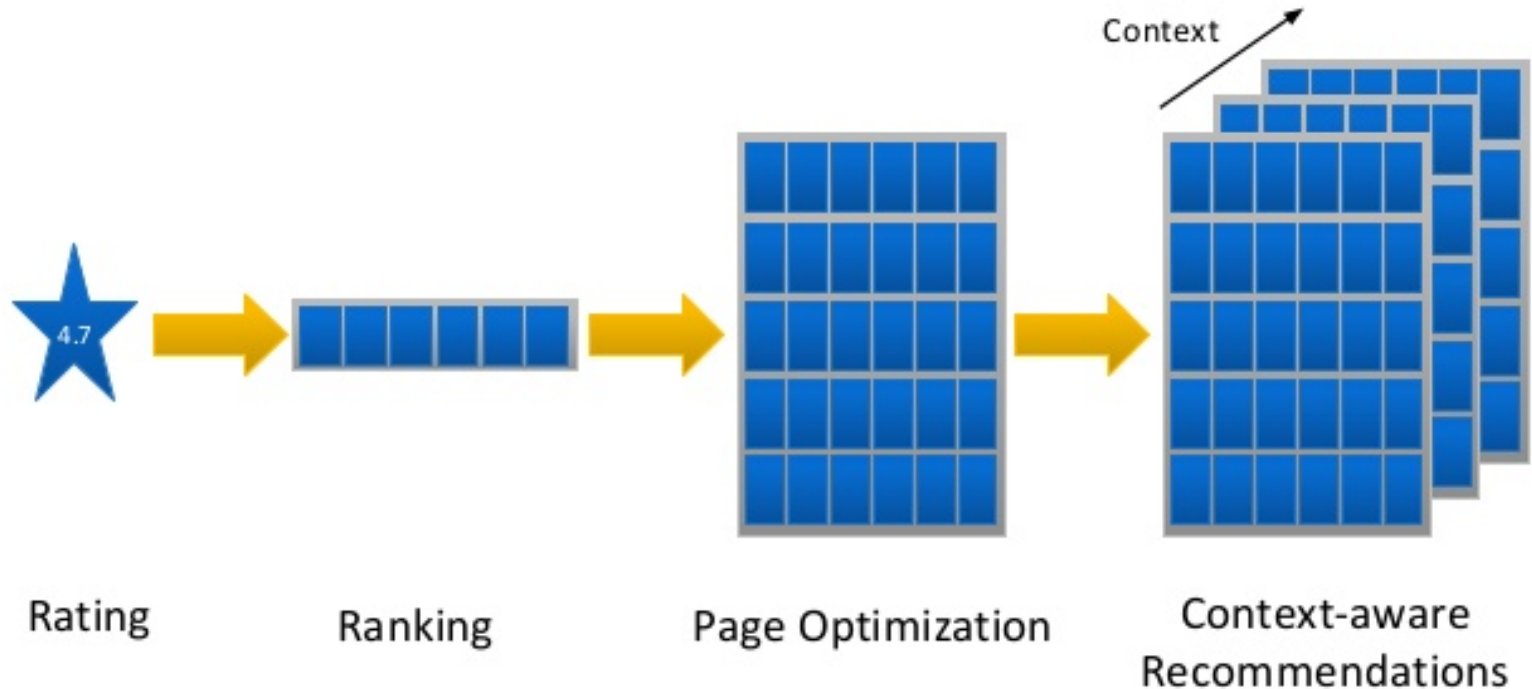
SGD: Factores Temporales

Los parametros afectados por el tiempo son b_x y b_i dado que los usuarios cambian de opinión a lo largo del tiempo y tambien la percepción de una pelicula.

$$r_{xi} = \mu + b_x(t) + b_i(t) + q_i \cdot p_x$$

El set de entrenamiento lo dividimos en intervalos temporales y hacemos la optimización teniendo en cuenta cuando fue hecha la estimacion.

Evolución del problema de recomendación



Más puntos de exploración

- Sparsity Problem (CF)
- Scalability Problems (CF)
- Diversidad de contenido.
- Cold-Start Problem
- Learning to Rank

Recursos

- <http://www.slideshare.net/xamat/qcon-sf-2013-machine-learning-recommender-systems-netflix-scale>
- <http://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu>
- <http://www.slideshare.net/xamat/kdd-2014-tutorial-the-recommender-problem-revisited>
- <http://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify>