Applying Subsequence Matching to Collaborative Filtering

Alejandro Bellogín **Pablo Sánchez**

Universidad Autónoma de Madrid
Escuela Politécnica Superior
Departamento de Ingeniería Informática

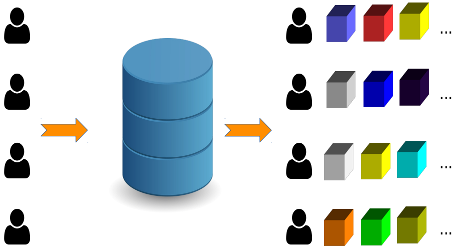
V Congreso Español de
Recuperación de Información (CERI 2018)

Outline

1. Recommender Systems
2. Sequential similarities
3. Experiments
4. Conclusions and future work

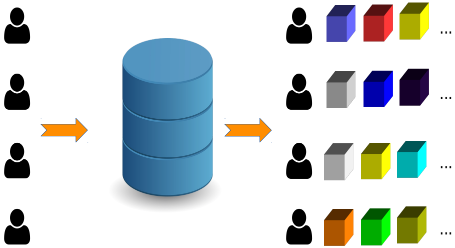
Outline

Recommender Systems



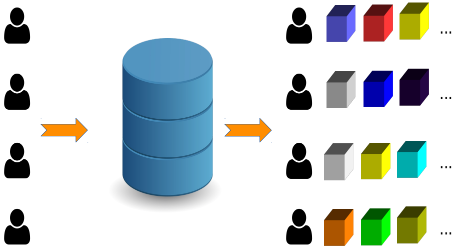
- Suggest **new items** to users based on their tastes and needs

Recommender Systems



- Suggest **new items** to users based on their tastes and needs
- Different methods to make recommendations (content-based, collaborative filtering, hybrids)

Recommender Systems



- Suggest **new items** to users based on their tastes and needs
- Different methods to make recommendations (content-based, collaborative filtering, hybrids)
- We will focus on neighborhood based collaborative filtering algorithms

Collaborative filtering

	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	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Collaborative filtering

	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	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

- Normally the User $\times$ Item matrix is very sparse (90%-99% of empty values)

Collaborative filtering

	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	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

- Normally the User $\times$ Item matrix is very sparse (90%-99% of empty values)
- Collaborative filtering try to fill the matrix either with latent factor models or neighborhood approaches

Collaborative filtering

Matrix factorization techniques

$$\min_{p*,q*} \sum_{u,i \in R} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \quad (1)$$

- q_i and p_u are the latent vectors of the user u and the item i
- R denotes all the training samples
- λ is the regularization parameter

Collaborative filtering

Matrix factorization techniques

$$\min_{p*,q*} \sum_{u,i \in R} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \quad (1)$$

- q_i and p_u are the latent vectors of the user u and the item i
- R denotes all the training samples
- λ is the regularization parameter

Neighborhood approaches

$$s_{u,i} \propto \sum_{v \in N_i(u)} w_{uv} r_{vi} \quad (2)$$

- r_{vi} is the rating of the neighbour v
- w_{uv} is the similarity between user u and v

Collaborative filtering

Matrix factorization techniques

$$\min_{p*,q*} \sum_{u,i \in R} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \quad (1)$$

- q_i and p_u are the latent vectors of the user u and the item i
- R denotes all the training samples
- λ is the regularization parameter

Neighborhood approaches

$$s_{u,i} \propto \sum_{v \in \boxed{N_i(u)}} \boxed{w_{uv}} r_{vi} \quad (2)$$

- r_{vi} is the rating of the neighbour v
- w_{uv} is the similarity between user u and v

Classic similarities

Pearson correlation

$$\mathrm{PC}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{3}$$

Cosine similarity

$$\cos(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}} \tag{4}$$

Jaccard index

$$\text{Jaccard(u,v)} = \frac{|I_u \cap I_v|}{|I_u \cup I_u|} \tag{5}$$

Outline

Longest Common Subsequence (LCS)

- Find the longest common subsequence (list of elements not necessary consecutive but maintaining the order) between 2 strings X and Y
- Used in DNA sequencing and file comparison
- Can be resolved applying dynamic programming filling a matrix of size $(|X| + 1) \times (|Y| + 1)$

Longest Common Subsequence (LCS)

- Find the longest common subsequence (list of elements not necessary consecutive but maintaining the order) between 2 strings X and Y
- Used in DNA sequencing and file comparison
- Can be resolved applying dynamic programming filling a matrix of size $(|X|+1) \times (|Y|+1)$

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j > 0 \text{ and } X_i \neq Y_j \end{cases} \quad (6)$$

- The last position in the matrix contains the length of the longest common subsequence

Longest Common Subsequence (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{7}$$

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

Longest Common Subsequence (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	$\emptyset$	A	G	G	T	A	C
$\emptyset$	0	0	0	0	0	0	0
G							
C							
G							
T							
G							
C							

Longest Common Subsequence (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j > 0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{7}$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j > 0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j > 0 \text{ and } X_i \neq Y_j \end{cases} \tag{7}$$

	∅	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 (LCS)

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (7)$$

	∅	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

Adapting LCS to Recommender Systems

- Users interactions can be interpreted as sequences of interactions

Adapting LCS to Recommender Systems

- Users interactions can be interpreted as sequences of interactions
- Different transformation functions on the user ratings $I(u)$:

Adapting LCS to Recommender Systems

- Users interactions can be interpreted as sequences of interactions
- Different transformation functions on the user ratings $I(u)$:
  - Using the item, i.e., $f_i : I(u) \rightarrow \Sigma = \mathcal{I}, f_i(x) = x(i)$.
  - Using the value of the interaction, i.e., $f_r : I(u) \rightarrow \mathcal{R}, f_r(x) = x(r)$.
  - Using a combination of the item and the interaction value, i.e., $f_{ir} : I(u) \rightarrow \mathcal{I} \times \mathcal{R}, f_{ir}(x) = (x(i), x(r))$.

Adapting LCS to Recommender Systems

- Users interactions can be interpreted as sequences of interactions
- Different transformation functions on the user ratings $I(u)$:
    - Using the item, i.e., $f_i : I(u) \to \Sigma = \mathcal{I}, f_i(x) = x(i)$.
    - Using the value of the interaction, i.e., $f_r : I(u) \to \mathcal{R}, f_r(x) = x(r)$.
    - Using a combination of the item and the interaction value, i.e., $f_{ir} : I(u) \to \mathcal{I} \times \mathcal{R}, f_{ir}(x) = (x(i), x(r))$.
- We used integers as symbols for the transformations

Adapting LCS to Recommender Systems

- Users interactions can be interpreted as sequences of interactions
- Different transformation functions on the user ratings $I(u)$:
  - Using the item, i.e., $f_i : I(u) \rightarrow \Sigma = \mathcal{I}, f_i(x) = x(i)$.
  - Using the value of the interaction, i.e., $f_r : I(u) \rightarrow \mathcal{R}, f_r(x) = x(r)$.
  - Using a combination of the item and the interaction value, i.e., $f_{ir} : I(u) \rightarrow \mathcal{I} \times \mathcal{R}, f_{ir}(x) = (x(i), x(r))$.
- We used integers as symbols for the transformations
- These transformations generate a pure collaborative filtering approach but they are easily extensible to use content information

Toy example

	$\times$	$\bigcirc$	$\triangle$	$\square$	$\diamond$
u	4		5	3	1
v	4	5		4	4

Table: Interaction (ratings) data between two users and five items.

f	$f(u)$	$f(v)$
f_i	$(\times, \triangle, \square, \diamond)$	$(\times, \bigcirc, \square, \diamond)$
f_r	(4,5,3,1)	(4,5,4,4)
f_{ir}	$(\times 4, \triangle 5, \square 3, \diamond 1)$	$(\times 4, \bigcirc 5, \square 4, \diamond 4)$

Table: Representation of the interactions for different transformation functions

Adapting LCS to Recommender Systems

- Different possible orderings for the users sequences. As a first approach, we will order the items by id

Adapting LCS to Recommender Systems

- Different possible orderings for the users sequences. As a first approach, we will order the items by id
- Allow differences in users rating by a threshold (δ)

Adapting LCS to Recommender Systems

- Different possible orderings for the users sequences. As a first approach, we will order the items by id
- Allow differences in users rating by a threshold (δ)
- Normalize the value LCS in the [0, 1] interval

$$sim_1^{f,\delta} = \text{LCS\_CF}(u, v, f, \delta) \tag{8}$$

$$sim_2^{f,\delta} = (sim_1^{f,\delta})^2 / (|f(u)| \cdot |f(v)|) \tag{9}$$

Adapting LCS to Recommender Systems

- Different possible orderings for the users sequences. As a first approach, we will order the items by id
- Allow differences in users rating by a threshold (δ)
- Normalize the value LCS in the [0, 1] interval

$$sim_1^{f,\delta} = \text{LCS\_CF}(u, v, f, \delta) \tag{8}$$

$$sim_2^{f,\delta} = (sim_1^{f,\delta})^2 / (|f(u)| \cdot |f(v)|) \tag{9}$$

- Using the pure item transformation (f_i) and a global ordering, we obtain an equivalence with the binary cosine:

$$cos_b(u, v) = |I(u, v)| / \sqrt{(|f(u)| \cdot |f(v)|)} \tag{10}$$

Adapting LCS to Recommender Systems

- Different possible orderings for the users sequences. As a first approach, we will order the items by id
- Allow differences in users rating by a threshold (δ)
- Normalize the value LCS in the [0, 1] interval

$$sim_1^{f,\delta} = \mathrm{LCS\_CF}(u, v, f, \delta) \tag{8}$$

$$sim_2^{f,\delta} = (sim_1^{f,\delta})^2 / (|f(u)| \cdot |f(v)|) \tag{9}$$

- Using the pure item transformation (f_i) and a global ordering, we obtain an equivalence with the binary cosine:

$$cos_b(u, v) = |I(u, v)| / \sqrt{(|f(u)| \cdot |f(v)|)} \tag{10}$$

- For more information, see **Bellogín and Sánchez (2017)**

Toy example

Movie (id)	**Director (id)**	**Genres (ids)**	u_1	u_2
The Wild Bunch (M1)	Sam Peckinpah (D1)	$\left.\begin{array}{l}\text{Western (G1)}\\\text{Robbery (G2)}\end{array}\right\}$	5	
Seven Samurais (M2)	Akira Kurosawa (D2)	$\left.\begin{array}{l}\text{Action (G3)}\\\text{Drama (G4)}\\\text{Adventure (G5)}\end{array}\right\}$	4	5
The Iron Cross (M3)	Sam Peckinpah (D1)	War (G6)}	3	
Gladiator (M4)	Riddley Scott (D3)	$\left.\begin{array}{l}\text{Action (G3)}\\\text{Drama (G4)}\\\text{Adventure (G5)}\end{array}\right\}$	4	2
Alien (M5)	Riddley Scott (D3)	$\left.\begin{array}{l}\text{Sci-Fi (G7)}\\\text{Terror (G8)}\end{array}\right\}$		5
The Magnificent Seven (M8)	John Sturges (D4)	$\left.\begin{array}{l}\text{Western (G1)}\\\text{Adventure (G5)}\end{array}\right\}$		4

- $f_i : u_1 = (1, 2, 3, 4), u_2 = (2, 4, 5, 8)$
- $f_{ir} : u_1 = (15, 24, 33, 44), u_2 = (25, 42, 55, 84)$

Toy example

Movie (id)	Director (id)	Genres (ids)	u_1	u_2
The Wild Bunch (M1)	Sam Peckinpah (D1)	Western (G1) Robbery (G2)	5	
Seven Samurais (M2)	Akira Kurosawa (D2)	Action (G3) Drama (G4) Adventure (G5)	4	5
The Iron Cross (M3)	Sam Peckinpah (D1)	War (G6)	3	
Gladiator (M4)	Riddley Scott (D3)	Action (G3) Drama (G4) Adventure (G5)	4	2
Alien (M5)	Riddley Scott (D3)	Sci-Fi (G7) Terror (G8)		5
The Magnificent Seven (M8)	John Sturges (D4)	Western (G1) Adventure (G5)		4

- $f_i : u_1 = (1, 2, 3, 4), u_2 = (2, 4, 5, 8)$
  - $sim_1 = 2, sim_2 = 0.25$
- $f_{ir} : u_1 = (15, 24, 33, 44), u_2 = (25, 42, 55, 84)$
  - $\delta = 1, sim_1 = 1, sim_2 = 1/16$
  - $\delta = 0, sim_1 = 0, sim_2 = 0$

Outline

Experiments

Table: Statistics about the datasets used in the experiments.

Dataset	users	items	ratings	Density
Lastfm HetRec	1, 892	17, 632	92, 834	0.28%
MovieLens HetRec	2, 113	10, 197	855, 598	3.97%

- 5-fold cross-validation
- Analyze both relevance (Precision, MAP, nDCG and Recall) and novelty and diversity, cutoff @5
- Reported results from RankSys and Mahout frameworks
- Different baselines analyzed: Popularity, UB (different similarities, including JMSD from **Bobadilla et al. (2010)**), IB (different similarities), MF (HKV version from **Hu et al. (2008)**)

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance
- Usually better than other UB approaches (actual baselines to beat)

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance
- Usually better than other UB approaches (actual baselines to beat)
- Good tradeoff between novelty, diversity, and relevance

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance
- Usually better than other UB approaches (actual baselines to beat)
- Good tradeoff between novelty, diversity, and relevance
- Our approach is highly competitive in the Lastfm dataset, being able to beat all recommenders except for the HKV in terms of relevance

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance
- Usually better than other UB approaches (actual baselines to beat)
- Good tradeoff between novelty, diversity, and relevance
- Our approach is highly competitive in the Lastfm dataset, being able to beat all recommenders except for the HKV in terms of relevance
- In Movielens, LCS is better than most baselines, except for the HKV and two UB approaches

Experiments

- In general, applying the normalization in LCS brings better results in terms of relevance
- Usually better than other UB approaches (actual baselines to beat)
- Good tradeoff between novelty, diversity, and relevance
- Our approach is highly competitive in the Lastfm dataset, being able to beat all recommenders except for the HKV in terms of relevance
- In Movielens, LCS is better than most baselines, except for the HKV and two UB approaches
- Very different performance between RankSys and Mahout frameworks

Outline

Conclusions

- We have defined a new UB similarity based on the LCS algorithm

Conclusions

- We have defined a new UB similarity based on the LCS algorithm
- We have shown that the basic approach is equivalent to the binary cosine similarity metric

Conclusions

- We have defined a new UB similarity based on the LCS algorithm
- We have shown that the basic approach is equivalent to the binary cosine similarity metric
- Our approach is competitive in two datasets with respect to other state-of-the-art algorithms in relevance, novelty, and diversity metrics

Conclusions

- We have defined a new UB similarity based on the LCS algorithm
- We have shown that the basic approach is equivalent to the binary cosine similarity metric
- Our approach is competitive in two datasets with respect to other state-of-the-art algorithms in relevance, novelty, and diversity metrics
- Our LCS-based similarity can be easily extended to use content-based and temporal information allowing us to model the user profiles better

Future work

- The LCS-based similarity may incorporate repetitions in a natural way

Future work

- The LCS-based similarity may incorporate repetitions in a natural way
- Perform experiments considering both content-based and temporal information

Future work

- The LCS-based similarity may incorporate repetitions in a natural way
- Perform experiments considering both content-based and temporal information
- The LCS algorithm can be also used in evaluation, to assess the quality of the recommendations when considering the ordering of the user interactions in the test set

Applying Subsequence Matching to Collaborative Filtering

Alejandro Bellogín **Pablo Sánchez**

Universidad Autónoma de Madrid
Escuela Politécnica Superior
Departamento de Ingeniería Informática

V Congreso Español de
Recuperación de Información (CERI 2018)

Thank you

Experiments. Lastfm: RankSys

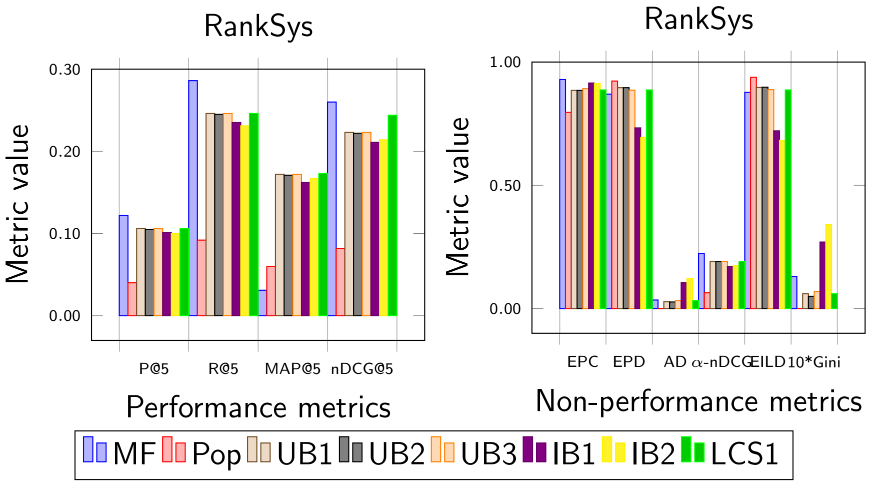


Figure: Performance results in the Lastfm dataset for RankSys framework.

Experiments. Lastfm: Mahout

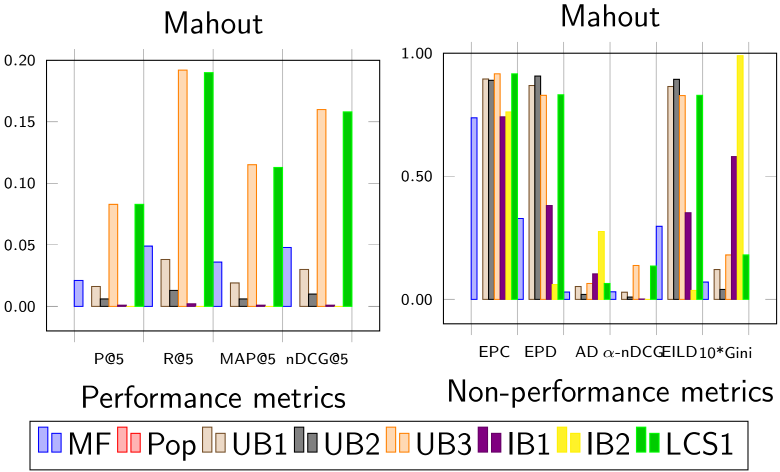


Figure: Performance results in the Lastfm dataset for Mahout framework.

Experiments. Movielens: RankSys

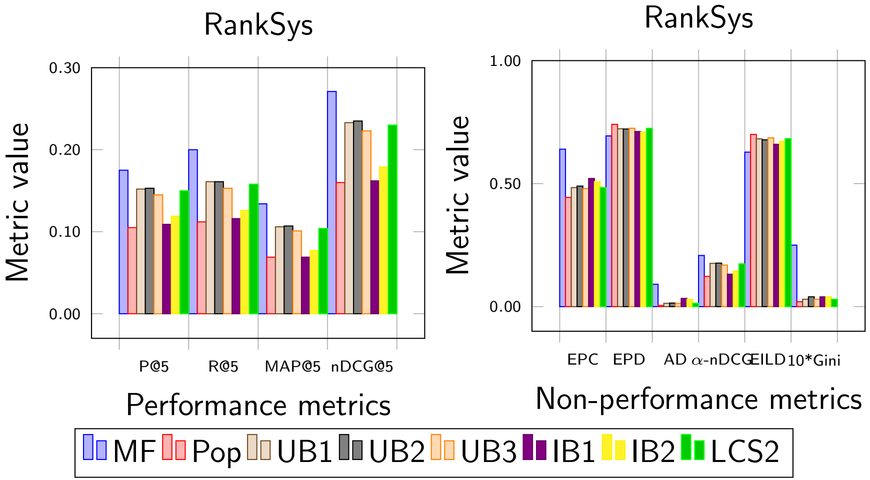
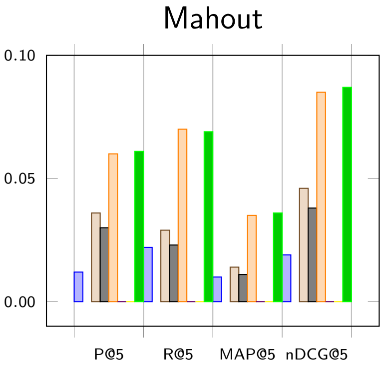
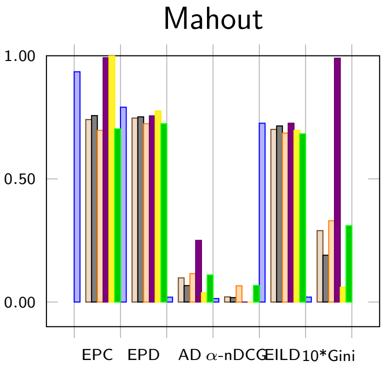


Figure: Performance results in the MovieLens dataset for RankSys framework.

Experiments. Movielens: Mahout



Performance metrics



Non-performance metrics

Figure: Performance results in the MovieLens dataset for RankSys framework.

References I

Bellogín, A. and Sánchez, P. (2017). Collaborative filtering based on subsequence matching: A new approach. *Inf. Sci.*, 418:432–446.

Bobadilla, J., Serradilla, F., and Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowl.-Based Syst.*, 23(6):520–528.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272. IEEE Computer Society.