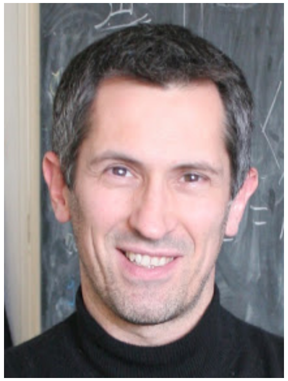
Sequence-Aware Recommenders

Massimo Quadrana
Pandora

Paolo Cremonesi
Politecnico di Milano

About today's presenter(s)

- Paolo Cremonesi
  - Politecnico di Milano, Italy



- Massimo Quadrana
  - Pandora, Italy

Today’s tutorial based on

M. Quadrana, P. Cremonesi, D. Jannach,
"*Sequence-Aware Recommender Systems*“
ACM Computing Surveys, 2018

bit.ly/sequence-aware-rs

About you?

Agenda

- 14:00 – 14:45 Introduction & Problem Definition (Paolo)
- 14:45 – 15:15 Evaluation (Paolo)
- 15:15 – 15:30 Algorithms I (Massimo)
- 15:30 – 16:00 Coffee break
- 16:00 – 16:45 Algorithms II (Massimo)
- 16:45 – 17:20 Hands-on (Massimo)
- 17:20 – 17:30 Conclusion / Questions

Introduction
&
Problem Definition

Common problem abstraction (1): matrix completion

- Goal
  - Learn missing ratings
- Quality assessment
  - Error between true and estimated ratings

	Item1	Item2	Item3	Item4	Item5
Alice	5	?	4	4	?
Paolo	3	1	?	3	3
Susan	?	3	?	?	5
Bob	3	?	1	5	4
George	?	5	5	?	1

Common problem abstraction (2): learning to rank

- Goal
  - Learn relative ranking of items
- Quality assessment
  - Error between true and estimated ranking

Real-world problem situations

- User intent
- Short-term intent/context vs. long term taste
- Order can matter
- Order matters (constraints)
- Interest drift
- Reminders
- Repeated purchases

User intent

- Our user searched and listened to “Last Christmas” by Wham!
- Should we, …
  - Play more songs by Wham!?
  - More pop Christmas songs?
  - More popular songs from the 1980s?
  - Play more songs with controversial user feedback?

- Knowing the user’s intention can be crucial

Short-term intent/context vs. long term taste

- Here's what the customer purchased during the last weeks

- Now, the user return to the shop and browse these items

Short-term intent/context vs. long term taste

- What to recommend?
- Some plausible options
  - **Only shoes**
  - Mostly **Nike shoes**
  - Maybe also some T-shirts

products purchased
in the past

products browsed
in the current session

Short-term intent/context vs. long term taste

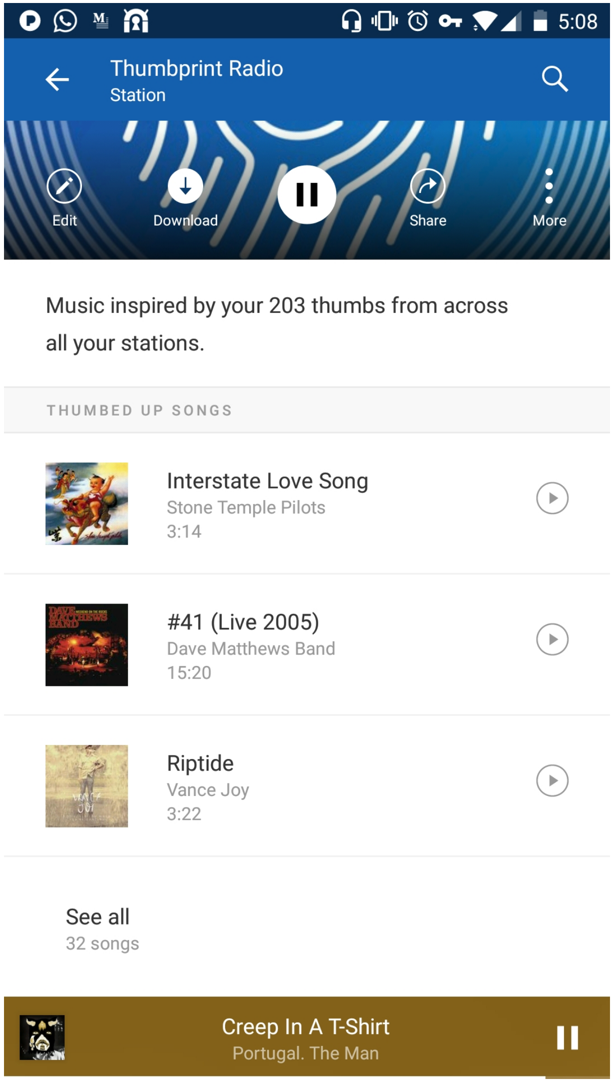
- Using the matrix completion formulation
  - One trains a model based only on past actions
  - Without the context, the algorithm will probably most recommend
  - Mostly (Nike) **T-shirts and some trousers**
  - Is this what you expect?

products purchased
in the past

products browsed
in the current session

Order can matter

- Next track recommendation
- What to recommend next should suit the previous tracks



Order matters (order constraints)

- Is it meaningful to recommend Star Wars III, if the user has not seen the previous episodes?

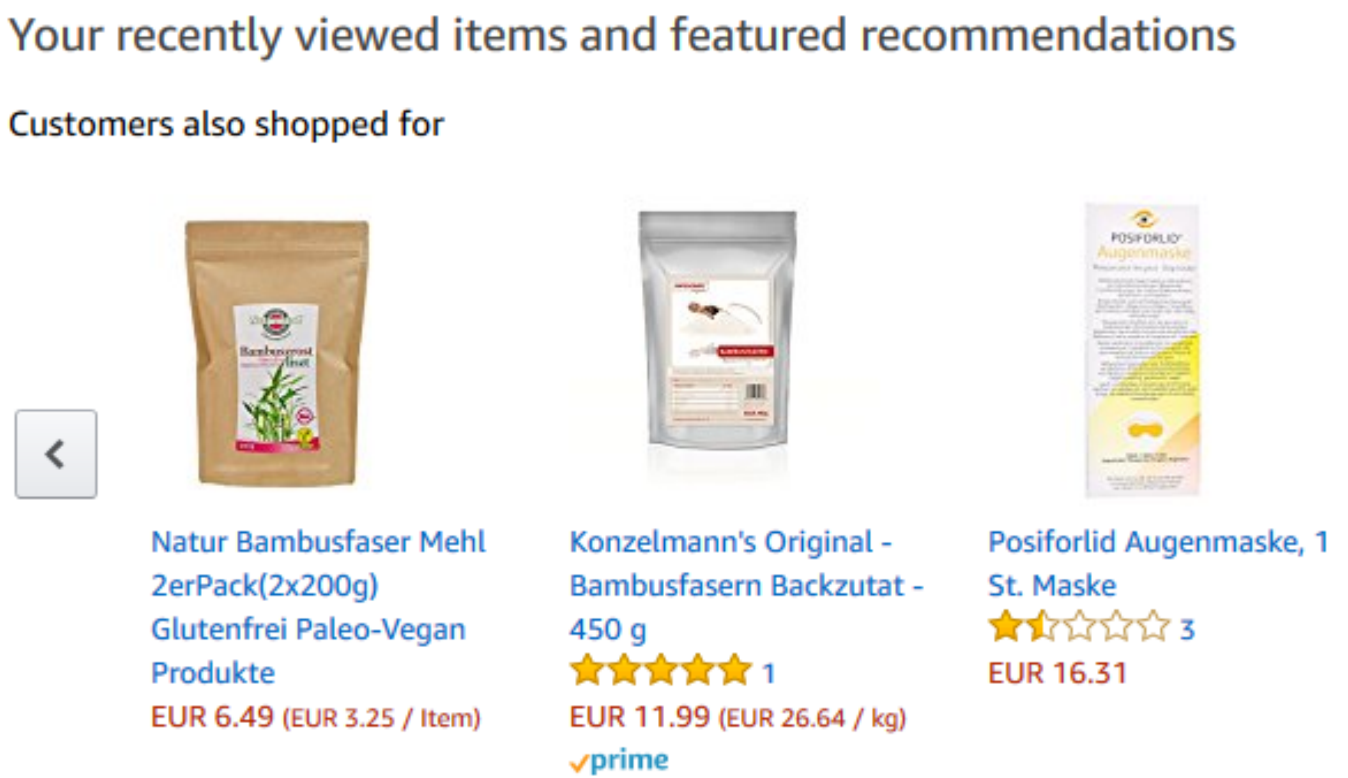
Interest drift

Before having a family

After having a family

Reminders

- Should we recommend items that the user already knows?
- Amazon does



Repeated purchases

- When should we remind users (through recommendations?)

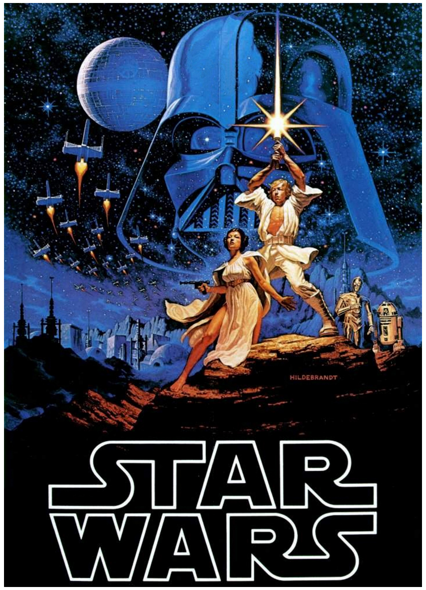
Algorithms depends on …

The choice of the **best** recommender algorithm depends on

- Goal (user task, application domain, context, …)
- Data available
- Target quality

Examples based on goal …

- Movie recommendation
    - if you watched a SF movie …



Examples based on goal …

- Movie recommendation
  - if you watched a SF movie …
  - … you recommend another SF movie
- **Algorithm: most similar item**

Examples based on goal …

- On-line store recommendations
  - if you bought a coffee maker …

Examples based on goal …

- On-line store recommendations
  - if you bought a coffee maker …
  - … you recommend cups or coffee beans (not another coffee maker)
- **Algorithm: frequently bought together**
  user who bought … also bought

Example based on data …

- If you have
  - all items with many ratings
  - no item attributes
- **You need CF**

- If you have
  - no ratings
  - all item with many attributes
- **You need CBF**

Examples for sequence-aware algorithms …

Goal

- next track recommendation, …

Data

- new users, …

Need for quality

- context, intent, …

Implications for research

Many of the scenarios cannot be addressed by a matrix completion or learning-to-rank problem formulation

- No short-term context
- Only one single user-item interaction pair
- Often only one type of interaction (e.g., ratings)
- Rating timestamps sometimes exist, but might be disconnected from actually experiencing/using the item

Sequence-Aware Recommenders

A family of recommenders that

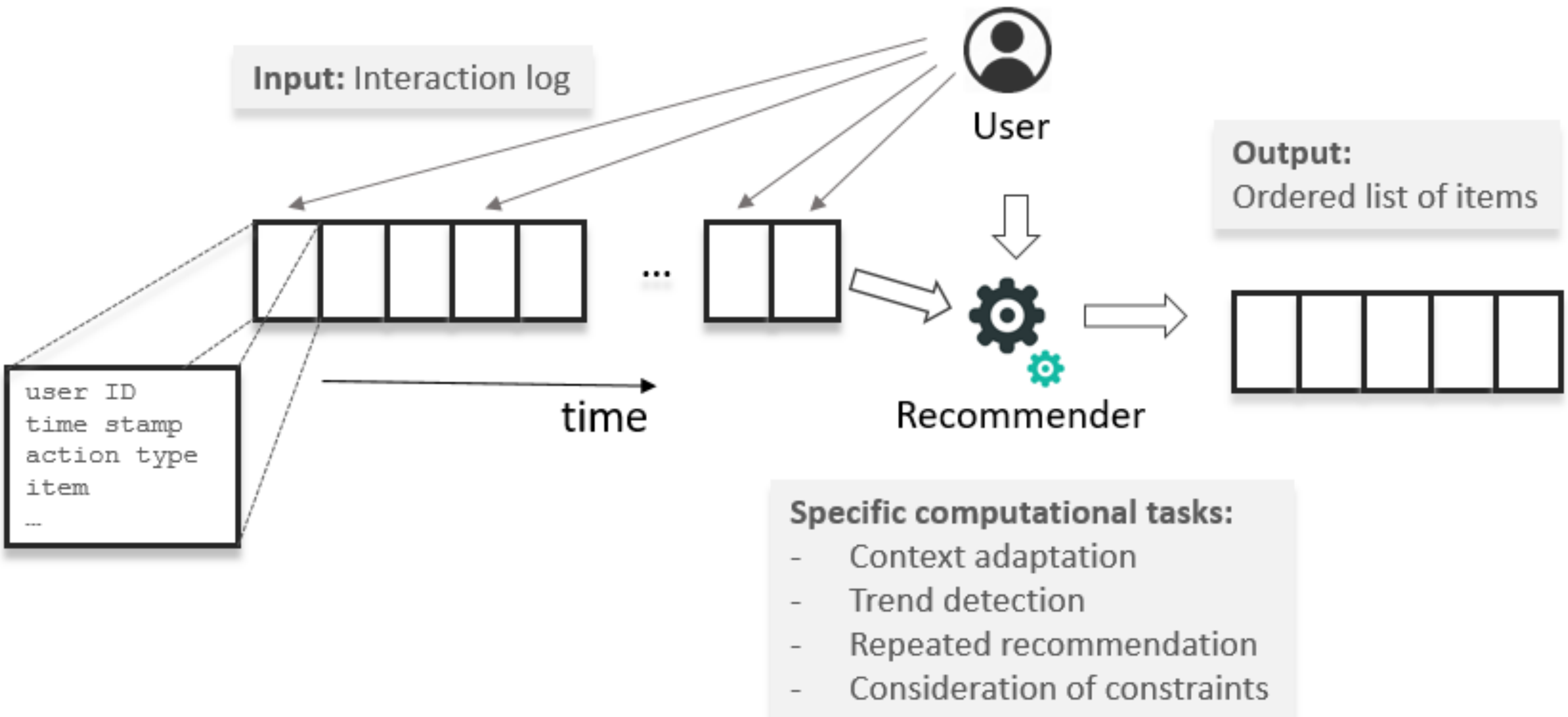
- uses different input data

- often bases the recommendations on certain types of sequential patterns in the data

- addresses the mentioned practical problem settings

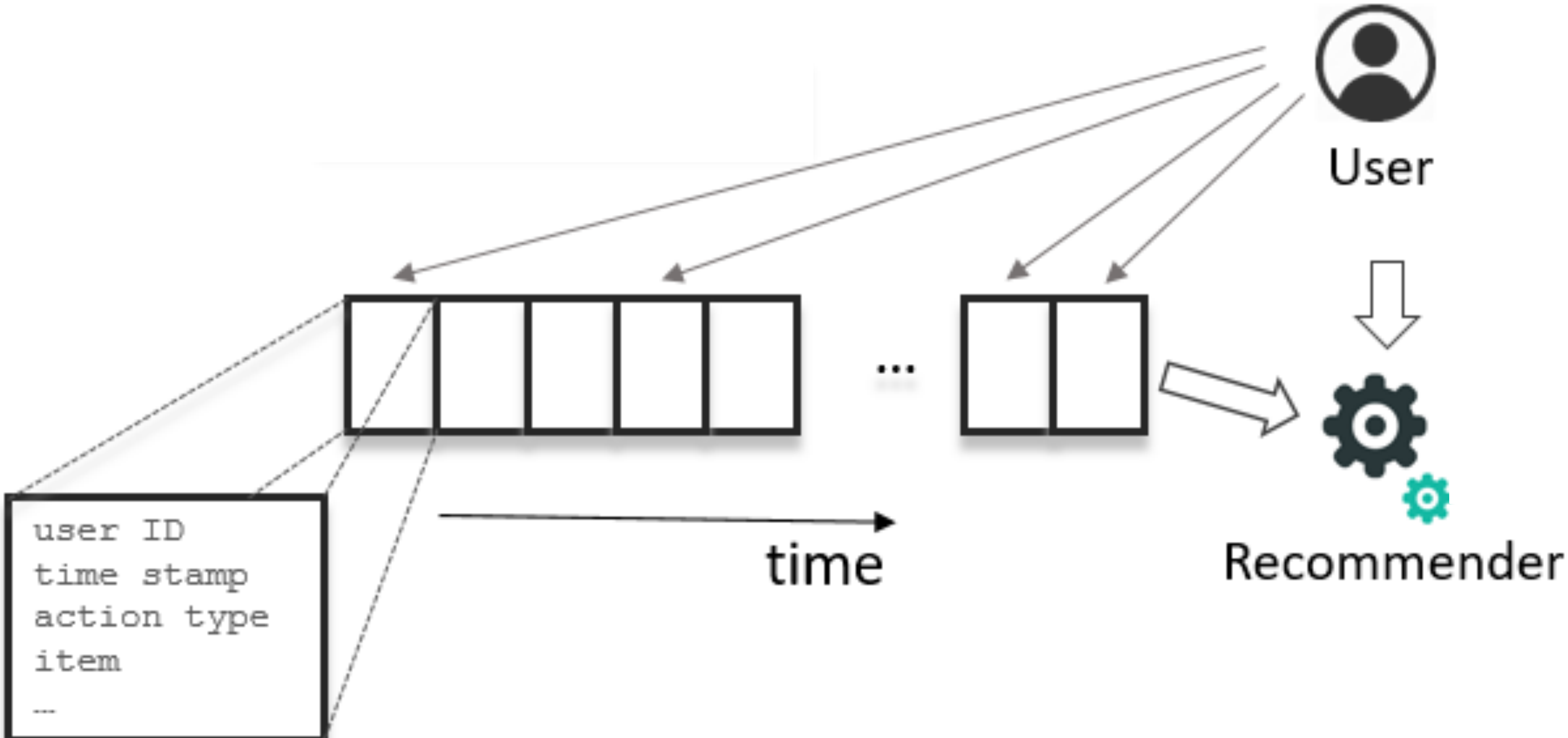
Problem Characterization

Characterizing Sequence-Aware Recommender Systems



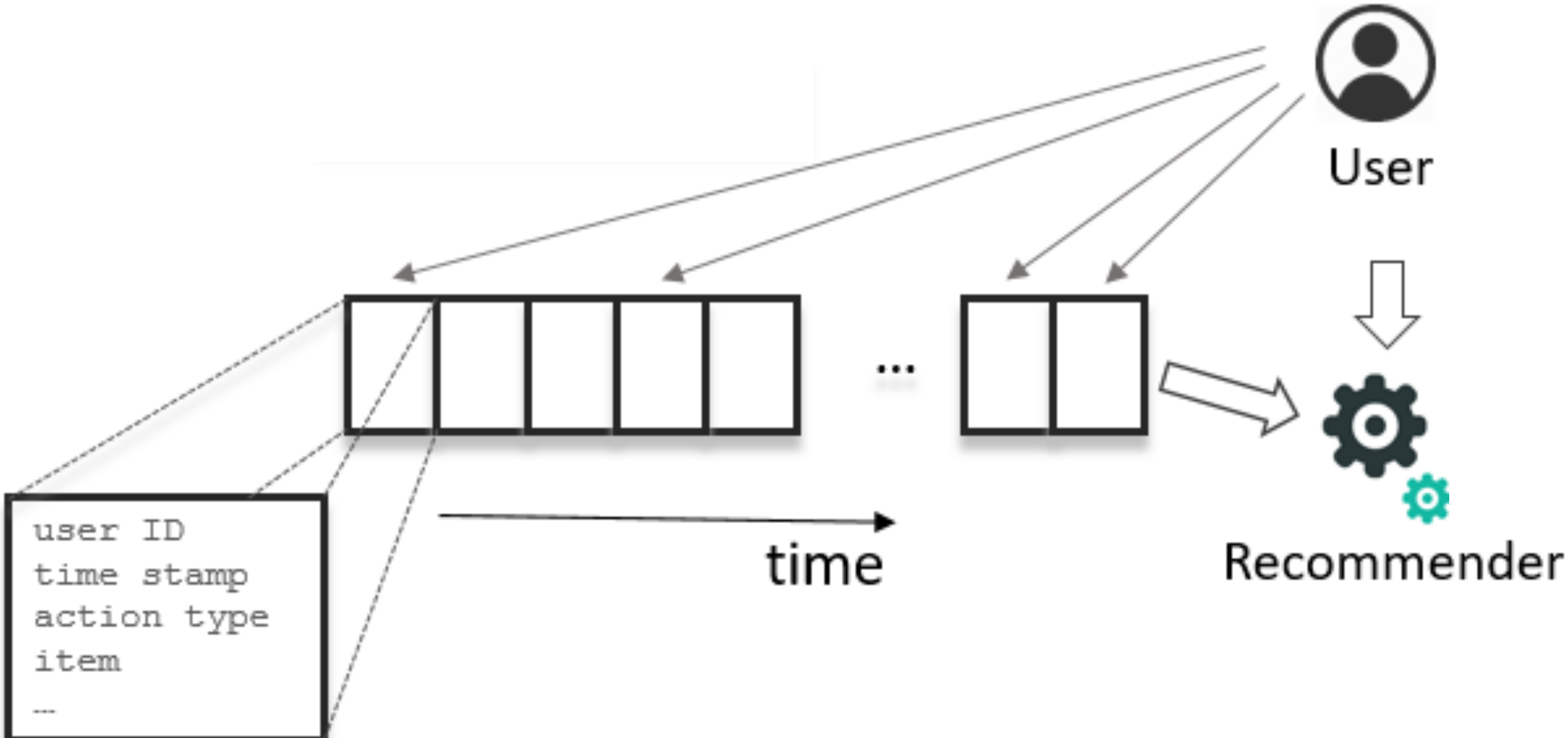
Inputs

- Ordered (and time-stamped) set of user actions
  - Known or anonymous (beyond the session)

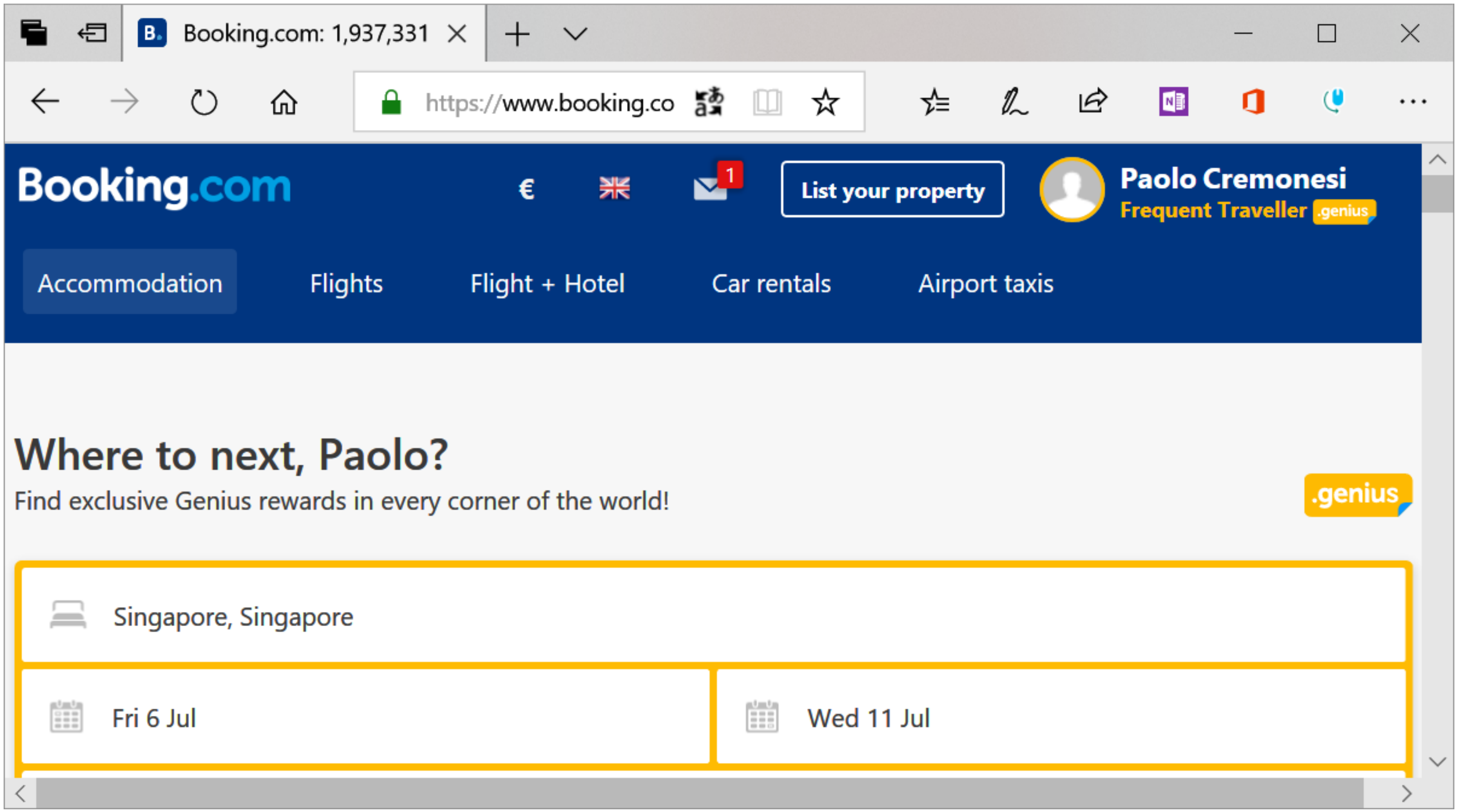


Inputs

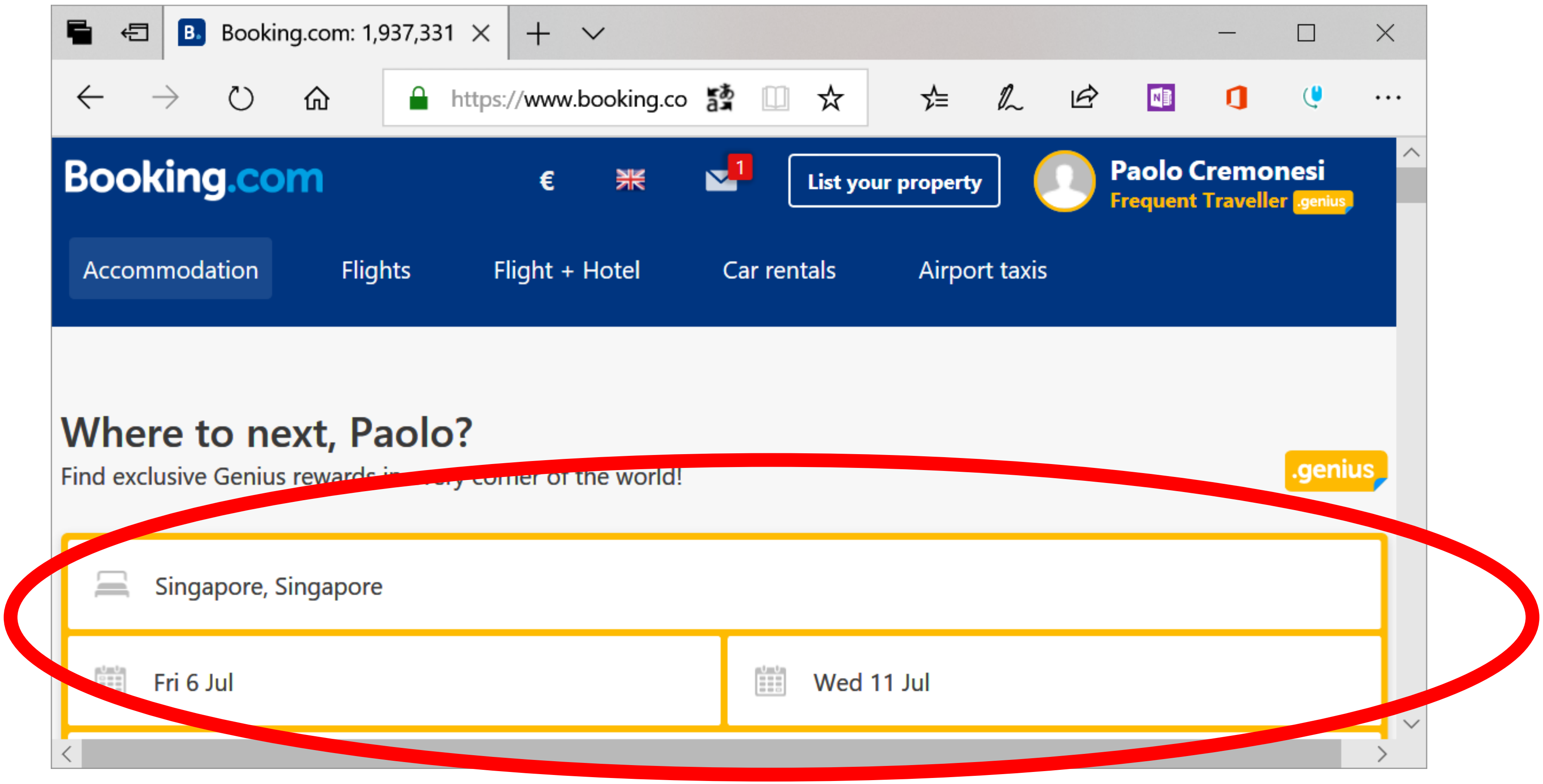
- Actions are usually connected with items
  - Exceptions: search terms, category navigation, …



Inputs

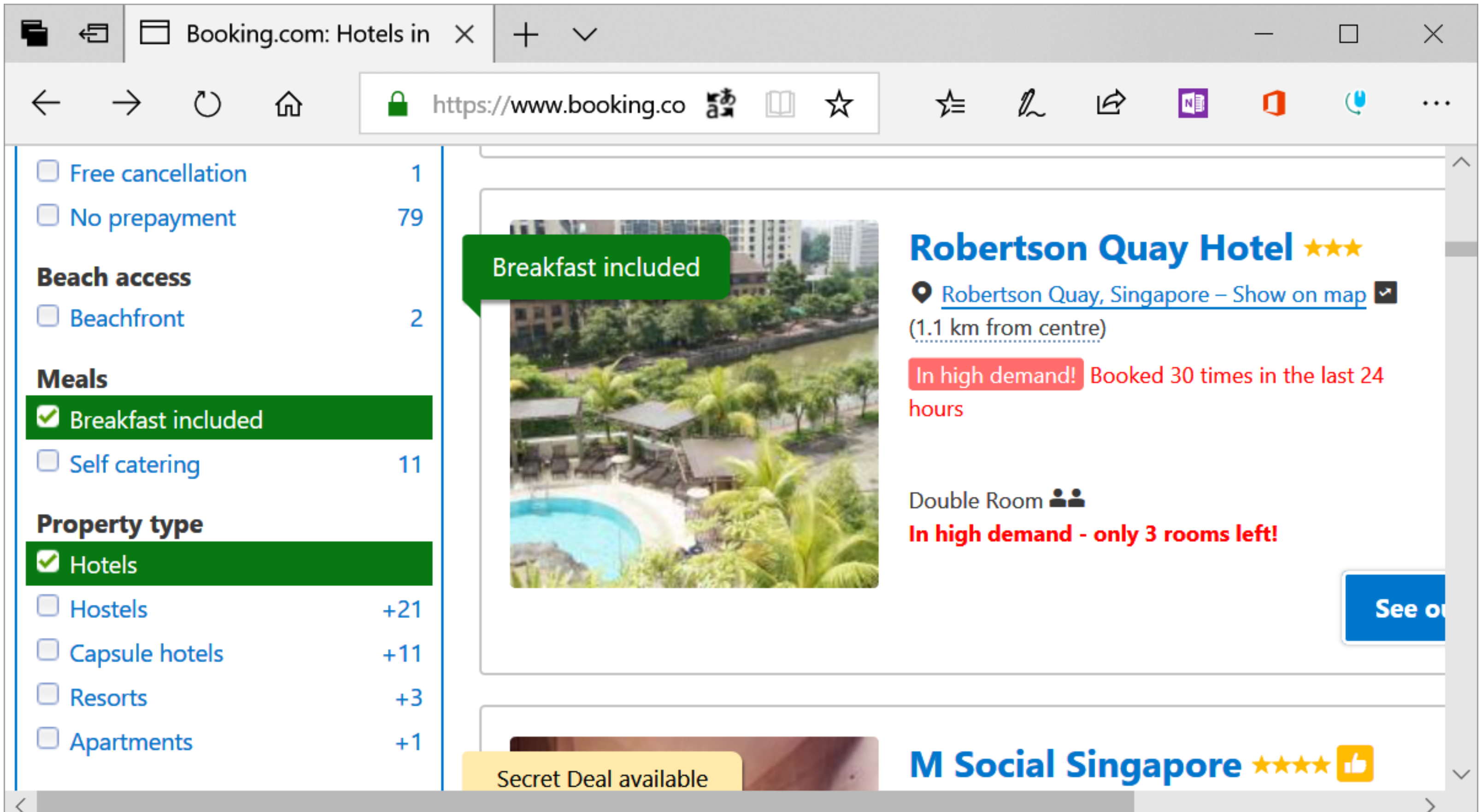

Booking.com: 1,937,331
https://www.booking.co
Booking.com
€
1
List your property
Paolo Cremonesi
Frequent Traveller .genius
Accommodation
Flights
Flight + Hotel
Car rentals
Airport taxis
Where to next, Paolo?
Find exclusive Genius rewards in every corner of the world!
.genius
Singapore, Singapore
Fri 6 Jul
Wed 11 Jul

Inputs



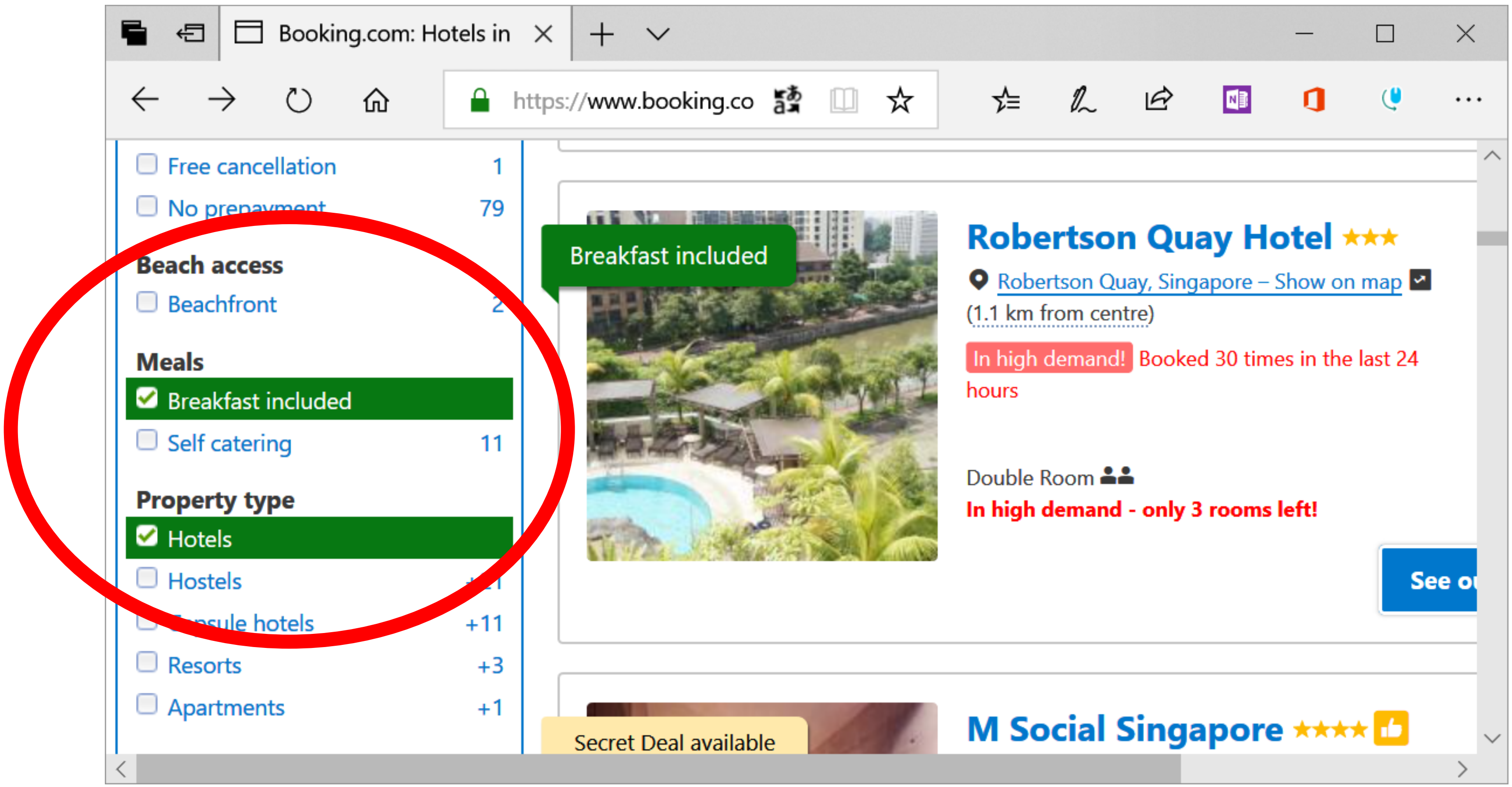
Booking.com: 1,937,331
https://www.booking.co
Booking.com
€
1
List your property
Paolo Cremonesi
Frequent Traveller
.genius
Accommodation
Flights
Flight + Hotel
Car rentals
Airport taxis
Where to next, Paolo?
.genius
Singapore, Singapore
Fri 6 Jul
Wed 11 Jul

Inputs



Booking.com: Hotels in
https://www.booking.co
Free cancellation 1
No prepayment 79
Beach access
Beachfront 2
Meals
Breakfast included
Self catering 11
Property type
Hotels
Hostels +21
Capsule hotels +11
Resorts +3
Apartments +1
Breakfast included
Robertson Quay Hotel
Robertson Quay, Singapore – Show on map
(1.1 km from centre)
In high demand! Booked 30 times in the last 24 hours
Double Room
In high demand - only 3 rooms left!
See o
Secret Deal available
M Social Singapore

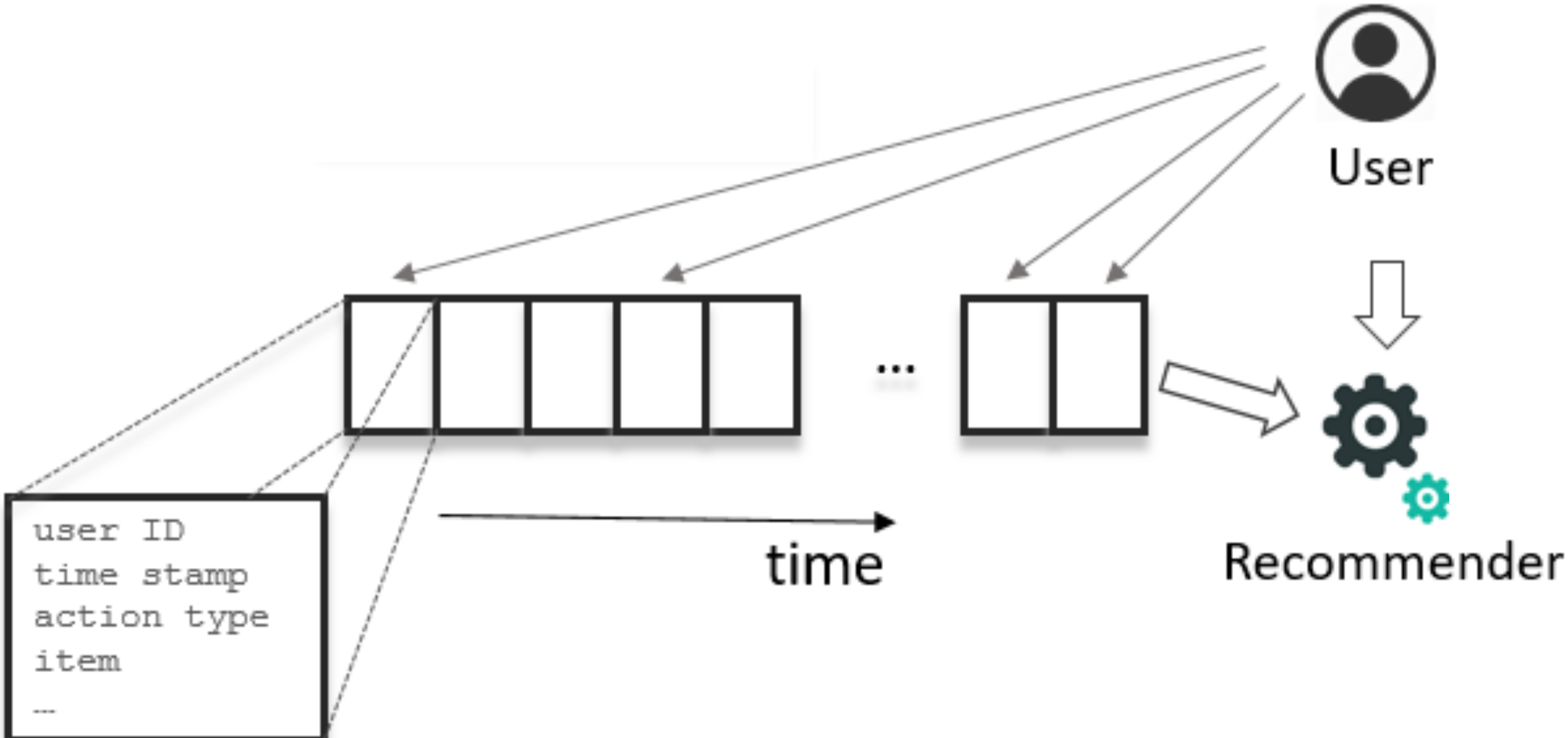
Inputs



Booking.com: Hotels in
https://www.booking.co
Free cancellation 1
No prepayment 79
Beach access
Beachfront 2
Meals
Breakfast included
Self catering 11
Property type
Hotels
Hostels
Resorts +3
Apartments +1
Breakfast included
Robertson Quay Hotel
Robertson Quay, Singapore – Show on map
(1.1 km from centre)
In high demand! Booked 30 times in the last 24 hours
Double Room
In high demand - only 3 rooms left!
Secret Deal available
M Social Singapore

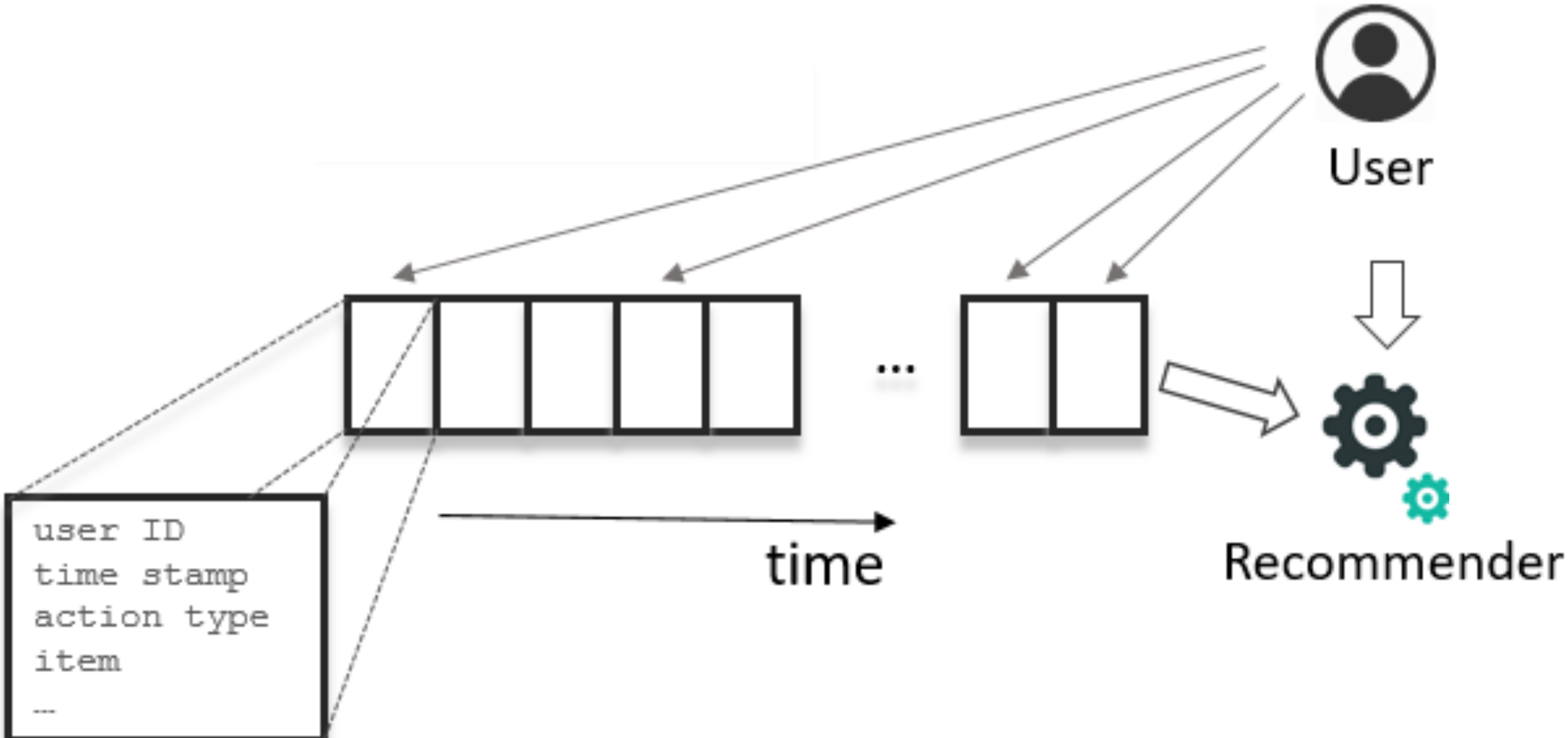
Inputs

- Different types of actions
  - Item purchase/consumption, item view, add-to-catalog, add-to-wish-list, …



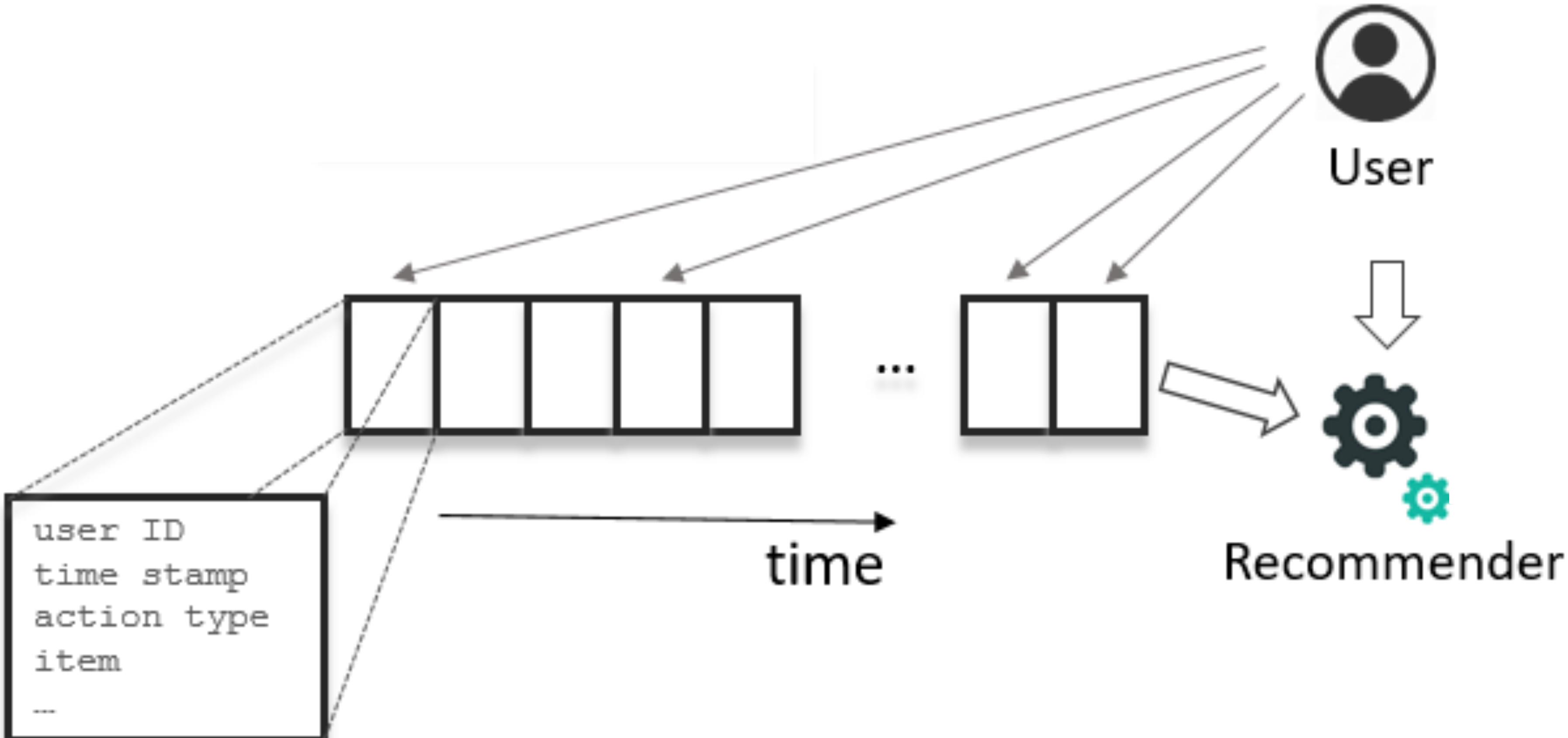
Inputs

- Attributes of actions: user/item details, context
  - Dwelling times, item discounts, etc.



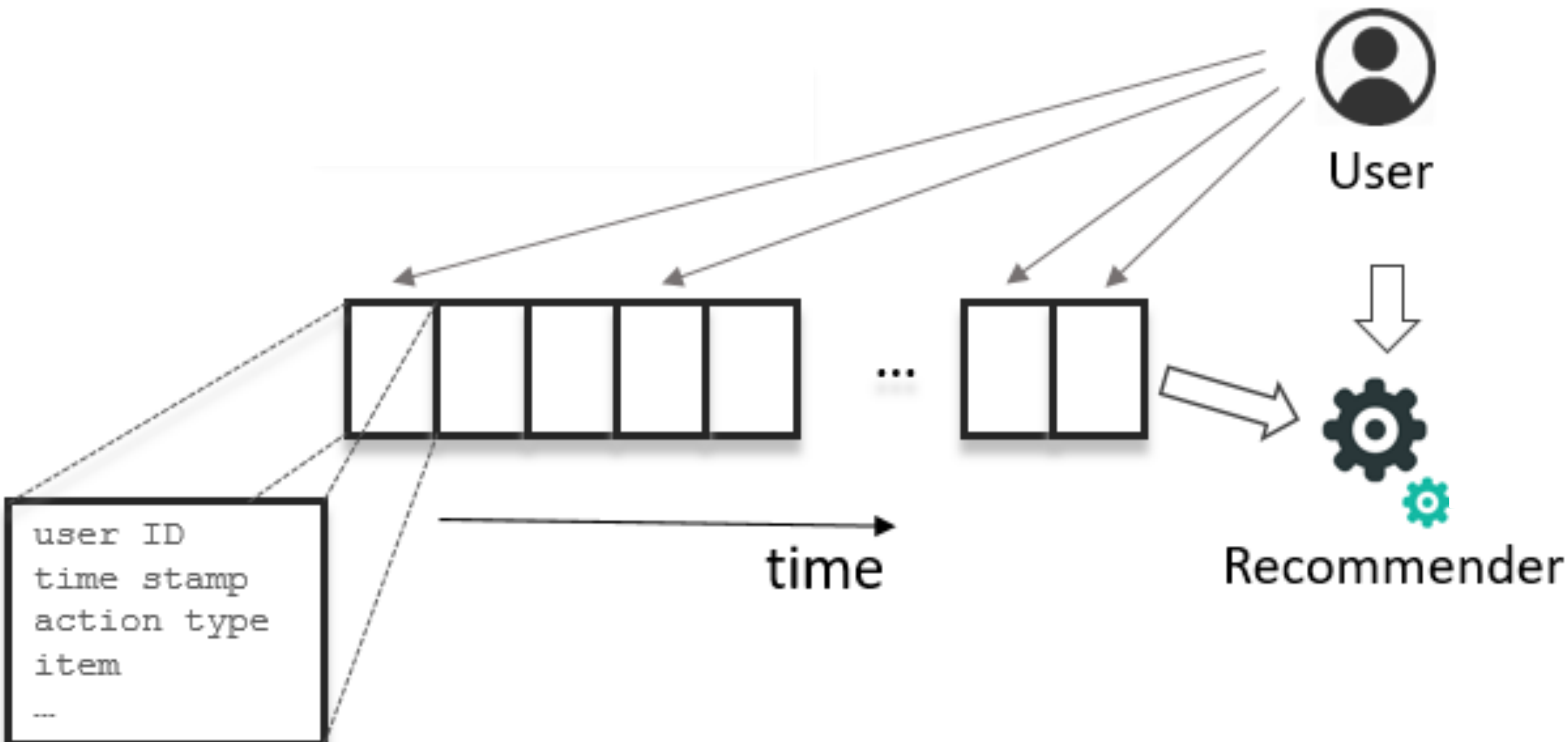
Inputs

- Typical: Enriched clickstream data



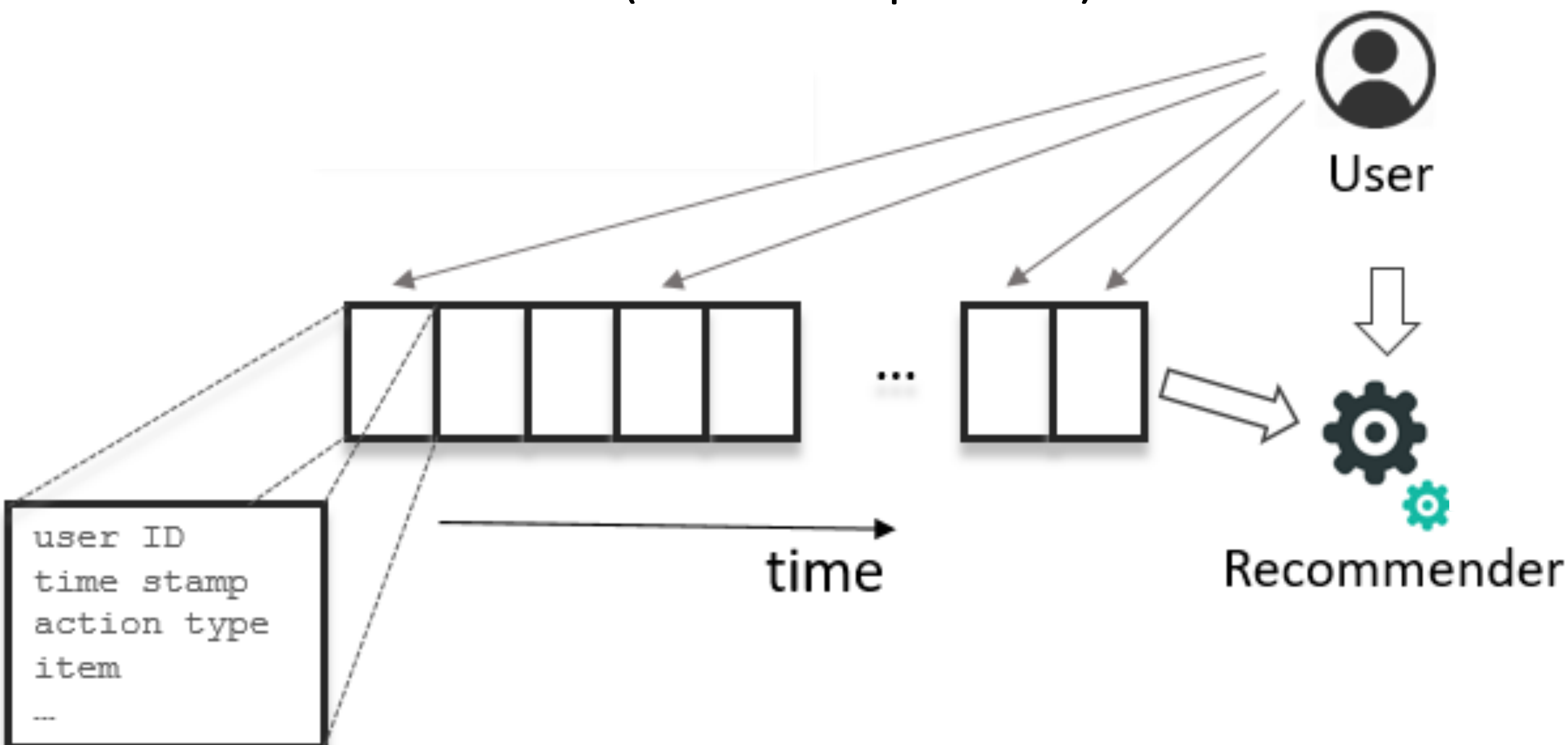
Inputs: differences with traditional RSs

- for each user, all sequences contain **one single action (item)**



Inputs: differences with traditional RSs (with time-stamp)

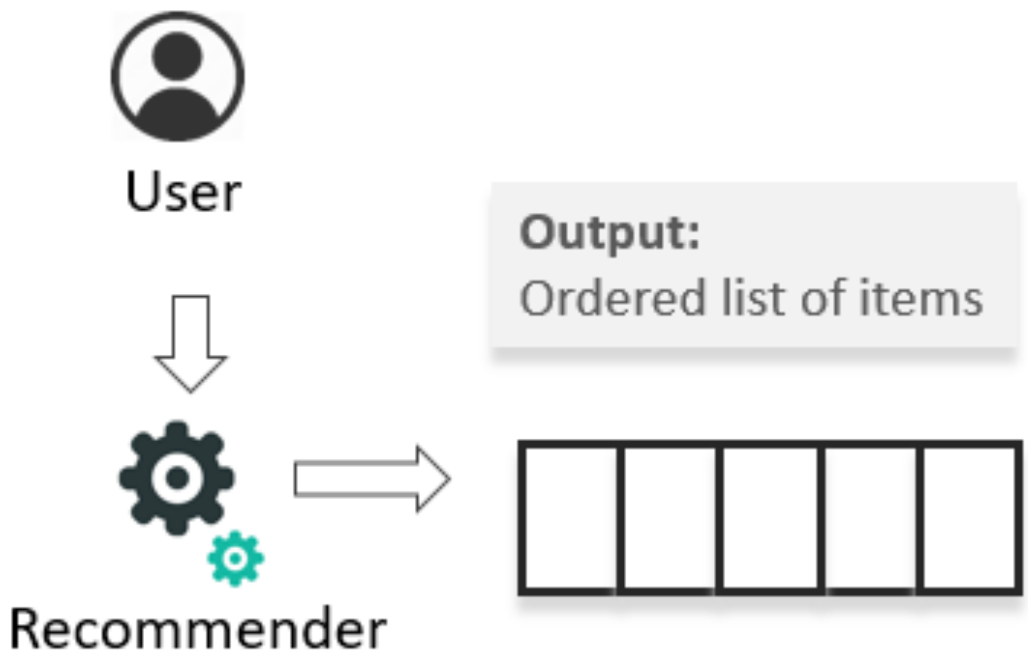
- for each user, there is
  one single sequence
  with several actions (one action per item)



Output (1)

- One (or more) ordered list of items
- The list can have different interpretations, based on goal, domain, application scenario

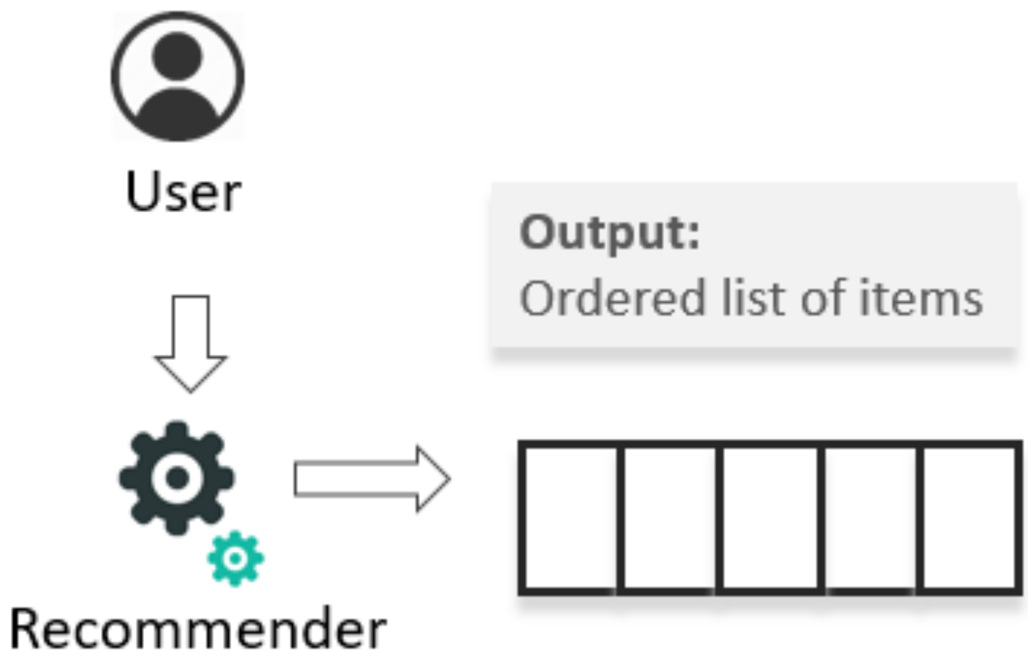
- Usual item-ranking tasks
  - list of alternatives for a given item
  - complements or accessories



Output (2)

- An ordered list of items
- The list can have different interpretations, based on goal, domain, application scenario

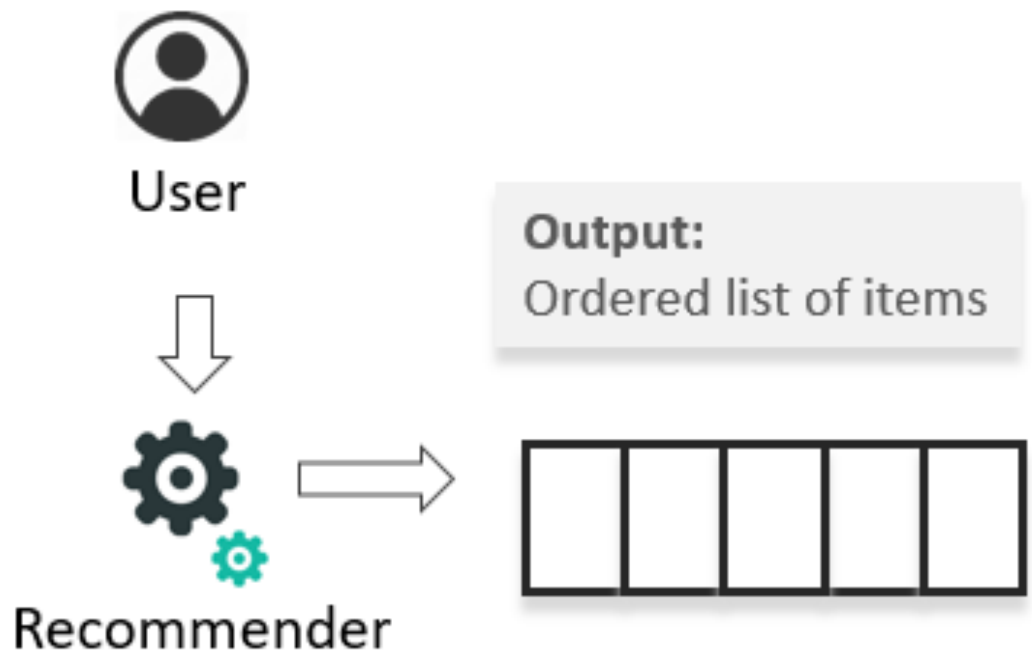
- Suggested sequence of actions
  - next-track music recommendations



Output (3)

- An ordered list of items
- The list can have different interpretations, based on goal, domain, application scenario

- Strict sequence of actions
  - course learning recommendations



Typical computational tasks

- Find sequence-related in the data, e.g.,
  - co-occurrence patterns
  - sequential patterns
  - distance patterns
- Reasoning about order constraints
  - weak and strong constraints
- Relate patterns with user profile and current point in time
  - e.g., items that match the current session

Abstract problem characterization: **Item-ranking** or **list-generation**

- Some definitions
  - U : users
  - I : items
  - L : ordered list of items of length k
  - L^* : set of all possible lists L of length up to k
  - $f(u, L)$: utility function, with $u \in U$ and $L \in L^*$

$$L_u = \operatorname*{argmax}_{L \in L^*} f(u, L) \quad u \in U$$

  - Task: learn $f(u, L)$ from sequences A of past user actions

Abstract problem characterization

- Utility function is not limited to scoring individual items
- The utility of entire lists can be assessed
  - including, e.g., transition between objects, fulfillment of order constraints, diversity aspects
- The design of the **utility function depends on the** purpose of the system
  - provide logical continuation
  - show alternatives
  - show accessories
  - …

[Jannach and Adomavicius] "***Recommendations with a Purpose***". RecSys 2016

On recommendation purposes

- Often, researchers are not explicit about the purpose
  - Traditionally, could be information filtering or discovery, with conflicting goals
- Commonly used abstraction
  - e.g., predict hidden rating
- For sequence-aware recommenders
  - often: predict next (hidden) action, e.g., for a given session beginning

Relation to other areas

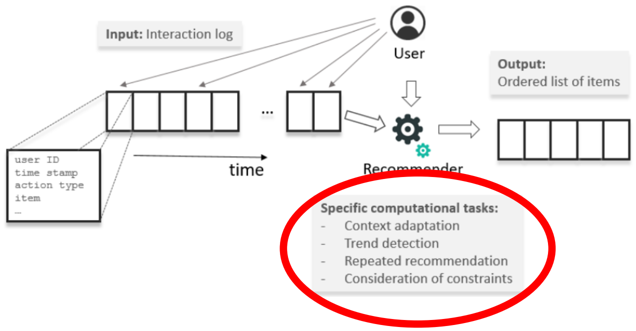
- Implicit feedback recommender systems
  - Sequence-aware recommenders are often built on implicit feedback signals (action logs)
  - Problem formulation is however not based on matrix completion
- Context-aware recommender systems
  - Sequence-aware recommenders often are special forms of context-aware systems
  - Here: Interactional context is relevant, which is only implicitly defined through the user's actions

Relation to other areas

- Time-Aware RSs
  - Sequence-aware recommenders do not necessarily need explicit timestamps
  - Time-Aware RSs use explicit time
    - e.g., to detect long-term user drifts
- Other:
  - interest drift
  - user-modeling

Categorization

Categorization of tasks



- Four main categories
  - Context Adaptation
  - Trend detection
  - Repeated recommendation
  - Consideration of order constraints and sequential patterns
- Notes
  - Categories are not mutually exclusive
  - All types of problems based on the same problem characterization, but with different utility functions, and using the data in different ways

Context Adaptation (CA)

- Traditional context-aware recommenders are often based on the representational context
  - defined set of variables and observations, e.g., weather, time of the day etc.
- Here, the interactional context is relevant
  - no explicit representation of the variables
  - contextual situation has to be inferred from user actions

CA: How much past information is considered?

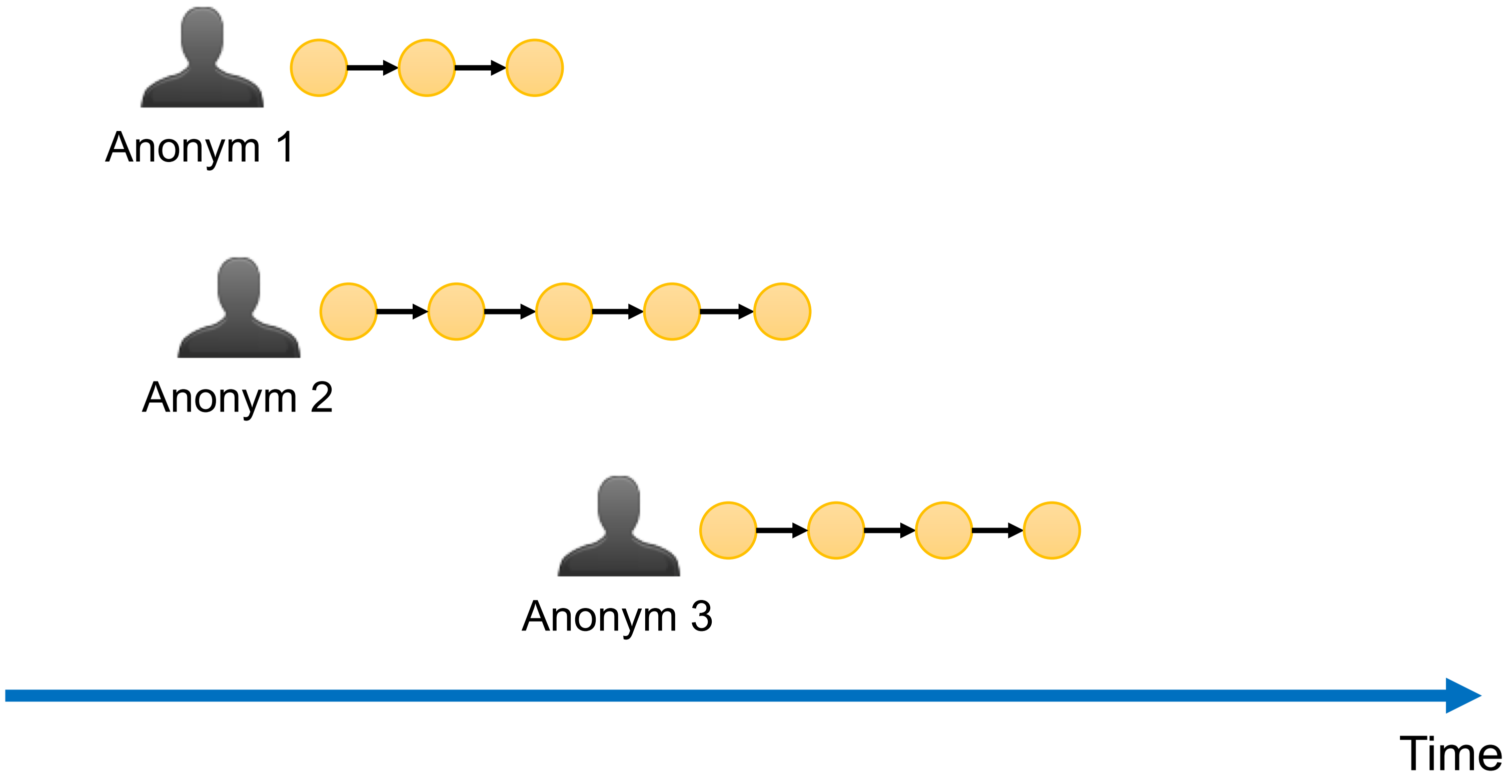
- Last-N interactions based recommendation:
  - Often used in Next-Point-Of-Interest recommendation scenarios
  - In many cases only the very last visited location is considered
  - Also: “Customers who bought … also bought”
- Reasons to limit oneself:
  - Not more information available
  - Previous information not relevant

CA: How much past information is considered?

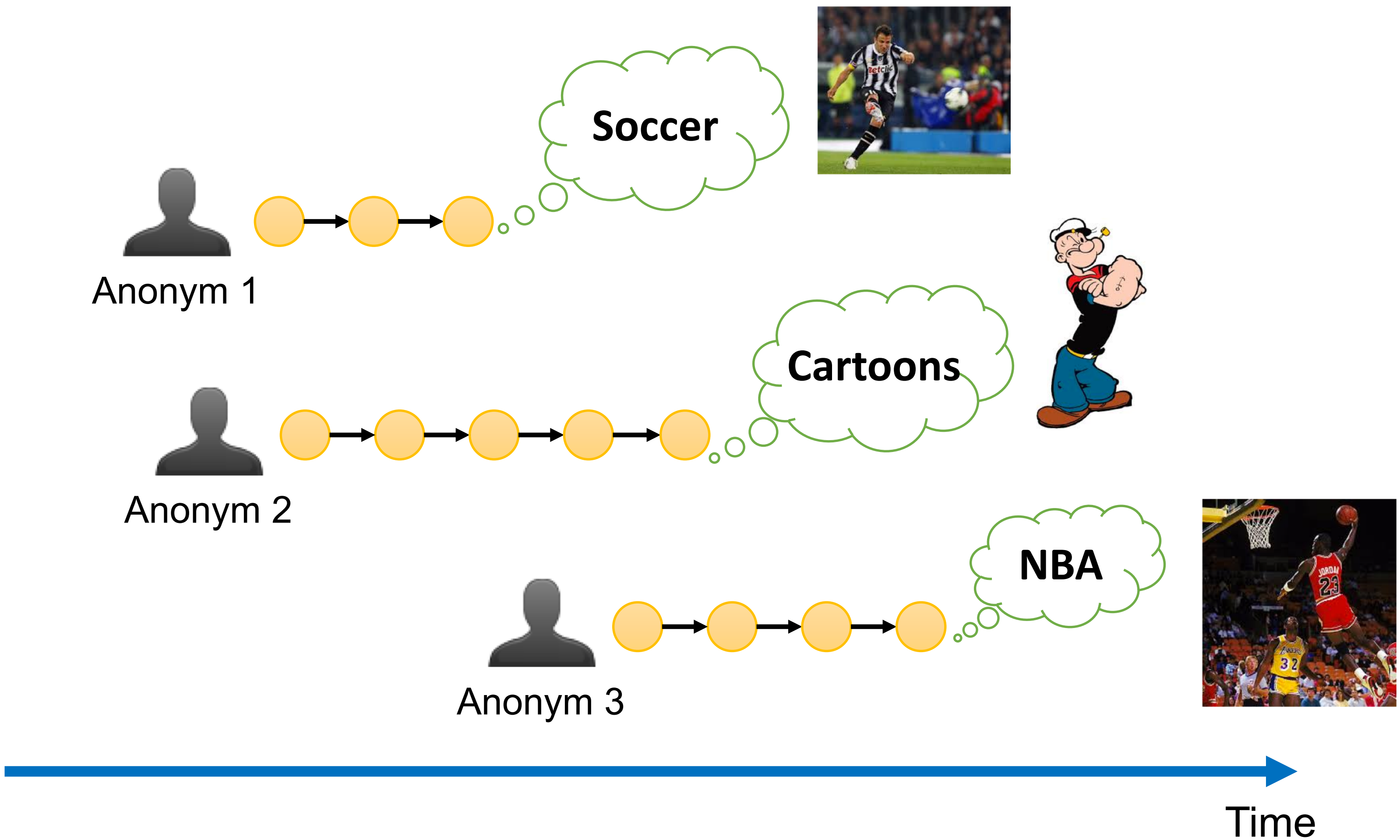
- Session-based recommendation:
  - **Short-term only**
  - Only last sequence of actions of the current user is known
  - User might be anonymous
- Session-aware recommendation:
  - **Short-term + Long-term**
  - In addition, past session of the current user are known
  - Allows for personalized session-based recommendation

[Quadrana et al.] **"Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks"** RecSys 2017

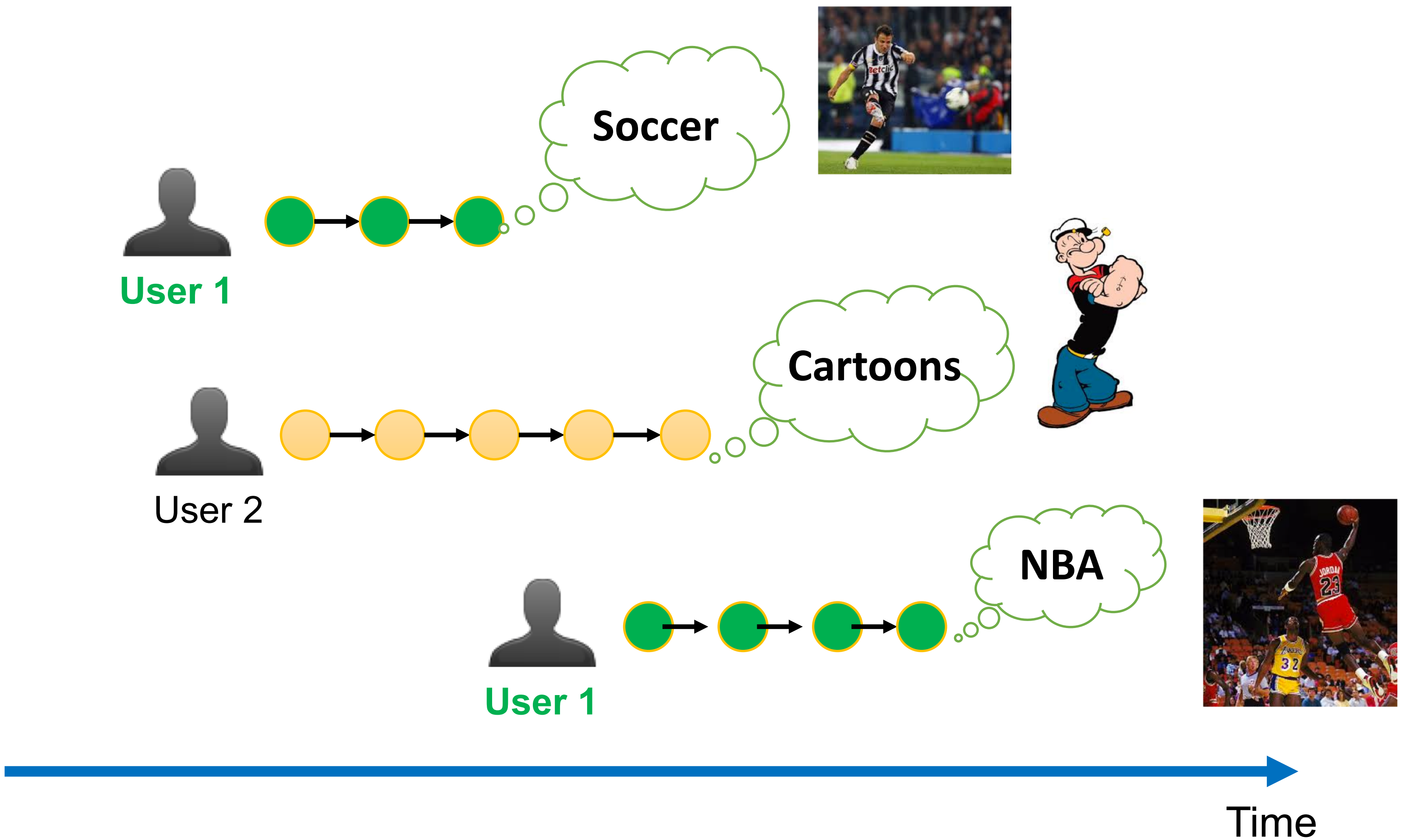
CA: Session-based recommendation



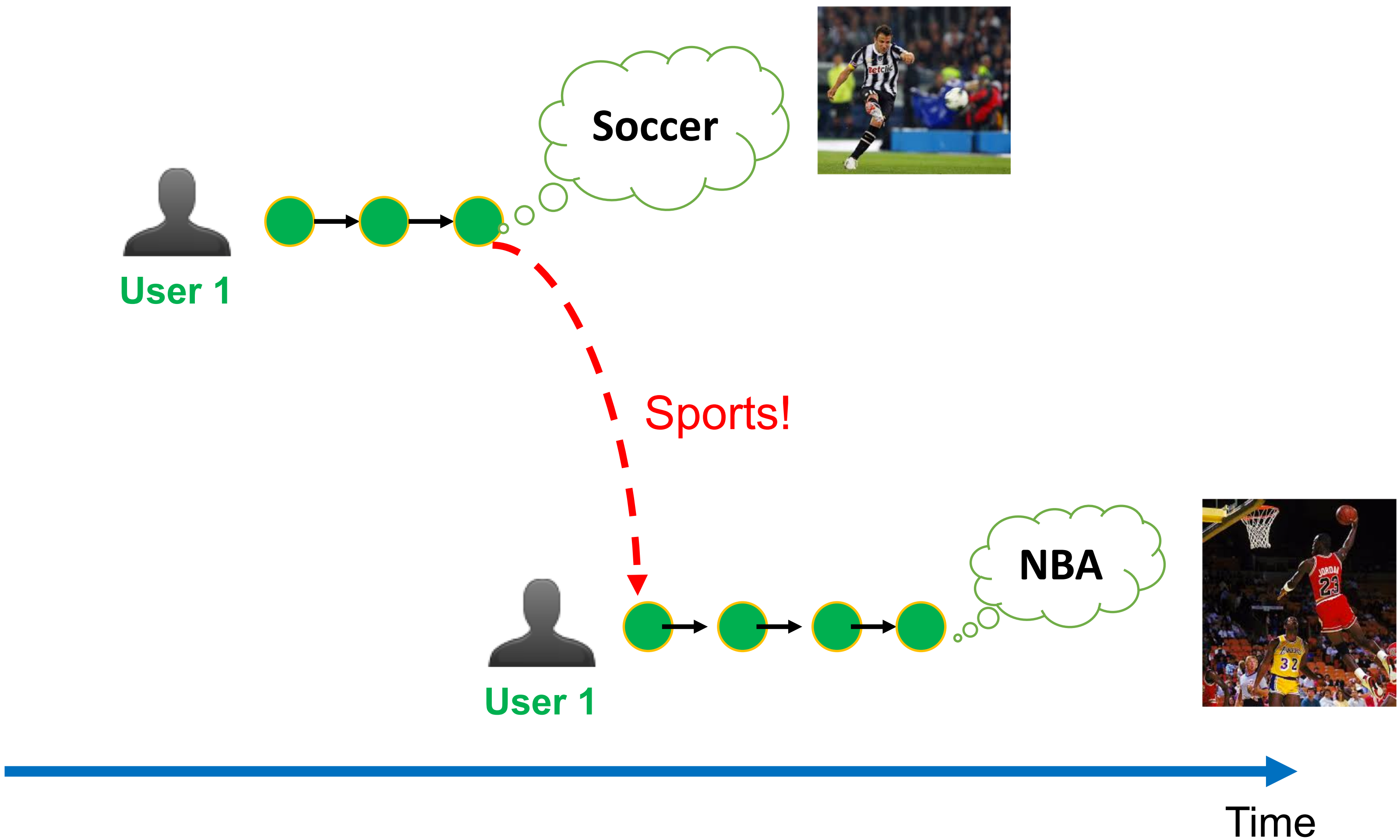
CA: Session-based recommendation



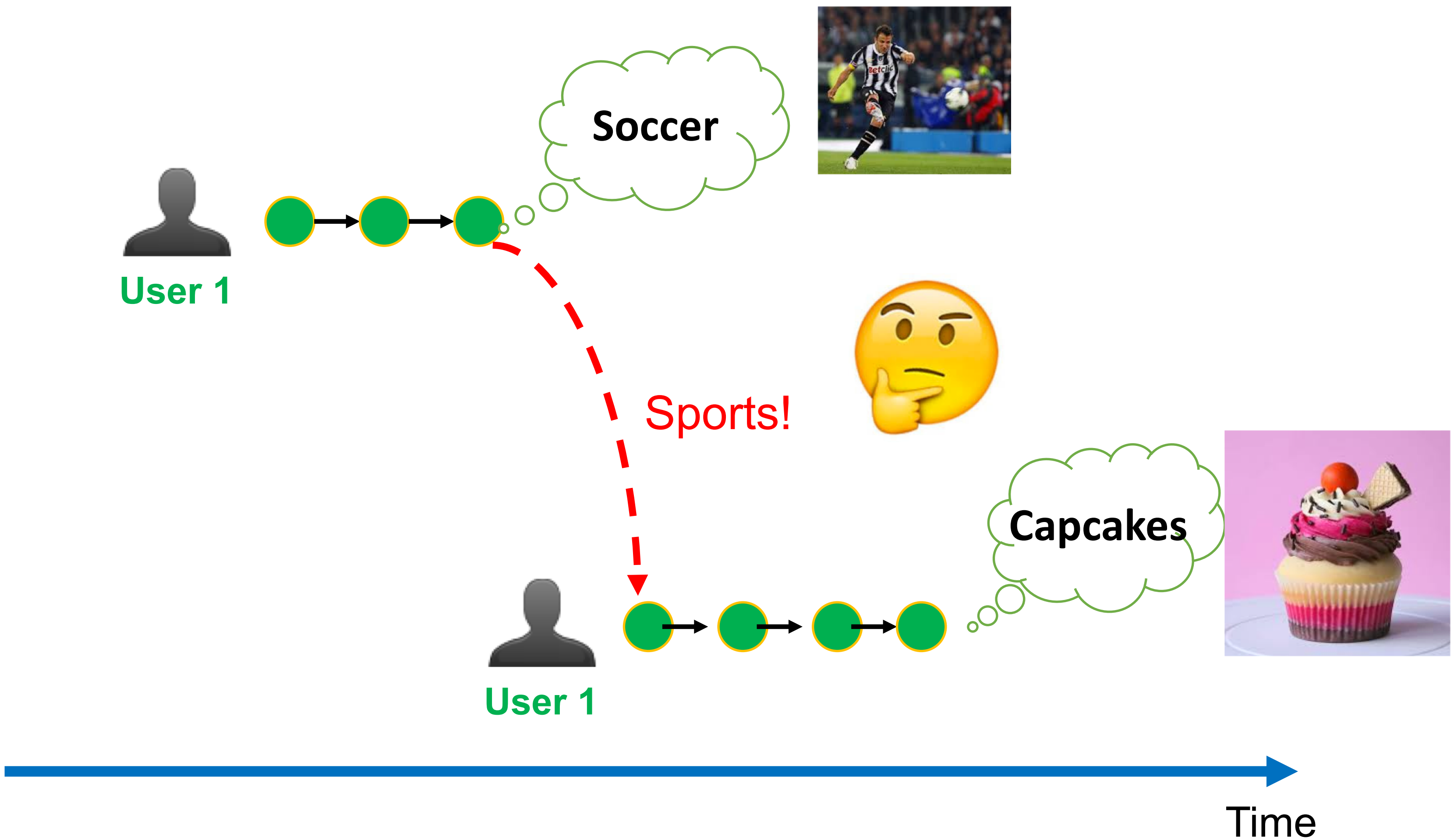
CA: Session-aware recommendation



CA: Session-aware recommendation



CA: Session-aware recommendation



CA: What to find?

- Next
- Alternatives
- Complements
- Continuations

CA: What to pick

- One
- All

Trend detection

- Less explored than context adaptation
- Community trends:
  - Consider the recent or seasonal popularity of items, e.g., in the fashion domain and, in particular, in the news domain
- Individual trends:
  - E.g., natural interest drift
    - Over time, because of influence of other people, because of a recent purchase, because something new was discovered (e.g., a new artist)

[Jannach et al.] "**Session-based Item Recommendation in E-Commerce: On Short-Term Intents, Reminders, Trends, and Discounts**". UMAP 2017

Repeated Recommendation

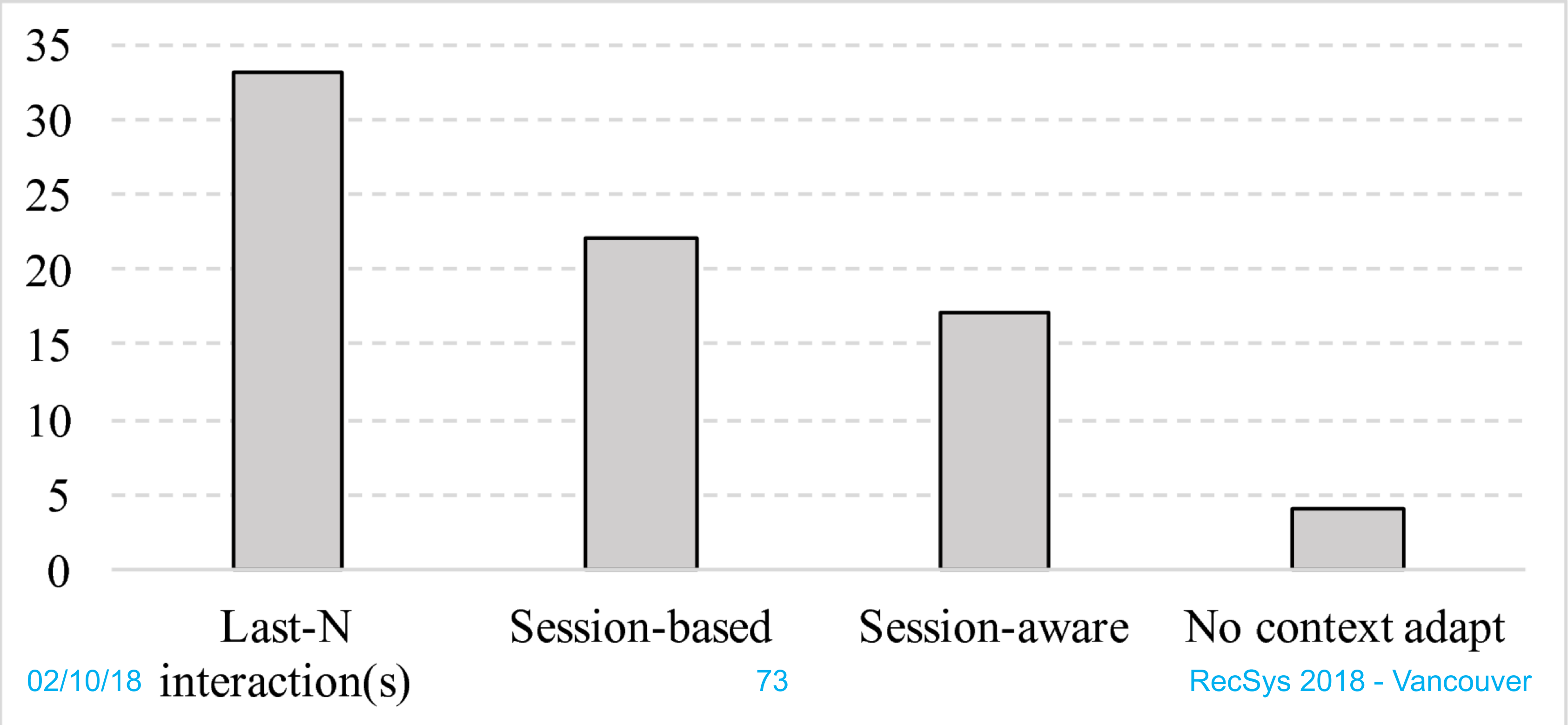
- Identifying repeated user behavior patterns:
  - Recommendation of repeat purchases or actions
    - E.g., ink for a printer, next app to open after call on mobile
  - Patterns can be mined from the individual or the community as a whole
- Repeated recommendations as reminders
  - Remind users of things they found interesting in the past
    - To remind the of things they might have forgotten
    - As navigational shortcuts, e.g., in a decision situation
- Timing as an interesting question in both situations

Consideration of Order Constraints and Observed Sequential Patterns

- Two types of sequentiality information
  - External domain knowledge: strict or weak ordering constraints
    - Strict, e.g., sequence of learning courses
    - Weak, e.g., when recommending sequels to movies
  - Information that is mined from the user behavior
    - Learn that one movie is always consumed after another
    - Predict next web page, e.g., using sequential pattern mining techniques

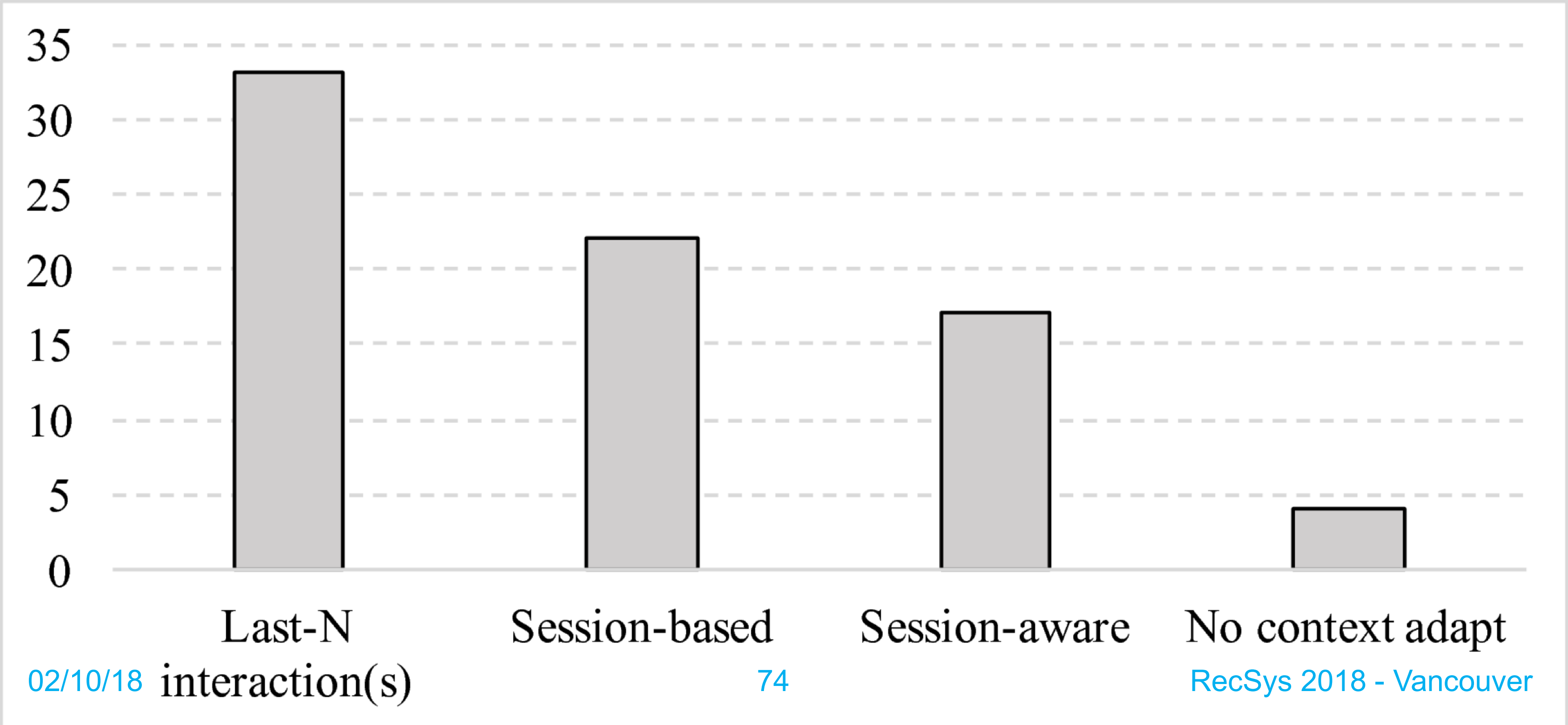
Review of existing works (1)

- 100+ papers reviewed
  - Focus on last N interactions common
    - But more emphasis on session-based / session-aware approaches in recent years



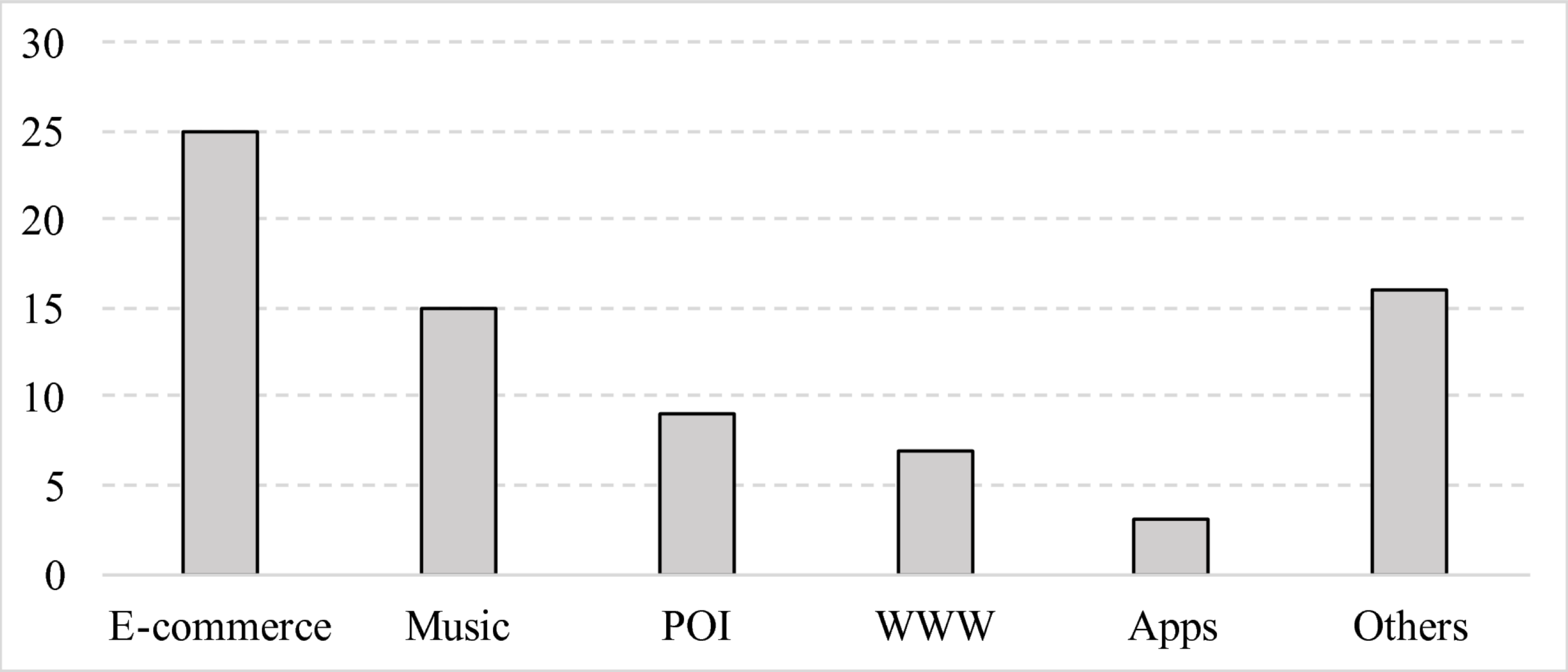
Review of existing works (2)

- 100+ papers reviewed
  - Very limited work for other problems:
    - Repeated recommendation, trend detection, consideration of constraints



Review of existing works (3)

- Application domains



Summary of first part

- Matrix completion abstraction not well suited for many practical problems
- In reality, rich user interaction logs are available
- Different types of information can be derived from the sequential information in the logs
  - and used for special recommendation tasks, in particular for the prediction of the next-action
- Coming next:
  - Algorithms and evaluation

Evaluation and Datasets

Agenda

- 14:00 – 14:45 Introduction & Problem Definition (Paolo)
- 14:45 – 15:15 Evaluation (Paolo)
- 15:15 – 15:30 Algorithms I (Massimo)
- 15:30 – 16:00 Coffee break
- 16:00 – 16:45 Algorithms II (Massimo)
- 16:45 – 17:20 Hands-on (Massimo)
- 17:20 – 17:30 Conclusion / Questions

Traditional evaluation approaches

- Off-line
  - evaluation metrics
    - Error metrics: RMSE, MAE
    - Classification metrics: Precision, Recall, …
    - Ranking metrics: MAP, MRR, NDCG, …
  - dataset partitioning (hold-out, leave-one-out, …)
  - available datasets
- On-line evaluation
  - users studies
  - field test

Dataset partitioning

- Splitting the data into training and test sets

event-level

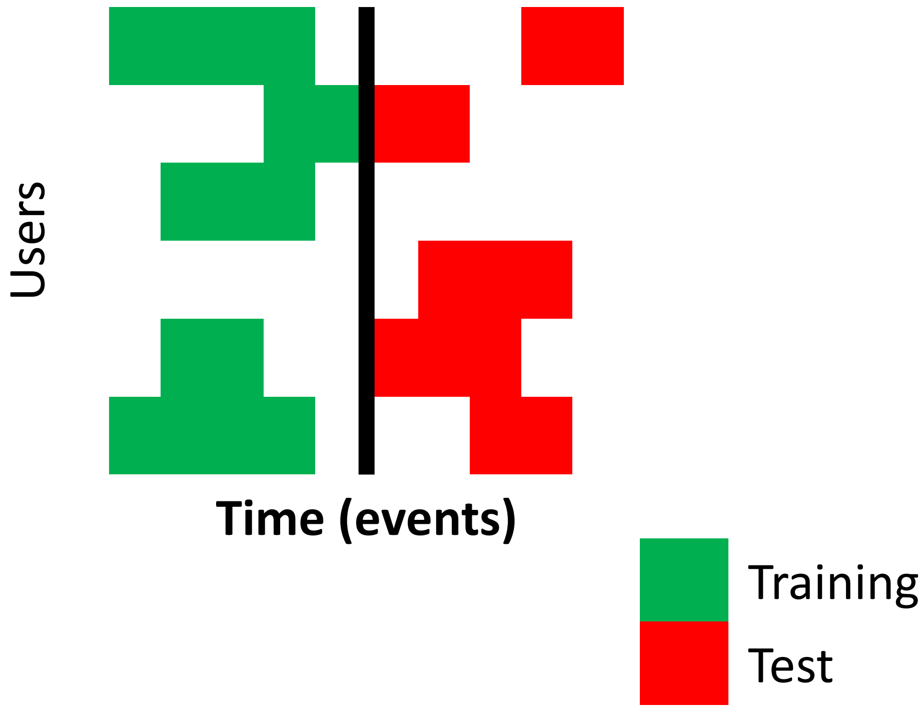
session-level

Dataset partitioning

- Splitting the data into training and test sets

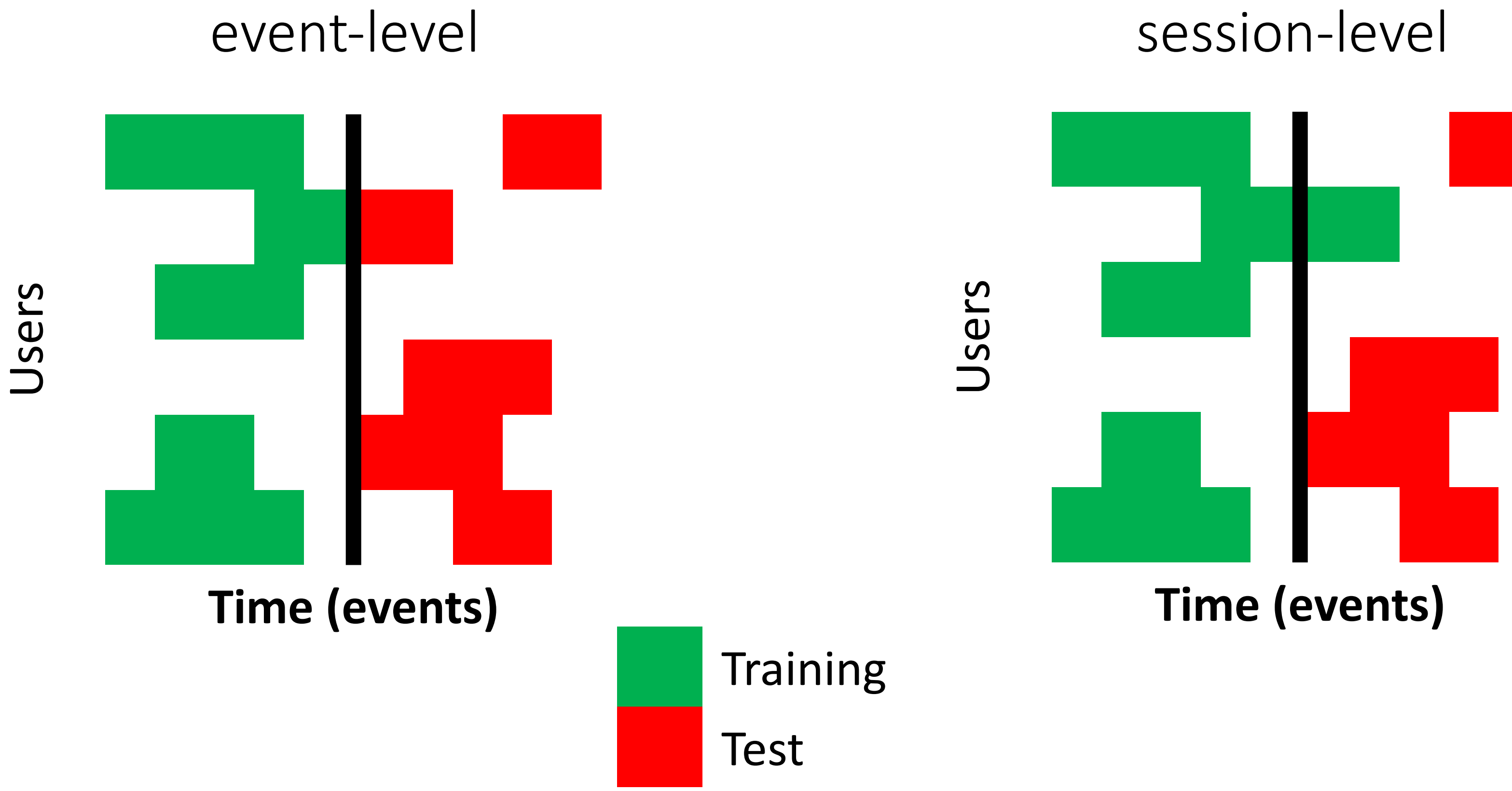
event-level

session-level



Dataset partitioning

- Splitting the data into training and test sets



Dataset partitioning

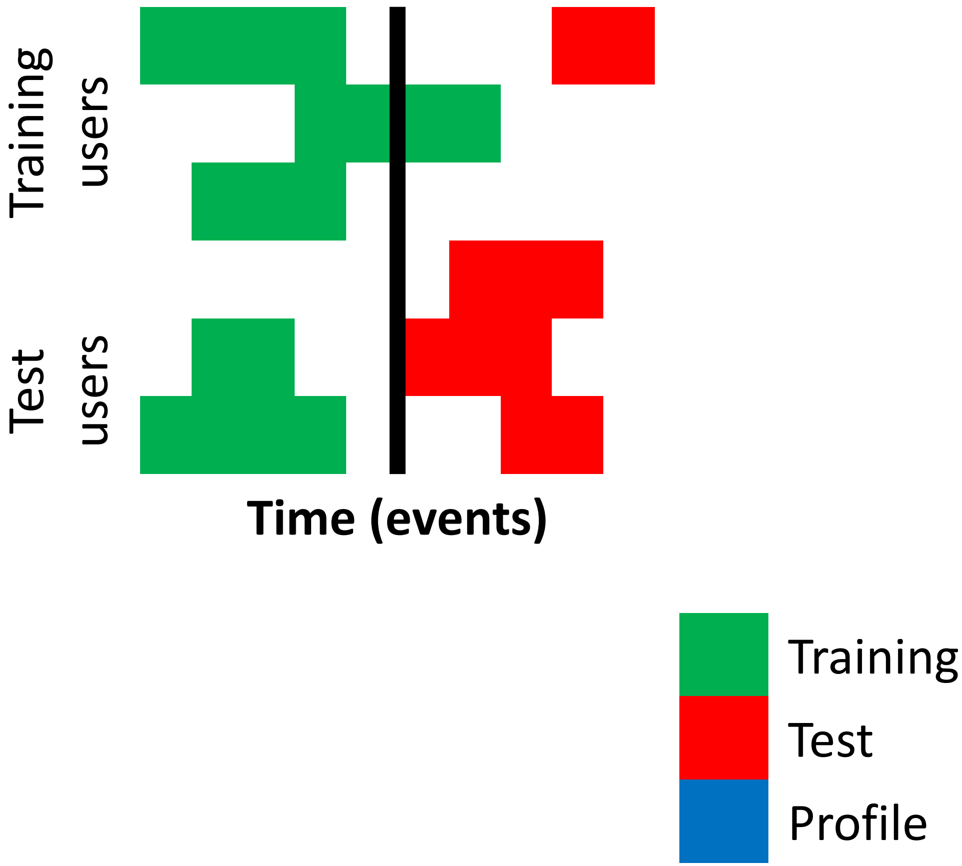
Community Level

User-level

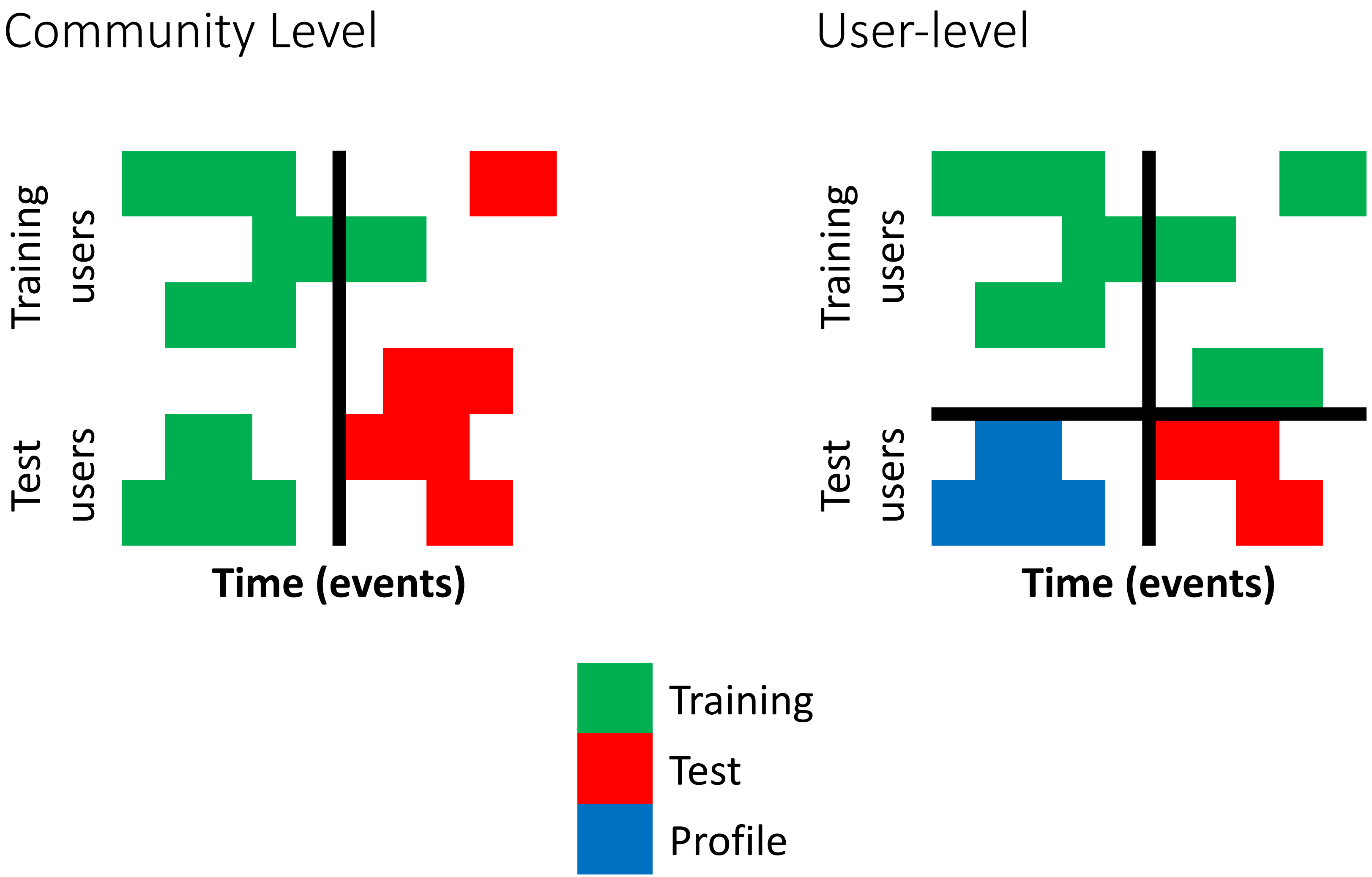
Dataset partitioning

Community Level

User-level

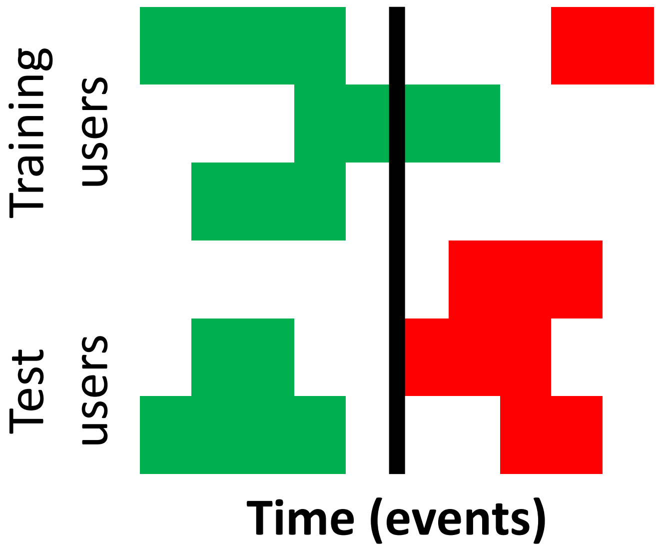


Dataset partitioning

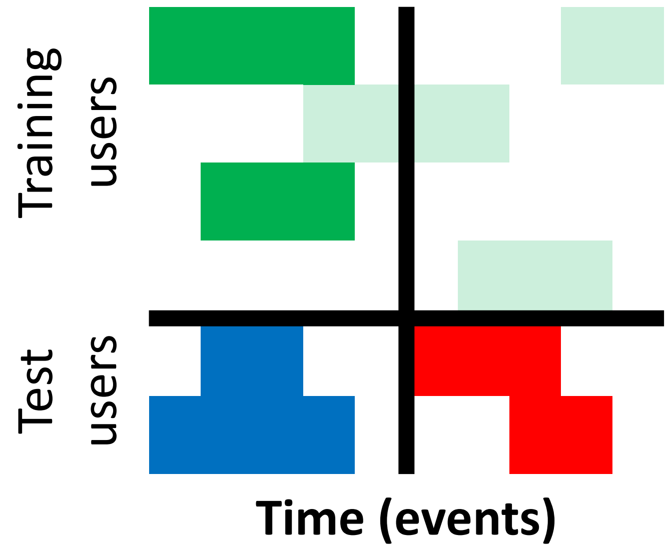


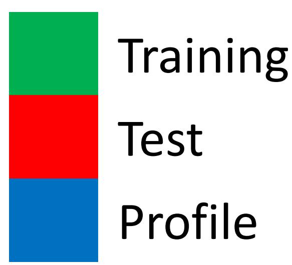
Dataset partitioning

Community Level



User-level





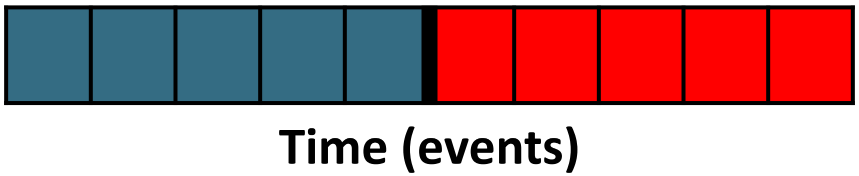
Dataset partitioning

- session-based or session-aware task
  → use **session-level** partitioning
  - better mimic real systems trained on past sessions

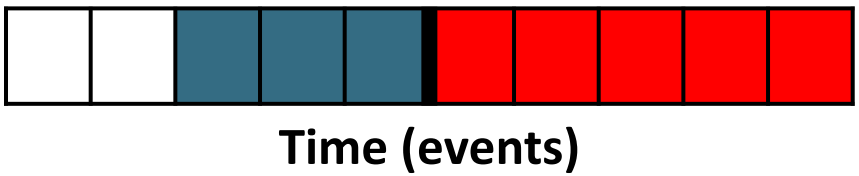
- session-based task
  → use session-level + **user-level**
  - to test for new users

Past interactions (user profile)

- All past interactions



- Given-N



Target Items

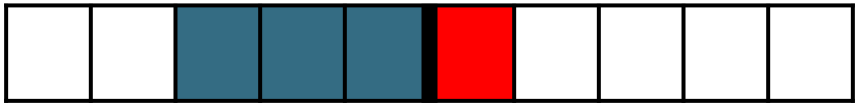
- Sequence agnostic prediction
  - e.g., similar to matrix completion

Time (events)

Time (events)

Target Items

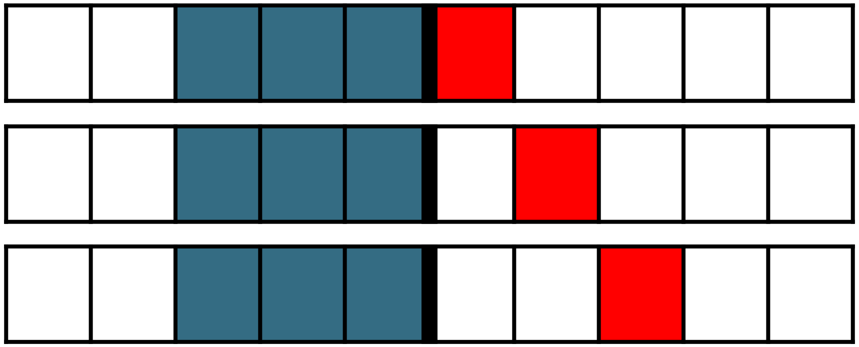
- **Given-N next item prediction**
  - e.g., next track prediction



Time (events)

Target Items

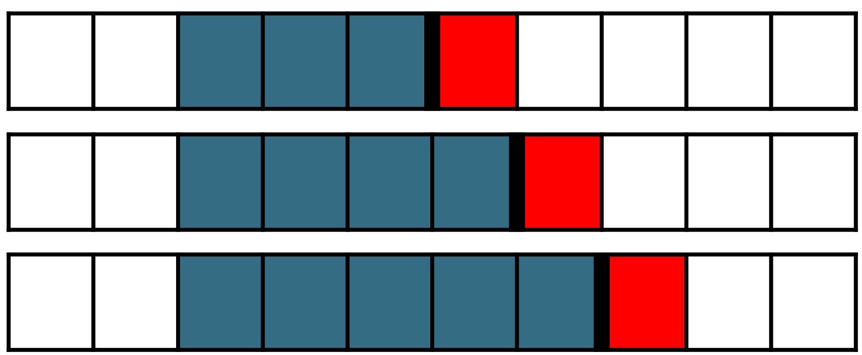
- Repeated given-N next item prediction
  - e.g., predict a sequence of events



Time (events)

Target Items

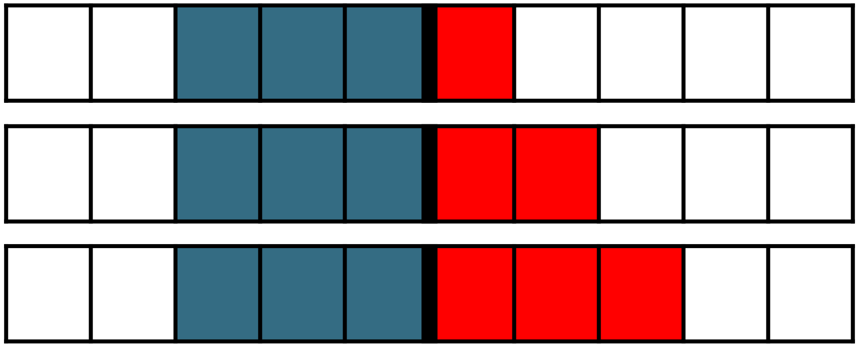
- Repeated given-N next item prediction with N incremental
  - e.g., predict a sequence of events



Time (events)

Target Items

- Given-N next item prediction with look-ahead size K



Datasets

Name	Domain	Users	Items	Events	Sessions
Amazon	EC	20M	6M	143M	
Epinions	EC	40k	110k	181k	
Ta-feng	EC	32k	24k	829k	
TMall	EC	1k	10k	5k	
Retailrocket	EC	1.4M	235k	2.7M	
Microsoft	WWW	27k	8k	55k	32k
MSNBC	WWW	1.3M	1k	476k	
Delicious	WWW	8.8k	3.3k	60k	45k
CiteULike	WWW	53k	1.8k	2.1M	40k
Outbrain	WWW	700M	560	2B	

Datasets

Name	Domain	Users	Items	Events	Sessions
AOL	Query	650k		17M	2.5M
Adressa	News	15k	923	2.7M	
Foursquare_2	POI	225k		22.5M	
Gowalla_1	POI	54k	367k	4M	
Gowalla_2	POI	196k		6.4M	
30Music	Music	40k	5.6M	31M	2.7M

An example of thorough evaluation

Evaluation of Session-based Recommendation Algorithms

MALTE LUDEWIG, TU Dortmund, Germany
DIETMAR JANNACH, AAU Klagenfurt, Austria

Recommender systems help users find relevant items of interest, for example on e-commerce or media streaming sites. Most academic research is concerned with approaches that personalize the recommendations according to long-term user profiles. In many real-world applications, however, such long-term profiles often do not exist and recommendations therefore have to be made solely based on the observed behavior of a user during an ongoing session. Given the high practical relevance of the problem, an increased interest in this problem can be observed in recent years, leading to a number of proposals for *session-based recommendation algorithms* that typically aim to predict the user's immediate next actions.

In this work, we present the results of an in-depth performance comparison of a number of such algorithms, using a variety of datasets and evaluation measures. Our comparison includes the most recent approaches based on recurrent neural networks like GRU4REC, factorized Markov model approaches such as FISM or FOSSIL, as well as more simple methods based, e.g., on nearest neighbor schemes. Our experiments reveal that algorithms of this latter class, despite their sometimes almost trivial nature, often perform equally well or significantly better than today's more complex approaches based on deep neural networks. Our results therefore suggest that there is substantial room for improvement regarding the development of more sophisticated session-based recommendation algorithms.

[Ludewig and Jannach] **Evaluatation of Session-based Recommendation Algorithms.** ArXiv

Algorithms

Agenda

- 14:00 – 14:45 Introduction & Problem Definition (Paolo)
- 14:45 – 15:15 Evaluation (Paolo)
- 15:15 – 15:30 Algorithms I (Massimo)
- 15:30 – 16:00 Coffee break
- 16:00 – 16:45 Algorithms II (Massimo)
- 16:45 – 17:20 Hands-on (Massimo)
- 17:20 – 17:30 Conclusion / Questions

Taxonomy

- Sequence Learning
  - Frequent Pattern Mining
  - Sequence Modeling
  - Distributed Item Representations
  - Supervised Models with Sliding Window
- Sequence-aware Matrix Factorization
- Hybrids
  - Factorized Markov Chains
  - LDA/Clustering + sequence learning
- Others
  - Graph-based, Discrete-optimization

Taxonomy

- Sequence Learning
  - Frequent Pattern Mining
  - Sequence Modeling
  - Distributed Item Representations
  - Supervised Models with Sliding Window
- Sequence-aware Matrix Factorization
- Hybrids
  - Factorized Markov Chains
  - LDA/Clustering + sequence learning
- Others
  - Graph-based, Discrete-optimization

Sequence Learning (SL)

- Useful in application domains where input data has an inherent sequential nature
  - Natural Language Processing
  - Time-series prediction
  - DNA modelling
  - Automatic generation of Garfield comic strips ☺
  - Sequence-Aware Recommendation

Frequent Pattern Mining (FPM)

1. Discover **user consumption patterns**
   - Association rules, (Contiguous) Sequential Patterns
2. Look for patterns matching partial transactions
3. Rank items by confidence of matched rules

Size 1	Size 2	Size 3
$< A >$ (5)	$< A, B >$ (4)	$< A, B, E >$ (4)
$< B >$ (6)	$< A, C >$ (4)	$< A, E, C >$ (4)
$< C >$ (4)	$< A, E >$ (4)	
$< E >$ (5)	$< B, C >$ (4)	
	$< B, E >$ (5)	
	$< C, E >$ (4)	

Table 3: Frequent Sequential Patterns

Credits (Nakagawa and Mobasher, 2003)

Applications

- Page prefetching and recommendations
- Personalized FPM for next-item recommendation
- Next-app prediction

FPM

- Association Rule Mining
  - items co-occurring within the same sessions
    - no check on order
    - if you like A and B, you also like C (aka: learning to rank)
- Sequential Pattern Mining
  - Items co-occurring in the same order
    - no check on distance
    - If you watch A and **later** watch B, you will **later** watch C
- Contiguous Sequential Pattern Mining
  - Item co-occurring in the same order and distance
    - If you watch A and B **one after the other**, if **now** watch C

[Mobasher et al] **Using sequential and non-sequential patterns in predictive web usage mining tasks**. ICDM '02

FPM

- Two steps approach
  1. Offline: rule mining
  2. Online: rule matching (with current user session)

- Rules have
  - Support: number of examples (main parameter)
  - Confidence: conditional probability

- Lower thresholds -> fewer rules
  - few rules: difficult to find rules matching a session
  - many rules: noisy rules (low quality)

Frequent Pattern Mining (FPM)

- Pros
  - Easy to implement
  - Explainable predictions

- Cons
  - Choice of the minimum support/confidence thresholds
  - Data sparsity
  - Limited scalability

[C. Lu et al] **Mining mobile application sequential patterns for usage prediction**. GrC '14
[Nakagawa and Mobasher] **Impact of site characteristics on recommendation models based on association rules and sequential patterns**. IJCAI '03
[Yap et al.] **Effective next-items recommendation via personalized sequential pattern mining**. DASFAA '12
[Zhou et al.] **An intelligent recommender system using sequential web access patterns**. CIS '04


DEEP LEARNING
IS COMING
imgflip.com

Sequence Modeling

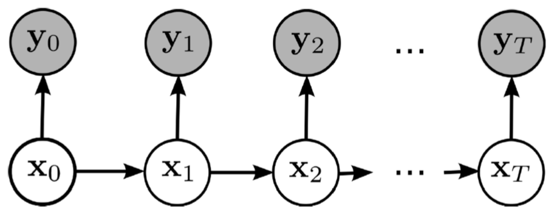
- Sequences of past user actions as time series with **discrete** observations
  - Timestamps used only to **order** user actions (optionally to model time intervals)

- Aim to learn models from past observations to predict future ones

- Categories of SM models
  - Markov Models
  - Reinforcement Learning
  - Recurrent Neural Networks

Markov Models

- Stochastic processes over **discrete random variables**
  - Finite history (= order of the model)
    → user actions depend on a limited # of most recent actions
  - Extensions: Variable Order MM / Context Trees, HMMs



HMM (Harvard University, Source)

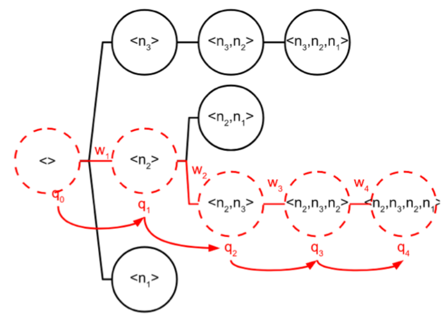


Figure 2: VMM context tree for the sequence s = $\langle n_1, n_2, n_3, n_2 \rangle$. **Nodes in red-dashed are active experts** $\mu \in A(\mathbf{s})$.

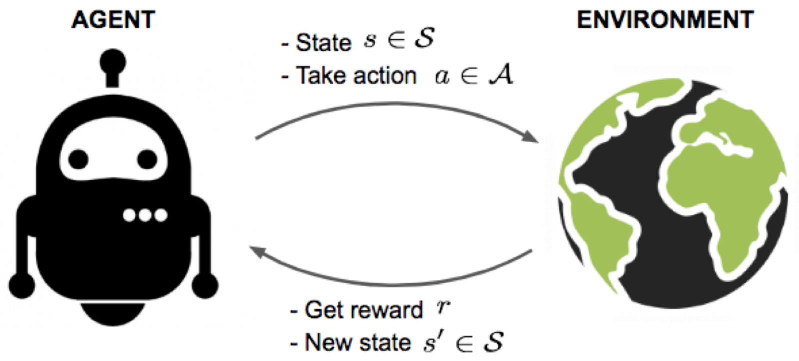
Credits (Garcin et al, 2013)

- Applications
  - Online shops
  - Playlist generation
  - **Variable Order Markov Models** for news recommendation
  - **Hidden Markov Models** for contextual next track prediction

[Garcin et al.] **Personalized news recommendation with context trees**. RecSys ’13,
[He et al.] **Web query recommendation via sequential query prediction**. ICDE ’09
[Hosseinzadeh Aghdam et al.] **Adapting recommendations to contextual changes using hierarchical hidden markov models**. RecSys ’15
[McFee and Lanckriet] **The natural language of playlists**. ISMIR ’11
[Shani et al.] **An MDP-based recommender system**. J. Mach. Learn. Res. 2005.

Reinforcement Learning

- Learn by **sequential interactions** with the **environment**
- Generate recommendations (**actions**) and collect user feedback (**reward**)
- Markov Decisions Processes (MDPs)



Credits lilianweng.github.io

- Applications
    - Online e-commerce services
    - Sequential relationships between the attributes of items explored in a user session

[Moling et al.] **Optimal radio channel recommendations with explicit and implicit feedback**. RecSys '12
[Shani et al.] **An MDP-based recommender system**. J. Mach. Learn. Res. 2005.
[Tavakol and Brefeld] **Factored mdps for detecting topics of user sessions**. RecSys '14

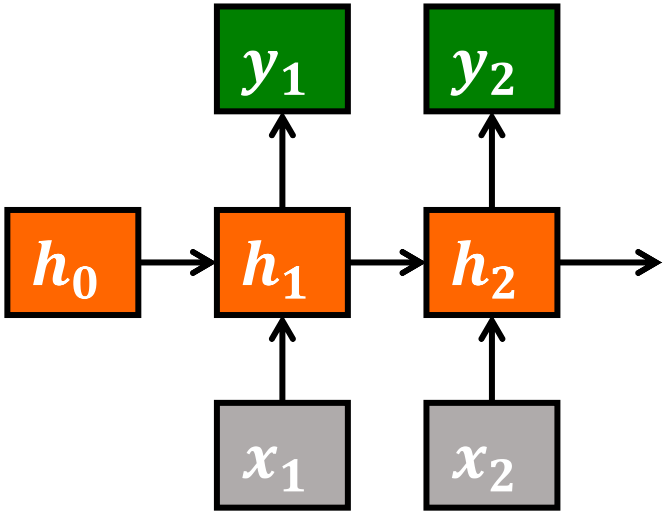
Recurrent Neural Networks (RNN)

- Distributed real-valued hidden state models with non-linear dynamics
  - Hidden state: latent representation of user state within/across sessions
  - Update the hidden state on the current input and its previous value, then use it to predict the probability for the next action

- Applications
  - Next-click prediction with RNNs
  - Session-based recommendation
  - Long-term/short-term user modeling

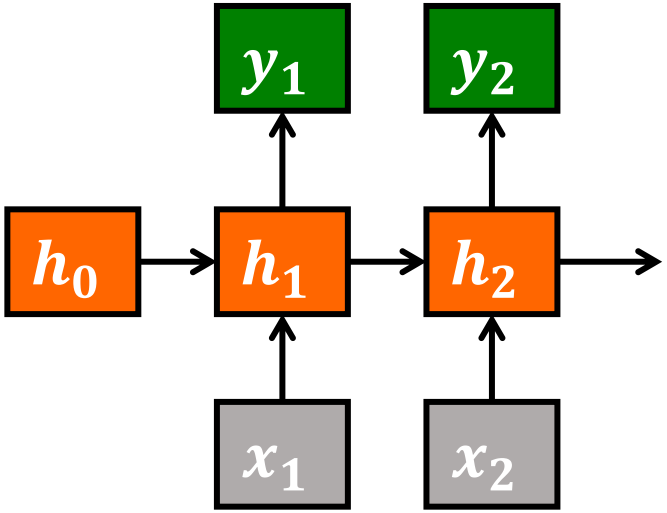
Simple Recurrent Neural Network

- Hidden state → used to predict the output
  - Computed from next input and previous hidden state



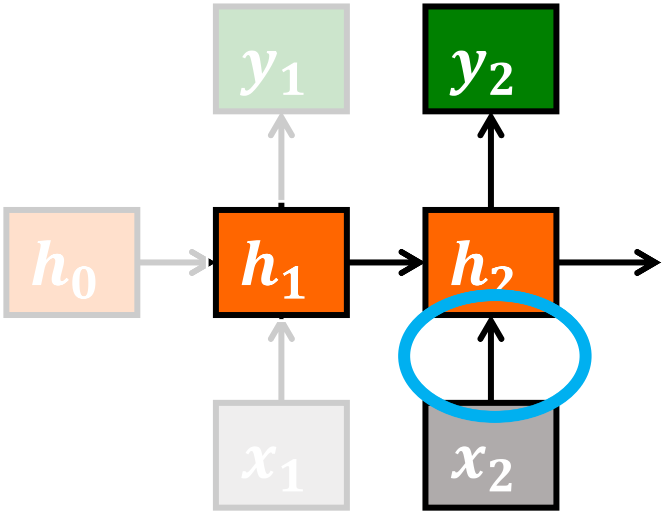
Simple Recurrent Neural Network

- Hidden state → used to predict the output
  - Computed from next input and previous hidden state
- Three weight matrices
  - $h_t = f(x^T W_x + h_{t-1}^T W_h + b_h)$
  - $y_t = f\left(h_t^T W_y\right)$



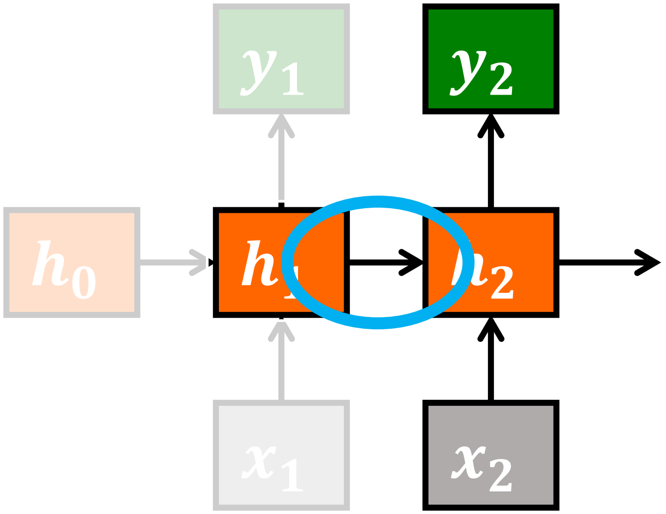
Simple Recurrent Neural Network

- Hidden state → used to predict the output
  - Computed from next input and previous hidden state
- Three weight matrices
  - $h_t = f(x^T W_x + h_{t-1}^T W_h + b_h)$
  - $y_t = f\left(h_t^T W_y\right)$



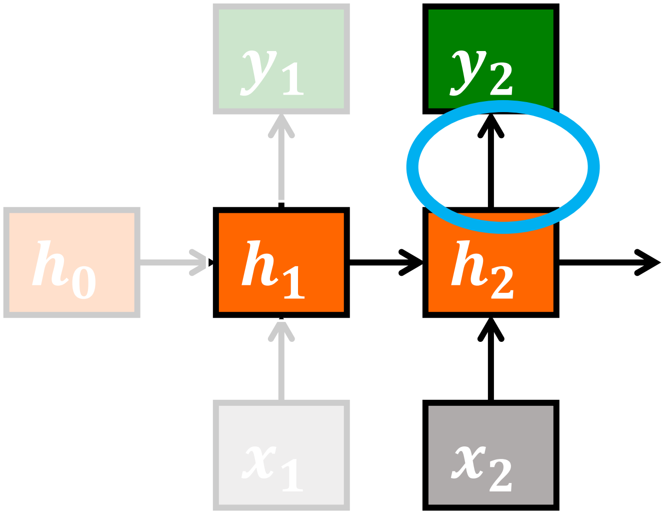
Simple Recurrent Neural Network

- Hidden state → used to predict the output
  - Computed from next input and previous hidden state
- Three weight matrices
  - $h_t = f(x^T W_x + h_{t-1}^T \boxed{W_h} + b_h)$
  - $y_t = f\left(h_t^T W_y\right)$



Simple Recurrent Neural Network

- Hidden state → used to predict the output
  - Computed from next input and previous hidden state
- Three weight matrices
  - $h_t = f(x^T W_x + h_{t-1}^T W_h + b_h)$
  - $y_t = f\left(h_t^T W_y\right)$



Simple Recurrent Neural Network

- Item subject to user interaction as **1-of-N** coded vector

- Example:
  - Output: item 4

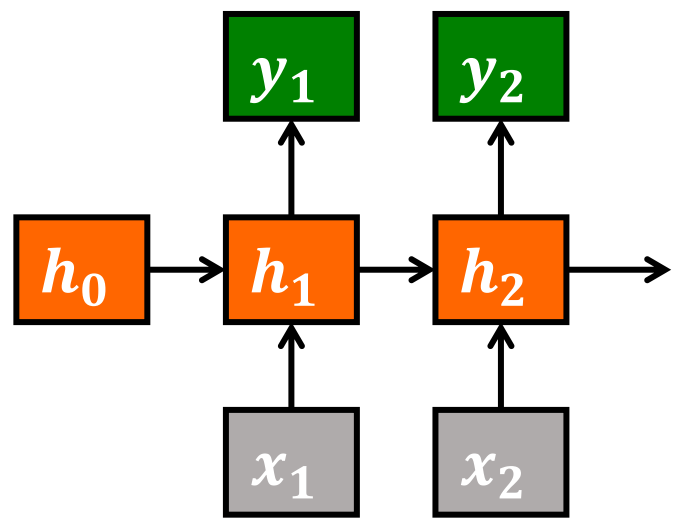
y_2 =

0	0	0	1	0

  - Input: item 3

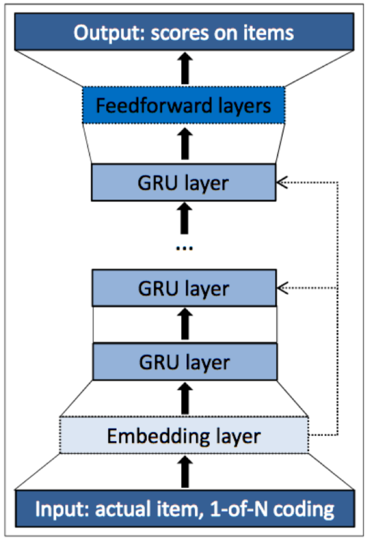
x_2 =

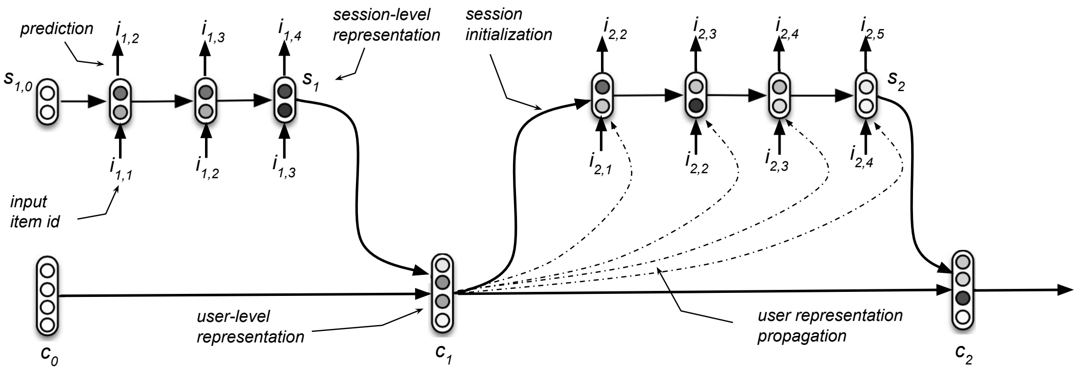
0	0	1	0	0



RNNs for Session-based Recommendation

- GRU4Rec/GRU4Rec+ [Hidasi, 2018]
  - Gated Recurrent Units (GRU), 1-hot item encoding, parallel mini-batching
  - Optimized **negative item sampling** and **ranking loss function**
- HGRURec [Quadrana, 2017]
  - Hierarchical RNN, **long+short-term** user profiling, **personalized** in-session recommendations

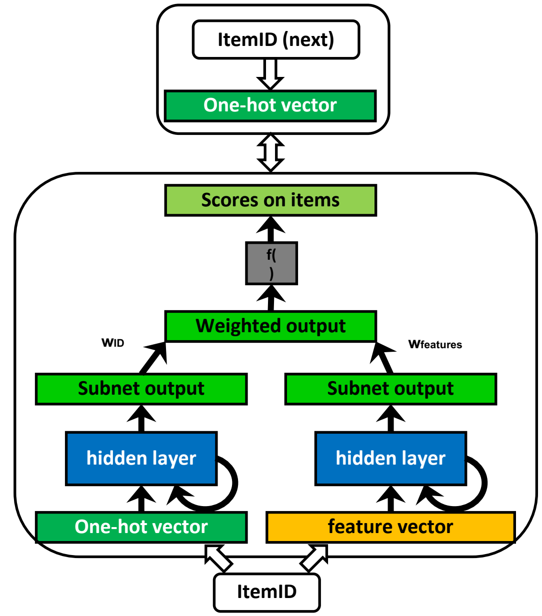




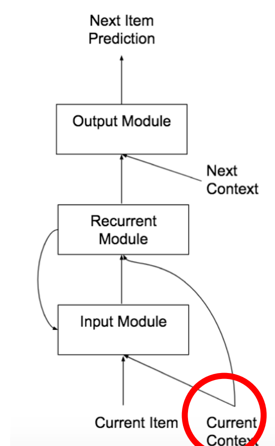
[Quadrana et al.] **Personalizing session-based recommendations with hierarchical recurrent neural networks**. RecSys '17
[Hidasi and Karatzoglou] **Recurrent Neural Networks with Top-k Gains for Session-based Recommendations**. CIKM 2018

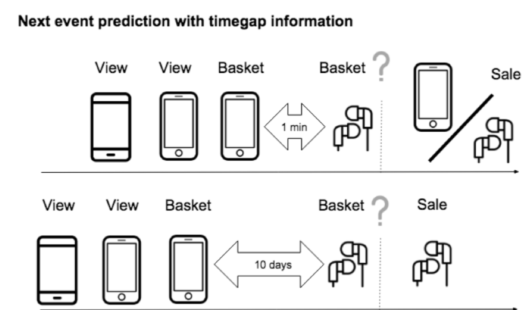
RNNs for Session-based Recommendation

- Parallel RNNs [Hidasi, 2016]
  - Jointly modelling **item features** and **identifiers**
  - **Alternated training** procedure to learn robust feature representations
  - Experiments with product **images** and **descriptions**



- Contextual Sequence Modeling [Smirnova, 2016]
  - Temporal ctx, actic $[x_t; c_t]$:, etc.
  - Concatenation $x_t \odot C c_t$
  - Mult. interaction $[x_t \odot C c_t; c_t]$
  - Combined

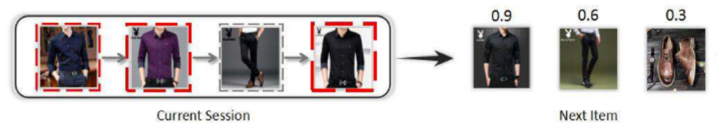


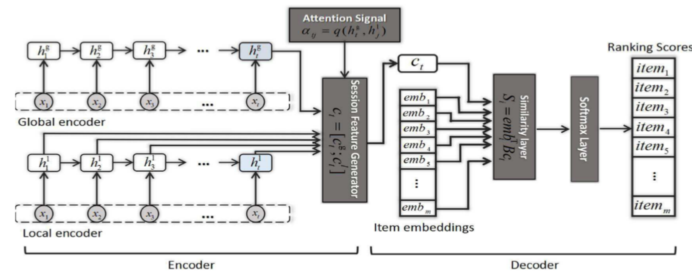


[Liu et al.] **Predicting the next location: A recurrent model with spatial and temporal contexts**. AAAI '16
[Smirnova and Vasile] **Contextual RNNs for Recommendation**. DLRS 2017

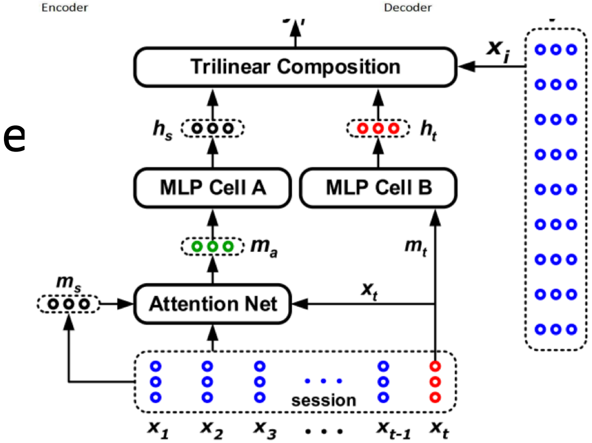
RNNs for Session-based Recommendation

- Neural Attention Networks [Liu, 2017]
  - Standard RNN → **global** session repr.
  - Attention RNN → **main purpose** of the session
  - Combine local/global features with bi-linear matching scheme





- Attentive memory networks [Liu, 2018]
  - Replace RNN with **attention/memory module**
  - Memory retains short-term user interests
  - **Weighted attention** to memory
  - Only Feed-forward networks!

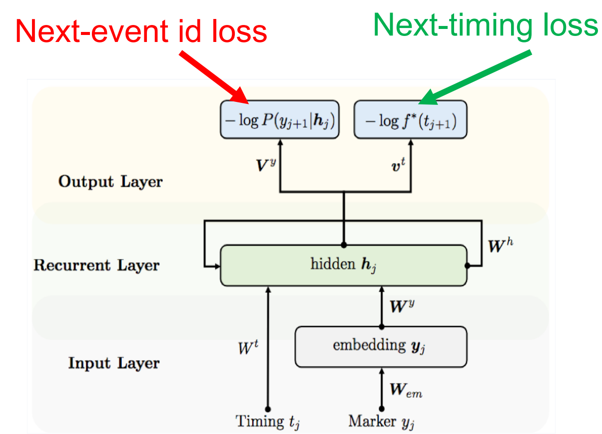


[Li et al.] **Neural Attentive Session-based Recommendation**. CIKM 2017

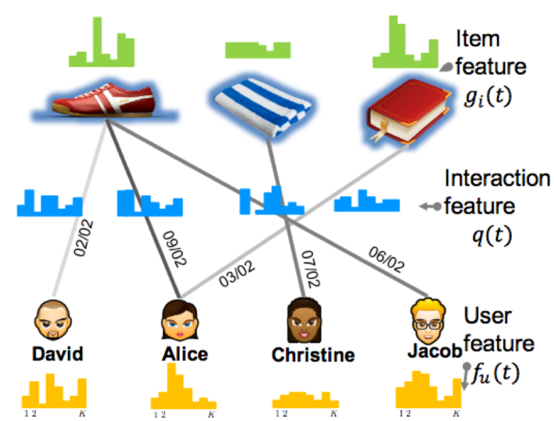
[Liu et al.] **STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation**. KDD 2018

Other RNN Applications

- Recurrent Marked Temporal Point Process (RNN) [Du, 2016]
  - Joint learning **next target item** and **its timing** with a combination of Hawkes process + RNNs
  - Predict **when and what** event will likely occur next



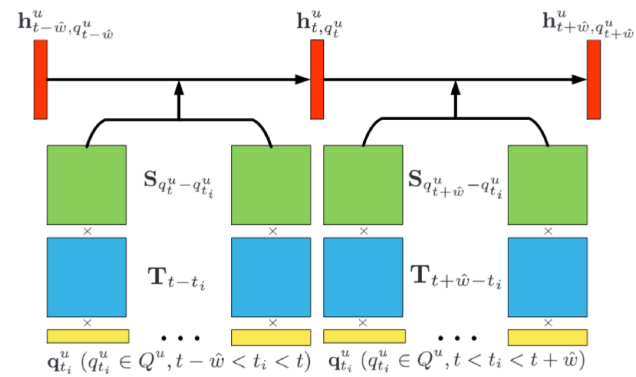
- Deep Co-evolutionary networks [Dai, 2016]
  - Jointly model **user-item feature co-evolution** and **influence each-other** over time



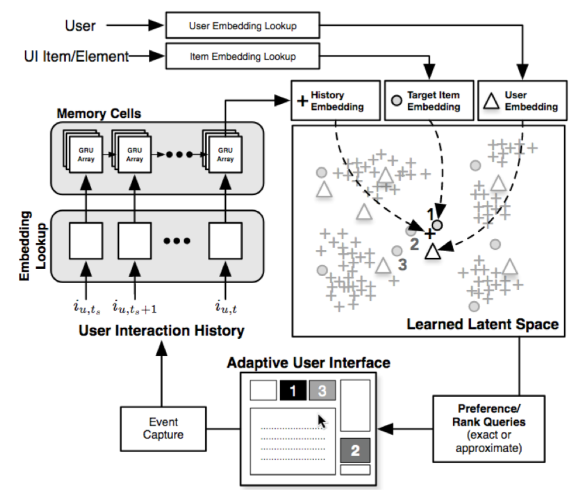
[Du et al.] **Recurrent Marked Temporal Point Processes: Embedding Event History to Vector**. KDD 2016
[Dai et al.] **Deep Coevolutionary Network: Embedding User and Item Features for Recommendation**. DLRS 2016

Other RNN Applications

- Spatio-Temporal RNN (ST-RNN) [Liu, 2016]
  - Next-location prediction with custom Spatial and Temporal hidden-to-hidden transition matrices



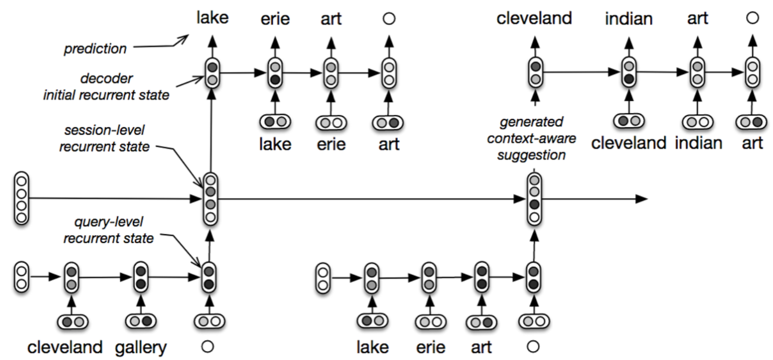
- Sequential recommendation for dynamic adaptation of user interfaces [Sho, 2017]



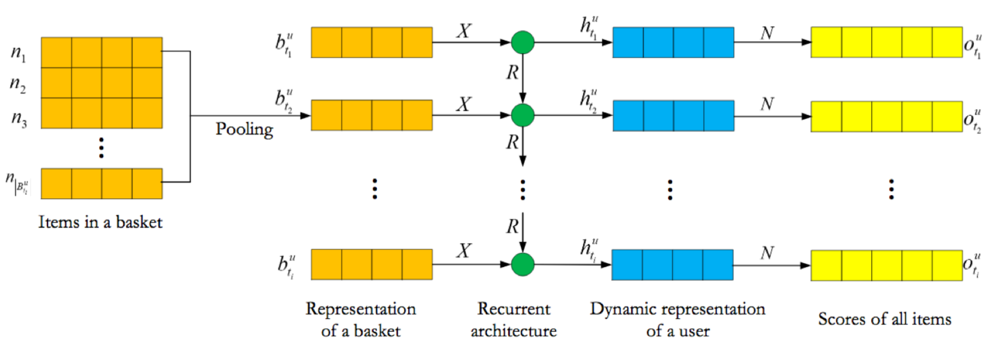
[Liu et al.] **Predicting the next location: A recurrent model with spatial and temporal contexts**. AAAI '16, 2016.
[Soh et al.] **Deep sequential recommendation for personalized adaptive user interfaces**. IUI '17

Other RNN Applications

- Generative context-aware next-query recommendation [Sordoni, 2015]
  - Hierarchical RNN to model query-level + session-level features



- Dynamic REcurrent bAsket Model (DREAM) [Yu, 2016]
  - Pooling to generate basket representations



[Sordoni et al.] **A hierarchical recurrent encoder-decoder for generative context-aware query suggestion**. CIKM ’15
[Yu et al.] **A dynamic recurrent model for next basket recommendati**on. SIGIR ’16

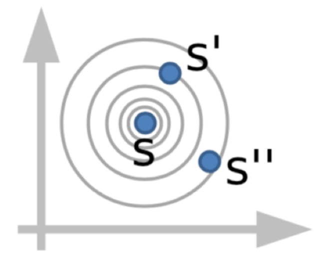
Distributed Item Representations

- Dense, lower-dimensional representations of items
  - derived from sequences of events
  - preserve the sequential relationships between items
- Similar to latent factor models
  - “similar” items are projected to similar vectors
  - every item is associated with a real-valued embedding vector
  - its projection into a lower-dimensional space
  - certain item transition properties are preserved
    - e.g., co-occurrence of items in similar contexts
- Different approaches are possible
  - Latent Markov Embedding (LME)
  - Prod2Vec

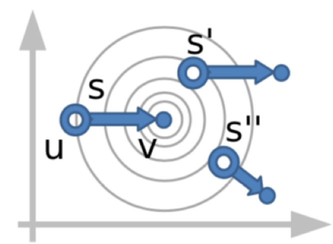
Latent Markov Embeddings

- Transition probability related with the **Euclidean distance** between the **embeddings** of **subsequent** items

$$\Pr(p^{[i]} | p^{[i-1]}) = \frac{e^{\boxed{-||X(p^{[i]}) - X(p^{[i-1]})||_2^2}}}{\sum_{j=1}^{|S|} e^{-||X(s_j) - X(p^{[i-1]})||_2^2}}$$



  - Symmetric transitions → **single-point model**
  - Asymmetric transitions → **dual-point model**
  - MLE over the existing sequences
  - Recommendation through **sampling**



- Applications:
  - Playlist generation
  - POI recommendation

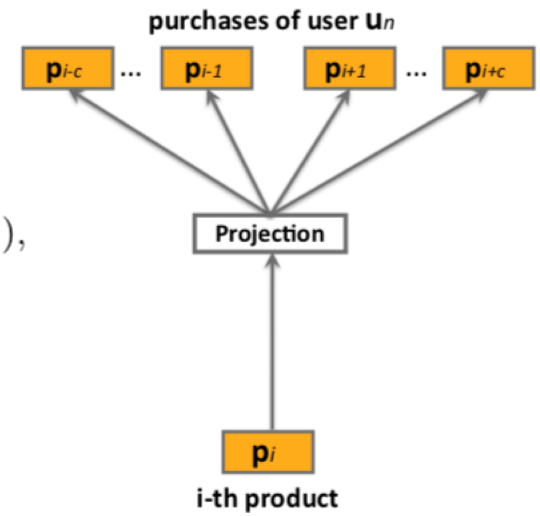
[Chen et al.] **Playlist prediction via metric embedding**. KDD '12
[Chen et al.] **Multi-space probabilistic sequence modeling**. KDD '13
[Feng et al.] **Personalized ranking metric embedding for next new POI recommendation**. IJCAI '15
[Wu et al.] **Personalized next-song recommendation in online karaokes**. RecSys '13

Prod2Vec

- Transition probability related to the **dot product** between the **embeddings** of items within the **context window**

$$\mathbb{P}(p_{i+j}|p_i) = \frac{\exp\boxed{(\mathbf{v}_{p_i}^\top \mathbf{v}'_{p_{i+j}})}}{\sum_{p=1}^{P} \exp(\mathbf{v}_{p_i}^\top \mathbf{v}'_p)},$$

$$\mathcal{L} = \sum_{s \in \mathcal{S}} \sum_{p_i \in s} \sum_{-c \le j \le c, j \ne 0} \log \mathbb{P}(p_{i+j}|p_i),$$

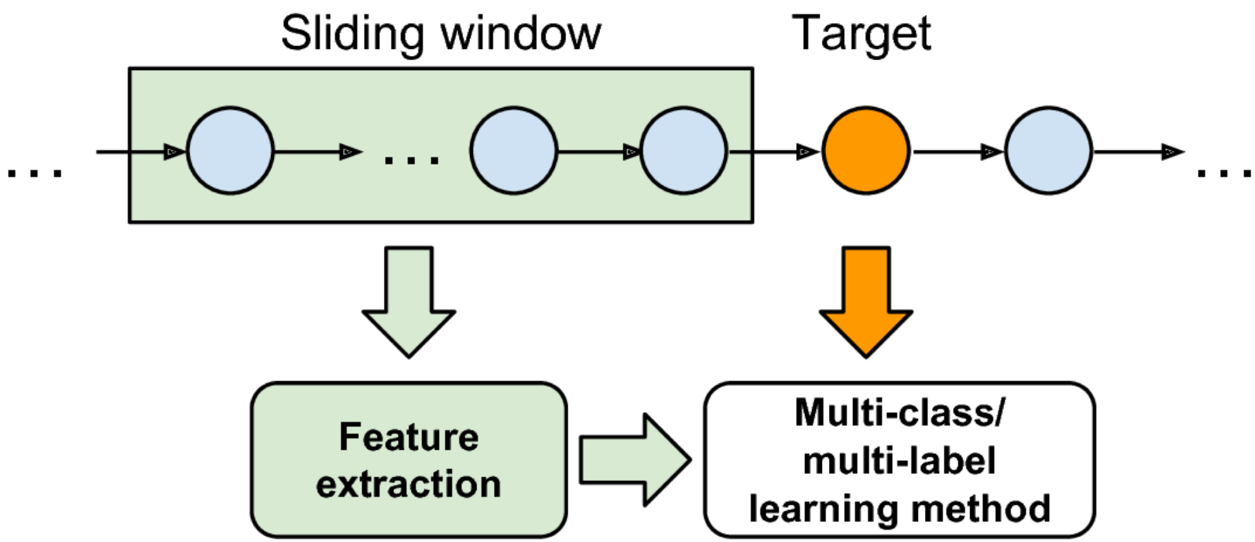


  - MLE + **Skip-gram**
  - Recommendation with (decayed) **KNN**
- Variants
  - Bagged-prod2vec [Grbovic, 2015]
  - Meta-prod2vec [Vasile, 2016]
  - User representation with paragraph vectors [Tagami,2015][Grbovic,2015]
- Applications
  - E-commerce, music, pretrained embeddings for other sequence models

[Grbovic et al.] **E-commerce in your inbox: Product recommendations at scale**. KDD '15
[Tagami et al.] **Modeling user activities on the web using paragraph vector**. WWW '15
[Vasile et al.] **Meta-prod2vec: Product embeddings using side-information for recommendation**. RecSys '16

Supervised Learning w/ Sliding Windows

- Frame sequential recommendation as a classification problem



[Baeza-Yates et al.] **Predicting the next app that you are going to use**. WSDM '15
[Zimdars et al.] **Using temporal data for making recommendations**. UAI '01

Taxonomy

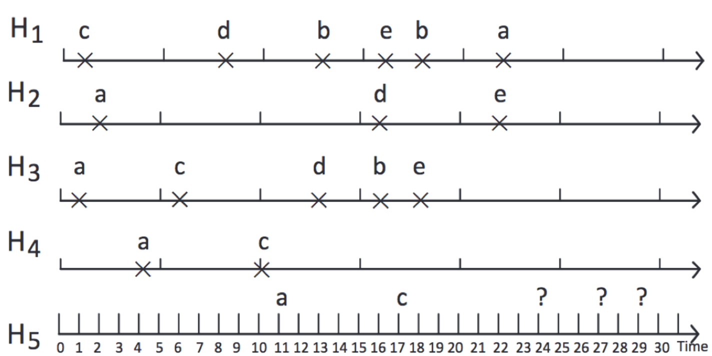
- Sequence Learning
  - Frequent Pattern Mining
  - Sequence Modeling
  - Distributed Item Representations
  - Supervised Models with Sliding Window
- Sequence-aware Matrix Factorization
- Hybrids
  - Factorized Markov Chains
  - LDA/Clustering + sequence learning
- Others
  - Graph-based, Discrete-optimization

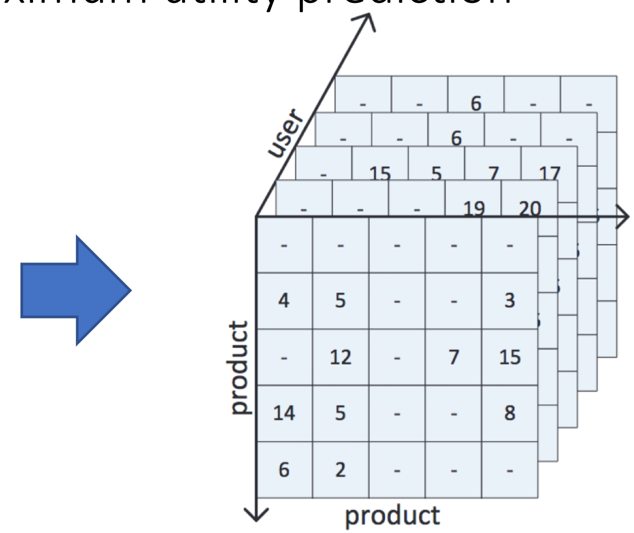
Sequence-aware Matrix Factorization

- Sequence information usually derived from timestamps

  → Time-aware recommender systems

- Purchase sequences → personalized purchase interval prediction
  - Framed as factorized (SVD) maximum utility prediction





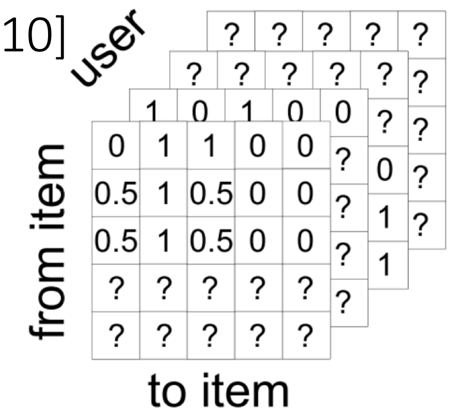
[Zhao et al.] **Increasing temporal diversity with purchase intervals**. SIGIR ’12
[Zhao et al.] **Utilizing purchase intervals in latent clusters for product recommendation**. SNAKDD ’14

Taxonomy

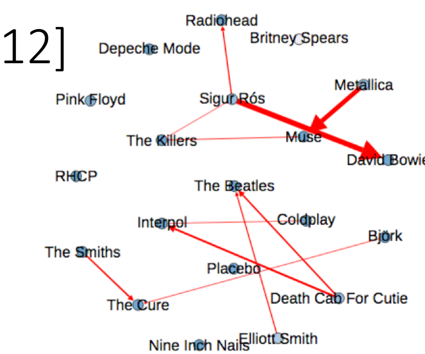
- Sequence Learning
  - Frequent Pattern Mining
  - Sequence Modeling
  - Distributed Item Representations
  - Supervised Models with Sliding Window
- Sequence-aware Matrix Factorization
- Hybrids
  - Factorized Markov Chains
  - LDA/Clustering + sequence learning
- Others
  - Graph-based, Discrete-optimization

Hybrid Methods

- Sequence Learning + Matrix Completion (CF or CBF)
  - Factored Personalized Markov Chains (FPMC) [Rendle, 2010]
    - Transition cube factorization with Pairwise Loss



- Topic modelling/clustering → sequence learning
  - **LDA** on sequences → **FPM** on topic sequences [Hariri, 2012]
  - Clustering over order-1 Markov transition matrices (**behavioural clustering**) → personalized Page-rank [Natarajan, 2013]

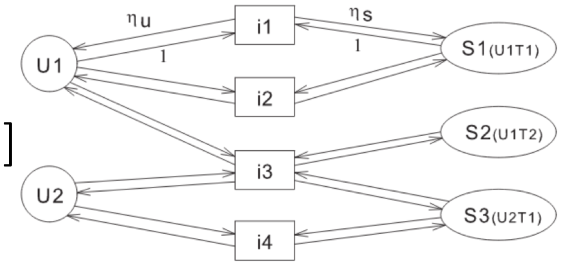


(a) Cluster 1.

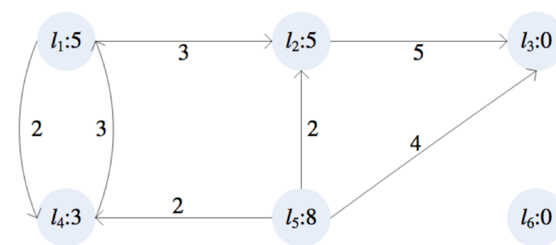
[Rendle et al.] **Factorizing personalized markov chains for next-basket recommendation**. WWW '10
[Hariri et al.] **Context-aware music recommendation based on latent topic sequential patterns**. RecSys '12
[Natarajan et al.] **Which app will you use next?: Collaborative filtering with interactional context**. RecSys '13

Other approaches

- Graph-based
  - Session-based Transition Graph [Xiang, 2010]



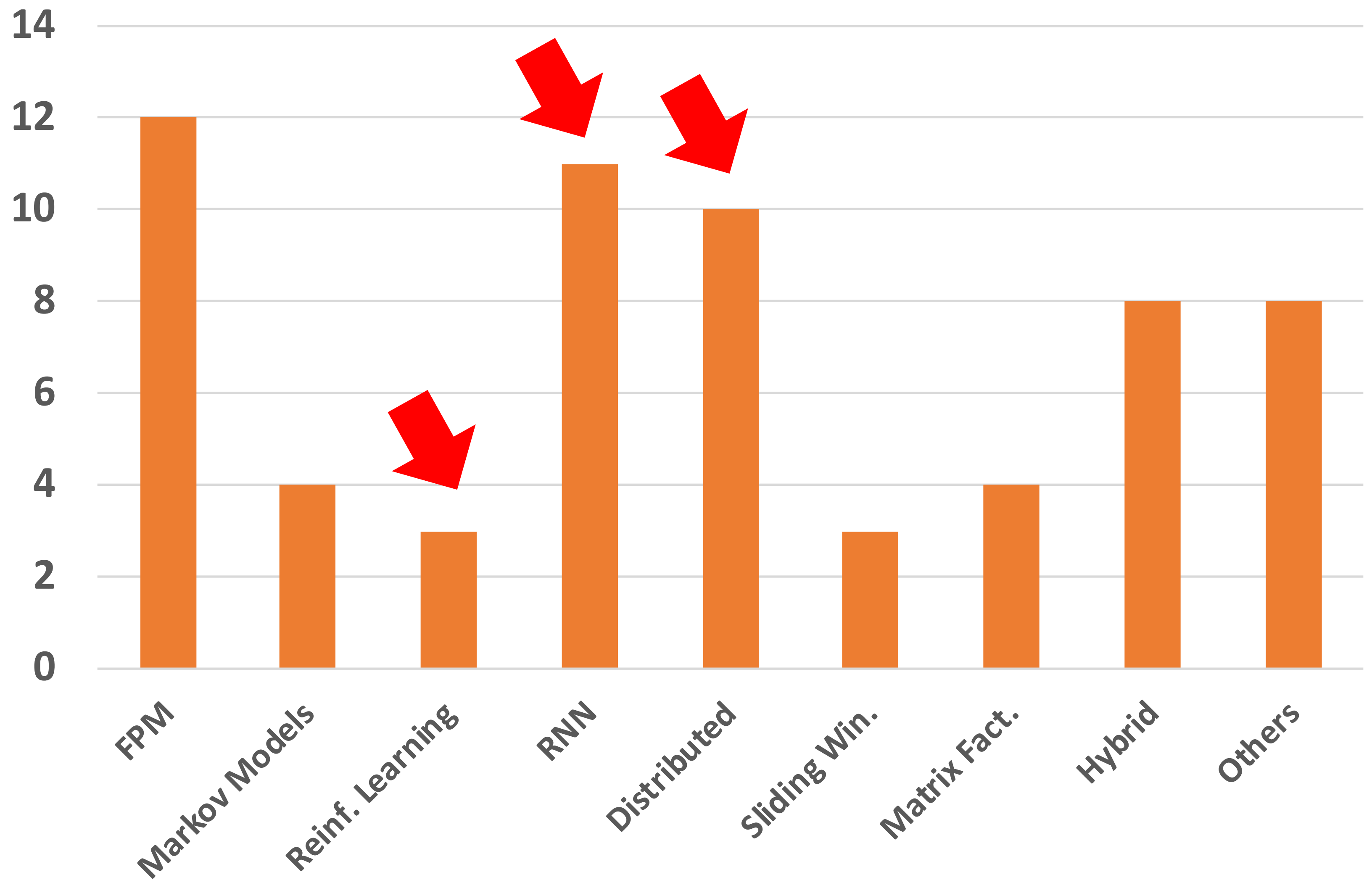
  - Location-location Transition Graph [Zhang, 2014]



- Discrete optimization via constraint satisfaction [Jannach, 2015][Paws, 2006][Xu, 2016]

[Xiang et al.] **Temporal recommendation on graphs via long- and short-term preference fusion**. KDD '10
[Zhang et al.] **Lore: Exploiting sequential influence for location recommendations**. SIGSPATIAL '14
[Jannach et al.] **Beyond "hitting the hits": Generating coherent music playlist continuations with the right tracks**. RecSys '15
[Pauws et al.] **Fast generation of optimal music playlists using local search**. ISMIR '06
[Xu et al.] **Personalized course sequence recommendations**. IEEE Trans. Signal Process. 2016.

Statistics



14
12
10
8
6
4
2
0
FPM
Markov Models
Reinf. Learning
RNN
Distributed
Sliding Win.
Matrix Fact.
Hybrid
Others

Algorithm Summary

Algorithm	Main idea	Pros	Cons
FPM	Discover patterns in user action sequences	• Easy implementation • Explainable results	• Complex configuration • Suffers from data sparsity • Limited scalability
MC	Compute transition probabilities over fixed-length sequences	• Explainable results	• Fixed transition order • Suffers from data sparsity • Limited scalability
VMM	Compute transition probabilities over variable-length sequences	• Variable transition orders • Explainable results	• Suffers from data sparsity
HMM	Model the causal factors in user sequences as transitions between *discrete* hidden states	• Learns from variable-length inputs • Robust to data sparsity	• Limited explainability • Huge number of discrete parameters
RL	Directly maximize the customer and seller reward over time	• Dynamically adapt recommendations to future (unknown) rewards • Under active research	• MDP-based approaches have same issues as MCs • Limited explainability
RNN	Model the causal factors in user sequences with *non-linear* transitions between *continuous* hidden states	• Learns from variable-length inputs • Learns long-term dependencies • Robust to data sparsity • Compact hidden states • Under active research	• Complex configuration • Limited explainability • Benefits not fully clear in some domains
EMB	Embed items into latent spaces that preserves sequential transition properties	• Robust to data sparsity • Visually interpetable embeddings • Under active research	• Need auxiliary methods to make recommendations • Limited explainability
SL	Use supervised learning over features extracted from fixed-size sliding windows over sequences	• Easy implementation • Use off-the-shelf supervised algorithms	• Explainability depends on the chosen supervised method • Feature engineering
MF	Define new inputs and loss functions for MF to handle sequences	• Extensive literature available • Robust to data sparsity	• Non-trivial input and loss design • Concerns regarding scalability

Algorithm: FPM: Frequent Pattern Mining, MC: Markov Chains, VMM: Variable-order Markov Models, HMM: Hidden Markov Models, RL: Reinforcement Learning, RNN: Recurrent Neural Networks, EMB: Distributed Item Representations, SL: Supervised Learning w/ Sliding Windows, MF: Matrix Factorization

Agenda

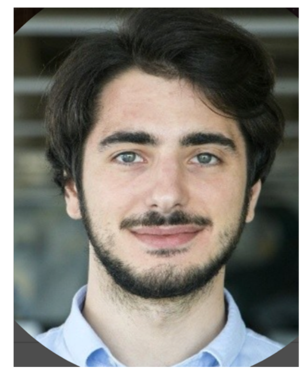
- 14:00 – 14:45 Introduction & Problem Definition (Paolo)
- 14:45 – 15:15 Evaluation (Paolo)
- 15:15 – 15:30 Algorithms I (Massimo)
- 15:30 – 16:00 Coffee break
- 16:00 – 16:45 Algorithms II (Massimo)
- 16:45 – 17:20 Hands-on (Massimo)
- 17:20 – 17:30 Conclusion / Questions

GitHub repo: git.io/fxTtV

Special thanks to...

- Dietmar Jannach

- Umberto Di Fabrizio



- Tutorial Organizers

RecSys

Special issue

UMUAI: Special Issue on Session-based and Sequential Recommender Systems

Abstracts due: November 18, 2018
Paper submission deadline: March 10, 2019

http://tinyurl.com/umuai-si-sessions

Thank you for the attention!
Questions?

Massimo Quadrana

Pandora

mquadrana@pandora.com

@mxqdr

Paolo Cremonesi

Politecnico di Milano

paolo.cremonesi@polimi.it

@crmpla64