Exploiting subsequence matching in Recommender Systems

Pablo Sánchez Pérez

Universidad Autónoma de Madrid

pablo.sanchezp@estudiante.uam.es

3 de Julio 2017

Índice

El problema de la recomendación

$\mathcal{U}$

El problema de la recomendación

$\mathcal{U}$

Sistemas de recomendación

-Filtrado colaborativo
-Basado en contenido
-Híbridos
...

El problema de la recomendación

$\mathcal{U}$

Sistemas de recomendación

-Filtrado colaborativo
-Basado en contenido
-Híbridos
...

$\mathcal{I}$

El problema de la recomendación

$\mathcal{U}$

$\mathcal{I}$

	i_1	i_2	i_3	i_4	$\cdots$
u_1	-	-	5	3	$\cdots$
u_2	4	-	4	-	$\cdots$
u_3	5	5	-	-	$\cdots$
u_4	-	2	1	-	$\cdots$
u_5	2	-	-	5	$\cdots$
u_6	-	1	-	1	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Objetivo: Recomendaciones **Relevantes** y **Personalizadas**

Nueva similitud

Trabajar con el algoritmo de la subcadena común más larga para definir la similitud entre usuarios

Nueva similitud

Trabajar con el algoritmo de la subcadena común más larga para definir la similitud entre usuarios

Generación de (sub) cadenas o secuencias

Investigar diferentes maneras de transformar a los usuarios en secuencias de artículos consumidos. Ver si es posible (y cómo) añadir información auxiliar para ser empleada en la generación de secuencias

Nueva similitud

Trabajar con el algoritmo de la subcadena común más larga para definir la similitud entre usuarios

Generación de (sub) cadenas o secuencias

Investigar diferentes maneras de transformar a los usuarios en secuencias de artículos consumidos. Ver si es posible (y cómo) añadir información auxiliar para ser empleada en la generación de secuencias

Evaluación

Obtener resultados empíricos de evaluación de ranking, novedad y diversidad. Comparar estos nuevos recomendadores con otros conocidos en el área

1: **procedure** $\mathrm{LCS}(x, y)$ ▷ La LCS de x e y
2: $L[0 \cdots m, 0 \cdots n] \leftarrow 0$
3: **for** $i \leftarrow 1, m$ **do**
4: **for** $j \leftarrow 1, n$ **do**
5: **if** $x_i = y_j$ **then** ▷ Hay una coincidencia
6: $L[i, j] \leftarrow L[i-1, j-1] + 1$
7: **else**
8: $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$
9: **end if**
10: **end for**
11: **end for**
12: **return** $L[m, n]$ ▷ $L[m, n]$ contiene la longitud de LCS entre $x_1 \ldots x_i$ y $y_1 \ldots y_j$
13: **end procedure**

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅							
G							
C							
G							
T							
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G							
C							
G							
T							
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C							
G							
T							
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G							
T							
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T							
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G							
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C							

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{si i=0 o j=0} \\ L[i-1,j-1]+1 & \text{si } i,j>0 \text{ y } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{si } i,j>0 \text{ y } X_i \neq Y_j \end{cases} \quad (1)$$

Utilizada para comparar similitudes entre dos cadenas de ADN

La subsecuencia puede no ser única

	∅	A	G	G	T	A	C
∅	0	0	0	0	0	0	0
G	0	0	1	1	1	1	1
C	0	0	1	1	1	1	2
G	0	0	1	2	2	2	2
T	0	0	1	2	3	3	3
G	0	0	1	2	3	3	3
C	0	0	1	2	3	3	4

Hipótesis

El algoritmo de LCS puede emplearse como medida de similitud entre usuarios y ser incluido en un sistema de recomendación de filtrado colaborativo basado en vecinos ($k - NN$)

Hipótesis

El algoritmo de LCS puede emplearse como medida de similitud entre usuarios y ser incluido en un sistema de recomendación de filtrado colaborativo basado en vecinos ($k - NN$)

Filtrado colaborativo basado en vecinos

$$\hat{r}_{ui} \approx \frac{\sum_{v \in \mathcal{N}_i(u)} r_{vi} w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \approx \sum_{v \in \mathcal{N}_i(u)} r_{vi} w_{uv} \tag{2}$$

Hipótesis

El algoritmo de LCS puede emplearse como medida de similitud entre usuarios y ser incluido en un sistema de recomendación de filtrado colaborativo basado en vecinos ($k - NN$)

Filtrado colaborativo basado en vecinos

$$\hat{r}_{ui} \approx \frac{\sum_{v \in \mathcal{N}_i(u)} r_{vi} w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \approx \sum_{v \in \mathcal{N}_i(u)} r_{vi} \boxed{w_{uv}} \tag{2}$$

- El objetivo es sustituir el peso de los vecinos por el valor de LCS obtenido al comparar las secuencias del usuario u y v:

$$w_{uv} = LCS(u, v)$$

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Analizar otras adaptaciones del algoritmo

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Analizar otras adaptaciones del algoritmo

- Aprovechar sesgos del usuario

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Analizar otras adaptaciones del algoritmo

- Aprovechar sesgos del usuario
- Favorecer los artículos más relevantes de cada usuario

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Analizar otras adaptaciones del algoritmo

- Aprovechar sesgos del usuario
- Favorecer los artículos más relevantes de cada usuario
- Considerar la confianza de los vecinos

Generación de secuencias

Definir un método genérico para obtener secuencias a partir de los datos del usuario

Analizar otras adaptaciones del algoritmo

- Aprovechar sesgos del usuario
- Favorecer los artículos más relevantes de cada usuario
- Considerar la confianza de los vecinos
- Normalizar en el intervalo $[0, 1]$, al igual que en otras métricas conocidas como la correlación de Pearson o la similitud coseno

1: **procedure** LCS_RecSys(u, v, f, δ) ▷ La LCS de u y v aplicando la transformación f
2: $(x, y) \leftarrow \boxed{(f(u), f(v))}$ ▷ La cadena x contiene m símbolos
3: $L[0 \cdots m, 0 \cdots n] \leftarrow 0$
4: **for** $i \leftarrow 1, m$ **do**
5: **for** $j \leftarrow 1, n$ **do**
6: **if** $\boxed{\texttt{match}(x_i, y_j, \delta)}$ **then** ▷ Hay una coincidencia $\leq \delta$
7: $L[i, j] \leftarrow L[i-1, j-1] + 1$
8: **else**
9: $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$
10: **end if**
11: **end for**
12: **end for**
13: **return** $L[m, n]$
14: **end procedure**

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Función f como combinación de tres funciones

$$f = s \circ t \circ e$$

e: Extensión de la información
t: Transformación a símbolos interpretables por LCS
s: Ordenación

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Extensión de la información

A cada ítem se le asocia un conjunto de tuplas de elementos asociables a dicho artículo $e : \mathcal{I} \times \mathcal{R} \rightarrow \mathcal{I} \times \mathcal{T}^k$

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Extensión de la información

A cada ítem se le asocia un conjunto de tuplas de elementos asociables a dicho artículo $e : \mathcal{I} \times \mathcal{R} \rightarrow \mathcal{I} \times \mathcal{T}^k$

Extensión de la información. Géneros: $e_{gr}(i, r) = (i, \{G_j(i), r\}_j)$

(Star Wars, {{Aventura, 4}, {Sci-Fi, 4} })
(Alien, {{Sci-Fi, 2}, {Terror, 2} })
(Blade runner, {{Acción, 5}, {Sci-Fi, 5}, {Thriller, 5}})

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Símbolos interpretables por LCS

Trasformación de dichas tuplas en símbolos más manejables

$t : \mathcal{I} \times \mathcal{T}^k \rightarrow \mathcal{I} \times \mathbb{Z}^k$

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Símbolos interpretables por LCS

Trasformación de dichas tuplas en símbolos más manejables
$t : \mathcal{I} \times \mathcal{T}^k \rightarrow \mathcal{I} \times \mathbb{Z}^k$

Símbolos interpretables para LCS. $t_{gr}(g, r) = 10 \cdot id(g) + r$

(Star Wars, {14,44})
(Alien, {42,52})
(Blade runner, {25,45,65})

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Ordenación de secuencias

$$s(\{i_j, (n_{jk})_k\}_j) = ((n_{jk})_k)_{j=1}^{|\mathcal{I}|}$$

Película	**Rating**	**Fecha**	**Géneros**
Star Wars IV (id 1)	4	24/6/2017	Aventura (id 1), Sci-Fi (id 4)
Alien (id 8)	2	26/6/2017	Sci-Fi (id 4), Terror (id 5)
Blade Runner (id 16)	5	25/6/2017	Acción (id 2), Sci-Fi (id 4) Thiller (id 6)

Ordenación de secuencias

$$s(\{i_j, (n_{jk})_k\}_j) = ((n_{jk})_k)_{j=1}^{|\mathcal{I}|}$$

Representación final. Ordenación por ids y temporal

Id (s_i) = (14,44,42,52,25,45,65)
Temporal (s_T) = (14,44,25,45,65,42,52)

Muchas funciones de similitud están acotadas entre [-1,1] o [0,1]. Para conseguir esto con LCS, se pueden emplear las siguientes:

$$\text{sim}_1^{f,\delta}(u,v) = \text{LCS\_RecSys}(u,v,f,\delta) \tag{3.1}$$

$$\text{sim}_2^{f,\delta}(u,v) = \frac{\text{sim}_1^{f,\delta}(u,v)^2}{|f(u)| \cdot |f(v)|} \tag{3.2}$$

$$\text{sim}_3^{f,\delta}(u,v) = \frac{2 \cdot \text{sim}_1^{f,\delta}(u,v)}{|f(u)| + |f(v)|} \tag{3.3}$$

$$\text{sim}_4^{f,\delta}(u,v) = \frac{\text{sim}_1^{f,\delta}(u,v)}{\max(|f(u)|, |f(v)|)} \tag{3.4}$$

$$\text{sim}_5^{f,\delta}(u,v) = \frac{\text{sim}_1^{f,\delta}(u,v)}{\min(|f(u)|, |f(v)|)} \tag{3.5}$$

Configuración	**Notación**	**Descripción**	**Efecto**
Generación de secuencias	f_{ir}, f_{gr} f_i, f_{dr} $\dots$	Generación de distintas secuencias empleando información de contenido o colaborativa	Obtención de recomendadores híbridos o puramente colaborativos
Preferencia	γ	Considera sólo los artículos que han sido votado con una una nota $\geq \gamma$	Reducción del coste computacional Reducción de cobertura
Confianza	τ	Sólo considera los vecinos que superan un determinado valor de similitud	Calidad de los vecinos aumentada Reducción de cobertura
Umbral	δ	Dos artículos son iguales si han sido puntuados con una diferencia menor o igual que el valor de dicho umbral	Calidad de los vecinos decrementada Aumento de la cobertura
Normalizaciones	sim_x	Acotar la escala de las similitudes	Similitud en el intervalo [0,1]

Evaluación de ranking

Objetivo: Devolver una lista de artículos relevantes al usuario
Métricas: Precision, Recall, MAP y nDCG
Valores más cercanos a 1, mejores

Evaluación de ranking

Objetivo: Devolver una lista de artículos relevantes al usuario
Métricas: Precision, Recall, MAP y nDCG
Valores más cercanos a 1, mejores

Novedad

Objetivo: Recomendar al usuario artículos que no está acostumbrado a consumir
Métricas: EPC y EPD
Valores más cercanos a 1, mejores

Evaluación de ranking

Objetivo: Devolver una lista de artículos relevantes al usuario
Métricas: Precision, Recall, MAP y nDCG
Valores más cercanos a 1, mejores

Novedad

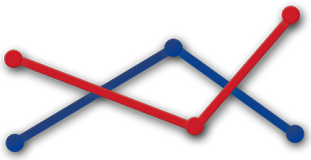
Objetivo: Recomendar al usuario artículos que no está acostumbrado a consumir
Métricas: EPC y EPD
Valores más cercanos a 1, mejores

Diversidad

Objetivo: Recomendar al usuario artículos distintos entre ellos
Métricas: EILD, Gini, Aggregate Diversity y $\alpha-$nDCG
Valores más altos, mejores

- Sólo se consideran los primeros 5 artículos devueltos (cutoff @5)
- Relevantes aquellos artículos que han sido puntuados con ≥ 5

RiVal



Mahout

RankSys

- RiVal para evaluación de ránking. RankSys para evaluación de novedad y diversidad
- RankSys para programar recomendadores
- Mahout descartado por bajos resultados

Dataset	Usuarios	Artículos	Ratings	Densidad
MovielensHetRec	2.113	10.197	855.598	3,97%
*LastFm	1.892	17.632	92.834	0,28%
MovieTweetings	45.324	26.087	541.304	0,045%

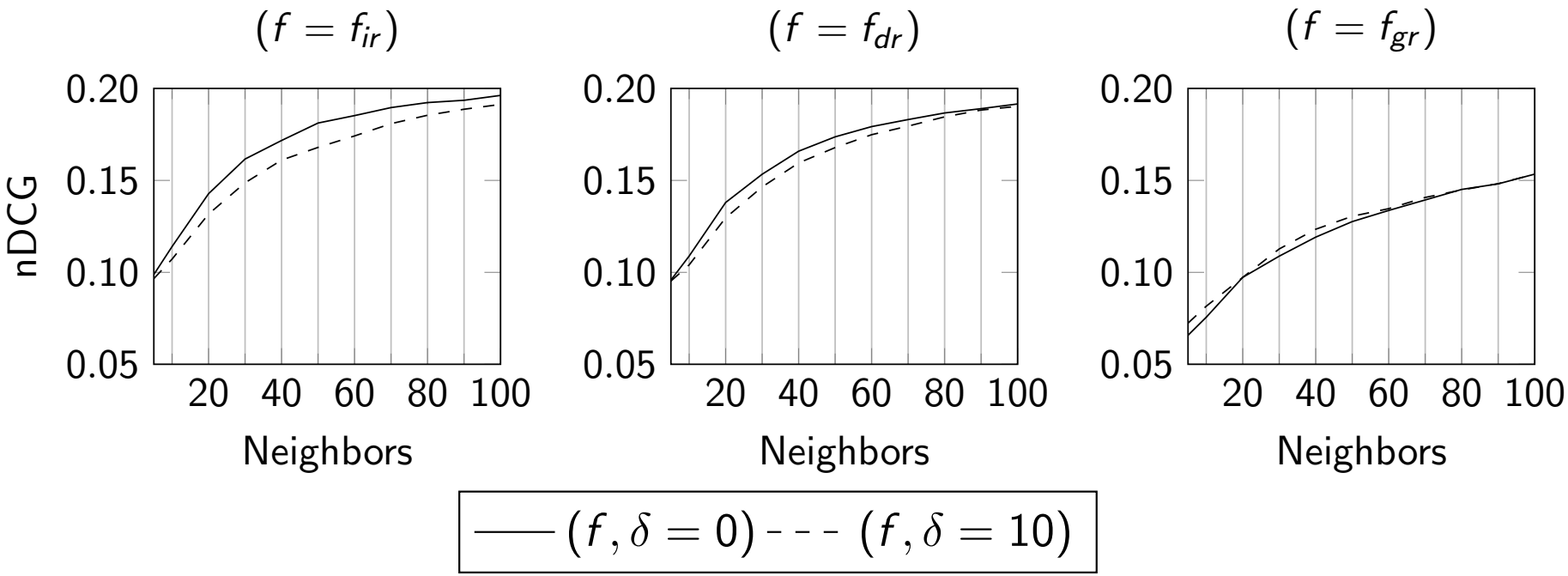
$$*\tilde{r}_{ui} \sim \left\lceil 5 \cdot \frac{\mathcal{F}_{ui}}{\max \mathcal{F}_u} \right\rceil$$

- MovielensHetRec y Lastfm validación cruzada con 5 folds
- MovieTweetings split temporal

Resultados: efectos de umbral y distintas secuencias (MovielensHetRec)

Configuración	Notación	Descripción	Efecto
Generación de secuencias	f_{ir}, f_{gr} f_i, f_{dr} $\dots$	Generación de distintas secuencias empleando información de contenido o colaborativa	Obtención de recomendadores híbridos o puramente colaborativos
Preferencia	γ	Considera sólo los artículos que han sido votado con una una nota $\geq \gamma$	Reducción del coste computacional Reducción de cobertura
Confianza	τ	Sólo considera los vecinos que superan un determinado valor de similitud	Calidad de los vecinos aumentada Reducción de cobertura
Umbral	δ	Dos artículos son iguales si han sido puntuados con una diferencia menor o igual que el valor de dicho umbral	Calidad de los vecinos decrementada Aumento de la cobertura
Normalizaciones	sim_x	Acotar la escala de las similitudes	Similitud en el intervalo [0,1]

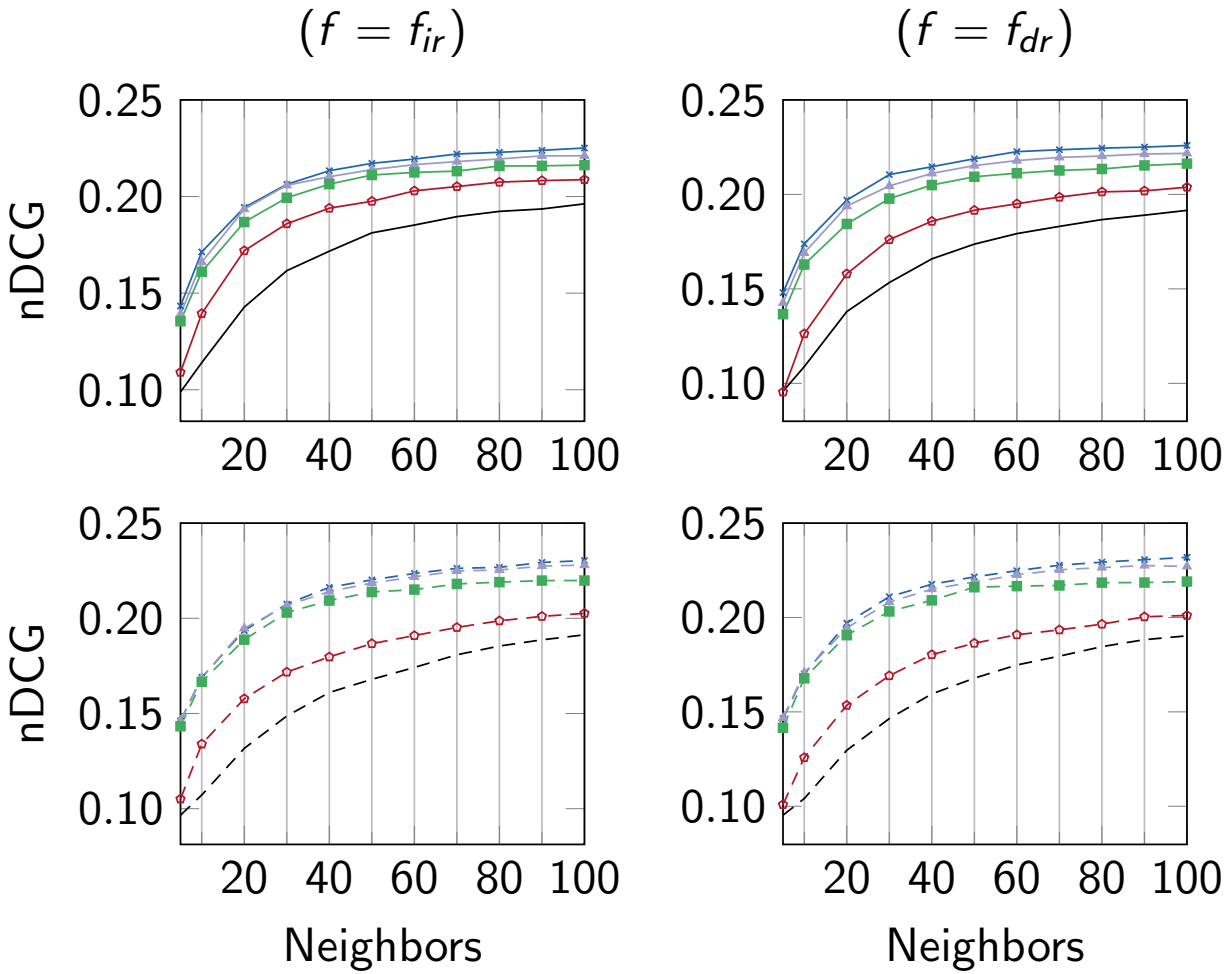
Secuencias basadas en géneros obtienen un rendimiento peor



Resultados: efecto de normalizaciones (MovielensHetRec)

Configuración	Notación	Descripción	Efecto
Generación de secuencias	f_{ir}, f_{gr} f_i, f_{dr} $\dots$	Generación de distintas secuencias empleando información de contenido o colaborativa	Obtención de recomendadores híbridos o puramente colaborativos
Preferencia	γ	Considera sólo los artículos que han sido votado con una una nota $\geq \gamma$	Reducción del coste computacional Reducción de cobertura
Confianza	τ	Sólo considera los vecinos que superan un determinado valor de similitud	Calidad de los vecinos aumentada Reducción de cobertura
Umbral	δ	Dos artículos son iguales si han sido puntuados con una diferencia menor o igual que el valor de dicho umbral	Calidad de los vecinos decrementada Aumento de la cobertura
Normalizaciones	sim_x	Acotar la escala de las similitudes	Similitud en el intervalo [0,1]

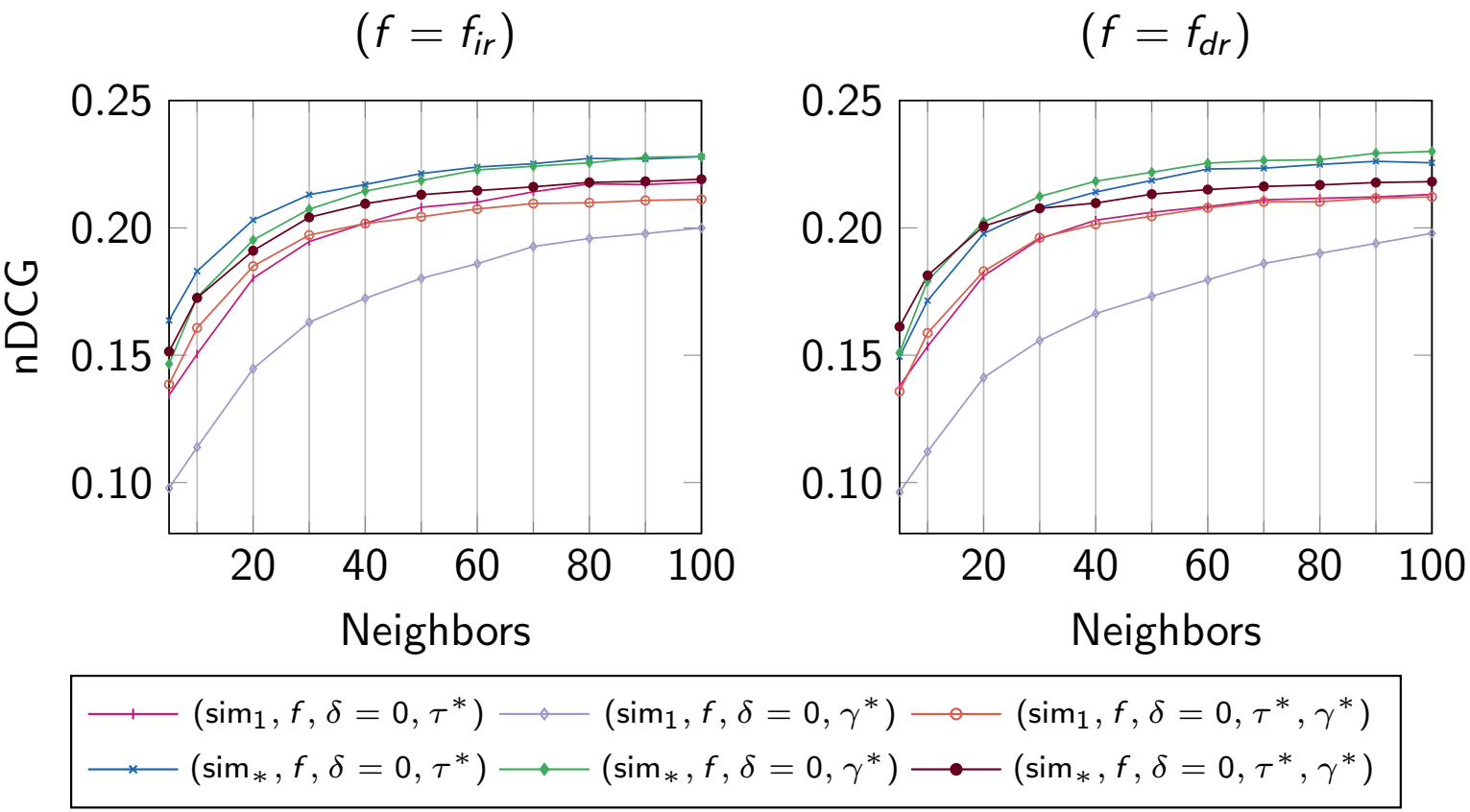
Normalizaciones permiten incrementar el valor de nDCG



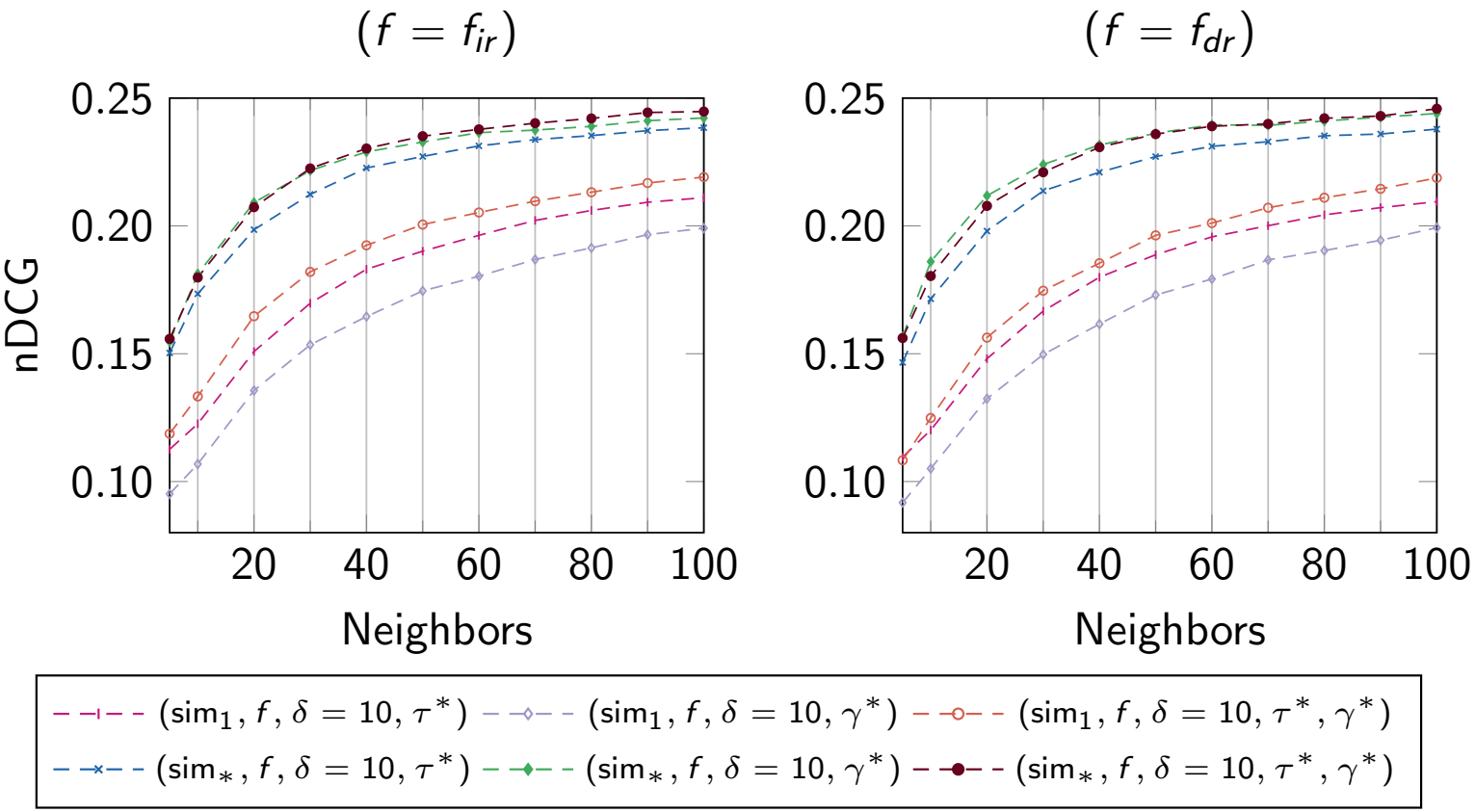
Resultados: efecto de las mejores combinaciones (MovielensHetRec)

Configuración	Notación	Descripción	Efecto
Generación de secuencias	f_{ir}, f_{gr} f_i, f_{dr} $\dots$	Generación de distintas secuencias empleando información de contenido o colaborativa	Obtención de recomendadores híbridos o puramente colaborativos
Preferencia	γ	Considera sólo los artículos que han sido votado con una una nota $\geq \gamma$	Reducción del coste computacional Reducción de cobertura
Confianza	τ	Sólo considera los vecinos que superan un determinado valor de similitud	Calidad de los vecinos aumentada Reducción de cobertura
Umbral	δ	Dos artículos son iguales si han sido puntuados con una diferencia menor o igual que el valor de dicho umbral	Calidad de los vecinos decrementada Aumento de la cobertura
Normalizaciones	sim_x	Acotar la escala de las similitudes	Similitud en el intervalo [0,1]

Combinación de parámetros: mejores soluciones en términos de ranking-evaluation



Combinación de parámetros: mejores resultados con $\delta = 10$



Mejor recomendador: MF
LCS altamente competitivo (comparación con UB)

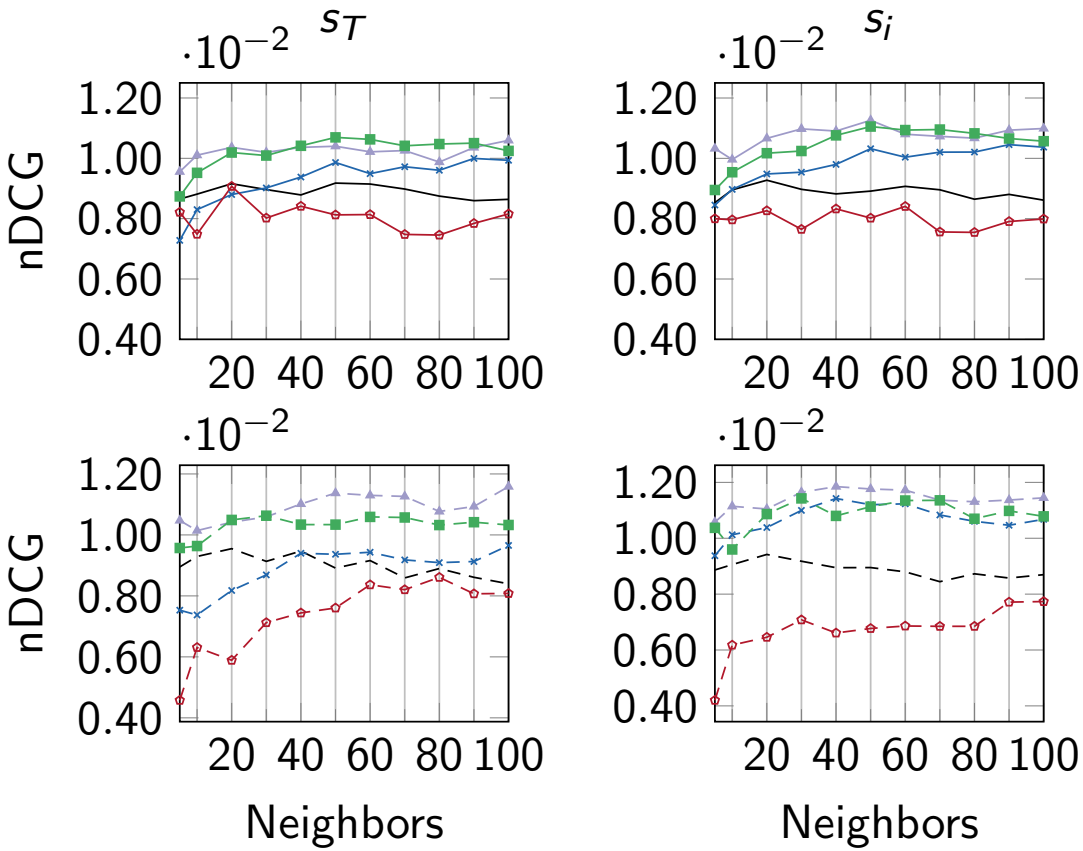
Recommender	nDCG	P	R	MAP	EPC	EPD	AD	α-nDCG	EILD	Gini
Pop	0.160	0.105	0.112	0.069	0.444	**0.741**	7.66%	0.123	**0.700**	0.002
UB1	0.233	0.152	0.161	0.106	0.484	0.723	12.94%	0.177	0.682	0.003
UB2	0.235	0.153	0.161	0.107	0.490	0.722	12.94%	0.177	0.678	0.004
IB1	0.162	0.109	0.116	0.069	0.521	0.712	65.03%	0.132	0.660	0.004
IB2	0.179	0.119	0.126	0.077	0.508	0.710	**67.04%**	0.145	0.672	0.004
MF	**0.271**	0.176	**0.200**	**0.133**	0.635	0.694	36.50%	**0.207**	0.626	**0.025**
PureCB	0.010	0.007	0.010	0.005	**0.853**	0.739	53.69%	0.028	0.658	0.020
CBCF	0.254	0.165	0.180	0.120	0.504	0.722	13.06%	0.192	0.666	0.004
BestLCS	0.246	**0.179**	0.152	0.101	0.406	0.571	25.88%	0.150	0.538	0.005

$$\text{BestLCS} = (\text{sim}_2, f_{dr}, 10, 30, \bar{u})$$

Resultados: efecto de ordenaciones y normalizaciones (MovieTweetings)

Configuración	**Notación**	**Descripción**	**Efecto**
Generación de secuencias	f_{ir}, f_{gr} f_i, f_{dr} $\dots$	Generación de distintas secuencias empleando información de contenido o colaborativa	Obtención de recomendadores híbridos o puramente colaborativos
Preferencia	γ	Considera sólo los artículos que han sido votado con una una nota $\geq \gamma$	Reducción del coste computacional Reducción de cobertura
Confianza	τ	Sólo considera los vecinos que superan un determinado valor de similitud	Calidad de los vecinos aumentada Reducción de cobertura
Umbral	δ	Dos artículos son iguales si han sido puntuados con una diferencia menor o igual que el valor de dicho umbral	Calidad de los vecinos decrementada Aumento de la cobertura
Normalizaciones	sim_x	Acotar la escala de las similitudes	Similitud en el intervalo [0,1]

Ordenación temporal (s_T) no produce ventajas respecto a la de id (s_i)
Resultados muy bajos en comparación con otros datasets



MC: mejor recomendador temporal. Fossil no obtiene buenos resultados.
LCS sigue siendo competitivo en sparse datasets

Recommender	nDCG	P	R	MAP	EPC	EPD	AD	α-nDCG	EILD	Gini
Pop	0.003	0.006	0.003	0.001	0.938	**0.393**	1.84%	0.006	**0.751**	0.000
UB1	0.011	0.016	0.006	0.003	0.459	0.330	39.05%	0.008	0.305	0.007
UB2	0.010	0.015	0.006	0.003	0.463	0.333	35.07%	0.008	0.311	0.007
IB1	0.009	0.015	0.006	0.003	0.484	0.331	**61.10%**	0.007	0.299	0.015
IB2	0.009	0.016	0.006	0.003	0.484	0.334	43.68%	0.008	0.311	**0.020**
MF	0.006	0.009	0.004	0.002	**0.986**	0.329	11.76%	0.009	0.517	0.003
Fossil	0.008	0.012	0.004	0.002	0.425	0.324	14.08%	0.005	0.327	0.001
MC	**0.013**	**0.021**	**0.008**	**0.004**	0.428	0.323	26.61%	**0.011**	0.312	0.002
$(\text{sim}_2, f_{ir}, 10, s_i)$	0.011	0.016	0.007	0.004	0.462	0.332	40.02%	0.008	0.310	0.008

Conclusiones

1. El algoritmo LCS es competitivo frente a otras similitudes

Conclusiones

1. El algoritmo LCS es competitivo frente a otras similitudes
2. El algoritmo LCS es altamente adaptable y configurable

Conclusiones

1. El algoritmo LCS es competitivo frente a otras similitudes
2. El algoritmo LCS es altamente adaptable y configurable
3. Las normalizaciones resultan imprescindibles para mejorar el rendimiento del algoritmo LCS

Conclusiones

1. El algoritmo LCS es competitivo frente a otras similitudes
2. El algoritmo LCS es altamente adaptable y configurable
3. Las normalizaciones resultan imprescindibles para mejorar el rendimiento del algoritmo LCS
4. Las secuencias que emplean géneros han obtenido unos resultados peores que las puramente colaborativas

Conclusiones

1. El algoritmo LCS es competitivo frente a otras similitudes
2. El algoritmo LCS es altamente adaptable y configurable
3. Las normalizaciones resultan imprescindibles para mejorar el rendimiento del algoritmo LCS
4. Las secuencias que emplean géneros han obtenido unos resultados peores que las puramente colaborativas
5. La ordenación secuencial a priori no parece tener ventajas frente a la ordenación por id. No obstante, este resultado al ser contraintuitivo, debe confirmarse con más experimentos

Más características

Ver si es posible emplear datos demográficos u otras características para generar recomendadores basados en LCS

Más características

Ver si es posible emplear datos demográficos u otras características para generar recomendadores basados en LCS

Variaciones de LCS

En lugar de sumar siempre el mismo valor si pasa de un cierto threshold, se podría modificar el algoritmo para que tenga en cuenta los ratings de forma distinta

Más características

Ver si es posible emplear datos demográficos u otras características para generar recomendadores basados en LCS

Variaciones de LCS

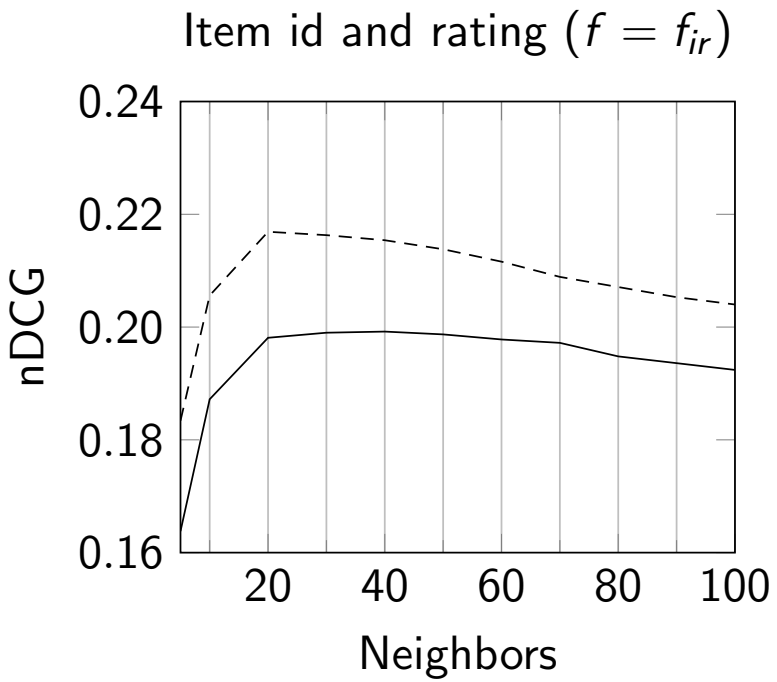
En lugar de sumar siempre el mismo valor si pasa de un cierto threshold, se podría modificar el algoritmo para que tenga en cuenta los ratings de forma distinta

Reformulación de los k-NN

Tener en cuenta la última interacción entre los vecinos para recomendar artículos cercanos a ella. Consideramos que el algoritmo de LCS puede aplicarse para esta nueva aproximación empleándose también como medida de similitud

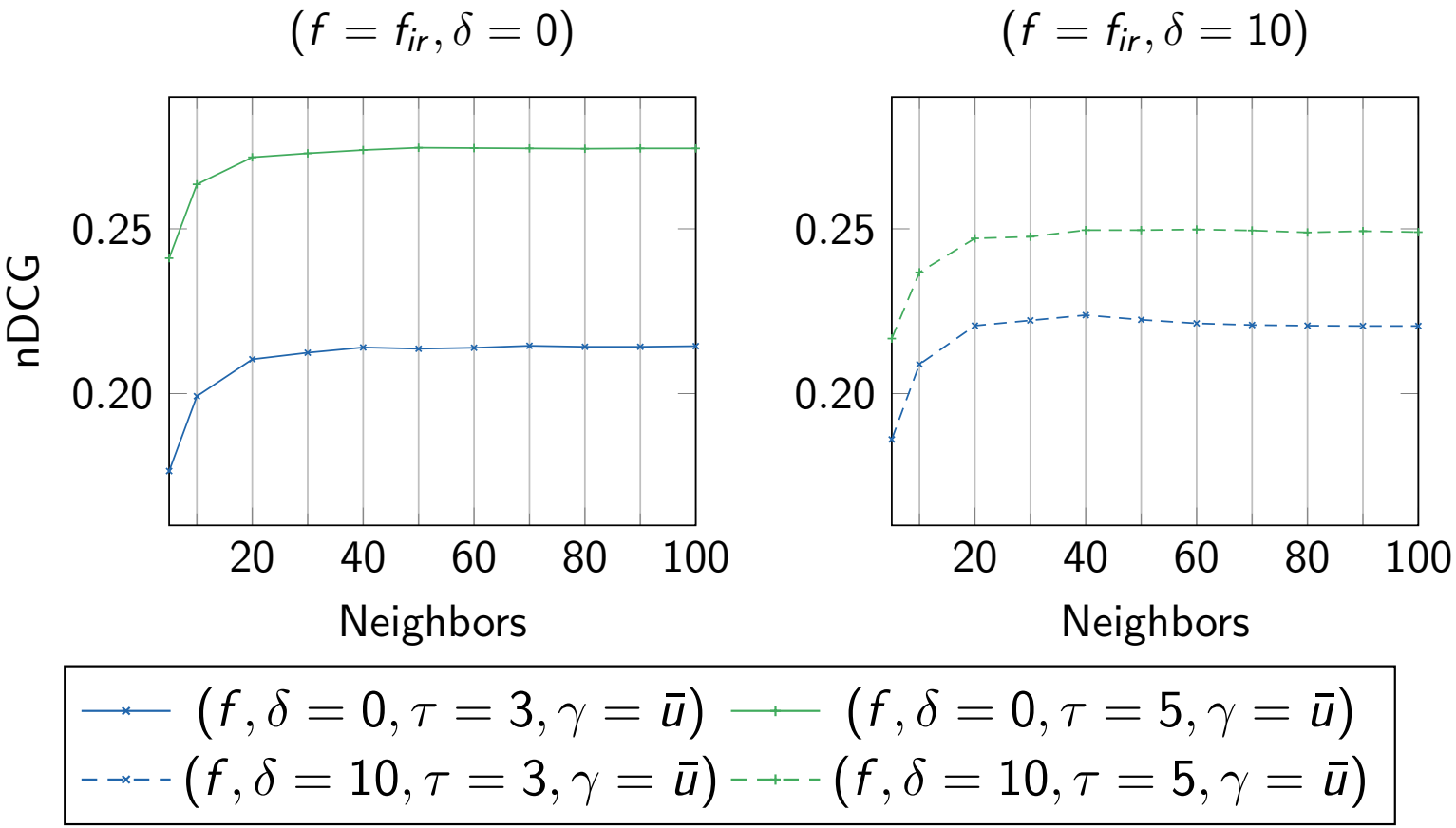
Gracias por vuestra atención

El uso de threshold en este caso mejora los resultados



$(\text{sim}_1, f, \delta = 0)$ $(\text{sim}_1, f, \delta = 10)$

Preferencia y confianza aumentan el rendimiento



Pero reducen cobertura

Recommender	**Coverage**
$(f_{ir}, 0, 3, \overline{u})$	$1.587,6$
$(f_{ir}, 0, 5, \overline{u})$	$700,6$
$(f_{ir}, 10, 3, \overline{u})$	$1.771,0$
$(f_{ir}, 10, 5, \overline{u})$	$1.243,2$

Incluso en versiones puramente colaborativas LCS es un recomendador competitivo

Recommender	nDCG	P	R	MAP	EPC	EPD	AD	α-nDCG	EILD	Gini
Pop	0.082	0.040	0.093	0.060	0.792	**0.922**	1.35%	0.064	**0.933**	0.000
UB1	0.223	0.106	0.246	0.172	0.883	0.895	53.60%	0.191	0.896	0.006
UB2	0.222	0.106	0.245	0.171	0.883	0.895	52.70%	0.191	0.896	0.005
IB1	0.211	0.101	0.235	0.162	0.913	0.732	81.93%	0.171	0.721	0.027
IB2	0.214	0.100	0.231	0.167	0.912	0.694	**86.81%**	0.175	0.681	**0.034**
MF	**0.261**	**0.123**	**0.288**	**0.203**	**0.925**	0.870	26.77%	**0.223**	0.874	0.014
$(\text{sim}_1, f_{ir}, 0)$	0.199	0.094	0.219	0.154	0.866	0.906	49.72%	0.171	0.906	0.004
$(\text{sim}_2, f_{ir}, 0)$	0.204	0.096	0.223	0.157	0.868	0.900	56.99%	0.174	0.899	0.004
$(\text{sim}_1, f_{ir}, 10)$	0.215	0.102	0.237	0.166	0.873	0.901	47.90%	0.186	0.902	0.004
$(\text{sim}_2, f_{ir}, 10)$	0.222	0.106	0.245	0.171	0.879	0.887	59.22%	0.190	0.887	0.006

Géneros son demasiado amplios

- Godzilla: Acción, Sci-Fi, Thriller
- Alien: Acción, Horror, Sci-Fi, Thriller
- Soy leyenda: Acción, Horror, Sci-Fi, Thriller

Todos estos artículos contienen prácticamente los mismos géneros, dando lugar a secuencias prácticamente iguales.

Pearson en LCS

1: **procedure** LCS_RecSys2($u, v, f, \delta, \vec{r}_u, \vec{r}_v$) ▷ La LCS de los usuarios u y v aplicando la transformación f, usando sus ratings $\vec{r}_u$ y $\vec{r}_v$
2: $(x, y) \leftarrow (f(u), f(v))$ ▷ La cadena x contiene m símbolos
3: $(m_u, m_v) \leftarrow (\text{avg}(\vec{r}_u), \text{avg}(\vec{r}_v))$ ▷ Media de los ratings
4: $L[0 \cdots m, 0 \cdots n] \leftarrow 0$
5: $(s, s_u, s_v) \leftarrow 0$
6: **for** $i \leftarrow 1, m$ **do**
7: **for** $j \leftarrow 1, n$ **do**
8: $(t_u, t_v) \leftarrow 0$
9: **if** $\texttt{match}(x_i, y_j, \delta)$ **then**
10: $L[i, j] \leftarrow L[i-1, j-1] + 1$ ▷ Hay una concidencia-δ
11: $t_u \leftarrow r_{ui} - m_u$
12: $t_v \leftarrow r_{vj} - m_v$
13: **else**
14: $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$
15: **end if**
16: $s \leftarrow s + t_u \cdot t_v$
17: $s_u \leftarrow s_u + t_u \cdot t_u$
18: $s_v \leftarrow s_v + t_v \cdot t_v$
19: **end for**
20: **end for**
21: **return** $s / \sqrt{s_u \cdot s_v}$
22: **end procedure**

Resultados para MovieTweetings. Backward-Forward. Baselines.

Recommender	nDCG	P	R	MAP	EPC	EPD	AD	α-nDCG	EILD	Gini
Pop	0.003	0.006	0.003	0.001	0.938	**0.393**	1.84%	0.006	**0.751**	0.000
UB1	0.011	0.016	0.006	0.003	0.459	0.330	39.05%	0.008	0.305	0.007
UB2	0.010	0.015	0.006	0.003	0.463	0.333	35.07%	0.008	0.311	0.007
IB1	0.009	0.015	0.006	0.003	0.484	0.331	**61.10%**	0.007	0.299	0.015
IB2	0.009	0.016	0.006	0.003	0.484	0.334	43.68%	0.008	0.311	**0.020**
MF	0.006	0.009	0.004	0.002	**0.986**	0.329	11.76%	0.009	0.517	0.003
Fossil	0.008	0.012	0.004	0.002	0.425	0.324	14.08%	0.005	0.327	0.001
MC	0.013	0.021	0.008	0.004	0.428	0.323	26.61%	0.011	0.312	0.002
$(\text{sim}_2, f_{ir}, 20, s_T)$	**0.031**	**0.058**	**0.021**	**0.011**	0.484	0.353	30.55%	**0.031**	0.288	0.002
$(\text{sim}_2, f_{ir}, 10, s_T)$	0.030	0.058	0.020	0.011	0.483	0.352	32.25%	0.029	0.286	0.002

Test			**Recommendation**		
User	Item	Rating	User	Item	Score
175	2	5	175	7	5
175	20	4	175	2	4,5
190	36	5	175	10	4
190	47	4	190	36	5
190	6	5	190	10	4,8
267	2	5	190	47	4
267	7	5	267	2	5
267	36	3	267	11	4
			267	10	3

Precision

$$\text{Precision} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Retrieved}|} = \frac{\frac{1}{3} + \frac{2}{3} + \frac{1}{3}}{3} = 0,4444 \quad (3)$$

Test			**Recommendation**		
User	Item	Rating	User	Item	Score
175	2	5	175	7	5
175	20	4	175	2	4,5
190	36	5	175	10	4
190	47	4	190	36	5
190	6	5	190	10	4,8
267	2	5	190	47	4
267	7	5	267	2	5
267	36	3	267	11	4
			267	10	3

Recall

$$\text{Recall} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Relevant}|} = \frac{\frac{1}{2} + \frac{2}{3} + \frac{1}{3}}{3} = 0,5 \quad (3)$$

Ejemplo de métricas. Evaluación de ránking

Test			**Recommendation**		
User	Item	Rating	User	Item	Score
175	2	5	175	7	5
175	20	4	175	2	4,5
190	36	5	175	10	4
190	47	4	190	36	5
190	6	5	190	10	4,8
267	2	5	190	47	4
267	7	5	267	2	5
267	36	3	267	11	4
			267	10	3

MAP

$$\text{AP} = \frac{1}{|\text{Relevant}|} \sum_{\{k:d_k \in \text{Relevant}\}} P@k = \frac{\frac{\frac{1}{2}}{2} + \frac{1+\frac{2}{3}}{3} + \frac{1}{3}}{3} = 0,3796 \qquad (3)$$

Ejemplo de métricas. Evaluación de ránking

Test			**Recommendation**		
User	Item	Rating	User	Item	Score
175	2	5	175	7	5
175	20	4	175	2	4,5
190	36	5	175	10	4
190	47	4	190	36	5
190	6	5	190	10	4,8
267	2	5	190	47	4
267	7	5	267	2	5
267	36	3	267	11	4
			267	10	3

NDCG

$$\text{DCG@}p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} = \frac{\frac{31}{31+19{,}55+3.5} + \frac{31+7.5}{31+19{,}55+7{,}5} + \frac{19.55}{31+9.46}}{3} = 0,573 \quad (3)$$