UAM
Universidad Autónoma
de Madrid

Exploring attributes, sequences, and time in Recommender Systems: From classical to Point-of-Interest recommendation

Pablo Sánchez Pérez

Under the supervision of
Alejandro Bellogín Kouki

Information Retrieval Group
Department of Computer Science
Universidad Autónoma de Madrid, Spain

July 8, 2021

Overview

Overview

Introduction

Introduction

amazon

NETFLIX

Spotify

STEAM

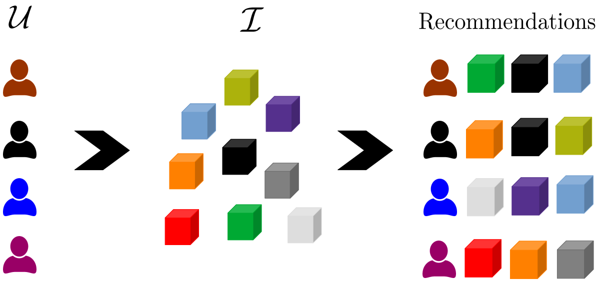
twitter

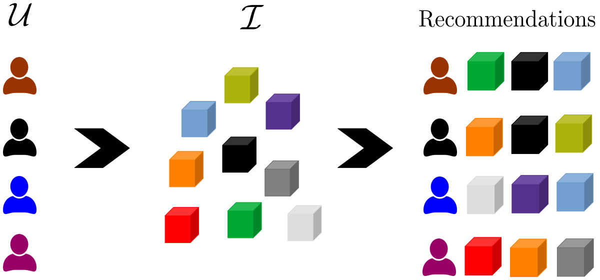
yelp

YouTube

LinkedIn

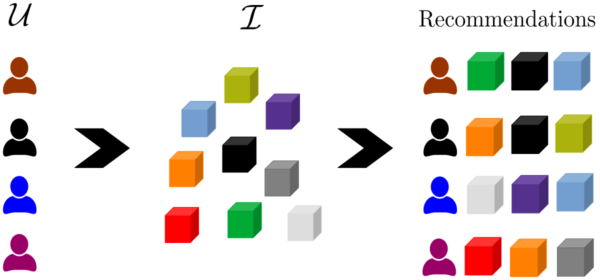
foursquare


$\mathcal{U}$
$\mathcal{I}$
Recommendations



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

Sparsity $\sim$ 99%



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

Objective: maximize the **usefulness** of the items for the target user ($\max \boldsymbol{g(u,i)}$)

Recommender Systems

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u, i) \tag{1}$$

—[Adomavicius and Tuzhilin, 2005]

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u, i) \tag{1}$$

—[Adomavicius and Tuzhilin, 2005]

- Domains: **movies**, **Point-of-Interest**, music, dates, ...

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u, i) \tag{1}$$

—[Adomavicius and Tuzhilin, 2005]

- Domains: **movies**, **Point-of-Interest**, music, dates, ...
- Types of Recommender Systems: **content-based, collaborative filtering, hybrids**, demographic, knowledge-based

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u, i) \tag{1}$$

—[Adomavicius and Tuzhilin, 2005]

- Domains: **movies**, **Point-of-Interest**, music, dates, ...
- Types of Recommender Systems: **content-based, collaborative filtering, hybrids**, demographic, knowledge-based
- Contextual information: **temporal, sequential, geographical**, social, weather, ...

Recommender Systems

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u,i) \qquad (1)$$

—[Adomavicius and Tuzhilin, 2005]

- Domains: **movies**, **Point-of-Interest**, music, dates, ...
- Types of Recommender Systems: **content-based, collaborative filtering, hybrids**, demographic, knowledge-based
- Contextual information: **temporal, sequential, geographical**, social, weather, ...
- Evaluation: **offline**, online, user studies

$$i^*(u) = \arg\max_{i \in \mathcal{I}} g(u, i) \quad (1)$$

—[Adomavicius and Tuzhilin, 2005]

- Domains: **movies**, **Point-of-Interest**, music, dates, ...
- Types of Recommender Systems: **content-based, collaborative filtering, hybrids**, demographic, knowledge-based
- Contextual information: **temporal, sequential, geographical**, social, weather, ...
- Evaluation: **offline**, online, user studies
- More information: **Chapter 2**

Recommender Systems: offline evaluation

Recommender Systems: offline evaluation

- Metrics:
  - Error metrics (rating prediction): MAE, RMSE, ...

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{R}_{test}|} \sum_{r_{ui} \in \mathcal{R}_{test}} (\hat{r}(u,i) - r_{ui})^2} \tag{2}$$

Recommender Systems: offline evaluation

- Metrics:
  - Error metrics (rating prediction): MAE, RMSE, ...
  - Ranking Accuracy (top-n evaluation): Precision (P), Recall (R), nDCG, ...

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{R}_{test}|} \sum_{r_{ui} \in \mathcal{R}_{test}} (\hat{r}(u,i) - r_{ui})^2} \tag{2}$$

$$\mathrm{P@k(u)} = \frac{\mathrm{Rel_u@k}}{\mathrm{k}} \tag{3}$$

Recommender Systems: offline evaluation

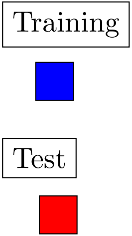
- Metrics:
  - Error metrics (rating prediction): MAE, RMSE, ...
  - Ranking Accuracy (top-n evaluation): Precision (P), Recall (R), nDCG, ...
  - Novelty and diversity: Item Coverage (IC), Gini, EPC, ...

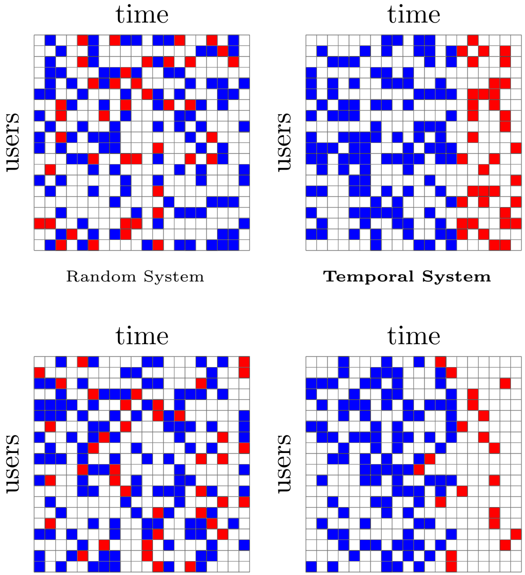
$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{R}_{test}|} \sum_{r_{ui} \in \mathcal{R}_{test}} (\hat{r}(u,i) - r_{ui})^2} \tag{2}$$

$$\mathrm{P@k(u)} = \frac{\mathrm{Rel_u@k}}{\mathrm{k}} \tag{3}$$

$$\mathrm{IC} = \left| \bigcup_{u \in \mathcal{U}} R_u \right| \tag{4}$$

Recommender Systems: types of data splitting





Random System

Temporal System

Random Per User

Temporal Per User

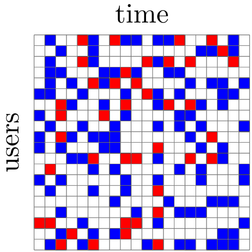
- Data splitting
  - Random vs Temporal
  - Per user vs system

Recommender Systems: types of data splitting

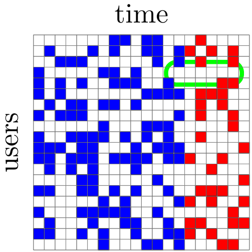
Training

Test

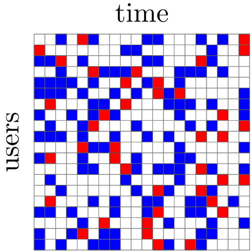
- Data splitting
  - Random vs Temporal
  - Per user vs system



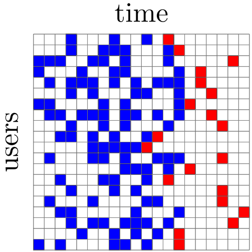
Random System



Temporal System



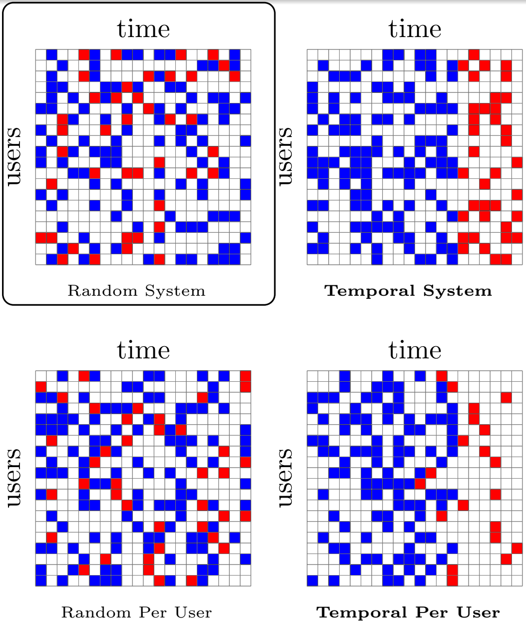
Random Per User



Temporal Per User

Training

Test



- Data splitting
  - Random vs Temporal
  - Per user vs system

Research Objectives

Research Objectives

- RO1: Integrating **additional dimensions beyond relevance** in evaluation metrics
  - We use temporal information, attributes, and low ratings for evaluating the recommenders
  - We obtain more complete results of the performance of the recommenders and we detect additional biases

- RO1: Integrating **additional dimensions beyond relevance** in evaluation metrics
  - We use temporal information, attributes, and low ratings for evaluating the recommenders
  - We obtain more complete results of the performance of the recommenders and we detect additional biases
- RO2: Incorporate **sequentiality** in **neighborhood-based recommenders**
  - We develop a sequential similarity metric and we redefine the formulation of k-NN recommenders
  - Our approaches are highly competitive in relevance, novelty and diversity

Research Objectives

- RO3: Review the **state-of-the-art on Point-of-Interest Recommender Systems**
  - We characterize POI recommendation works between 2011 and 2019 analyzing the algorithms, the information, and the evaluation methodology used

Research Objectives

- RO3: Review the **state-of-the-art on Point-of-Interest Recommender Systems**
  - We characterize POI recommendation works between 2011 and 2019 analyzing the algorithms, the information, and the evaluation methodology used
- RO4: **Improve the performance** of **POI** recommendation algorithms
  - We propose multi-city aggregation strategies to augment the information of the recommenders
  - We improve the performance of most recommenders by selecting the cities by proximity

Research Objectives

- RO3: Review the **state-of-the-art on Point-of-Interest Recommender Systems**
  - We characterize POI recommendation works between 2011 and 2019 analyzing the algorithms, the information, and the evaluation methodology used
- RO4: **Improve the performance** of **POI** recommendation algorithms
  - We propose multi-city aggregation strategies to augment the information of the recommenders
  - We improve the performance of most recommenders by selecting the cities by proximity
- RO5: **Generate full sequences** from **Location-Based Social Networks** data
  - We will use reranking techniques to generate routes from independent POIs
  - We demonstrate how we can improve the recommendations across different dimensions (category and/or distance) using our reranking approaches

Overview

Classical Recommender Systems evaluation

- Most **Recommender Systems** are only evaluated by **measuring the relevance** of the recommendations

Classical Recommender Systems evaluation

- Most **Recommender Systems** are only evaluated by **measuring the relevance** of the recommendations
- **Additional information** (temporal, sequential, ratings, etc.) is being used by most recommenders, but not in the **evaluation step**

Classical Recommender Systems evaluation

- Most **Recommender Systems** are only evaluated by **measuring the relevance** of the recommendations
- **Additional information** (temporal, sequential, ratings, etc.) is being used by most recommenders, but not in the **evaluation step**
- We focus on the first objective: we will show some examples of **incorporating additional information for measuring the quality** of the recommendations

Classical Recommender Systems evaluation

- Most **Recommender Systems** are only evaluated by **measuring the relevance** of the recommendations
- **Additional information** (temporal, sequential, ratings, etc.) is being used by most recommenders, but not in the **evaluation step**
- We focus on the first objective: we will show some examples of **incorporating additional information for measuring the quality** of the recommendations
    1. Temporal information
    2. Anti-relevance models
    3. User and item attributes

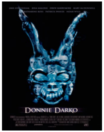
Classical Recommender Systems evaluation

- Most **Recommender Systems** are only evaluated by **measuring the relevance** of the recommendations
- **Additional information** (temporal, sequential, ratings, etc.) is being used by most recommenders, but not in the **evaluation step**
- We focus on the first objective: we will show some examples of **incorporating additional information for measuring the quality** of the recommendations
  1. Temporal information
  2. Anti-relevance models
  3. User and item attributes
- **Contributions published** in ECIR [Sánchez and Bellogín, 2018b] and RecSys [Sánchez and Bellogín, 2019a, Sánchez and Bellogín, 2018a]

Time-aware novelty metrics

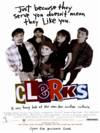
Time-aware novelty metrics

R_1



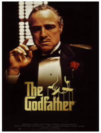
(2001)

(1994)

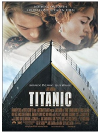


(1994)

R_2



(1972)



(1997)

(1993)

R_3

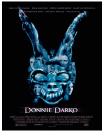
(2018)

(2017)

(2016)

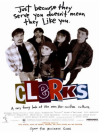
Time-aware novelty metrics

R_1



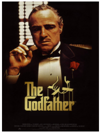
(2001)

(1994)

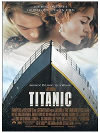


(1994)

R_2



(1972)



(1997)

(1993)

R_3

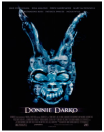
(2018)

(2017)

(2016)

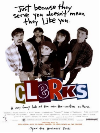
- Best in Relevance?

R_1



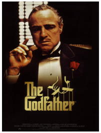
(2001)

(1994)

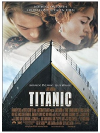


(1994)

R_2



(1972)



(1997)

(1993)

R_3

(2018)

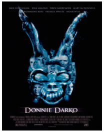
(2017)

(2016)

- Best in Relevance?
  - $R_2 > R_1 > R_3$

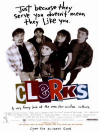
Time-aware novelty metrics

R_1



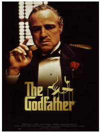
(2001)

(1994)

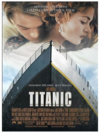


(1994)

R_2



(1972)



(1997)

(1993)

R_3

(2018)

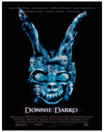
(2017)

(2016)

- Best in Relevance?
  - $R_2 > R_1 > R_3$
- Best in Novelty?

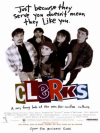
Time-aware novelty metrics

R_1



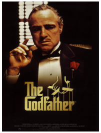
(2001)

(1994)

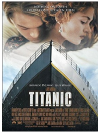


(1994)

R_2



(1972)



(1997)

(1993)

R_3

(2018)

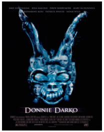
(2017)

(2016)

- Best in Relevance?
  - $R_2 > R_1 > R_3$
- Best in Novelty?
  - $R_1 > R_3 > R_2$

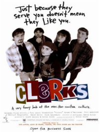
Time-aware novelty metrics

R_1



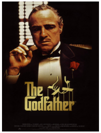
(2001)

(1994)

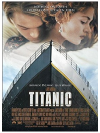


(1994)

R_2



(1972)



(1997)

(1993)

R_3

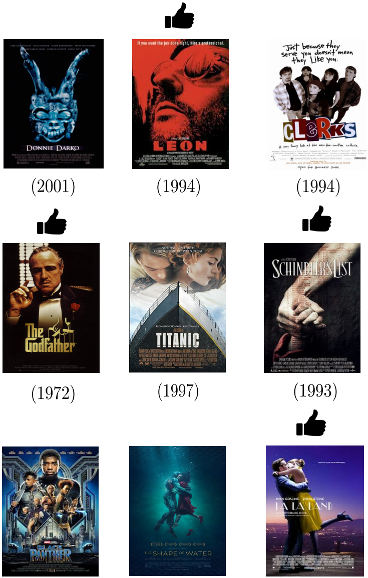
(2018)

(2017)

(2016)

- Best in Relevance?
  - $R_2 > R_1 > R_3$
- Best in Novelty?
  - $R_1 > R_3 > R_2$
- Best in **Temporal novelty**?

Time-aware novelty metrics



R_1: (2001) (1994) (1994)

R_2: (1972) (1997) (1993)

R_3: (2018) (2017) (2016)

- Best in Relevance?
  - $R_2 > R_1 > R_3$
- Best in Novelty?
  - $R_1 > R_3 > R_2$
- Best in **Temporal novelty**?
  - $R_3 > R_1 > R_2$

Time-aware novelty metrics

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \mathrm{nov}(i_n \mid \theta) \tag{5}$$

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \mathrm{nov}(i_n \mid \theta) \qquad (5)$$

- Where:
  - R_u items recommended to user u
  - θ contextual variable (e.g., the user profile)
  - $\mathrm{disc}(n)$ is a discount model (e.g. nDCG)
  - $p(rel \mid i_n, u)$ relevance component
  - $\mathrm{nov}(i_n \mid \theta)$ **novelty model**

Time-aware novelty metrics

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \mathrm{nov}(i_n \mid \theta) \qquad (6)$$

Time-aware novelty metrics

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \mathrm{nov}(i_n \mid \theta) \qquad (6)$$

- For example, when using $\mathrm{nov}(i_n \mid \theta) = (1 - p(\mathrm{seen}|i))$ we obtain the Expected Popularity Complement (EPC) metric

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \mathrm{nov}(i_n \mid \theta) \qquad (6)$$

- For example, when using $\mathrm{nov}(i_n \mid \theta) = (1 - p(\mathrm{seen}|i))$ we obtain the Expected Popularity Complement (EPC) metric
- However, all the metrics derived from this framework are *time-agnostic*

Time-aware novelty metrics

- Probabilistic framework from [Vargas and Castells, 2011]

$$m(R_u \mid \theta_t) = C \sum_{i_n \in R_u} \mathrm{disc}(n) p(rel \mid i_n, u) \boxed{\mathrm{nov}(i_n \mid \theta_t)} \qquad (6)$$

- For example, when using $\mathrm{nov}(i_n \mid \theta) = (1 - p(\mathrm{seen}|i))$ we obtain the Expected Popularity Complement (EPC) metric
- However, all the metrics derived from this framework are *time-agnostic*
- We propose to replace the novelty component defining new **time-aware novelty models**

- Every item in the system can be modeled with a temporal representation:

$$\theta_t = \{\theta_t(i)\} = \{(i, \langle t_1(i), \cdots, t_n(i)\rangle)\} \tag{7}$$

Time-aware novelty metrics

- Every item in the system can be modeled with a temporal representation:

$$\theta_t = \{\theta_t(i)\} = \{(i, \langle t_1(i), \cdots, t_n(i)\rangle)\} \quad (7)$$

- We will use the rating history of the items for generating the temporal representation

- Every item in the system can be modeled with a temporal representation:

$$\theta_t = \{\theta_t(i)\} = \{(i, \langle t_1(i), \cdots, t_n(i)\rangle)\} \tag{7}$$

- We will use the rating history of the items for generating the temporal representation
- We explore 4 aggregation functions: the first interaction (FIN), the last interaction (LIN), the **average of the ratings times (AIN)** and the **median of the ratings times (MIN)**

Time-aware novelty metrics

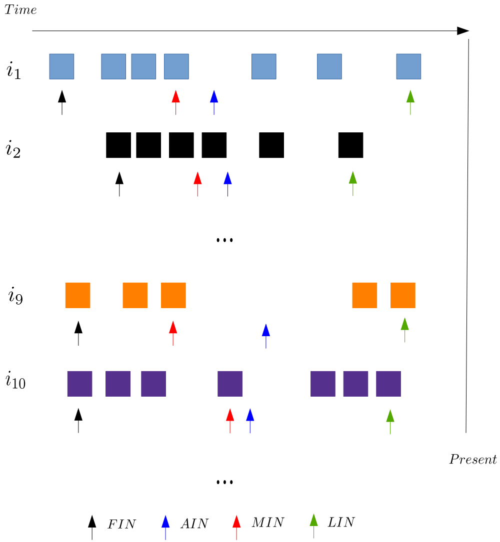
- Every item in the system can be modeled with a temporal representation:

$$\theta_t = \{\theta_t(i)\} = \{(i, \langle t_1(i), \cdots, t_n(i)\rangle)\} \tag{7}$$

- We will use the rating history of the items for generating the temporal representation
- We explore 4 aggregation functions: the first interaction (FIN), the last interaction (LIN), the **average of the ratings times (AIN)** and the **median of the ratings times (MIN)**
- We normalize the values to be suitable for the probabilistic framework (min-max normalization)

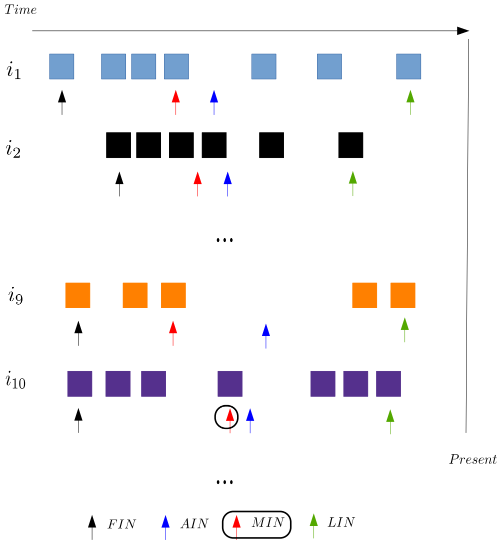
Time-aware novelty metrics

- Differences between the proposed aggregation functions



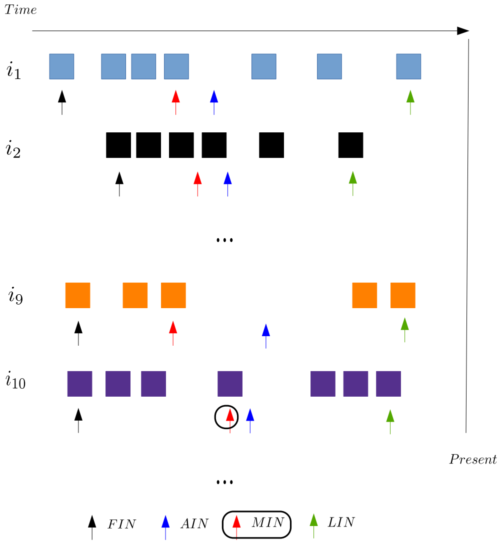
Time-aware novelty metrics

- Differences between the proposed aggregation functions



- MIN

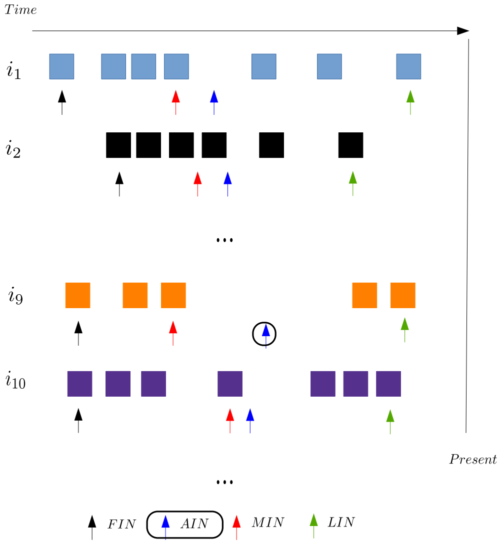
- Differences between the proposed aggregation functions



- MIN
  - $i_{10} > i_2 > i_9 > i_1$

Time-aware novelty metrics

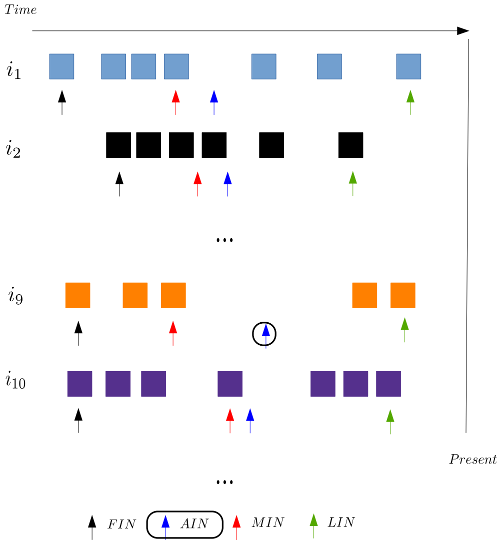
- Differences between the proposed aggregation functions



- MIN
  - $i_{10} > i_2 > i_9 > i_1$
- AIN

Time-aware novelty metrics

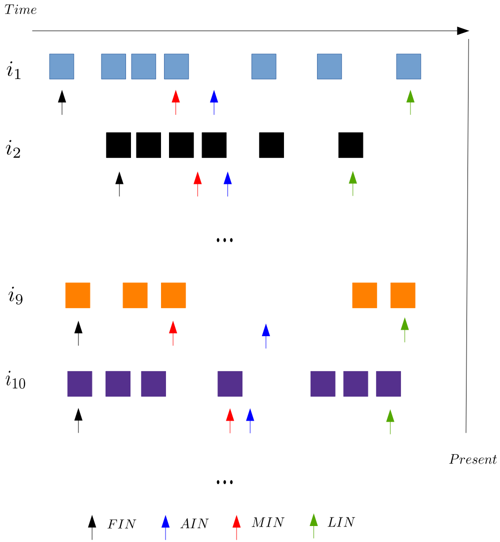
- Differences between the proposed aggregation functions



- MIN
  - $i_{10} > i_2 > i_9 > i_1$
- AIN
  - $i_9 > i_{10} > i_2 > i_1$

Time-aware novelty metrics

- Differences between the proposed aggregation functions



- MIN
  - $i_{10} > i_2 > i_9 > i_1$
- AIN
  - $i_9 > i_{10} > i_2 > i_1$
- FIN
  - $i_2 > i_{10} > i_9 > i_1$
- LIN
  - $i_9 > i_1 > i_{10} > i_2$

Time-aware novelty metrics: a summary

- We propose metrics that **exploit** the **temporal information** of the **interactions of the items**

Time-aware novelty metrics: a summary

- We propose metrics that **exploit** the **temporal information** of the **interactions of the items**
- Our metrics allow us to measure the **temporal novelty** of the items in the system

Time-aware novelty metrics: a summary

- We propose metrics that **exploit** the **temporal information** of the **interactions of the items**
- Our metrics allow us to measure the **temporal novelty** of the items in the system
- Our metrics are **integrated** in a previous defined **novelty framework**

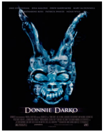
Time-aware novelty metrics: a summary

- We propose metrics that **exploit** the **temporal information** of the **interactions of the items**
- Our metrics allow us to measure the **temporal novelty** of the items in the system
- Our metrics are **integrated** in a previous defined **novelty framework**
- We believe that **AIN** and **MIN** are the strategies that capture better the **temporal evolution** of the items

Anti-relevance metrics

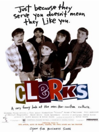
Anti-relevance metrics

R_1



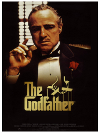
(2001)

(1994)

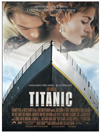


(1994)

R_2



(1972)



(1997)

(1993)

R_3

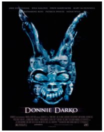
(2018)

(2017)

(2016)

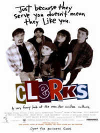
Anti-relevance metrics

R_1



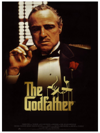
(2001)

(1994)

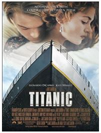


(1994)

R_2



(1972)



(1997)

(1993)

R_3

(2018)

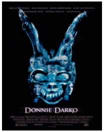
(2017)

(2016)

- Best recommmendation list?

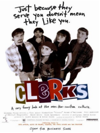
Anti-relevance metrics

R_1



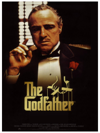
(2001)

(1994)

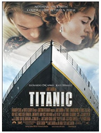


(1994)

R_2



(1972)



(1997)

(1993)

R_3

(2018)

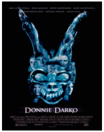
(2017)

(2016)

- Best recommmendation list?
- All lists return 1 relevant item

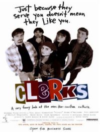
Anti-relevance metrics

R_1



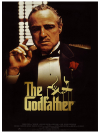
(2001)

(1994)

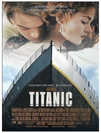


(1994)

R_2



(1972)



(1997)

(1993)

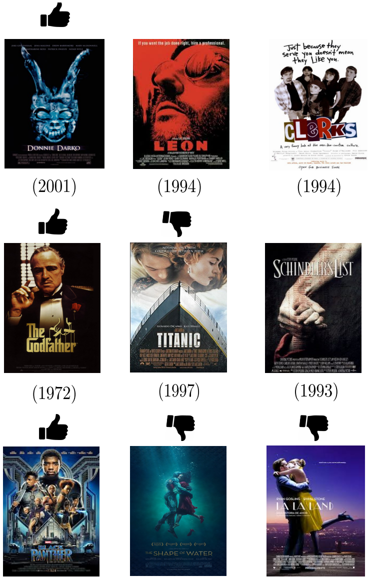
R_3

(2018)

(2017)

(2016)

- Best recommmendation list?
- All lists return 1 relevant item
- But R_3 return 2 bad items



R_1: (2001) (1994) (1994)

R_2: (1972) (1997) (1993)

R_3: (2018) (2017) (2016)

- Best recommmendation list?
- All lists return 1 relevant item
- But R_3 return 2 bad items
- We should also **measure** the **anti-relevant** items

Anti-relevance metrics

- The Probabilistic Ranking Principle (PRP):

> *If a system's response to a query is a ranking of documents in order of decreasing probability of relevance, the overall effectiveness of the system to its users will be maximized*
>
> —[Robertson, 1997]

Anti-relevance metrics

- The Probabilistic Ranking Principle (PRP):

> *If a system's response to a query is a ranking of documents in order of decreasing probability of relevance, the overall effectiveness of the system to its users will be maximized*
>
> —[Robertson, 1997]

- Most ranking-based accuracy metrics are formulated to estimate the classical PRP:

$$m(R_u|\theta_{rel}) = C \sum_{i \in R_u} m(\theta_{rel}(r_{ui})|u, i) \tag{8}$$

Anti-relevance metrics

- We study the dual PRP problem:

$$\overline{m}(R_u|\theta_{arel}) = C \sum_{i \in R_u} (1 - \overline{m}(\theta_{arel}(r_{ui})|u,i)) \propto$$
$$\propto 1 - C' \sum_{i \in R_u} m(\theta_{arel}(r_{ui})|u,i) = \boxed{1 - m(R_u|\theta_{arel})} \quad (9)$$

Anti-relevance metrics

- We study the dual PRP problem:

$$\overline{m}(R_u|\theta_{arel}) = C \sum_{i \in R_u} (1 - \overline{m}(\theta_{arel}(r_{ui})|u,i)) \propto$$
$$\propto 1 - C' \sum_{i \in R_u} m(\theta_{arel}(r_{ui})|u,i) = \boxed{1 - m(R_u|\theta_{arel})} \quad (9)$$

- Our anti-relevance metrics formulation is equivalent to computing any relevance-based metric using an anti-relevance model (where an item is relevant if $r_{ui} \leq \tau_{AR}$) and returning its complement. **Higher value implies less anti-relevant items recommended**

Anti-relevance metrics: a summary

- Relevance metrics **only measure** the amount of **highly relevant** items recommended by the user

Anti-relevance metrics: a summary

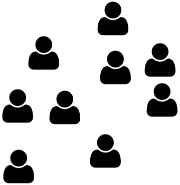
- Relevance metrics **only measure** the amount of **highly relevant** items recommended by the user
- However, we should also **measure** the number of items **with low ratings** that we are recommending to the users. Users tend to **penalize** the recommenders **mistakes**

Anti-relevance metrics: a summary

- Relevance metrics **only measure** the amount of **highly relevant** items recommended by the user
- However, we should also **measure** the number of items **with low ratings** that we are recommending to the users. Users tend to **penalize** the recommenders **mistakes**
- We can analyze the **anti-relevance** of the items by computing classical relevance metrics with an **anti-relevance models**

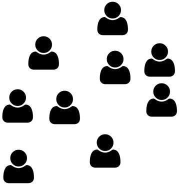
Incorporating user and item attributes in our metrics

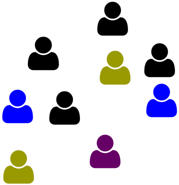
- We **usually** assume that all users in the system are **equal**



User attributes

- We **usually** assume that all users in the system are **equal**
- But some users may belong to **less represented groups** and our recommendations may be **biased** towards the **majority groups**





User attributes

- We **usually** assume that all users in the system are **equal**
- But some users may belong to **less represented groups** and our recommendations may be **biased** towards the **majority groups**

- Should not we be analyzing the **performance** of **specific groups** of users?

$$m(\theta) = C^{-1} \sum_{u \in \mathcal{U}} c(u) \cdot m(R_u, \theta) \qquad (10)$$

Item attributes

- In every recommendation we can distinguish the **items that appear in the test set $\boldsymbol{w^+}$**, non-relevant items that share similarity with the items in test w^* and the rest w^-

$$m(R_u, \theta) \propto \sum_{\boldsymbol{i \in I^+(u)}} \boldsymbol{w^+(u,i)} + \sum_{i \in I^*(u)} w^*(u,i) + \sum_{i \in I^-(u)} w^-(u,i) \tag{11}$$

- In every recommendation we can distinguish the items that appear in the test set w^{+}, non-relevant items that **share similarity** with the **items in test $\boldsymbol{w^{*}}$** and the rest w^{-}

$$m(R_u,\theta) \propto \sum_{i\in I^{+}(u)} w^{+}(u,i) + \sum_{\boldsymbol{i\in I^{*}(u)}} \boldsymbol{w^{*}(u,i)} + \sum_{i\in I^{-}(u)} w^{-}(u,i) \tag{11}$$

- In every recommendation we can distinguish the items that appear in the test set w^+, non-relevant items that share similarity with the items in test w^* and **the rest $\boldsymbol{w^-}$**

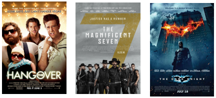
$$m(R_u, \theta) \propto \sum_{i \in I^+(u)} w^+(u,i) + \sum_{i \in I^*(u)} w^*(u,i) + \sum_{\boldsymbol{i \in I^-(u)}} \boldsymbol{w^-(u,i)} \tag{11}$$

- In every recommendation we can distinguish the items that appear in the test set w^+, non-relevant items that share similarity with the items in test w^* and the rest w^-

$$m(R_u, \theta) \propto \sum_{i \in I^+(u)} w^+(u,i) + \sum_{i \in I^*(u)} w^*(u,i) + \sum_{i \in I^-(u)} w^-(u,i) \tag{11}$$

Test

R_1



Test $\sim R_1$

$|I^+| = 0$

$|I^*| = 0$

$|I^-| = 3$

R_2

Test $\approx R_2$

$|I^+| = 0$

$|I^*| = 3$

$|I^-| = 0$

Attributes in evaluation: a summary

- We can use **user attributes** to **detect** possible **biases** in the algorithms. The recommendations might be biased toward the majority groups

Attributes in evaluation: a summary

- We can use **user attributes** to **detect** possible **biases** in the algorithms. The recommendations might be biased toward the majority groups
- **Item attributes** can be integrated into classical relevance metrics to consider more items in the recommendations as **partially relevant**

- Objective: **test our metrics in well-known datasets**

Experiments on new metrics

- Objective: **test our metrics in well-known datasets**
- Datasets:
  - **Movielens1M: 6k users, 3.7k items, 1M ratings (1-5)**
  - FS (Tokyo): 11.6k users, 51.1k items, 998k interactions

- Objective: **test our metrics in well-known datasets**
- Datasets:
  - **Movielens1M: 6k users, 3.7k items, 1M ratings (1-5)**
  - FS (Tokyo): 11.6k users, 51.1k items, 998k interactions
- Recommenders:
  - No personalized: Pop, Rnd
  - k-NN: UBCB, IB, UB
  - Matrix factorization: HKV, BPRMF
  - Temporal/Sequential: TD, MC, FPMC, Fossil, Caser
  - Skylines: Skyline

Experiments on new metrics

- Objective: **test our metrics in well-known datasets**
- Datasets:
  - **Movielens1M: 6k users, 3.7k items, 1M ratings (1-5)**
  - FS (Tokyo): 11.6k users, 51.1k items, 998k interactions
- Recommenders:
  - No personalized: Pop, Rnd
  - *k*-NN: UBCB, IB, UB
  - Matrix factorization: HKV, BPRMF
  - Temporal/Sequential: TD, MC, FPMC, Fossil, Caser
  - Skylines: Skyline
- Splits:
  - **Temporal system split (TS, 80% training)**
  - Random system split (RS, 80% training)

Experiments: time-aware novelty metrics

- Skyline obtain high results in our time-aware novelty metrics. Temporal novel items are relevant

Recommender	FIN	AIN	MIN	LIN
Rnd	**0.118**	**0.630**	**0.616**	0.971
Rnd_{CF}	0.112	0.626	0.611	0.972
Pop	0.000	0.614	0.592	†**1.000**
Pop_{CF}	0.000	0.613	0.591	1.000
UBCB	0.001	0.608	0.579	0.999
IB	0.001	0.605	0.570	0.999
UB	0.004	0.610	0.581	0.999
HKV	0.005	0.611	0.585	0.999
BPRMF	0.003	0.614	0.591	0.999
TD	0.004	0.612	0.587	0.999
MC	0.028	0.629	0.614	0.999
FPMC	0.001	0.606	0.577	0.999
Fossil	0.004	0.613	0.591	0.999
Caser	0.025	0.626	0.609	0.999
Skyline	0.136	0.666	0.661	0.998
$\text{Skyline}_{\text{CF}}$	†**0.145**	†**0.671**	†**0.670**	0.997

Experiments: time-aware novelty metrics. Metrics @5

- Some sequential recommenders do not obtain high results in time-aware novelty metrics

Recommender	FIN	AIN	MIN	LIN
Rnd	**0.118**	**0.630**	**0.616**	0.971
Rnd_{CF}	0.112	0.626	0.611	0.972
Pop	0.000	0.614	0.592	†**1.000**
Pop_{CF}	0.000	0.613	0.591	1.000
UBCB	0.001	0.608	0.579	0.999
IB	0.001	0.605	0.570	0.999
UB	0.004	0.610	0.581	0.999
HKV	0.005	0.611	0.585	0.999
BPRMF	0.003	0.614	0.591	0.999
TD	0.004	0.612	0.587	0.999
MC	0.028	0.629	0.614	0.999
FPMC	0.001	0.606	0.577	0.999
Fossil	0.004	0.613	0.591	0.999
Caser	0.025	0.626	0.609	0.999
Skyline	0.136	0.666	0.661	0.998
$Skyline_{CF}$	†**0.145**	†**0.671**	†**0.670**	0.997

- AIN and MIN are more useful than FIN and LIN. Most models achieve low values of FIN and high values of LIN

Recommender	**FIN**	**AIN**	**MIN**	**LIN**
Rnd	**0.118**	**0.630**	**0.616**	0.971
Rnd_{CF}	0.112	0.626	0.611	0.972
Pop	0.000	0.614	0.592	†**1.000**
Pop_{CF}	0.000	0.613	0.591	1.000
UBCB	0.001	0.608	0.579	0.999
IB	0.001	0.605	0.570	0.999
UB	0.004	0.610	0.581	0.999
HKV	0.005	0.611	0.585	0.999
BPRMF	0.003	0.614	0.591	0.999
TD	0.004	0.612	0.587	0.999
MC	0.028	0.629	0.614	0.999
FPMC	0.001	0.606	0.577	0.999
Fossil	0.004	0.613	0.591	0.999
Caser	0.025	0.626	0.609	0.999
Skyline	0.136	0.666	0.661	0.998
$\text{Skyline}_{\text{CF}}$	†**0.145**	†**0.671**	†**0.670**	0.997

Experiments: time-aware novelty metrics. Metrics @5

- AIN and MIN are more useful than FIN and LIN. Most models achieve low values of FIN and high values of LIN

Recommender	FIN	AIN	MIN	LIN
Rnd	**0.118**	**0.630**	**0.616**	0.971
Rnd_{CF}	0.112	0.626	0.611	0.972
Pop	0.000	0.614	0.592	†**1.000**
Pop_{CF}	0.000	0.613	0.591	1.000
UBCB	0.001	0.608	0.579	0.999
IB	0.001	0.605	0.570	0.999
UB	0.004	0.610	0.581	0.999
HKV	0.005	0.611	0.585	0.999
BPRMF	0.003	0.614	0.591	0.999
TD	0.004	0.612	0.587	0.999
MC	0.028	0.629	0.614	0.999
FPMC	0.001	0.606	0.577	0.999
Fossil	0.004	0.613	0.591	0.999
Caser	0.025	0.626	0.609	0.999
Skyline	0.136	0.666	0.661	0.998
$\text{Skyline}_{\text{CF}}$	†**0.145**	†**0.671**	†**0.670**	0.997

Experiments: anti-relevance metrics

- Rnd recommender achieves highest values in anti-relevance metrics

Recommender	**P**	$\overline{\textbf{P}}$	**nDCG**	$\overline{\textbf{nDCG}}$
Rnd	0.019	0.993	0.012	0.996
Rnd_{CF}	0.015	**0.994**	0.008	**0.997**
Pop	**0.281**	0.977	**0.221**	0.981
Pop_{CF}	0.210	0.979	0.161	0.983
UBCB	0.254	0.979	0.195	0.985
IB	0.234	0.979	0.177	0.987
UB	0.248	0.985	0.195	0.990
HKV	0.257	0.985	0.202	0.990
BPRMF	0.231	0.975	0.172	0.983
TD	0.248	0.987	0.194	0.990
MC	0.177	0.972	0.134	0.978
FPMC	0.212	0.979	0.159	0.985
Fossil	0.227	0.974	0.170	0.984
Caser	0.192	0.969	0.136	0.977
Skyline	†**0.943**	†**1.000**	†**1.000**	†**1.000**
$\text{Skyline}_{\text{CF}}$	0.911	1.000	0.999	1.000
$\overline{\text{Skyline}}$	0.000	0.189	0.000	0.001
$\overline{\text{Skyline}_{\text{CF}}}$	0.000	0.221	0.000	0.001

Experiments: anti-relevance metrics. Metrics @5

- Personalized recommenders sometimes fail in the recommendations

Recommender	**P**	$\overline{\mathbf{P}}$	**nDCG**	$\overline{\mathbf{nDCG}}$
Rnd	0.019	0.993	0.012	0.996
Rnd_{CF}	0.015	**0.994**	0.008	**0.997**
Pop	**0.281**	0.977	**0.221**	0.981
Pop_{CF}	0.210	0.979	0.161	0.983
UBCB	0.254	0.979	0.195	0.985
IB	0.234	0.979	0.177	0.987
UB	0.248	0.985	0.195	0.990
HKV	0.257	0.985	0.202	0.990
BPRMF	0.231	0.975	0.172	0.983
TD	0.248	0.987	0.194	0.990
MC	0.177	0.972	0.134	0.978
FPMC	0.212	0.979	0.159	0.985
Fossil	0.227	0.974	0.170	0.984
Caser	0.192	0.969	0.136	0.977
Skyline	†**0.943**	†**1.000**	†**1.000**	†**1.000**
$\text{Skyline}_{\text{CF}}$	0.911	1.000	0.999	1.000
$\overline{\text{Skyline}}$	0.000	0.189	0.000	0.001
$\overline{\text{Skyline}_{\text{CF}}}$	0.000	0.221	0.000	0.001

Experiments: user and item attributes

Experiments: user and item attributes (nDCG@5)

- In Movielens1M, users with more than 56 years (∼5%) tend to obtain lower results in terms of relevance

Family	Std	Gender		Age				Test Quartile			
		F	M	1	18	35	56	Q1	Q2	Q3	Q4
Rnd	0.012	0.011	0.012	0.011	0.014	0.009	0.005	0.003	0.004	0.016	0.023
Rnd_{CF}	0.008	0.010	0.008	0.003	0.009	0.009	0.000	0.002	0.003	0.006	0.027
Pop	0.221	0.177	**0.238**	0.192	**0.250**	**0.190**	0.132	0.055	0.160	0.260	**0.406**
Pop_{CF}	0.161	0.131	0.171	0.185	0.178	0.135	0.101	0.043	0.114	0.219	0.344
UBCB	0.195	0.177	0.202	0.195	0.206	0.180	0.164	0.057	**0.178**	0.264	0.368
IB	0.177	0.153	0.185	0.168	0.187	0.161	0.166	0.052	0.144	0.239	0.351
UB	0.195	0.173	0.202	0.194	0.208	0.176	0.160	0.067	0.165	0.274	0.352
HKV	**0.202**	**0.184**	0.209	**0.207**	0.213	0.185	**0.191**	**0.074**	0.166	**0.284**	0.366
BPRMF	0.172	0.166	0.175	0.180	0.179	0.164	0.144	0.056	0.144	0.232	0.330
TD	0.194	0.176	0.200	0.188	0.205	0.178	0.171	0.066	0.162	0.270	0.358
MC	0.134	0.127	0.137	0.127	0.142	0.123	0.122	0.052	0.109	0.170	0.257
FPMC	0.159	0.139	0.166	0.196	0.176	0.134	0.085	0.044	0.124	0.215	0.327
Fossil	0.170	0.178	0.168	0.160	0.177	0.160	0.172	0.062	0.134	0.221	0.333
Caser	0.136	0.141	0.135	0.114	0.143	0.128	0.129	0.044	0.109	0.202	0.248
Skyline	†**1.000**	†**1.000**	†**1.000**	†**1.000**	†**1.000**	†**0.999**	†**1.000**	†**1.000**	†**1.000**	†**0.999**	†**1.000**
$\text{Skyline}_{\text{CF}}$	0.999	1.000	0.999	1.000	1.000	0.999	1.000	1.000	1.000	0.998	1.000

- In Movielens1M and FS (Tokyo), females (∼27% and ∼11%) also tend to obtain lower in terms of relevance

Family	Std	Gender		Age				Test Quartile			
		F	M	1	18	35	56	Q1	Q2	Q3	Q4
Rnd	0.012	0.011	0.012	0.011	0.014	0.009	0.005	0.003	0.004	0.016	0.023
Rnd_{CF}	0.008	0.010	0.008	0.003	0.009	0.009	0.000	0.002	0.003	0.006	0.027
Pop	0.221	0.177	**0.238**	0.192	**0.250**	**0.190**	0.132	0.055	0.160	0.260	**0.406**
Pop_{CF}	0.161	0.131	0.171	0.185	0.178	0.135	0.101	0.043	0.114	0.219	0.344
UBCB	0.195	0.177	0.202	0.195	0.206	0.180	0.164	0.057	**0.178**	0.264	0.368
IB	0.177	0.153	0.185	0.168	0.187	0.161	0.166	0.052	0.144	0.239	0.351
UB	0.195	0.173	0.202	0.194	0.208	0.176	0.160	0.067	0.165	0.274	0.352
HKV	**0.202**	**0.184**	0.209	**0.207**	0.213	0.185	**0.191**	**0.074**	0.166	**0.284**	0.366
BPRMF	0.172	0.166	0.175	0.180	0.179	0.164	0.144	0.056	0.144	0.232	0.330
TD	0.194	0.176	0.200	0.188	0.205	0.178	0.171	0.066	0.162	0.270	0.358
MC	0.134	0.127	0.137	0.127	0.142	0.123	0.122	0.052	0.109	0.170	0.257
FPMC	0.159	0.139	0.166	0.196	0.176	0.134	0.085	0.044	0.124	0.215	0.327
Fossil	0.170	0.178	0.168	0.160	0.177	0.160	0.172	0.062	0.134	0.221	0.333
Caser	0.136	0.141	0.135	0.114	0.143	0.128	0.129	0.044	0.109	0.202	0.248
Skyline	**†1.000**	**†1.000**	**†1.000**	**†1.000**	**†1.000**	**†0.999**	**†1.000**	**†1.000**	**†1.000**	**†0.999**	**†1.000**
$\text{Skyline}_{\text{CF}}$	0.999	1.000	0.999	1.000	1.000	0.999	1.000	1.000	1.000	0.998	1.000

Experiments: user and item attributes (nDCG@5)

- The higher the test quartile, the higher the results obtained (more items in the test set)

		Gender		Age				Test Quartile			
Family	**Std**	**F**	**M**	**1**	**18**	**35**	**56**	**Q1**	**Q2**	**Q3**	**Q4**
Rnd	0.012	0.011	0.012	0.011	0.014	0.009	0.005	0.003	0.004	0.016	0.023
Rnd_{CF}	0.008	0.010	0.008	0.003	0.009	0.009	0.000	0.002	0.003	0.006	0.027
Pop	0.221	0.177	**0.238**	0.192	**0.250**	**0.190**	0.132	0.055	0.160	0.260	**0.406**
Pop_{CF}	0.161	0.131	0.171	0.185	0.178	0.135	0.101	0.043	0.114	0.219	0.344
UBCB	0.195	0.177	0.202	0.195	0.206	0.180	0.164	0.057	**0.178**	0.264	0.368
IB	0.177	0.153	0.185	0.168	0.187	0.161	0.166	0.052	0.144	0.239	0.351
UB	0.195	0.173	0.202	0.194	0.208	0.176	0.160	0.067	0.165	0.274	0.352
HKV	**0.202**	**0.184**	0.209	**0.207**	0.213	0.185	**0.191**	**0.074**	0.166	**0.284**	0.366
BPRMF	0.172	0.166	0.175	0.180	0.179	0.164	0.144	0.056	0.144	0.232	0.330
TD	0.194	0.176	0.200	0.188	0.205	0.178	0.171	0.066	0.162	0.270	0.358
MC	0.134	0.127	0.137	0.127	0.142	0.123	0.122	0.052	0.109	0.170	0.257
FPMC	0.159	0.139	0.166	0.196	0.176	0.134	0.085	0.044	0.124	0.215	0.327
Fossil	0.170	0.178	0.168	0.160	0.177	0.160	0.172	0.062	0.134	0.221	0.333
Caser	0.136	0.141	0.135	0.114	0.143	0.128	0.129	0.044	0.109	0.202	0.248
Skyline	**†1.000**	**†1.000**	**†1.000**	**†1.000**	**†1.000**	**†0.999**	**†1.000**	**†1.000**	**†1.000**	**†0.999**	**†1.000**
$\text{Skyline}_{\text{CF}}$	0.999	1.000	0.999	1.000	1.000	0.999	1.000	1.000	1.000	0.998	1.000

Experiments: user and item attributes (nDCG@5)

- Using the both main and secondary features we obtain higher results than the pure metric

Family	nDCG $\tau = 0$	τ_m	τ_s	τ_{ms}
Rnd	0.012	0.034	0.269	0.276
Rnd_{CF}	0.008	0.023	0.251	0.255
Pop	**0.221**	**0.244**	0.361	0.372
Pop_{CF}	0.161	0.189	0.308	0.322
UBCB	0.195	0.221	0.356	0.366
IB	0.177	0.206	0.322	0.337
UB	0.195	0.224	0.347	0.360
HKV	0.202	0.230	**0.364**	**0.375**
BPRMF	0.172	0.201	0.334	0.347
TD	0.194	0.223	0.347	0.361
MC	0.134	0.170	0.312	0.327
FPMC	0.159	0.181	0.314	0.325
Fossil	0.170	0.195	0.331	0.342
Caser	0.136	0.166	0.309	0.321
Skyline	†**1.000**	†**1.000**	†**1.000**	†**1.000**
$\text{Skyline}_{\text{CF}}$	0.999	0.999	0.999	0.999

$\tau = 0$: pure metric
τ_m: main feature (directors)
τ_s: secondary feature (genres)

Experiments: user and item attributes (nDCG@5)

- Misleading results might be obtained using higher values of the similarities (Rnd recommender becomes competitive)

$\tau = 0$: pure metric
τ_m: main feature (directors)
τ_s: secondary feature (genres)

	nDCG			
Family	$\tau = 0$	τ_m	τ_s	τ_{ms}
Rnd	0.012	0.034	0.269	0.276
Rnd_{CF}	0.008	0.023	0.251	0.255
Pop	**0.221**	**0.244**	0.361	0.372
Pop_{CF}	0.161	0.189	0.308	0.322
UBCB	0.195	0.221	0.356	0.366
IB	0.177	0.206	0.322	0.337
UB	0.195	0.224	0.347	0.360
HKV	0.202	0.230	**0.364**	**0.375**
BPRMF	0.172	0.201	0.334	0.347
TD	0.194	0.223	0.347	0.361
MC	0.134	0.170	0.312	0.327
FPMC	0.159	0.181	0.314	0.325
Fossil	0.170	0.195	0.331	0.342
Caser	0.136	0.166	0.309	0.321
Skyline	†**1.000**	†**1.000**	†**1.000**	†**1.000**
$\text{Skyline}_{\text{CF}}$	0.999	0.999	0.999	0.999

Summary

- Analyzing **only** the **relevance** of recommendations is **incomplete**

- Analyzing **only** the **relevance** of recommendations is **incomplete**
- There is a **relationship** between **time-aware novelty metrics** and **relevance**

Summary

- Analyzing **only** the **relevance** of recommendations is **incomplete**
- There is a **relationship** between **time-aware novelty metrics** and **relevance**
- The RS community should further **analyze** the **bad recommendations** of the algorithms

- Analyzing **only** the **relevance** of recommendations is **incomplete**
- There is a **relationship** between **time-aware novelty metrics** and **relevance**
- The RS community should further **analyze** the **bad recommendations** of the algorithms
- The RS community should **exploit** the **attributes** of both users and items to better analyze the performance of the recommenders

- Second objective: develop mechanisms to **incorporate sequentiality** in **k-NN recommender systems**

k-NN recommender systems

- Second objective: develop mechanisms to **incorporate sequentiality** in **k-NN recommender systems**
- We will define a **sequential similarity metric** based on LCS

k-NN recommender systems

- Second objective: develop mechanisms to **incorporate sequentiality** in **k-NN recommender systems**
- We will define a **sequential similarity metric** based on LCS
- We will also **redefine** the classical formulation of **k-NN recommender systems**

k-NN recommender systems

- Second objective: develop mechanisms to **incorporate sequentiality** in ***k*-NN recommender systems**
- We will define a **sequential similarity metric** based on LCS
- We will also **redefine** the classical formulation of ***k*-NN recommender systems**
- **Contributions published** in Information Processing and Management [Sánchez and Bellogín, 2020b] journal. Based on the future work of [Sánchez and Bellogín, 2019b] and [Bellogín and Sánchez, 2017]. Research conducted during the master's degree.

Defining a new similarity metric

k-NN recommender systems

- Classic formulation of k-NN recommender systems:

$$\hat{r}_{ui} = \sum_{v \in \mathcal{N}_i(u)} r_{vi} \boxed{w_{uv}} \tag{12}$$

k-NN recommender systems

- Classic formulation of k-NN recommender systems:

$$\hat{r}_{ui} = \sum_{v \in \mathcal{N}_i(u)} r_{vi} \boxed{w_{uv}} \tag{12}$$

$$\text{Cosine similarity}(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}} \tag{13}$$

$$\text{Pearson correlation}(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \overline{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \overline{r}_v)^2}} \tag{14}$$

- Classic formulation of k-NN recommender systems:

$$\hat{r}_{ui} = \sum_{v \in \mathcal{N}_i(u)} r_{vi} \boxed{w_{uv}} \tag{12}$$

$$\cancel{\text{Cosine similarity}(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}}} \tag{13}$$

$$\cancel{\text{Pearson correlation}(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \overline{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \overline{r}_v)^2}}} \tag{14}$$

- We propose a sequential similarity metric between users u and v:

$$\boldsymbol{w_{uv} \sim LCS(u,v)}$$

Longest Common Subsequence

- Applications on text comparison and DNA sequences

```
procedure LCS(x, y)                                  ▷ LCS between x and y
    L[0···m, 0···n] ← 0
    for i ← 1, m do
        for j ← 1, n do                              ▷ There is a match
            if x_i = y_j then
                L[i, j] ← L[i − 1, j − 1] + 1
            else
                L[i, j] ← max(L[i, j − 1], L[i − 1, j])
            end if
        end for
    end for
    return L[m, n]   ▷ L[m, n] contains the length of the LCS between x_1 ... x_i and y_1 ... y_j
end procedure
```

Longest Common Subsequence

- Applications on text comparison and DNA sequences

procedure LCS(x, y) ▷ LCS between x and y
    $L[0 \cdots m, 0 \cdots n] \leftarrow 0$
    for $i \leftarrow 1, m$ **do**
        for $j \leftarrow 1, n$ **do** ▷ There is a match
            if $x_i = y_j$ **then**
                $L[i, j] \leftarrow L[i-1, j-1] + 1$
            else
                $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$
            end if
        end for
    end for
    return $L[m, n]$ ▷ $L[m, n]$ contains the length of the LCS between $x_1 \ldots x_i$ and $y_1 \ldots y_j$
end procedure

Longest Common Subsequence: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (15)$$

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

Longest Common Subsequence: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

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

Longest Common Subsequence: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (15)$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	$\emptyset$	A	G	G	T	A	C
$\emptyset$	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \quad (15)$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

	∅	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: example

Longest Common Subsequence

$$L[i,j] = \begin{cases} 0 & \text{if i=0 or j=0} \\ L[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } X_i = Y_j \\ \max(L[i,j-1], L[i-1,j]) & \text{if } i,j>0 \text{ and } X_i \neq Y_j \end{cases} \tag{15}$$

The LCS may not be unique

	∅	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 for RS

procedure LCS_RecSys(u, v, f, δ) ▷ The LCS of users u and v applying transformation f
    $(x, y) \leftarrow (f(u), f(v))$ ▷ String x contains m symbols
    $L[0 \cdots m, 0 \cdots n] \leftarrow 0$
    for $i \leftarrow 1, m$ **do**
        for $j \leftarrow 1, n$ **do** ▷ There is a δ-matching
            if `match`(x_i, y_j, δ) **then**
                $L[i, j] \leftarrow L[i-1, j-1] + 1$
            else
                $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$
            end if
        end for
    end for
    return $L[m, n]$
end procedure

```
procedure LCS_RecSys(u, v, f, δ)    ▷ The LCS of users u and v applying transformation f
    (x, y) ← (f(u), f(v))           ▷ String x contains m symbols
    L[0···m, 0···n] ← 0
    for i ← 1, m do
        for j ← 1, n do             ▷ There is a δ-matching
            if match(x_i, y_j, δ) then
                L[i, j] ← L[i − 1, j − 1] + 1
            else
                L[i, j] ← max(L[i, j − 1], L[i − 1, j])
            end if
        end for
    end for
    return L[m, n]
end procedure
```

Longest Common Subsequence for RS

procedure LCS_RecSys(u, v, f, δ) $\triangleright$ The LCS of users u and v applying transformation f

    $(x, y) \leftarrow (f(u), f(v))$ $\triangleright$ String x contains m symbols

    $L[0 \cdots m, 0 \cdots n] \leftarrow 0$

    for $i \leftarrow 1, m$ **do**

        for $j \leftarrow 1, n$ **do** $\triangleright$ There is a δ-matching

            if $\boxed{\texttt{match}(x_i, y_j, \delta)}$ **then**

                $L[i, j] \leftarrow L[i-1, j-1] + 1$

            else

                $L[i, j] \leftarrow \max(L[i, j-1], L[i-1, j])$

            end if

        end for

    end for

    return $L[m, n]$

end procedure

LCS normalizations

- LCS algorithm obtain values in the $[0, min(|f(u)|, |f(v)|)]$ interval

$$\mathrm{sim}_1^{f,\delta}(u,v) \quad = \quad \text{LCS\_Recsys}(u,v,f,\delta) \tag{16}$$

LCS normalizations

- LCS algorithm obtain values in the $[0, min(|f(u)|, |f(v)|)]$ interval

$$\mathrm{sim}_1^{f,\delta}(u,v) = \mathrm{LCS\_Recsys}(u,v,f,\delta) \tag{16}$$

$$\mathrm{sim}_2^{f,\delta}(u,v) = \frac{\mathrm{sim}_1^{f,\delta}(u,v)^2}{|f(u)| \cdot |f(v)|} \tag{17}$$

$$\mathrm{sim}_3^{f,\delta}(u,v) = \frac{2 \cdot \mathrm{sim}_1^{f,\delta}(u,v)}{|f(u)| + |f(v)|} \tag{18}$$

$$\mathrm{sim}_4^{f,\delta}(u,v) = \frac{\mathrm{sim}_1^{f,\delta}(u,v)}{\max\left(|f(u)|, |f(v)|\right)} \tag{19}$$

$$\mathrm{sim}_5^{f,\delta}(u,v) = \frac{\mathrm{sim}_1^{f,\delta}(u,v)}{\min\left(|f(u)|, |f(v)|\right)} \tag{20}$$

Redefining k-NN recommender systems

Redefining k-NN RS: Backward-Forward algorithm

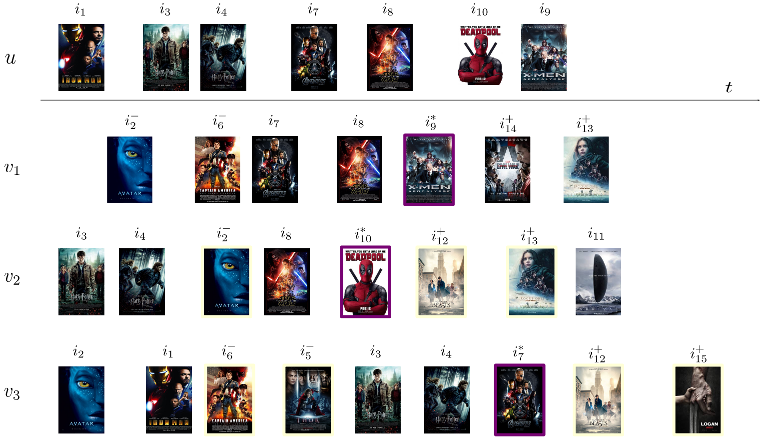
- Obtain the neighbors using any similarity metric (classical or sequential)

Redefining k-NN RS: Backward-Forward algorithm

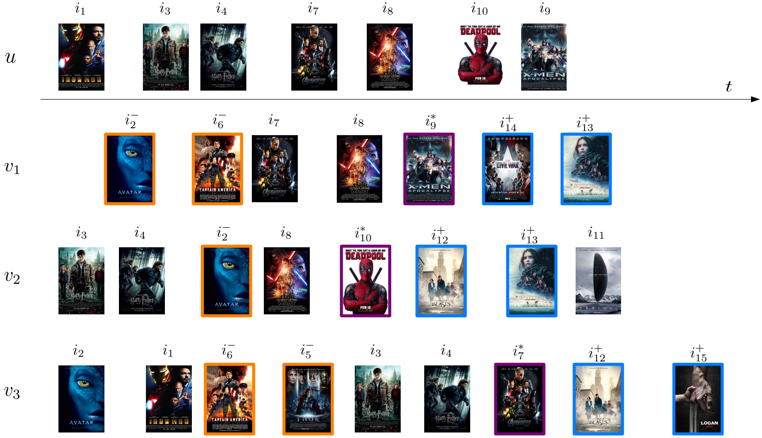
- Obtain the neighbors using any similarity metric (classical or sequential)
- Find the last common interactions between the user and her neighbors

Redefining k-NN RS: Backward-Forward algorithm

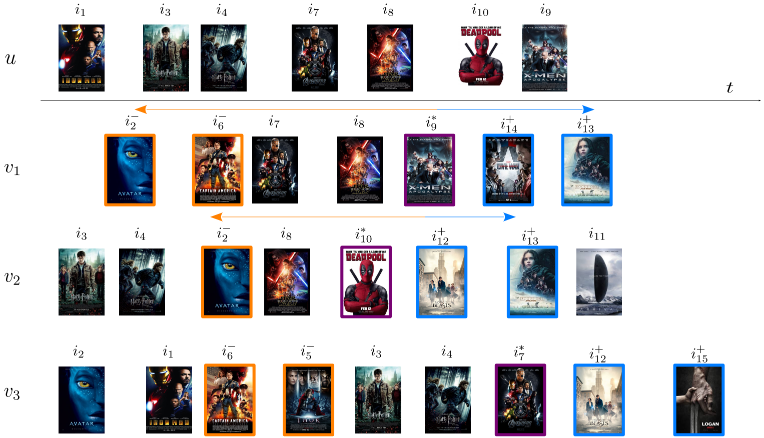
- Obtain the neighbors using any similarity metric (classical or sequential)
- Find the last common interactions between the user and her neighbors



Backward-Forward algorithm (2)

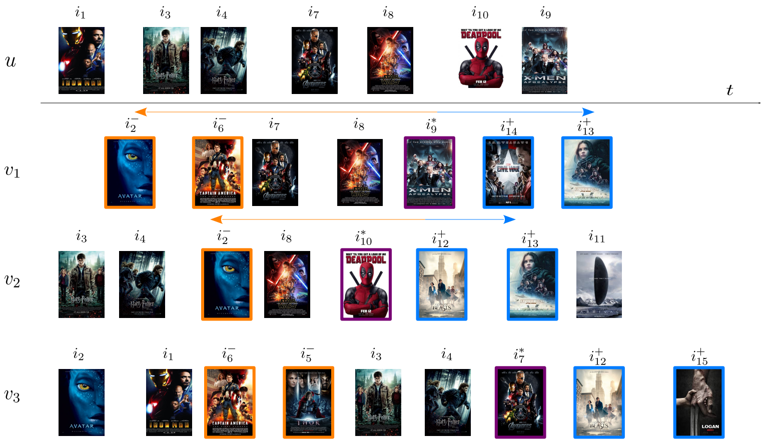


Backward-Forward algorithm (2)



$$L_2^+(v_1; u) = (i_{14}, i_{13}), L_2^-(v_1; u) = (i_6, i_2)$$
$$L_2^+(v_2; u) = (i_{12}, i_{13}), L_2^-(v_2; u) = (i_2)$$
$$L_2^+(v_3; u) = (i_{12}, i_{15}), L_2^-(v_3; u) = (i_5, i_6)$$

Backward-Forward algorithm (2)



$$\boldsymbol{L_2^+(v_1;u) = (i_{14}, i_{13}), L_2^-(v_1;u) = (i_6, i_2)}$$

$$L_2^+(v_2;u) = (i_{12}, i_{13}), L_2^-(v_2;u) = (i_2)$$

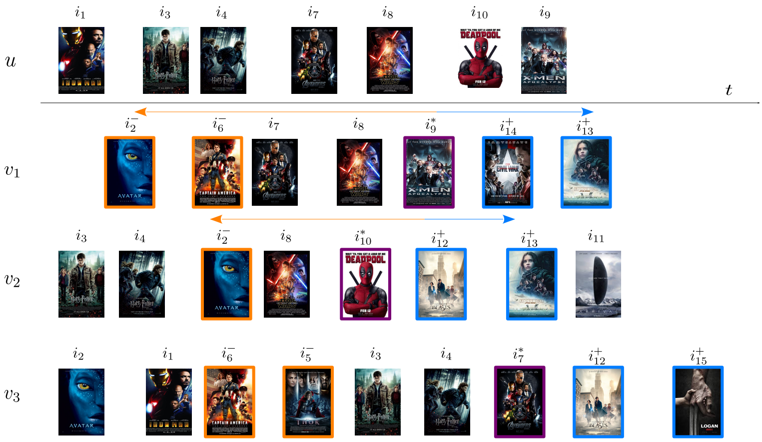
$$L_2^+(v_3;u) = (i_{12}, i_{15}), L_2^-(v_3;u) = (i_5, i_6)$$

- Normalize the rankings obtained for each neighbor
  - Standard normalization: $x' = \frac{x - x_{min}}{x_{min} - x_{max}}$
  - Rank normalization: $x' = 1 - \frac{rank(x) - 1}{|X|}$
  - Default normalization: $x' = x$
  - Other

Backward-Forward algorithm (3)

- Normalize the rankings obtained for each neighbor
  - Standard normalization: $x' = \frac{x - x_{min}}{x_{min} - x_{max}}$
  - Rank normalization: $x' = 1 - \frac{rank(x) - 1}{|X|}$
  - Default normalization: $x' = x$
  - Other
- Generate a single list for each user using her neighbors rankings
  - **Sum combiner**
  - Min combiner
  - Max combiner
  - Other

Backward-Forward algorithm (4)



$$\boldsymbol{BF_2^+ = \{i_{12}, i_{13}\}}$$

$$BF_2^- = \{i_2, i_6\}$$

- Objective: **test our BF approaches** in datasets with **realistic timestamps** following a **temporal split**

- Objective: **test our BF approaches** in datasets with **realistic timestamps** following a **temporal split**
- Foursquare from [He and McAuley, 2016] and MovieTweetings from [Dooms et al., 2016].

- Objective: **test our BF approaches** in datasets with **realistic timestamps** following a **temporal split**
- Foursquare from [He and McAuley, 2016] and MovieTweetings from [Dooms et al., 2016].
- **Temporal** split: per user vs system

Experiments: sequences in k-NN RS

- Objective: **test our BF approaches** in datasets with **realistic timestamps** following a **temporal split**
- Foursquare from [He and McAuley, 2016] and MovieTweetings from [Dooms et al., 2016].
- **Temporal** split: per user vs system

Dataset	**Users**	**Items**	**Ratings**	**Density**	**Scale**	**Unique times**	**Time interval**
Foursquare	16k	3k	105k	0.205%	1	102k	Dec 2011 - Apr 2012
MovieTweetings	15k	8k	519k	0.399%	0-10	517k	Feb 2013 - Apr 2017

- Relevance (nDCG), novelty (EPC), temporal-novelty (MIN), diversity (IC)

MovieTweetings

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	†**0.996**	0.410	†**0.949**
Rnd_{CF}	0.000	0.996	0.411	0.900
Pop	0.003	0.853	0.207	0.006
Pop_{CF}	0.003	0.854	0.210	0.006
IB	0.010	0.914	0.585	0.126
UB	0.016	0.907	0.585	0.030
HKV	0.024	0.934	0.573	0.081
BPRMF	0.016	0.923	0.579	0.125
TD	0.023	0.916	0.697	0.053
BFUB	0.031	0.927	0.728	0.077
BFsUB	**0.034**	0.936	†**0.828**	0.076
MC	0.031	0.919	0.707	0.043
FPMC	0.020	0.913	0.634	0.040
Fossil	0.025	0.915	0.647	0.028
Caser	0.026	0.939	0.771	0.129
Skyline	0.806	0.977	0.588	0.295
$\text{Skyline}_{\text{CF}}$	†**0.812**	0.977	0.616	0.251

Foursquare

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	0.998	0.615	†**1.000**
Rnd_{CF}	0.001	†**0.998**	0.612	1.000
Pop	0.130	0.879	0.515	0.004
Pop_{CF}	0.130	0.879	0.515	0.004
IB	0.155	0.952	0.613	0.828
UB	0.173	0.929	0.573	0.293
HKV	0.154	0.949	0.585	0.029
BPRMF	0.146	0.886	0.511	0.071
TD	0.170	0.929	0.582	0.307
BFUB	0.173	0.929	0.573	0.293
BFsUB	**0.174**	0.921	0.569	0.281
MC	0.133	0.945	0.624	0.269
FPMC	0.133	0.935	0.608	0.196
Fossil	0.163	0.938	**0.624**	0.131
Caser	0.170	0.929	0.610	0.301
Skyline	†**0.998**	0.960	†**0.671**	0.577
$\text{Skyline}_{\text{CF}}$	0.998	0.960	0.670	0.573

Experiments: BF. Temporal System. All metrics @5

- Sequential recommenders highly competitive in MovieTweetings but not in Foursquare

MovieTweetings

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	†**0.996**	0.410	†**0.949**
Rnd_{CF}	0.000	0.996	0.411	0.900
Pop	0.003	0.853	0.207	0.006
Pop_{CF}	0.003	0.854	0.210	0.006
IB	0.010	0.914	0.585	0.126
UB	0.016	0.907	0.585	0.030
HKV	0.024	0.934	0.573	0.081
BPRMF	0.016	0.923	0.579	0.125
TD	0.023	0.916	0.697	0.053
BFUB	0.031	0.927	0.728	0.077
BFsUB	**0.034**	0.936	†**0.828**	0.076
MC	0.031	0.919	0.707	0.043
FPMC	0.020	0.913	0.634	0.040
Fossil	0.025	0.915	0.647	0.028
Caser	0.026	0.939	0.771	0.129
Skyline	0.806	0.977	0.588	0.295
$\text{Skyline}_{\text{CF}}$	†**0.812**	0.977	0.616	0.251

Foursquare

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	0.998	0.615	†**1.000**
Rnd_{CF}	0.001	†**0.998**	0.612	1.000
Pop	0.130	0.879	0.515	0.004
Pop_{CF}	0.130	0.879	0.515	0.004
IB	0.155	0.952	0.613	0.828
UB	0.173	0.929	0.573	0.293
HKV	0.154	0.949	0.585	0.029
BPRMF	0.146	0.886	0.511	0.071
TD	0.170	0.929	0.582	0.307
BFUB	0.173	0.929	0.573	0.293
BFsUB	**0.174**	0.921	0.569	0.281
MC	0.133	0.945	0.624	0.269
FPMC	0.133	0.935	0.608	0.196
Fossil	0.163	0.938	**0.624**	0.131
Caser	0.170	0.929	0.610	0.301
Skyline	†**0.998**	0.960	†**0.671**	0.577
$\text{Skyline}_{\text{CF}}$	0.998	0.960	0.670	0.573

- Our Backward-Forward approaches are the best in terms of relevance and competitive in other dimensions

MovieTweetings

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	†**0.996**	0.410	†**0.949**
Rnd_{CF}	0.000	0.996	0.411	0.900
Pop	0.003	0.853	0.207	0.006
Pop_{CF}	0.003	0.854	0.210	0.006
IB	0.010	0.914	0.585	0.126
UB	0.016	0.907	0.585	0.030
HKV	0.024	0.934	0.573	0.081
BPRMF	0.016	0.923	0.579	0.125
TD	0.023	0.916	0.697	0.053
BFUB	0.031	0.927	0.728	0.077
BFsUB	**0.034**	0.936	†**0.828**	0.076
MC	0.031	0.919	0.707	0.043
FPMC	0.020	0.913	0.634	0.040
Fossil	0.025	0.915	0.647	0.028
Caser	0.026	0.939	0.771	0.129
Skyline	0.806	0.977	0.588	0.295
$Skyline_{CF}$	†**0.812**	0.977	0.616	0.251

Foursquare

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	0.998	0.615	†**1.000**
Rnd_{CF}	0.001	†**0.998**	0.612	1.000
Pop	0.130	0.879	0.515	0.004
Pop_{CF}	0.130	0.879	0.515	0.004
IB	0.155	0.952	0.613	0.828
UB	0.173	0.929	0.573	0.293
HKV	0.154	0.949	0.585	0.029
BPRMF	0.146	0.886	0.511	0.071
TD	0.170	0.929	0.582	0.307
BFUB	0.173	0.929	0.573	0.293
BFsUB	**0.174**	0.921	0.569	0.281
MC	0.133	0.945	0.624	0.269
FPMC	0.133	0.935	0.608	0.196
Fossil	0.163	0.938	**0.624**	0.131
Caser	0.170	0.929	0.610	0.301
Skyline	†**0.998**	0.960	†**0.671**	0.577
$Skyline_{CF}$	0.998	0.960	0.670	0.573

- Sequential recommenders are less competitive in this split for both datasets

MovieTweetings

Recommender	**nDCG**	**EPC**	**MIN**	**IC**
Rnd	0.000	†**0.996**	0.383	0.980
Rnd_{CF}	0.000	0.996	0.383	†**0.980**
Pop	0.024	0.870	0.159	0.005
Pop_{CF}	0.024	0.870	0.159	0.005
IB	0.050	0.919	0.402	0.185
UB	0.049	0.910	0.360	0.038
HKV	0.050	0.934	0.367	0.075
BPRMF	0.037	0.933	0.363	0.218
TD	0.081	0.916	0.451	0.077
BFUB	0.070	0.918	0.424	0.054
BFsUB	**0.111**	0.928	**0.518**	0.086
MC	0.062	0.905	0.436	0.073
FPMC	0.038	0.913	0.365	0.065
Fossil	0.050	0.909	0.386	0.045
Caser	0.083	0.928	0.483	0.158
Skyline	†**1.000**	0.962	†**0.525**	0.260
$\text{Skyline}_{\text{CF}}$	1.000	0.962	†**0.525**	0.260

Foursquare

Recommender	**nDCG**	**EPC**	**MIN**	**IC**
Rnd	0.001	0.998	0.540	†**1.000**
Rnd_{CF}	0.002	†**0.998**	0.538	1.000
Pop	0.133	0.878	0.501	0.004
Pop_{CF}	0.133	0.878	0.501	0.004
IB	0.186	0.950	0.535	0.829
UB	0.191	0.926	0.516	0.169
HKV	0.174	0.948	0.503	0.032
BPRMF	0.157	0.947	0.515	0.489
TD	0.185	0.929	0.536	0.232
BFUB	**0.192**	0.927	0.515	0.176
BFsUB	0.190	0.925	0.515	0.259
MC	0.159	0.940	0.558	0.185
FPMC	0.145	0.933	0.554	0.149
Fossil	0.177	0.939	**0.563**	0.080
Caser	0.182	0.932	0.559	0.307
Skyline	†**1.000**	0.960	†**0.568**	0.687
$\text{Skyline}_{\text{CF}}$	1.000	0.960	†**0.568**	0.687

Experiments: BF. Temporal Per User. All metrics @5.

- Our Backward-Forward approaches are still competitive against state-of-the-art algorithms

MovieTweetings

Recommender	nDCG	EPC	MIN	IC
Rnd	0.000	†**0.996**	0.383	0.980
Rnd_{CF}	0.000	0.996	0.383	†**0.980**
Pop	0.024	0.870	0.159	0.005
Pop_{CF}	0.024	0.870	0.159	0.005
IB	0.050	0.919	0.402	0.185
UB	0.049	0.910	0.360	0.038
HKV	0.050	0.934	0.367	0.075
BPRMF	0.037	0.933	0.363	0.218
TD	0.081	0.916	0.451	0.077
BFUB	0.070	0.918	0.424	0.054
BFsUB	**0.111**	0.928	**0.518**	0.086
MC	0.062	0.905	0.436	0.073
FPMC	0.038	0.913	0.365	0.065
Fossil	0.050	0.909	0.386	0.045
Caser	0.083	0.928	0.483	0.158
Skyline	†**1.000**	0.962	†**0.525**	0.260
$\text{Skyline}_{\text{CF}}$	1.000	0.962	†**0.525**	0.260

Foursquare

Recommender	nDCG	EPC	MIN	IC
Rnd	0.001	0.998	0.540	†**1.000**
Rnd_{CF}	0.002	†**0.998**	0.538	1.000
Pop	0.133	0.878	0.501	0.004
Pop_{CF}	0.133	0.878	0.501	0.004
IB	0.186	0.950	0.535	0.829
UB	0.191	0.926	0.516	0.169
HKV	0.174	0.948	0.503	0.032
BPRMF	0.157	0.947	0.515	0.489
TD	0.185	0.929	0.536	0.232
BFUB	**0.192**	0.927	0.515	0.176
BFsUB	0.190	0.925	0.515	0.259
MC	0.159	0.940	0.558	0.185
FPMC	0.145	0.933	0.554	0.149
Fossil	0.177	0.939	**0.563**	0.080
Caser	0.182	0.932	0.559	0.307
Skyline	†**1.000**	0.960	†**0.568**	0.687
$\text{Skyline}_{\text{CF}}$	1.000	0.960	†**0.568**	0.687

Summary

- We have defined a **sequential similarity metric** based on the LCS algorithm

Summary

- We have defined a **sequential similarity metric** based on the LCS algorithm
- We have **redefined** the k-NN recommenders by exploiting the **last common interactions** between the neighbors named Backward-Forward (BF)

Summary

- We have defined a **sequential similarity metric** based on the LCS algorithm
- We have **redefined** the k-NN recommenders by exploiting the **last common interactions** between the neighbors named Backward-Forward (BF)
- Our Backward-Forward algorithm can be used with **any kind** of **similarity** (sequential or not sequential)

Summary

- We have defined a **sequential similarity metric** based on the LCS algorithm
- We have **redefined** the k-NN recommenders by exploiting the **last common interactions** between the neighbors named Backward-Forward (BF)
- Our Backward-Forward algorithm can be used with **any kind** of **similarity** (sequential or not sequential)
- Our approach is **highly competitive** in two datasets using a time-aware evaluation

Overview

Point-Of-Interest (POI) recommendation

- We will address the objectives regarding the **analysis of current POI recommendation works** and **improve the performance** of **POI recommenders**

Point-Of-Interest (POI) recommendation

- We will address the objectives regarding the **analysis of current POI recommendation works** and **improve the performance** of **POI recommenders**
- We conduct a **survey** characterizing the POI recommendation works between 2011 and 2019

- We will address the objectives regarding the **analysis of current POI recommendation works** and **improve the performance** of **POI recommenders**
- We conduct a **survey** characterizing the POI recommendation works between 2011 and 2019
- We develop mechanisms to **increase the performance** of the recommenders in POI recommendation by using cross-domain techniques

Point-Of-Interest (POI) recommendation

- We will address the objectives regarding the **analysis of current POI recommendation works** and **improve the performance** of **POI recommenders**
- We conduct a **survey** characterizing the POI recommendation works between 2011 and 2019
- We develop mechanisms to **increase the performance** of the recommenders in POI recommendation by using cross-domain techniques
- **Contributions under review** in ACM Computing Surveys journal **(2º round of review)** and **published** in the Information Processing and Management journal [Sánchez and Bellogín, 2021] **(new)**

- Recommending **new venues** to the users when they arrive a city

- Recommending **new venues** to the users when they arrive a city
- Differences with classical recommendation:

- Recommending **new venues** to the users when they arrive a city
- Differences with classical recommendation:
  - **Greater sparsity**: Movielens20M (0.539%) and Netflix (1.177%) density vs Foursquare (0.0034%) and Gowalla (0.0047%) density

Point-Of-Interest (POI) Recommendation

- Recommending **new venues** to the users when they arrive a city
- Differences with classical recommendation:
  - **Greater sparsity**: Movielens20M (0.539%) and Netflix (1.177%) density vs Foursquare (0.0034%) and Gowalla (0.0047%) density
  - **Implicit** and **repeated interactions**: users visit the same places more than once

Point-Of-Interest (POI) Recommendation

- Recommending **new venues** to the users when they arrive a city
- Differences with classical recommendation:
  - **Greater sparsity**: Movielens20M (0.539%) and Netflix (1.177%) density vs Foursquare (0.0034%) and Gowalla (0.0047%) density
  - **Implicit** and **repeated interactions**: users visit the same places more than once
  - **External influences**: **geographical**, temporal, social, and sequential influences

Everything is related to everything else, but near things are more related than distant things —[Miller, 2004]

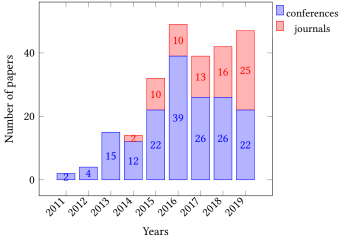
Point-Of-Interest recommendation: a survey

- Types of algorithms: based on similarities, factorization machines, neural networks, ...
- Information used: geographical, temporal, sequential, social, ...
- Evaluation methodology: metrics, splits, validation, datasets, ...

Point-Of-Interest recommendation: a survey

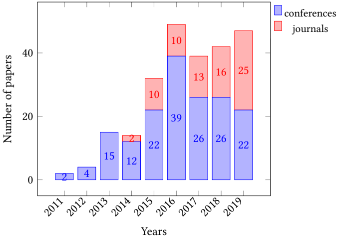
- Types of algorithms: based on similarities, factorization machines, neural networks, ...
- Information used: geographical, temporal, sequential, social, ...
- Evaluation methodology: metrics, splits, validation, datasets, ...

Source	Papers retrieved	Valid papers
Scopus	321	238
ScienceDirect	36	22
ACM	46	24
Unique papers	347	**244**



- Types of algorithms: based on similarities, factorization machines, neural networks, ...
- Information used: geographical, temporal, sequential, social, ...
- Evaluation methodology: metrics, splits, validation, datasets, ...

Source	Papers retrieved	Valid papers
Scopus	321	238
ScienceDirect	36	22
ACM	46	24
Unique papers	347	**244**



- More information in **Chapter 3**

Point-Of-Interest recommendation: a survey

Details			Evaluation configuration							Baselines			Split type			Split level	
Year	Reference	Acronym	Filter data	Validation	Error	Ranking	Region Split	Check-in(✓) POI(✗)	Cold Start Analysis	C. Non Personalized	C. Personalized	Geographical	Random	Temporal	Other	System	Per User
2011	[Ye et al., 2011]	USG				✓		✗	✓		✓	✓	✓				✓
2012	[Levandoski et al., 2012]	LARS					✓	✗			✓	✓	✓			✓	
2012	[Bao et al., 2012]	(N.A.)	✓			✓	✓	✗			✓				✓		
2013	[Yang et al., 2013]	LBSMF	✓		✓		✓	✓			✓		✓			✓	
2013	[Liu et al., 2013]	GT-BNMF				✓	✓	✓			✓		✓			✓	
2013	[Yuan et al., 2013]	UTE+SE	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Ying et al., 2014]	UPOI-Walk			✓	✓	✓	?			✓	✓					
2014	[Yuan et al., 2014]	GTAG	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Lian et al., 2014]	GeoMF	✓			✓		✗			✓		✓				✓
2015	[Yin et al., 2015]	LA-LDA		✓		✓		✗	✓			✓	✓				✓
2015	[Li et al., 2015]	RankGeoFM		✓		✓	✓	✓			✓	✓		✓			✓
2015	[Zhang and Chow, 2015]	GeoSoCa				✓	✓	✓				✓		✓		✓	
2015	[Feng et al., 2015]	PRME-G	✓	✓		✓	✓	✓		✓	✓			✓		✓	
2016	[Li et al., 2016]	ASMF	✓			✓	✓	✗	✓		✓	✓		✓			✓
2016	[Zhao et al., 2016]	STELLAR	✓			✓		✓			✓			✓		✓	
2017	[Zhao et al., 2017]	Geo-Teaser	✓			✓		✓			✓	✓		✓			✓
2017	[Yang et al., 2017]	PACE	✓			✓		✓				✓		✓			✓
2017	[Ren et al., 2017]	TGSC-PMF	✓		✓	✓		✓				✓	✓			✓	
2018	[Ma et al., 2018]	SAE-NAD	✓			✓	✓	✗			✓	✓	✓				✓
2018	[Gao et al., 2018]	GeoEISo	✓			✓	✓	✓			✓	✓	✓			✓	
2018	[Wang et al., 2018]	GeoIE	✓	✓		✓		✓				✓		✓			✓
2019	[Ying et al., 2019]	MEAP-T	✓	✓		✓	✓	✓		✓	✓			✓			✓
2019	[Si et al., 2019]	APRA-SA			✓	✓		✓			✓	✓	✓				✓
2019	[Qian et al., 2019]	STA		✓		✓		✓	✓			✓		✓			✓
Most Representatives			24	10	5	38	23	C:21 P:16	7	3	30	26	18	17	1	13	22
Total			135	37	22	229	135	C:150 P:66	27	29	147	142	123	82	14	101	104

- Most POI models use **ranking based accuracy metrics**

Details			Evaluation configuration							Baselines			Split type			Split level	
Year	Reference	Acronym	Filter data	Validation	Error	Ranking	Region Split	Check-in(✓) POI(✗)	Cold Start Analysis	C. Non Personalized	C. Personalized	Geographical	Random	Temporal	Other	System	Per User
2011	[Ye et al., 2011]	USG				✓		✗	✓		✓	✓	✓				✓
2012	[Levandoski et al., 2012]	LARS					✓	✗			✓	✓	✓			✓	
2012	[Bao et al., 2012]	(N.A.)	✓			✓	✓	✗			✓				✓		
2013	[Yang et al., 2013]	LBSMF	✓		✓		✓	✓			✓		✓			✓	
2013	[Liu et al., 2013]	GT-BNMF				✓	✓	✓			✓		✓			✓	
2013	[Yuan et al., 2013]	UTE+SE	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Ying et al., 2014]	UPOI-Walk			✓	✓	✓	?			✓	✓					
2014	[Yuan et al., 2014]	GTAG	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Lian et al., 2014]	GeoMF	✓			✓		✗			✓		✓				✓
2015	[Yin et al., 2015]	LA-LDA		✓		✓		✗	✓			✓	✓				✓
2015	[Li et al., 2015]	RankGeoFM		✓		✓	✓	✓			✓	✓		✓			✓
2015	[Zhang and Chow, 2015]	GeoSoCa				✓	✓	✓				✓		✓		✓	
2015	[Feng et al., 2015]	PRME-G	✓	✓		✓	✓	✓		✓	✓			✓		✓	
2016	[Li et al., 2016]	ASMF	✓			✓	✓	✗	✓		✓	✓		✓			✓
2016	[Zhao et al., 2016]	STELLAR	✓			✓		✓			✓			✓		✓	
2017	[Zhao et al., 2017]	Geo-Teaser	✓			✓		✓			✓	✓		✓			✓
2017	[Yang et al., 2017]	PACE	✓			✓		✓				✓		✓			✓
2017	[Ren et al., 2017]	TGSC-PMF	✓		✓	✓		✓				✓	✓			✓	
2018	[Ma et al., 2018]	SAE-NAD	✓			✓	✓	✗			✓	✓	✓				✓
2018	[Gao et al., 2018]	GeoEISo	✓			✓	✓	✓			✓	✓	✓			✓	
2018	[Wang et al., 2018]	GeoIE	✓	✓		✓		✓				✓		✓			✓
2019	[Ying et al., 2019]	MEAP-T	✓	✓		✓	✓	✓		✓	✓			✓			✓
2019	[Si et al., 2019]	APRA-SA			✓	✓		✓			✓	✓	✓				✓
2019	[Qian et al., 2019]	STA		✓		✓		✓	✓			✓		✓			✓
Most Representatives			24	10	5	38	23	C:21 P:16	7	3	30	26	18	17	1	13	22
Total			135	37	22	229	135	C:150 P:66	27	29	147	142	123	82	14	101	104

- Some researchers **apply some data filtering**

Point-Of-Interest recommendation: a survey

Details			Evaluation configuration							Baselines			Split type			Split level	
Year	Reference	Acronym	Filter data	Validation	Error	Ranking	Region Split	Check-in(✓) POI(✗)	Cold Start Analysis	C. Non Personalized	C. Personalized	Geographical	Random	Temporal	Other	System	Per User
2011	[Ye et al., 2011]	USG				✓		✗	✓		✓	✓	✓				✓
2012	[Levandoski et al., 2012]	LARS					✓	✗			✓	✓	✓			✓	
2012	[Bao et al., 2012]	(N.A.)	✓			✓	✓	✗			✓				✓		
2013	[Yang et al., 2013]	LBSMF	✓		✓		✓	✓			✓		✓			✓	
2013	[Liu et al., 2013]	GT-BNMF				✓	✓	✓			✓		✓			✓	
2013	[Yuan et al., 2013]	UTE+SE	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Ying et al., 2014]	UPOI-Walk			✓	✓	✓	?			✓	✓					
2014	[Yuan et al., 2014]	GTAG	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Lian et al., 2014]	GeoMF	✓			✓		✗			✓		✓				✓
2015	[Yin et al., 2015]	LA-LDA		✓		✓		✗	✓			✓	✓				✓
2015	[Li et al., 2015]	RankGeoFM		✓		✓	✓	✓			✓	✓		✓			✓
2015	[Zhang and Chow, 2015]	GeoSoCa				✓	✓	✓				✓		✓		✓	
2015	[Feng et al., 2015]	PRME-G	✓	✓		✓	✓	✓		✓	✓			✓		✓	
2016	[Li et al., 2016]	ASMF	✓			✓	✓	✗	✓		✓	✓		✓			✓
2016	[Zhao et al., 2016]	STELLAR	✓			✓		✓			✓			✓		✓	
2017	[Zhao et al., 2017]	Geo-Teaser	✓			✓		✓			✓	✓		✓			✓
2017	[Yang et al., 2017]	PACE	✓			✓		✓				✓		✓			✓
2017	[Ren et al., 2017]	TGSC-PMF	✓		✓	✓		✓				✓	✓			✓	
2018	[Ma et al., 2018]	SAE-NAD	✓			✓	✓	✗			✓	✓	✓				✓
2018	[Gao et al., 2018]	GeoEISo	✓			✓	✓	✓			✓	✓	✓			✓	
2018	[Wang et al., 2018]	GeoIE	✓	✓		✓		✓				✓		✓			✓
2019	[Ying et al., 2019]	MEAP-T	✓	✓		✓	✓	✓		✓	✓			✓			✓
2019	[Si et al., 2019]	APRA-SA			✓	✓		✓			✓	✓	✓				✓
2019	[Qian et al., 2019]	STA		✓		✓		✓	✓			✓		✓			✓
Most Representatives			24	10	5	38	23	C:21 P:16	7	3	30	26	18	17	1	13	22
Total			135	37	22	229	135	C:150 P:66	27	29	147	142	123	82	14	101	104

- It is **not common** to use a **validation split**

Point-Of-Interest recommendation: a survey

Details			Evaluation configuration							Baselines			Split type			Split level	
Year	Reference	Acronym	Filter data	Validation	Error	Ranking	Region Split	Check-in(✓) POI(✗)	Cold Start Analysis	C. Non Personalized	C. Personalized	Geographical	Random	Temporal	Other	System	Per User
2011	[Ye et al., 2011]	USG				✓		✗	✓		✓	✓	✓				✓
2012	[Levandoski et al., 2012]	LARS					✓	✗			✓	✓	✓			✓	
2012	[Bao et al., 2012]	(N.A.)	✓			✓	✓	✗			✓				✓		
2013	[Yang et al., 2013]	LBSMF	✓		✓		✓	✓			✓		✓			✓	
2013	[Liu et al., 2013]	GT-BNMF				✓	✓	✓			✓		✓			✓	
2013	[Yuan et al., 2013]	UTE+SE	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Ying et al., 2014]	UPOI-Walk			✓	✓	✓	?			✓	✓					
2014	[Yuan et al., 2014]	GTAG	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Lian et al., 2014]	GeoMF	✓			✓		✗			✓		✓				✓
2015	[Yin et al., 2015]	LA-LDA		✓		✓		✗	✓			✓	✓				✓
2015	[Li et al., 2015]	RankGeoFM		✓		✓	✓	✓			✓	✓		✓			✓
2015	[Zhang and Chow, 2015]	GeoSoCa				✓	✓	✓				✓		✓		✓	
2015	[Feng et al., 2015]	PRME-G	✓	✓		✓	✓	✓		✓	✓			✓		✓	
2016	[Li et al., 2016]	ASMF	✓			✓	✓	✗	✓		✓	✓		✓			✓
2016	[Zhao et al., 2016]	STELLAR	✓			✓		✓			✓			✓		✓	
2017	[Zhao et al., 2017]	Geo-Teaser	✓			✓		✓			✓	✓		✓			✓
2017	[Yang et al., 2017]	PACE	✓			✓		✓				✓		✓			✓
2017	[Ren et al., 2017]	TGSC-PMF	✓		✓	✓		✓				✓	✓			✓	
2018	[Ma et al., 2018]	SAE-NAD	✓			✓	✓	✗			✓	✓	✓				✓
2018	[Gao et al., 2018]	GeoEISo	✓			✓	✓	✓			✓	✓	✓			✓	
2018	[Wang et al., 2018]	GeoIE	✓	✓		✓		✓				✓		✓			✓
2019	[Ying et al., 2019]	MEAP-T	✓	✓		✓	✓	✓		✓	✓			✓			✓
2019	[Si et al., 2019]	APRA-SA			✓	✓		✓			✓	✓	✓				✓
2019	[Qian et al., 2019]	STA		✓		✓		✓	✓			✓		✓			✓
Most Representatives			24	10	5	38	23	C:21 P:16	7	3	30	26	18	17	1	13	22
Total			135	37	22	229	135	C:150 P:66	27	29	147	142	123	82	14	101	104

- **No standard procedure** for evaluating the models

Point-Of-Interest recommendation: a survey

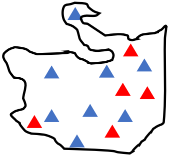
Details			Evaluation configuration							Baselines			Split type			Split level	
Year	Reference	Acronym	Filter data	Validation	Error	Ranking	Region Split	Check-in(✓) POI(✗)	Cold Start Analysis	C. Non Personalized	C. Personalized	Geographical	Random	Temporal	Other	System	Per User
2011	[Ye et al., 2011]	USG				✓		✗	✓		✓	✓	✓				✓
2012	[Levandoski et al., 2012]	LARS					✓	✗			✓	✓	✓			✓	
2012	[Bao et al., 2012]	(N.A.)	✓			✓	✓	✗			✓				✓		
2013	[Yang et al., 2013]	LBSMF	✓		✓		✓	✓			✓		✓			✓	
2013	[Liu et al., 2013]	GT-BNMF				✓	✓	✓			✓		✓			✓	
2013	[Yuan et al., 2013]	UTE+SE	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Ying et al., 2014]	UPOI-Walk			✓	✓	✓	?			✓	✓					
2014	[Yuan et al., 2014]	GTAG	✓	✓		✓	✓	✗			✓	✓	✓				✓
2014	[Lian et al., 2014]	GeoMF	✓			✓		✗			✓		✓				✓
2015	[Yin et al., 2015]	LA-LDA		✓		✓		✗	✓			✓	✓				✓
2015	[Li et al., 2015]	RankGeoFM		✓		✓	✓	✓			✓	✓		✓			✓
2015	[Zhang and Chow, 2015]	GeoSoCa				✓	✓	✓				✓		✓		✓	
2015	[Feng et al., 2015]	PRME-G	✓	✓		✓	✓	✓		✓	✓			✓		✓	
2016	[Li et al., 2016]	ASMF	✓			✓	✓	✗	✓		✓	✓		✓			✓
2016	[Zhao et al., 2016]	STELLAR	✓			✓		✓			✓			✓		✓	
2017	[Zhao et al., 2017]	Geo-Teaser	✓			✓		✓			✓	✓		✓			✓
2017	[Yang et al., 2017]	PACE	✓			✓		✓				✓		✓			✓
2017	[Ren et al., 2017]	TGSC-PMF	✓		✓	✓		✓				✓	✓			✓	
2018	[Ma et al., 2018]	SAE-NAD	✓			✓	✓	✗			✓	✓	✓				✓
2018	[Gao et al., 2018]	GeoEISo	✓			✓	✓	✓			✓	✓	✓			✓	
2018	[Wang et al., 2018]	GeoIE	✓	✓		✓		✓				✓		✓			✓
2019	[Ying et al., 2019]	MEAP-T	✓	✓		✓	✓	✓		✓	✓			✓			✓
2019	[Si et al., 2019]	APRA-SA			✓	✓		✓			✓	✓	✓				✓
2019	[Qian et al., 2019]	STA		✓		✓		✓	✓			✓		✓			✓
Most Representatives			24	10	5	38	23	C:21 P:16	7	3	30	26	18	17	1	13	22
Total			135	37	22	229	135	C:150 P:66	27	29	147	142	123	82	14	101	104

- Some researchers use some kind of **region/city split**

Improving POI recommendation performance

- Some researchers tend to consider each city/region as an independent dataset (same city/region for training and test)

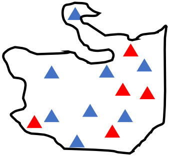
Training with one city
and test with the same city



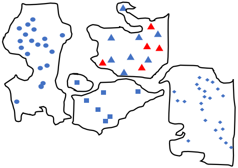
Improving POI recommendation performance

- Some researchers tend to consider each city/region as an independent dataset (same city/region for training and test)

Training with one city
and test with the same city



Training with many cities
and test with one city



- We propose different strategies to select the training cities: based on distance (N-MCA and C-MCA) and based on the number of check-ins (most popular, P-MCA)

- Objective: **test our MCA strategies** in a LBSN dataset

Experiments: POI recommendation

- Objective: **test our MCA strategies** in a LBSN dataset
- **8 different cities** from the Global-scale check-in dataset [Yang et al., 2016]: Istanbul, Jakarta, Kuala Lumpur, Mexico City, Moscow, Santiago, São Paulo and Tokyo

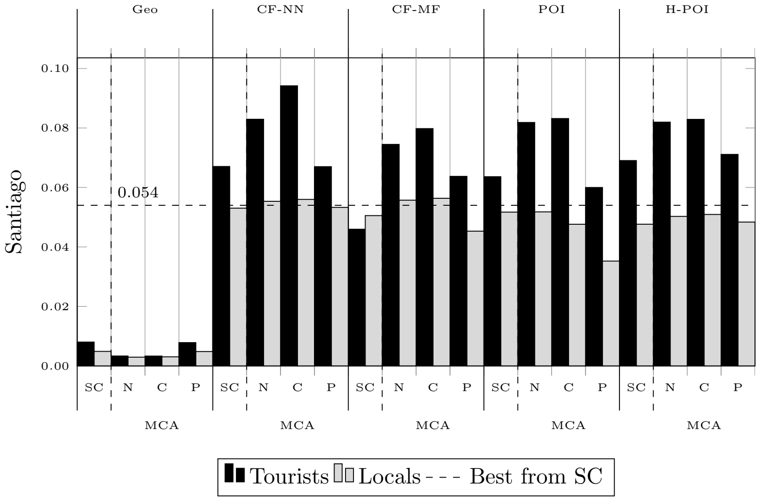
- Objective: **test our MCA strategies** in a LBSN dataset
- **8 different cities** from the Global-scale check-in dataset [Yang et al., 2016]: Istanbul, Jakarta, Kuala Lumpur, Mexico City, Moscow, Santiago, São Paulo and Tokyo
- **Temporal system split**

- Objective: **test our MCA strategies** in a LBSN dataset
- **8 different cities** from the Global-scale check-in dataset [Yang et al., 2016]: Istanbul, Jakarta, Kuala Lumpur, Mexico City, Moscow, Santiago, São Paulo and Tokyo
- **Temporal system split**
- **3 different MCA** strategies: the test set is always formed by the target city

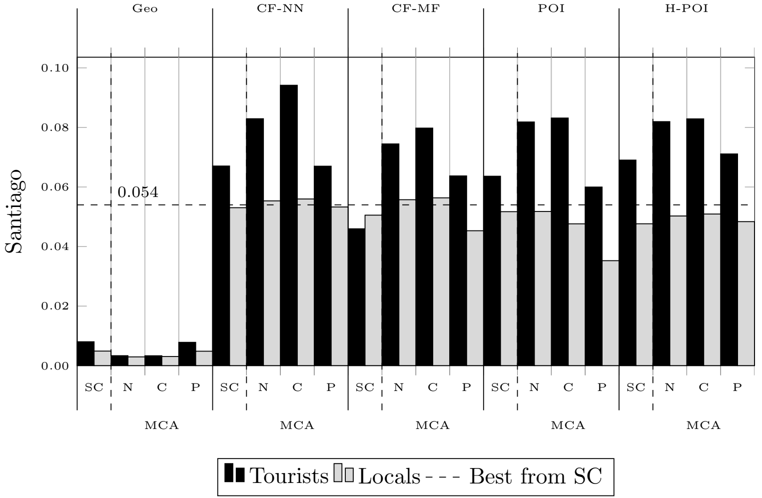
Experiments: POI recommendation

- Objective: **test our MCA strategies** in a LBSN dataset
- **8 different cities** from the Global-scale check-in dataset [Yang et al., 2016]: Istanbul, Jakarta, Kuala Lumpur, Mexico City, Moscow, Santiago, São Paulo and Tokyo
- **Temporal system split**
- **3 different MCA** strategies: the test set is always formed by the target city
- **2 different** groups of users: **tourists and locals**

Experiments: POI recommendation. nDCG@5

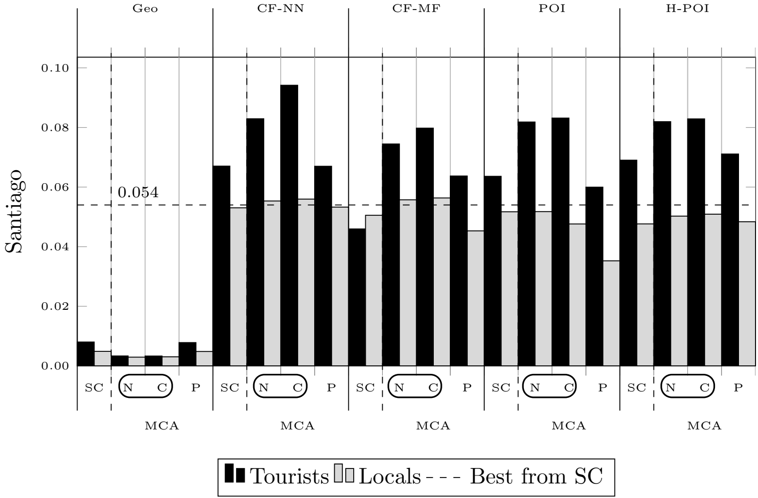


- Families: Geo, CF-NN, CF-MF, POI, H-POI



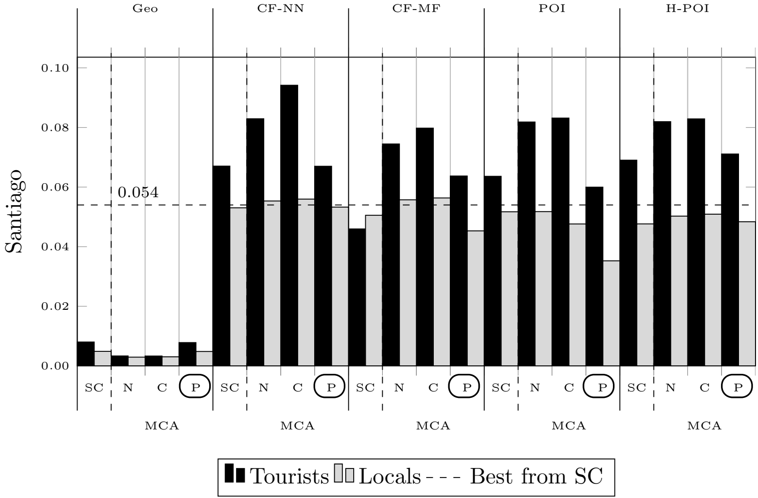
- N-MCA (closest), C-MCA (country), P-MCA (popular)

Experiments: POI recommendation. nDCG@5



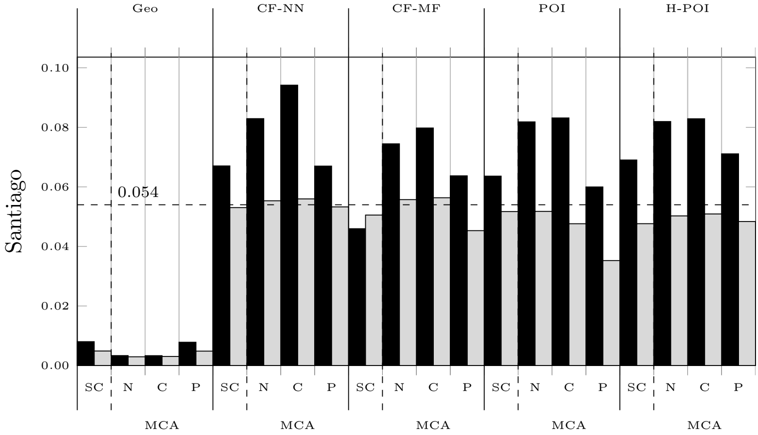
- N-MCA and C-MCA increase relevance

Experiments: POI recommendation. nDCG@5



- P-MCA sometimes decreases relevance

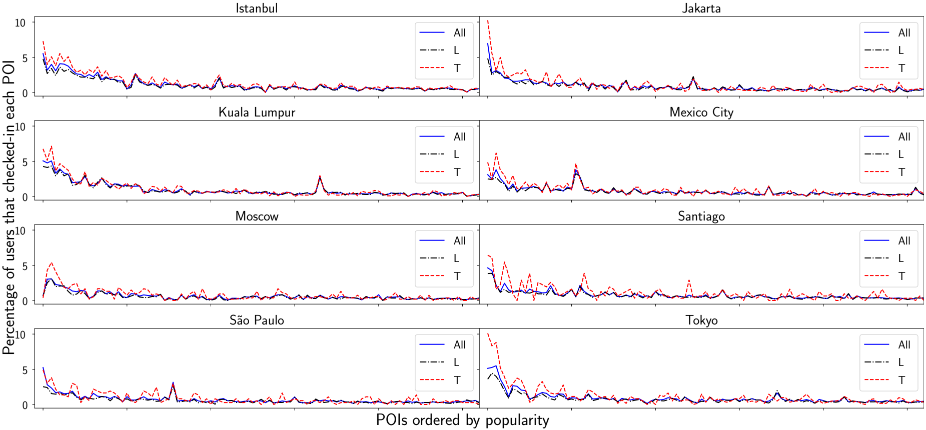
Experiments: POI recommendation. nDCG@5



Tourists Locals - - - Best from SC

- Great differences between tourists and locals

Experiments: popularity bias in tourists



- Tourist tend to visit the most popular venues

- **Most POI** recommendation algorithms **are not comparable** between them

Summary

- **Most POI** recommendation algorithms **are not comparable** between them
- POI recommendation **is highly affected** by the **geographical influence** and its **sparsity**

Summary

- **Most POI** recommendation algorithms **are not comparable** between them
- POI recommendation **is highly affected** by the **geographical influence** and its **sparsity**
- We can **improve** the **performance** of the recommenders by using **multi-city aggregation** strategies

Summary

- **Most POI** recommendation algorithms **are not comparable** between them
- POI recommendation **is highly affected** by the **geographical influence** and its **sparsity**
- We can **improve** the **performance** of the recommenders by using **multi-city aggregation** strategies
- **Quality** over quantity (of the data)

Overview

Sequences in POI recommendation

- We address the last objective: **generate routes** from POI data

Sequences in POI recommendation

- We address the last objective: **generate routes** from POI data
- We use data from **different sources** (Foursquare, Tripbuilder, and SemanticTrails)

Sequences in POI recommendation

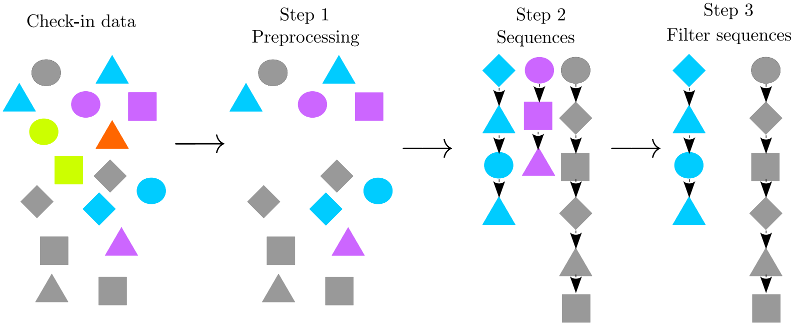
- We address the last objective: **generate routes** from POI data
- We use data from **different sources** (Foursquare, Tripbuilder, and SemanticTrails)
- We apply **reranking techniques** to generate routes

Sequences in POI recommendation

- We address the last objective: **generate routes** from POI data
- We use data from **different sources** (Foursquare, Tripbuilder, and SemanticTrails)
- We apply **reranking techniques** to generate routes
- **Contributions published** in User Modeling and User-Adapted Interaction [Sánchez and Bellogín, 2020a] journal

Generate sequences from check-in data

- From LBSN like Foursquare or Gowalla

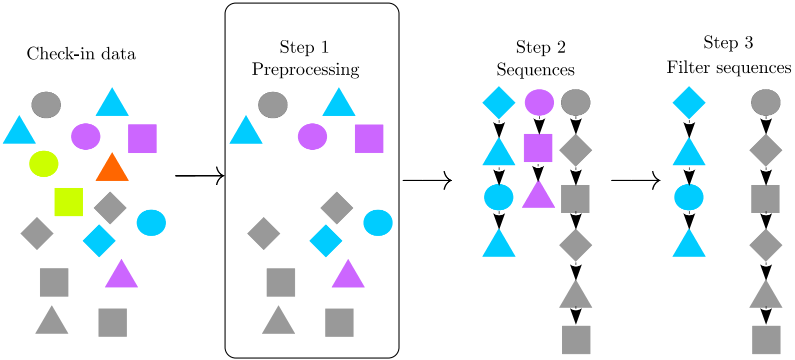


Each color represents a user

Each figure represents a POI category

Generate sequences from check-in data

- From LBSN like Foursquare or Gowalla

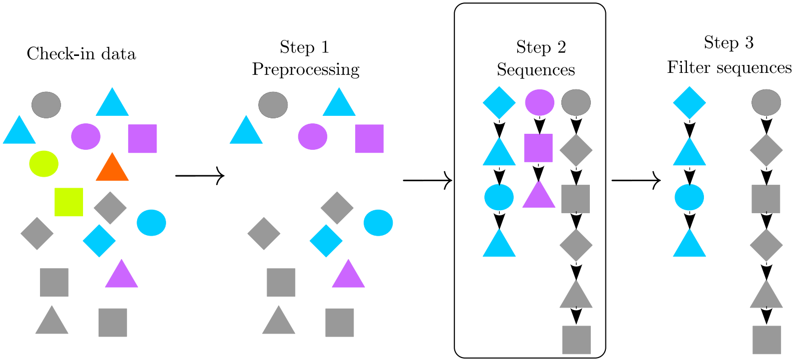


Each color represents a user

Each figure represents a POI category

Generate sequences from check-in data

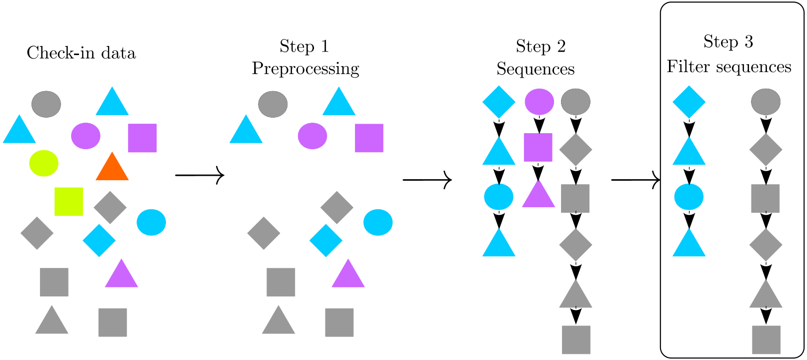
- From LBSN like Foursquare or Gowalla



Each color represents a user

Each figure represents a POI category

- From LBSN like Foursquare or Gowalla



Each color represents a user

Each figure represents a POI category

Using reranking techniques to generate routes

$$f_{obj}(u, i, R_u) = \lambda \cdot f_{rec}(u, i) + (1 - \lambda) \cdot f_{seq}(u, i, R_u)$$

Using reranking techniques to generate routes

$$f_{obj}(u, i, R_u) = \lambda \cdot f_{rec}(u, i) + (1 - \lambda) \cdot f_{seq}(u, i, R_u)$$

- We propose three different families of rerankers:

Using reranking techniques to generate routes

$$f_{obj}(u, i, R_u) = \lambda \cdot f_{rec}(u, i) + (1 - \lambda) \cdot f_{seq}(u, i, R_u)$$

- We propose three different families of rerankers:
  1. Independent
     - Random: $f_{seq}^{rnd}(u, i, R_u) = \text{rnd} \in [0, 1]$
     - Recommender-based: $f_{seq}^{rec}(u, i, R_u) = r(u, i)$

Using reranking techniques to generate routes

$$f_{obj}(u, i, R_u) = \lambda \cdot f_{rec}(u, i) + (1 - \lambda) \cdot f_{seq}(u, i, R_u)$$

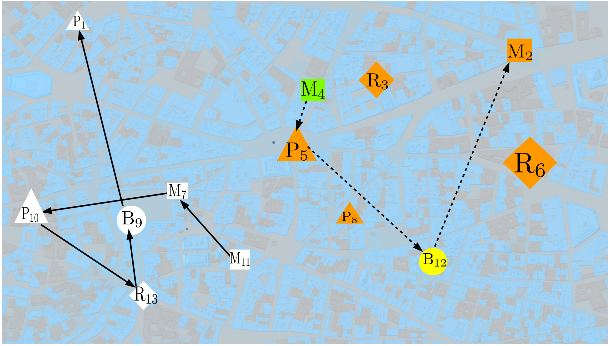
- We propose three different families of rerankers:
  1. Independent
     - Random: $f_{seq}^{rnd}(u, i, R_u) = \text{rnd} \in [0, 1]$
     - Recommender-based: $f_{seq}^{rec}(u, i, R_u) = r(u, i)$
  2. Dependent on the last item
     - Distance: $f_{seq}^{dist}(u, i, R_u) = 1/dist(i_{n-1}, i)$
     - Feature Markov Chain: $f_{seq}^{feat}(u, i, R_u) = p(i^a | i_{n-1}^a)$
     - Item Markov Chain: $f_{seq}^{item}(u, i, R_u) = p(i | i_{n-1})$

Using reranking techniques to generate routes

$$f_{obj}(u,i,R_u) = \lambda \cdot f_{rec}(u,i) + (1-\lambda) \cdot f_{seq}(u,i,R_u)$$

- We propose three different families of rerankers:
  1. Independent
     - Random: $f_{seq}^{rnd}(u,i,R_u) = \text{rnd} \in [0,1]$
     - Recommender-based: $f_{seq}^{rec}(u,i,R_u) = r(u,i)$
  2. Dependent on the last item
     - Distance: $f_{seq}^{dist}(u,i,R_u) = 1/dist(i_{n-1},i)$
     - Feature Markov Chain: $f_{seq}^{feat}(u,i,R_u) = p(i^a|i_{n-1}^a)$
     - Item Markov Chain: $f_{seq}^{item}(u,i,R_u) = p(i|i_{n-1})$
  3. Dependent on the whole sequence
     - LCS-based: $f_{seq}^{lcs}(u,i,R_u) = lcs((R_u+i)^a, u^a)$
     - Suffix tree: $f_{seq}^{stree}(u,i,R_u) = \delta_{ST(u^a)}(\{(R_u+i)^a\}_m)$
     - Oracle: $f_{seq}^{oracle}(u,i,R_u) = order_{test}(u,i)$

Using reranking techniques to generate routes



f^{lcs}_{seq} $M_4 \rightarrow P_5 \rightarrow R_3 \rightarrow P_8$

f^{dist}_{seq} $M_4 \rightarrow P_5 \rightarrow P_8 \rightarrow R_3 \rightarrow M_2 \rightarrow R_6$

f^{stree}_{seq} $M_4 \rightarrow P_5 \rightarrow R_3$

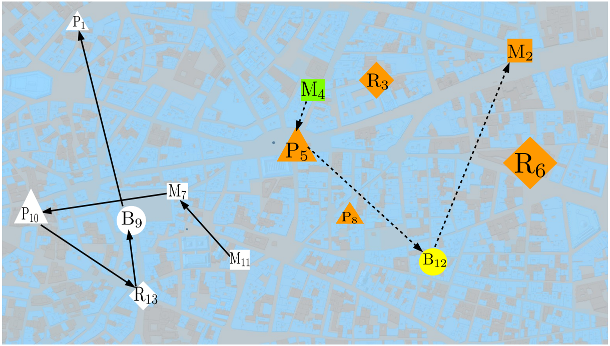
f^{rec}_{seq} $M_4 \rightarrow R_6 \rightarrow P_5 \rightarrow R_3 \rightarrow M_2 \rightarrow P_8$

f^{oracle}_{seq} $M_4 \rightarrow P_5 \rightarrow M_2$

Training   Recommended venues   First venue in the sequence

Venue in test not recommended   Test route of the user

Using reranking techniques to generate routes



f_{seq}^{lcs} $M_4 \rightarrow P_5 \rightarrow R_3 \rightarrow P_8$

$\boxed{f_{seq}^{dist} \; M_4 \rightarrow P_5 \rightarrow P_8 \rightarrow R_3 \rightarrow M_2 \rightarrow R_6}$

f_{seq}^{stree} $M_4 \rightarrow P_5 \rightarrow R_3$

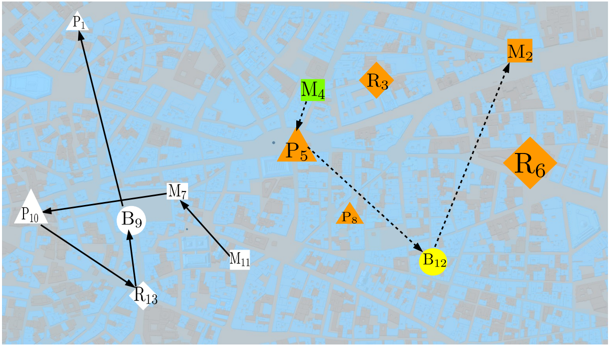
f_{seq}^{rec} $M_4 \rightarrow R_6 \rightarrow P_5 \rightarrow R_3 \rightarrow M_2 \rightarrow P_8$

f_{seq}^{oracle} $M_4 \rightarrow P_5 \rightarrow M_2$

Training Recommended venues First venue in the sequence

Venue in test not recommended Test route of the user

Using reranking techniques to generate routes



f_{seq}^{lcs} $M_4 \to P_5 \to R_3 \to P_8$

f_{seq}^{dist} $M_4 \to P_5 \to P_8 \to R_3 \to M_2 \to R_6$

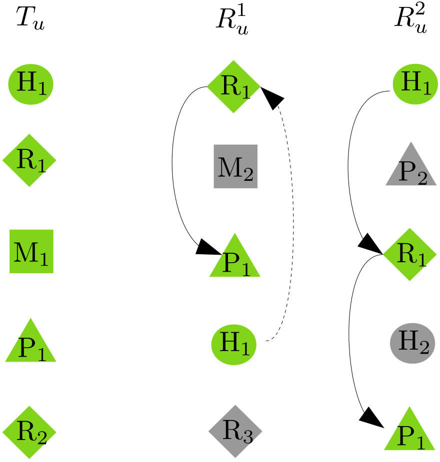
f_{seq}^{stree} $M_4 \to P_5 \to R_3$

$\boxed{f_{seq}^{rec} \quad M_4 \to R_6 \to P_5 \to R_3 \to M_2 \to P_8}$

f_{seq}^{oracle} $M_4 \to P_5 \to M_2$

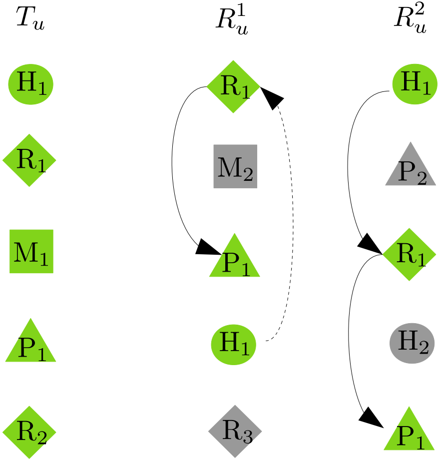
Training   Recommended venues   First venue in the sequence

Venue in test not recommended   Test route of the user



$$\mathrm{P}(R_u^1) = \mathrm{P}(R_u^2) = 3/5$$

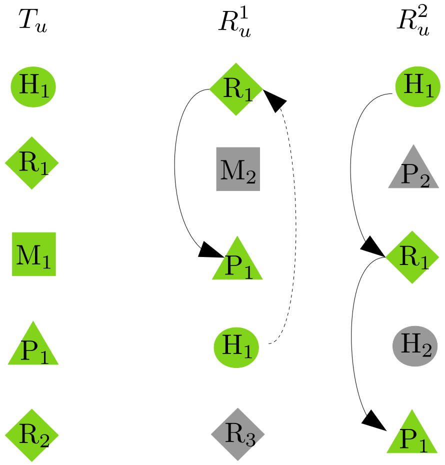
Sequential evaluation



$$\mathrm{P}(R_u^1) = \mathrm{P}(R_u^2) = 3/5$$

$$LCS(T_u, R_u^1) = 2$$
$$LCS(T_u, R_u^2) = 3$$

Sequential evaluation



$$\mathrm{P}(R_u^1) = \mathrm{P}(R_u^2) = 3/5$$

$$LCS(T_u, R_u^1) = 2$$
$$LCS(T_u, R_u^2) = 3$$

$$\mathrm{P_s}\ (R_u^2) = \mathrm{P}(R_u^2) = 3/5$$
$$\mathrm{P_s}\ (R_u^1) = 2/5$$

- Objective: check if our **reranking strategies improve** the performance of the recommenders

Experiments: sequences in POI recommendation

- Objective: check if our **reranking strategies improve** the performance of the recommenders
- 3 different datasets and 4 different cities: Foursquare (**New York**, Tokyo, generating our own routes), Semantic Trails (Petaling Jaya), Trip Builder (**Rome**)

Experiments: sequences in POI recommendation

- Objective: check if our **reranking strategies improve** the performance of the recommenders
- 3 different datasets and 4 different cities: Foursquare (**New York**, Tokyo, generating our own routes), Semantic Trails (Petaling Jaya), Trip Builder (**Rome**)
- **Last session** of the users to the test set, rest to training

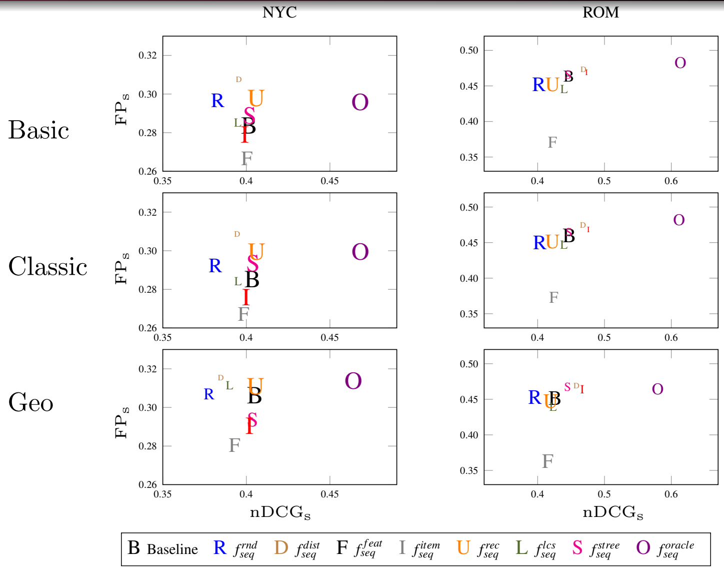
Experiments: sequences in POI recommendation

- Objective: check if our **reranking strategies improve** the performance of the recommenders
- 3 different datasets and 4 different cities: Foursquare (**New York**, Tokyo, generating our own routes), Semantic Trails (Petaling Jaya), Trip Builder (**Rome**)
- **Last session** of the users to the test set, rest to training
- **Families** of recommenders: Basic, Classic, Temporal, Geo, Tour

Experiments: sequences in POI recommendation

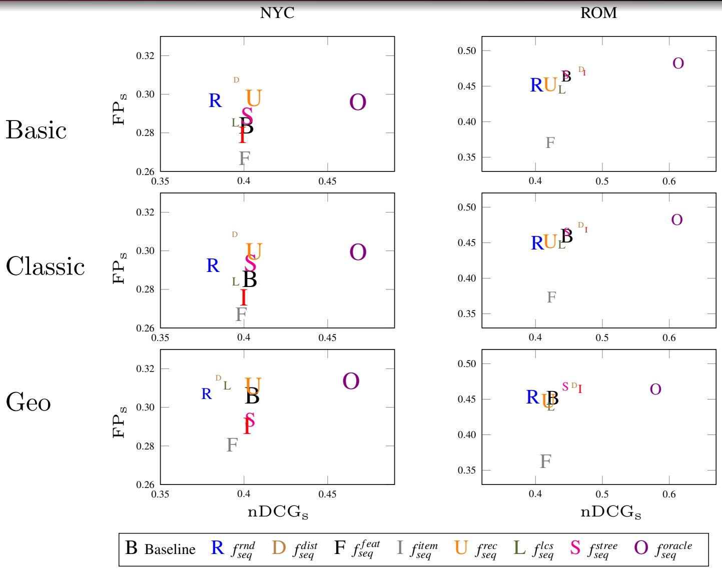
- Objective: check if our **reranking strategies improve** the performance of the recommenders
- 3 different datasets and 4 different cities: Foursquare (**New York**, Tokyo, generating our own routes), Semantic Trails (Petaling Jaya), Trip Builder (**Rome**)
- **Last session** of the users to the test set, rest to training
- **Families** of recommenders: Basic, Classic, Temporal, Geo, Tour
- Analysis on relevance, **sequential relevance**, novelty, diversity, **attribute evaluation** and **distance**

Experiments: effect of rerankers



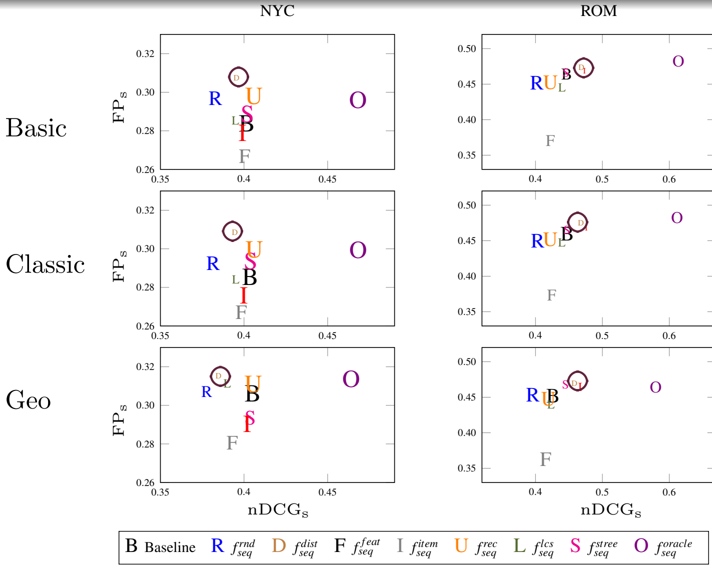
- Accuracy (x-axis), feature accuracy (y-axis), distance (size)

Experiments: effect of rerankers



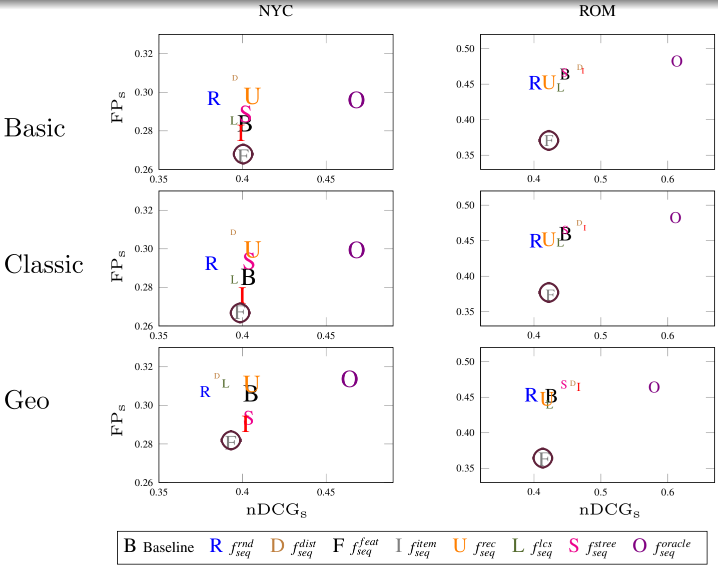
- There is always at least one reranker that improves the baseline

Experiments: effect of rerankers



- Distance reranker often improves the relevance

Experiments: effect of rerankers



- Categorical rerankers do not always obtain better results

- We have shown how to **generate sequences** from **POI recommendation** data

Summary

- We have shown how to **generate sequences** from **POI recommendation** data
- We proposed to **evaluate** the recommendations using **sequential metrics**

Summary

- We have shown how to **generate sequences** from **POI recommendation** data
- We proposed to **evaluate** the recommendations using **sequential metrics**
- We have shown how we can use **reranking techniques** for **generating routes** optimizing different criteria

Conclusions

- RO1: **Recommender Systems evaluation**

Conclusions

- RO1: **Recommender Systems evaluation**
  - There is a clear **relationship** between the temporal novelty of the items and their relevance

Conclusions

- RO1: **Recommender Systems evaluation**
  - There is a clear **relationship** between the temporal novelty of the items and their relevance
  - Importance of analyzing the **anti-relevance** of the items. **Personalized** recommendations often return **anti-relevant items** for the users

Conclusions

- RO1: **Recommender Systems evaluation**
  - There is a clear **relationship** between the temporal novelty of the items and their relevance
  - Importance of analyzing the **anti-relevance** of the items. **Personalized** recommendations often return **anti-relevant items** for the users
  - With the **user attributes** we may detect **biases** in specific **groups of users**. With **item attributes** we can **increase** the performance of the recommenders in very sparse datasets

Conclusions (II)

- RO2: **Sequences in k-NN recommenders**

Conclusions (II)

- RO2: **Sequences in k-NN recommenders**
  - We showed how to incorporate **sequential** information in **k-NN recommenders** by defining a **similarity metric** and by **reformulating** them

Conclusions (II)

- RO2: **Sequences in k-NN recommenders**
  - We showed how to incorporate **sequential** information in **k-NN recommenders** by defining a **similarity metric** and by **reformulating** them
  - Our reformulation of k-NN recommenders is **intuitive**, easy to explain and allows us to work with any **similarity metric**

Conclusions (II)

- RO2: **Sequences in k-NN recommenders**
  - We showed how to incorporate **sequential** information in **k-NN recommenders** by defining a **similarity metric** and by **reformulating** them
  - Our reformulation of k-NN recommenders is **intuitive**, easy to explain and allows us to work with any **similarity metric**
  - Our proposal was **highly competitive** against other **state-of-the-art** algorithms in **different dimensions**

Conclusions (III)

- RO3: **Review POI algorithms**

Conclusions (III)

- RO3: **Review POI algorithms**
  - Most **POI** approaches are **not comparable** as they use very different **evaluation protocols**

- RO3: **Review POI algorithms**
  - Most **POI** approaches are **not comparable** as they use very different **evaluation protocols**
  - Very few **researchers** provide the **source code** of their models

Conclusions (IV)

- RO4: **Improve the performance of POI RS**

Conclusions (IV)

- RO4: **Improve the performance of POI RS**
  - **Cross-domain** techniques **increase** the performance of the recommenders in terms of **relevance** and **user coverage**

Conclusions (IV)

- RO4: **Improve the performance of POI RS**
  - **Cross-domain** techniques **increase** the performance of the recommenders in terms of **relevance** and **user coverage**
  - **Augment the information** using the **cities by distance** obtain **better results** that using the most popular cities

Conclusions (IV)

- RO4: **Improve the performance of POI RS**
  - **Cross-domain** techniques **increase** the performance of the recommenders in terms of **relevance** and **user coverage**
  - **Augment the information** using the **cities by distance** obtain **better results** that using the most popular cities
  - **Useful information** is better than more information

Conclusions (V)

- RO5: **Generate routes** from Location-Based Social Networks data

Conclusions (V)

- RO5: **Generate routes** from Location-Based Social Networks data
  - We can generate **meaningful routes** from LBSNs data

- RO5: **Generate routes** from Location-Based Social Networks data
    - We can generate **meaningful routes** from LBSNs data
    - We can use **reranking techniques** for generating routes improving dimensions like **feature precision** and/or **distance**

Future Work (I)

- On Recommender Systems evaluation

Future Work (I)

- On Recommender Systems evaluation
  - Test our **novelty metrics** in **online environments**

Future Work (I)

- On Recommender Systems evaluation
  - Test our **novelty metrics** in **online environments**
  - Test our **anti-relevance models** in domains with **implicit information**

Future Work (I)

- On Recommender Systems evaluation
  - Test our **novelty metrics** in **online environments**
  - Test our **anti-relevance models** in domains with **implicit information**
  - Detect **biases in different groups** of users and in other recommendation domains

Future Work (I)

- On Recommender Systems evaluation
  - Test our **novelty metrics** in **online environments**
  - Test our **anti-relevance models** in domains with **implicit information**
  - Detect **biases in different groups** of users and in other recommendation domains
  - Apply our **attribute metrics** in **other domains** like music

Future Work (II)

- On sequential-based k-NN recommenders

Future Work (II)

- On sequential-based k-NN recommenders
  - Extend the **LCS similarity** to incorporate item **features** or the **distance** of the POIs

Future Work (II)

- On sequential-based k-NN recommenders
  - Extend the **LCS similarity** to incorporate item **features** or the **distance** of the POIs
  - Perform more complete analysis using different **validation** subsets

Future Work (II)

- On sequential-based k-NN recommenders
  - Extend the **LCS similarity** to incorporate item **features** or the **distance** of the POIs
  - Perform more complete analysis using different **validation** subsets
- On Point-of-Interest recommendation

Future Work (II)

- On sequential-based k-NN recommenders
  - Extend the **LCS similarity** to incorporate item **features** or the **distance** of the POIs
  - Perform more complete analysis using different **validation** subsets
- On Point-of-Interest recommendation
  - Perform a survey focusing on the **reproducibility** of the model under different **evaluation methodologies** (splits, datasets, etc.)

- On sequential-based k-NN recommenders
  - Extend the **LCS similarity** to incorporate item **features** or the **distance** of the POIs
  - Perform more complete analysis using different **validation** subsets
- On Point-of-Interest recommendation
  - Perform a survey focusing on the **reproducibility** of the model under different **evaluation methodologies** (splits, datasets, etc.)
  - Propose different **aggregation strategies** and use other algorithms based on **items similarities** like Factored Item Similarity Models (FISM) or Sparse Linear Methods (SLIM)

Exploring attributes, sequences, and time in Recommender Systems: From classical to Point-of-Interest recommendation

Pablo Sánchez Pérez

Under the supervision of
Alejandro Bellogín Kouki

Information Retrieval Group
Department of Computer Science
Universidad Autónoma de Madrid, Spain

July 8, 2021

Thank you

`https://bitbucket.org/PabloSanchezP`

Publications: journals

1. **Pablo Sánchez** and Alejandro Bellogín. (2020). Point-of-Interest Recommender Systems: A Survey from an Experimental Perspective. Submitted to *ACM Computing Surveys*. Under Review (2nd round of review)
2. **Pablo Sánchez** and Alejandro Bellogín. On the effects of aggregation strategies for different groups of users in venue recommendation. *Information Processing and Management*, 58(5):102609, 2021.
3. **Pablo Sánchez** and Alejandro Bellogín. Applying reranking strategies to route recommendation using sequence-aware evaluation. *User Modeling and User-Adapted Interaction*, 30(4):659-725, 2020
4. **Pablo Sánchez** and Alejandro Bellogín. Time and sequence awareness in similarity metrics for recommendation. *Information Processing and Management*, 57(3):102228, 2020
5. **Pablo Sánchez** and Alejandro Bellogín. Building user profiles based on sequences for content and collaborative filtering. *Information Processing and Management*, 56(1):192-211, 2019.
6. Alejandro Bellogín and **Pablo Sánchez**. Collaborative filtering based on subsequence matching: A new approach. *Information Sciences*, 418:432-446, 2017

Publications: conferences

1. **Pablo Sánchez** and Alejandro Bellogín. Time-aware novelty metrics for recommender systems. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *Advances in Information Retrieval - 40th European Conference on IR Research*, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings, volume 10772 of Lecture Notes in Computer Science, pages 357-370. Springer, 2018
2. **Pablo Sánchez** and Alejandro Bellogín. Measuring anti-relevance: a study on when recommendation algorithms produce bad suggestions. In Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan, editors, *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys 2018, Vancouver, BC, Canada, October. 2-7, 2018, pages 367-371. ACM, 2018
3. **Pablo Sánchez** and Alejandro Bellogín. Attribute-based evaluation for recommender systems: incorporating user and item attributes in evaluation metrics. In Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019., pages 378-382. ACM, 2019

References I

Adomavicius, G. and Tuzhilin, A. (2005).
Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.
IEEE Trans. Knowl. Data Eng., 17(6):734–749.

Bao, J., Zheng, Y., and Mokbel, M. F. (2012).
Location-based and preference-aware recommendation using sparse geo-social networking data.
In Cruz, I. F., Knoblock, C. A., Kröger, P., Tanin, E., and Widmayer, P., editors, *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012*, pages 199–208. ACM.

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

Dooms, S., Bellogín, A., Pessemier, T. D., and Martens, L. (2016).
A framework for dataset benchmarking and its application to a new movie rating dataset.
ACM TIST, 7(3):41:1–41:28.

Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., and Yuan, Q. (2015).
Personalized ranking metric embedding for next new POI recommendation.
In Yang, Q. and Wooldridge, M. J., editors, *Proceedings of the Twenty-Fourth International Joint Conference on*

Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, pages 2069–2075. AAAI Press.

Gao, R., Li, J., Li, X., Song, C., and Zhou, Y. (2018).
A personalized point-of-interest recommendation model via fusion of geo-social information.
Neurocomputing, 273:159–170.

He, R. and McAuley, J. (2016).
Fusing similarity models with markov chains for sparse sequential recommendation.
In Bonchi, F., Domingo-Ferrer, J., Baeza-Yates, R. A., Zhou, Z., and Wu, X., editors, *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 191–200. IEEE.

Levandoski, J. J., Sarwat, M., Eldawy, A., and Mokbel, M. F. (2012).
LARS: A location-aware recommender system.
In Kementsietsidis, A. and Salles, M. A. V., editors, *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 450–461. IEEE Computer Society.

Li, H., Ge, Y., Hong, R., and Zhu, H. (2016).
Point-of-interest recommendations: Learning potential check-ins from friends.
In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 975–984. ACM.

Li, X., Cong, G., Li, X., Pham, T. N., and Krishnaswamy, S. (2015).
Rank-geofm: A ranking based geographical factorization method for point of interest recommendation.
In Baeza-Yates, R. A., Lalmas, M., Moffat, A., and Ribeiro-Neto, B. A., editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 433–442. ACM.

Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., and Rui, Y. (2014).
Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation.
In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W., and Ghani, R., editors, *The 20th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pages 831–840. ACM.

Liu, B., Fu, Y., Yao, Z., and Xiong, H. (2013).
Learning geographical preferences for point-of-interest recommendation.
In Dhillon, I. S., Koren, Y., Ghani, R., Senator, T. E., Bradley, P., Parekh, R., He, J., Grossman, R. L., and Uthurusamy, R., editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 1043–1051. ACM.

Ma, C., Zhang, Y., Wang, Q., and Liu, X. (2018).
Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence.
In Cuzzocrea, A., Allan, J., Paton, N. W., Srivastava, D., Agrawal, R., Broder, A. Z., Zaki, M. J., Candan, K. S., Labrinidis, A., Schuster, A., and Wang, H., editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 697–706. ACM.

Miller, H. J. (2004).
Tobler's first law and spatial analysis.
Annals of the Association of American Geographers, 94(2):284–289.

Qian, T., Liu, B., Nguyen, Q. V. H., and Yin, H. (2019).
Spatiotemporal representation learning for translation-based POI recommendation.
ACM Trans. Inf. Syst., 37(2):18:1–18:24.

Ren, X., Song, M., E, H., and Song, J. (2017).
Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation.
Neurocomputing, 241:38–55.

Robertson, S. E. (1997).
Readings in information retrieval.
In Sparck Jones, K. and Willett, P., editors, *Readings in Information Retrieval*, chapter The Probability Ranking Principle in IR, pages 281–286. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sánchez, P. and Bellogín, A. (2018a).
Measuring anti-relevance: a study on when recommendation algorithms produce bad suggestions.
In Pera, S., Ekstrand, M. D., Amatriain, X., and O'Donovan, J., editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 367–371. ACM.

Sánchez, P. and Bellogín, A. (2018b).
Time-aware novelty metrics for recommender systems.
In Pasi, G., Piwowarski, B., Azzopardi, L., and Hanbury, A., editors, *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume

10772 of *Lecture Notes in Computer Science*, pages 357–370. Springer.

Sánchez, P. and Bellogín, A. (2019a).
Attribute-based evaluation for recommender systems: incorporating user and item attributes in evaluation metrics.
In Bogers, T., Said, A., Brusilovsky, P., and Tikk, D., editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2017, Copenhagen, Denmark, September 16-20, 2019.*, pages 378–382. ACM.

Sánchez, P. and Bellogín, A. (2019b).
Building user profiles based on sequences for content and collaborative filtering.
Inf. Process. Manage., 56(1):192–211.

Sánchez, P. and Bellogín, A. (2020a).
Applying reranking strategies to route recommendation using sequence-aware evaluation.
User Model. User Adapt. Interact., 30(4):659–725.

Sánchez, P. and Bellogín, A. (2020b).
Time and sequence awareness in similarity metrics for recommendation.
Inf. Process. Manag., 57(3):102228.

Si, Y., Zhang, F., and Liu, W. (2019).
An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features.
Knowl.-Based Syst., 163:267–282.

Sánchez, P. and Bellogín, A. (2021).
On the effects of aggregation strategies for different groups of users in venue recommendation.
Information Processing and Management, 58(5):102609.

Vargas, S. and Castells, P. (2011).
Rank and relevance in novelty and diversity metrics for recommender systems.
In Mobasher, B., Burke, R. D., Jannach, D., and Adomavicius, G., editors, *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, pages 109–116. ACM.

- Wang, H., Shen, H., Ouyang, W., and Cheng, X. (2018). Exploiting poi-specific geographical influence for point-of-interest recommendation. In Lang, J., editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3877–3883. ijcai.org.

- Yang, C., Bai, L., Zhang, C., Yuan, Q., and Han, J. (2017). Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 1245–1254. ACM.

Yang, D., Zhang, D., and Qu, B. (2016).
Participatory cultural mapping based on collective behavior data in location-based social networks.
ACM TIST, 7(3):30:1–30:23.

Yang, D., Zhang, D., Yu, Z., and Wang, Z. (2013).
A sentiment-enhanced personalized location recommendation system.
In Stumme, G. and Hotho, A., editors, *24th ACM Conference on Hypertext and Social Media (part of ECRC), HT '13, Paris, France - May 02 - 04, 2013*, pages 119–128. ACM.

Ye, M., Yin, P., Lee, W., and Lee, D. L. (2011).
Exploiting geographical influence for collaborative point-of-interest recommendation.
In Ma, W., Nie, J., Baeza-Yates, R. A., Chua, T., and Croft, W. B., editors, *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 325–334. ACM.

Yin, H., Cui, B., Chen, L., Hu, Z., and Zhang, C. (2015).
Modeling location-based user rating profiles for personalized recommendation.
TKDD, 9(3):19:1–19:41.

Ying, H., Wu, J., Xu, G., Liu, Y., Liang, T., Zhang, X., and Xiong, H. (2019).
Time-aware metric embedding with asymmetric projection for successive POI recommendation.
World Wide Web, 22(5):2209–2224.

Ying, J. J., Kuo, W., Tseng, V. S., and Lu, E. H. (2014).
Mining user check-in behavior with a random walk for urban point-of-interest recommendations.
ACM TIST, 5(3):40:1–40:26.

Yuan, Q., Cong, G., Ma, Z., Sun, A., and Magnenat-Thalmann, N. (2013).
Time-aware point-of-interest recommendation.
In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *The 36th International ACM SIGIR conference on research and development in Information*

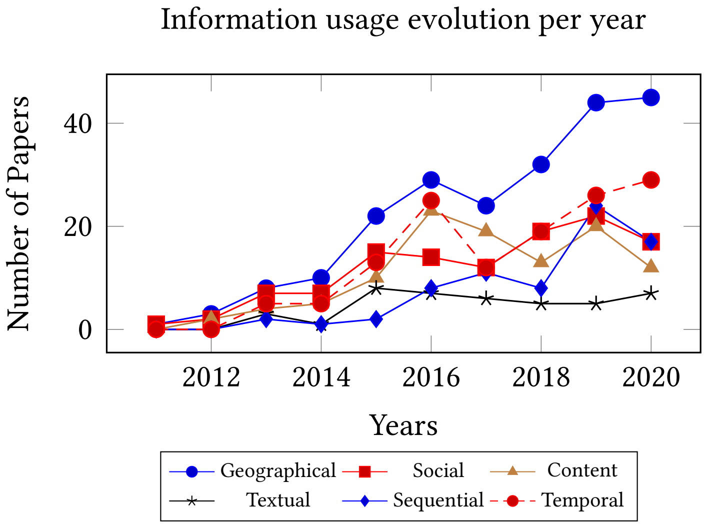
Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013, pages 363–372. ACM.

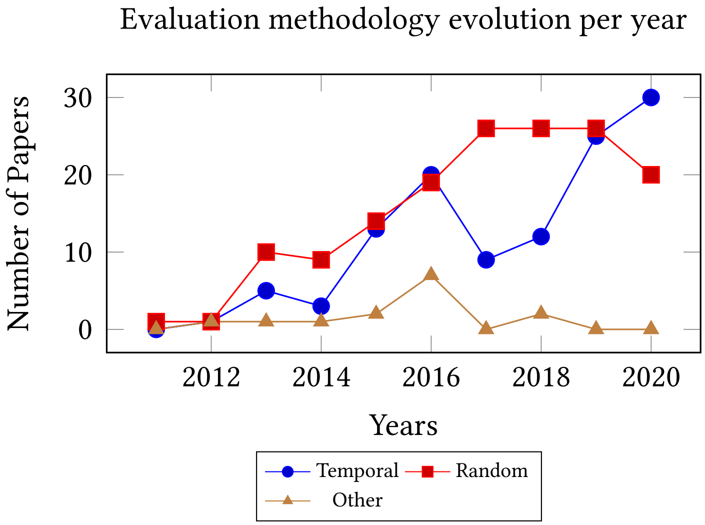
Yuan, Q., Cong, G., and Sun, A. (2014).
Graph-based point-of-interest recommendation with geographical and temporal influences.
In Li, J., Wang, X. S., Garofalakis, M. N., Soboroff, I., Suel, T., and Wang, M., editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 659–668. ACM.

Zhang, J. and Chow, C. (2015).
Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations.
In Baeza-Yates, R. A., Lalmas, M., Moffat, A., and Ribeiro-Neto, B. A., editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 443–452. ACM.

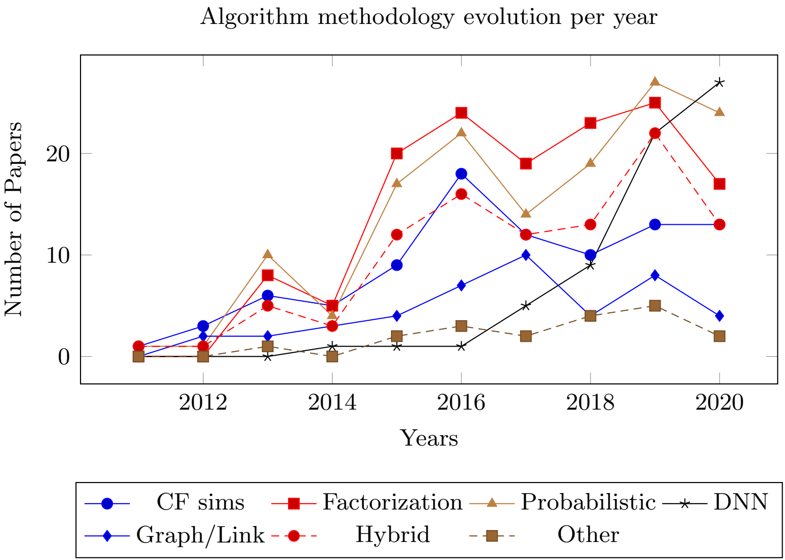
Zhao, S., Zhao, T., King, I., and Lyu, M. R. (2017).
Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation.
In Barrett, R., Cummings, R., Agichtein, E., and Gabrilovich, E., editors, *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 153–162. ACM.

Zhao, S., Zhao, T., Yang, H., Lyu, M. R., and King, I. (2016).
STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation.
In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 315–322. AAAI Press.

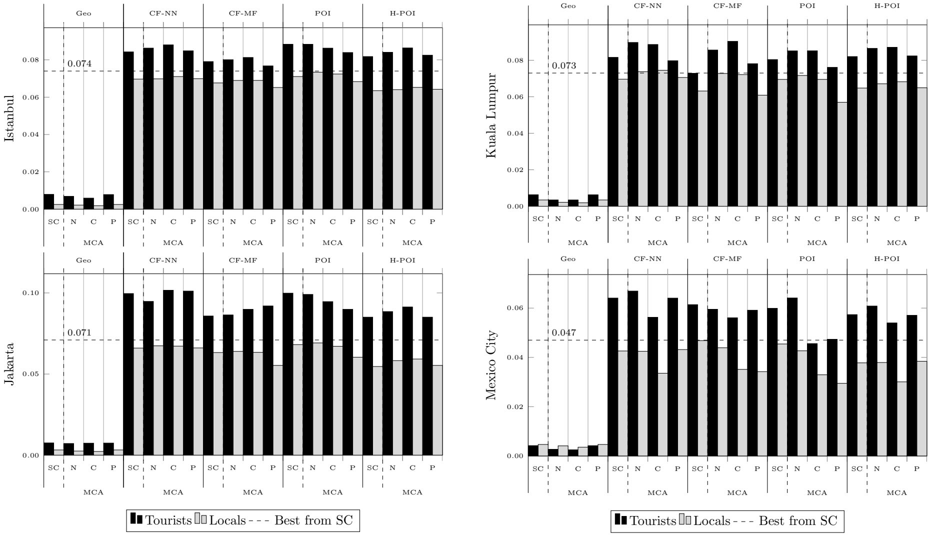

Information usage evolution per year
Number of Papers
0
20
40
2012
2014
2016
2018
2020
Years
Geographical
Social
Content
Textual
Sequential
Temporal


Evaluation methodology evolution per year
Number of Papers
30
20
10
0
2012
2014
2016
2018
2020
Years
Temporal
Random
Other

Point-Of-Interest recommendation: a survey



Experiments: POI recommendation. nDCG@5



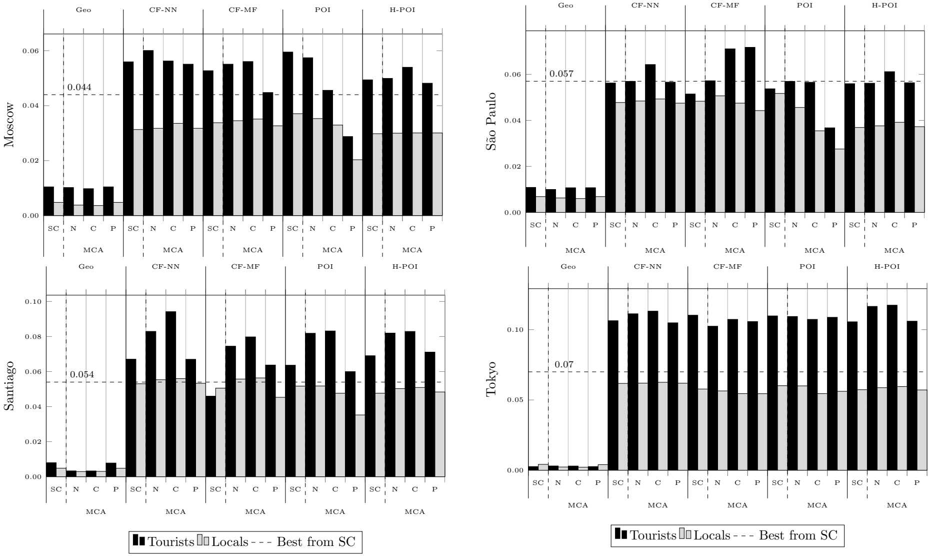

Moscow
Geo
CF-NN
CF-MF
POI
H-POI
0.044
SC N C P
MCA
São Paulo
0.057
Santiago
0.054
Tokyo
0.07
Tourists
Locals
Best from SC

Table: Performance in terms of nDCG@5 of the Popularity recommender in all cities in both Tourists and Locals.

City	All Users	Tourists	Locals	Δ Tourists (%)	Δ Locals (%)
Istanbul	0.054	0.064	0.048	19.04	−9.77
Jakarta	0.066	0.091	0.053	38.33	−19.92
Kuala Lumpur	0.066	0.077	0.060	17.34	−8.46
Mexico City	0.041	0.059	0.034	45.69	−15.70
Moscow	0.027	0.037	0.026	34.02	−4.48
Santiago	0.051	0.067	0.044	30.47	−13.21
São Paulo	0.053	0.061	0.031	14.85	−40.33
Tokyo	0.069	0.106	0.056	53.48	−18.73

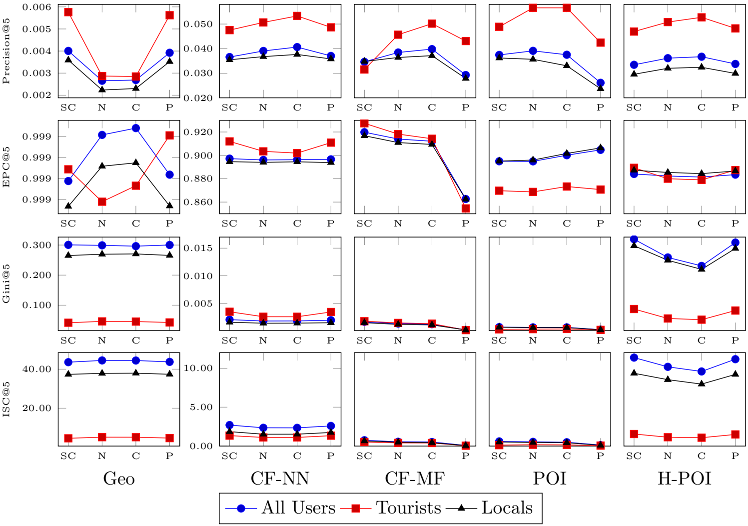


Figure: Results of tourists (7.62% of the users) and local (72.43% of the users) users in Santiago.

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Basic	Baseline	0.402	0.284	43.9	0.447	0.464	5.0	0.404	0.245	35.0
	f_{seq}^{rnd}	0.383	0.297	28.0	0.402	0.452	5.9	0.387	0.274	29.5
	f_{seq}^{dist}	0.396	**▲0.308**	**▲4.1**	0.469	**†0.474**	**▲1.4**	**†0.409**	**▲0.296**	**▲7.2**
	f_{seq}^{feat}	0.400	0.267	33.3	0.422	0.371	5.0	0.402	0.267	33.2
	f_{seq}^{item}	0.399	0.279	37.8	**†0.473**	0.469	1.8	0.408	0.262	19.5
	f_{seq}^{rec}	**†0.406**	0.298	42.4	0.422	0.452	6.0	0.407	0.271	26.3
	f_{seq}^{lcs}	0.395	0.285	17.8	0.440	0.446	2.3	0.403	0.274	14.8
	f_{seq}^{stree}	0.402	0.289	38.4	0.446	0.466	3.2	0.403	0.263	25.9
	f_{seq}^{oracle}	**▲0.468**	0.296	43.2	**▲0.614**	**▲0.482**	4.2	**▲0.456**	0.247	34.2
Classic	Baseline	0.404	0.285	45.3	0.447	0.460	6.3	0.408	0.270	30.0
	f_{seq}^{rnd}	0.382	0.292	30.3	0.403	0.450	5.9	0.394	0.278	30.7
	f_{seq}^{dist}	0.395	**0.309**	**▲4.2**	0.468	**†0.475**	**▲1.4**	**†0.410**	**▲0.294**	**▲7.4**
	f_{seq}^{feat}	0.398	0.267	33.5	0.424	0.373	5.0	0.402	0.269	34.0
	f_{seq}^{item}	0.400	0.276	38.0	**†0.476**	0.468	1.8	0.409	0.268	18.5
	f_{seq}^{rec}	**†0.406**	0.300	42.4	0.422	0.452	6.0	0.407	0.273	26.5
	f_{seq}^{lcs}	0.395	0.284	17.9	0.440	0.447	2.3	0.405	0.279	13.2
	f_{seq}^{stree}	0.404	0.294	38.6	0.447	0.465	3.7	0.405	0.275	22.1
	f_{seq}^{oracle}	**▲0.468**	0.300	44.3	**▲0.612**	**▲0.482**	4.9	**▲0.455**	0.269	29.0
Temporal	Baseline	**†0.404**	0.302	42.4	0.447	**†0.469**	4.9	**†0.416**	0.285	26.6
	f_{seq}^{rnd}	0.379	0.317	25.3	0.409	0.449	6.0	0.383	0.308	28.1
	f_{seq}^{dist}	0.389	**▲0.319**	**▲3.5**	0.464	0.468	**▲1.4**	0.412	**▲0.326**	**▲5.6**
	f_{seq}^{feat}	0.388	0.272	30.6	0.421	0.375	5.0	0.397	0.291	30.2
	f_{seq}^{item}	0.400	0.293	37.2	**†0.474**	0.465	1.9	0.412	0.283	17.5
	f_{seq}^{rec}	0.403	0.309	41.5	0.422	0.452	6.1	0.407	0.292	26.1
	f_{seq}^{lcs}	0.388	0.314	10.8	0.441	0.447	2.3	0.407	0.311	10.9
	f_{seq}^{stree}	0.398	0.311	29.6	0.445	0.468	3.1	0.411	0.301	17.3
	f_{seq}^{oracle}	**▲0.462**	0.308	39.9	**▲0.608**	**▲0.482**	4.1	**▲0.457**	0.287	25.8

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Basic	Baseline	0.402	0.284	43.9	0.447	0.464	5.0	0.404	0.245	35.0
	f_{seq}^{rnd}	0.383	0.297	28.0	0.402	0.452	5.9	0.387	0.274	29.5
	f_{seq}^{dist}	0.396	▲**0.308**	▲**4.1**	0.469	†**0.474**	▲**1.4**	†**0.409**	▲**0.296**	▲**7.2**
	f_{seq}^{feat}	0.400	0.267	33.3	0.422	0.371	5.0	0.402	0.267	33.2
	f_{seq}^{item}	0.399	0.279	37.8	†**0.473**	0.469	1.8	0.408	0.262	19.5
	f_{seq}^{rec}	†**0.406**	0.298	42.4	0.422	0.452	6.0	0.407	0.271	26.3
	f_{seq}^{lcs}	0.395	0.285	17.8	0.440	0.446	2.3	0.403	0.274	14.8
	f_{seq}^{stree}	0.402	0.289	38.4	0.446	0.466	3.2	0.403	0.263	25.9
	f_{seq}^{oracle}	▲**0.468**	0.296	43.2	▲**0.614**	▲**0.482**	4.2	▲**0.456**	0.247	34.2
Classic	Baseline	0.404	0.285	45.3	0.447	0.460	6.3	0.408	0.270	30.0
	f_{seq}^{rnd}	0.382	0.292	30.3	0.403	0.450	5.9	0.394	0.278	30.7
	f_{seq}^{dist}	0.395	**0.309**	▲**4.2**	0.468	†**0.475**	▲**1.4**	†**0.410**	▲**0.294**	▲**7.4**
	f_{seq}^{feat}	0.398	0.267	33.5	0.424	0.373	5.0	0.402	0.269	34.0
	f_{seq}^{item}	0.400	0.276	38.0	†**0.476**	0.468	1.8	0.409	0.268	18.5
	f_{seq}^{rec}	†**0.406**	0.300	42.4	0.422	0.452	6.0	0.407	0.273	26.5
	f_{seq}^{lcs}	0.395	0.284	17.9	0.440	0.447	2.3	0.405	0.279	13.2
	f_{seq}^{stree}	0.404	0.294	38.6	0.447	0.465	3.7	0.405	0.275	22.1
	f_{seq}^{oracle}	▲**0.468**	0.300	44.3	▲**0.612**	▲**0.482**	4.9	▲**0.455**	0.269	29.0
Temporal	Baseline	†**0.404**	0.302	42.4	0.447	†**0.469**	4.9	†**0.416**	0.285	26.6
	f_{seq}^{rnd}	0.379	0.317	25.3	0.409	0.449	6.0	0.383	0.308	28.1
	f_{seq}^{dist}	0.389	▲**0.319**	▲**3.5**	0.464	0.468	▲**1.4**	0.412	▲**0.326**	▲**5.6**
	f_{seq}^{feat}	0.388	0.272	30.6	0.421	0.375	5.0	0.397	0.291	30.2
	f_{seq}^{item}	0.400	0.293	37.2	†**0.474**	0.465	1.9	0.412	0.283	17.5
	f_{seq}^{rec}	0.403	0.309	41.5	0.422	0.452	6.1	0.407	0.292	26.1
	f_{seq}^{lcs}	0.388	0.314	10.8	0.441	0.447	2.3	0.407	0.311	10.9
	f_{seq}^{stree}	0.398	0.311	29.6	0.445	0.468	3.1	0.411	0.301	17.3
	f_{seq}^{oracle}	▲**0.462**	0.308	39.9	▲**0.608**	▲**0.482**	4.1	▲**0.457**	0.287	25.8

Experiments: Sequences in POI recommendation

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Basic	Baseline	0.402	0.284	43.9	0.447	0.464	5.0	0.404	0.245	35.0
	f^{rnd}_{seq}	0.383	0.297	28.0	0.402	0.452	5.9	0.387	0.274	29.5
	f^{dist}_{seq}	0.396	**▲0.308**	**▲4.1**	0.469	**†0.474**	**▲1.4**	**†0.409**	**▲0.296**	**▲7.2**
	f^{feat}_{seq}	0.400	0.267	33.3	0.422	0.371	5.0	0.402	0.267	33.2
	f^{item}_{seq}	0.399	0.279	37.8	**†0.473**	0.469	1.8	0.408	0.262	19.5
	f^{rec}_{seq}	**†0.406**	0.298	42.4	0.422	0.452	6.0	0.407	0.271	26.3
	f^{lcs}_{seq}	0.395	0.285	17.8	0.440	0.446	2.3	0.403	0.274	14.8
	f^{stree}_{seq}	0.402	0.289	38.4	0.446	0.466	3.2	0.403	0.263	25.9
	f^{oracle}_{seq}	**▲0.468**	0.296	43.2	**▲0.614**	**▲0.482**	4.2	**▲0.456**	0.247	34.2
Classic	Baseline	0.404	0.285	45.3	0.447	0.460	6.3	0.408	0.270	30.0
	f^{rnd}_{seq}	0.382	0.292	30.3	0.403	0.450	5.9	0.394	0.278	30.7
	f^{dist}_{seq}	0.395	**0.309**	**▲4.2**	0.468	**†0.475**	**▲1.4**	**†0.410**	**▲0.294**	**▲7.4**
	f^{feat}_{seq}	0.398	0.267	33.5	0.424	0.373	5.0	0.402	0.269	34.0
	f^{item}_{seq}	0.400	0.276	38.0	**†0.476**	0.468	1.8	0.409	0.268	18.5
	f^{rec}_{seq}	**†0.406**	0.300	42.4	0.422	0.452	6.0	0.407	0.273	26.5
	f^{lcs}_{seq}	0.395	0.284	17.9	0.440	0.447	2.3	0.405	0.279	13.2
	f^{stree}_{seq}	0.404	0.294	38.6	0.447	0.465	3.7	0.405	0.275	22.1
	f^{oracle}_{seq}	**▲0.468**	0.300	44.3	**▲0.612**	**▲0.482**	4.9	**▲0.455**	0.269	29.0
Temporal	Baseline	**†0.404**	0.302	42.4	0.447	**†0.469**	4.9	**†0.416**	0.285	26.6
	f^{rnd}_{seq}	0.379	0.317	25.3	0.409	0.449	6.0	0.383	0.308	28.1
	f^{dist}_{seq}	0.389	**▲0.319**	**▲3.5**	0.464	0.468	**▲1.4**	0.412	**▲0.326**	**▲5.6**
	f^{feat}_{seq}	0.388	0.272	30.6	0.421	0.375	5.0	0.397	0.291	30.2
	f^{item}_{seq}	0.400	0.293	37.2	**†0.474**	0.465	1.9	0.412	0.283	17.5
	f^{rec}_{seq}	0.403	0.309	41.5	0.422	0.452	6.1	0.407	0.292	26.1
	f^{lcs}_{seq}	0.388	0.314	10.8	0.441	0.447	2.3	0.407	0.311	10.9
	f^{stree}_{seq}	0.398	0.311	29.6	0.445	0.468	3.1	0.411	0.301	17.3
	f^{oracle}_{seq}	**▲0.462**	0.308	39.9	**▲0.608**	**▲0.482**	4.1	**▲0.457**	0.287	25.8

Experiments: Sequences in POI recommendation

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Geo	Baseline	0.405	0.306	43.9	0.427	0.457	5.6	0.406	0.286	30.0
	f_{seq}^{rnd}	0.378	0.307	22.5	0.397	0.447	5.9	0.390	0.307	25.1
	f_{seq}^{dist}	0.385	0.315	▲**3.6**	0.456	†**0.468**	▲**1.4**	0.405	▲**0.315**	▲**5.8**
	f_{seq}^{feat}	0.393	0.281	32.8	0.414	0.364	5.3	0.397	0.282	26.8
	f_{seq}^{item}	0.402	0.291	37.1	†**0.467**	0.466	2.1	†**0.412**	0.270	18.8
	f_{seq}^{rec}	†**0.405**	0.311	42.0	0.417	0.453	6.0	0.407	0.290	25.8
	f_{seq}^{lcs}	0.390	0.311	11.9	0.426	0.440	2.2	0.401	0.308	10.5
	f_{seq}^{stree}	0.402	**0.321**	31.3	0.431	0.458	3.5	0.404	0.302	19.4
	f_{seq}^{oracle}	▲**0.464**	0.314	41.7	▲**0.586**	▲**0.472**	4.6	▲**0.449**	0.287	28.8
Tour	Baseline	0.391	0.279	44.9	†**0.477**	0.473	2.0	0.403	0.240	28.4
	f_{seq}^{rnd}	0.364	0.305	23.9	0.400	0.448	5.7	0.390	0.291	30.8
	f_{seq}^{dist}	0.381	0.311	▲**4.2**	0.467	†**0.474**	▲**1.4**	†**0.412**	▲**0.309**	▲**7.1**
	f_{seq}^{feat}	0.374	0.277	20.9	0.420	0.359	5.0	0.401	0.278	31.6
	f_{seq}^{item}	0.397	0.283	38.1	0.477	0.470	1.8	0.406	0.271	16.9
	f_{seq}^{rec}	†**0.403**	0.289	41.4	0.427	0.451	5.8	0.408	0.273	26.6
	f_{seq}^{lcs}	0.382	▲**0.312**	12.0	0.438	0.446	2.1	0.406	0.290	13.9
	f_{seq}^{stree}	0.386	0.295	32.4	0.457	0.466	2.4	0.403	0.272	21.5
	f_{seq}^{oracle}	▲**0.442**	0.285	44.4	▲**0.600**	▲**0.482**	3.0	▲**0.455**	0.244	28.0

Experiments: Sequences in POI recommendation

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Geo	Baseline	0.405	0.306	43.9	0.427	0.457	5.6	0.406	0.286	30.0
	f_{seq}^{rnd}	0.378	0.307	22.5	0.397	0.447	5.9	0.390	0.307	25.1
	f_{seq}^{dist}	0.385	0.315	**▲3.6**	0.456	**†0.468**	**▲1.4**	0.405	**▲0.315**	**▲5.8**
	f_{seq}^{feat}	0.393	0.281	32.8	0.414	0.364	5.3	0.397	0.282	26.8
	f_{seq}^{item}	0.402	0.291	37.1	**†0.467**	0.466	2.1	**†0.412**	0.270	18.8
	f_{seq}^{rec}	**†0.405**	0.311	42.0	0.417	0.453	6.0	0.407	0.290	25.8
	f_{seq}^{lcs}	0.390	0.311	11.9	0.426	0.440	2.2	0.401	0.308	10.5
	f_{seq}^{stree}	0.402	**0.321**	31.3	0.431	0.458	3.5	0.404	0.302	19.4
	f_{seq}^{oracle}	**▲0.464**	0.314	41.7	**▲0.586**	**▲0.472**	4.6	**▲0.449**	0.287	28.8
Tour	Baseline	0.391	0.279	44.9	**†0.477**	0.473	2.0	0.403	0.240	28.4
	f_{seq}^{rnd}	0.364	0.305	23.9	0.400	0.448	5.7	0.390	0.291	30.8
	f_{seq}^{dist}	0.381	0.311	**▲4.2**	0.467	**†0.474**	**▲1.4**	**†0.412**	**▲0.309**	**▲7.1**
	f_{seq}^{feat}	0.374	0.277	20.9	0.420	0.359	5.0	0.401	0.278	31.6
	f_{seq}^{item}	0.397	0.283	38.1	0.477	0.470	1.8	0.406	0.271	16.9
	f_{seq}^{rec}	**†0.403**	0.289	41.4	0.427	0.451	5.8	0.408	0.273	26.6
	f_{seq}^{lcs}	0.382	**▲0.312**	12.0	0.438	0.446	2.1	0.406	0.290	13.9
	f_{seq}^{stree}	0.386	0.295	32.4	0.457	0.466	2.4	0.403	0.272	21.5
	f_{seq}^{oracle}	**▲0.442**	0.285	44.4	**▲0.600**	**▲0.482**	3.0	**▲0.455**	0.244	28.0

Family	Reranker	New York nDCG$_s$	New York FP$_s$	New York Dist	Rome nDCG$_s$	Rome FP$_s$	Rome Dist	Petaling Jaya nDCG$_s$	Petaling Jaya FP$_s$	Petaling Jaya Dist
Geo	Baseline	0.405	0.306	43.9	0.427	0.457	5.6	0.406	0.286	30.0
	f_{seq}^{rnd}	0.378	0.307	22.5	0.397	0.447	5.9	0.390	0.307	25.1
	f_{seq}^{dist}	0.385	0.315	▲**3.6**	0.456	†**0.468**	▲**1.4**	0.405	▲**0.315**	▲**5.8**
	f_{seq}^{feat}	0.393	0.281	32.8	0.414	0.364	5.3	0.397	0.282	26.8
	f_{seq}^{item}	0.402	0.291	37.1	†**0.467**	0.466	2.1	†**0.412**	0.270	18.8
	f_{seq}^{rec}	†**0.405**	0.311	42.0	0.417	0.453	6.0	0.407	0.290	25.8
	f_{seq}^{lcs}	0.390	0.311	11.9	0.426	0.440	2.2	0.401	0.308	10.5
	f_{seq}^{stree}	0.402	**0.321**	31.3	0.431	0.458	3.5	0.404	0.302	19.4
	f_{seq}^{oracle}	▲**0.464**	0.314	41.7	▲**0.586**	▲**0.472**	4.6	▲**0.449**	0.287	28.8
Tour	Baseline	0.391	0.279	44.9	†**0.477**	0.473	2.0	0.403	0.240	28.4
	f_{seq}^{rnd}	0.364	0.305	23.9	0.400	0.448	5.7	0.390	0.291	30.8
	f_{seq}^{dist}	0.381	0.311	▲**4.2**	0.467	†**0.474**	▲**1.4**	†**0.412**	▲**0.309**	▲**7.1**
	f_{seq}^{feat}	0.374	0.277	20.9	0.420	0.359	5.0	0.401	0.278	31.6
	f_{seq}^{item}	0.397	0.283	38.1	0.477	0.470	1.8	0.406	0.271	16.9
	f_{seq}^{rec}	†**0.403**	0.289	41.4	0.427	0.451	5.8	0.408	0.273	26.6
	f_{seq}^{lcs}	0.382	▲**0.312**	12.0	0.438	0.446	2.1	0.406	0.290	13.9
	f_{seq}^{stree}	0.386	0.295	32.4	0.457	0.466	2.4	0.403	0.272	21.5
	f_{seq}^{oracle}	▲**0.442**	0.285	44.4	▲**0.600**	▲**0.482**	3.0	▲**0.455**	0.244	28.0