Sequential/Session-based Recommendations: Challenges, Approaches, Applications and Opportunities

Presenters: Shoujin Wang, Qi Zhang, Liang Hu, Zhongyuan Lai

Authors: Shoujin Wang, Qi Zhang, Liang Hu, Xiuzhen (Jenny) Zhang, Yan Wang, Charu Aggarwal

Time: Thursday, July 7, 11:00 - 14:30, Online, (CEST Time, UTC +2)

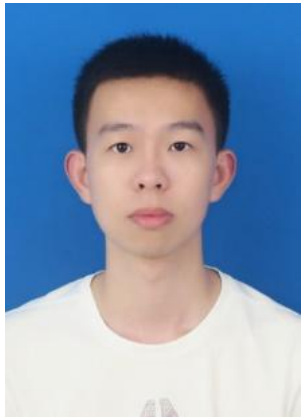
Website: https://neurec22.github.io/SRS&SBRS/ (download the slides)

Zoom link: https://pacifico-meetings.zoom.us/w/86872017204

Acknowledgement

- We thank Mr. Zijian He from Taiyuan University of Technology for his great support to the preparation of the delivery of this tutorial!

- We thank Prof. Xiuzhen Zhang from RMIT University, Prof. Yan Wang from Macquarie University, Dr. Charu Aggarwal from IBM T. J. Watson Research Center for their great support to the proposal of this tutorial!



Dr Shoujin Wang

- Shoujin Wang is a Research Fellow in Data Science at RMIT University, Australia. He is also a Causal Academic at Macquarie University.

- Shoujin holds a PhD in Data Science from the University of Technology Sydney, Australia.

- Shoujin's research interests include data mining, machine learning, user behavior analytics, recommender systems and fake news mitigation. He has more than 30 high-quality publications published at premier international conferences such as The Web Conf., AAAI, IJCAI, ECML and journals such as ACM Computing Surveys.

- Homepage: https://shoujinwang1.github.io/
  Email: shoujin.wang@mq.edu.au

Dr Qi Zhang

- Qi Zhang is the AI scientist in DeepBlue Academy of Sciences, and a Ph.D. candidate in Analytics at University of Technology Sydney, Australia.

- Qi Zhang received his first Ph.D. from the Department of Computer Science and Engineering, Beijing Institute of Technology, China in 2020.

- His research interests include recommender systems, learning to hash, machine learning and general artificial intelligence.

Dr Liang Hu

- Liang Hu is a professor with Tongji University, China and the Chief AI Scientist with Deep Academy of Sciences, Shanghai, China
- His research interests include recommender systems, machine learning, data science and general intelligence.
- He has published more than 40 papers in top-rank international conferences and journals, including WWW, IJCAI, AAAI, ICDM, ICWS, TOIS, IEEE-IS, and etc.

Dr Zhongyuan Lai

- Dr. Zhong Yuan Lai is a researcher at DeepBlue Academy of Sciences.
- Dr. Zhong Yuan Lai obtained his PhD from the University of Bonn, Germany in 2017.
- He was subsequently postdoc researcher at Fudan University before assuming his current position.

Key references

- Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR), 54(7), 1-38. [192 citations]
- Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. Sequential Recommender Systems: Challenges, Progress and Prospects. (2019). In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), 6332-6338. [158 citations]
- Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q. Z., Orgun, M. A., ... & Yu, P. S. (2021). Graph Learning based Recommender Systems: A Review. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), 4644-4652. [45 citations]


2022
Volume 54, Number 7

ACM
Computing Surveys

Article 135 (39 pages) Towards Interconnected Blockchains: A Comprehensive Review of the Role of Interoperability among Disparate Blockchains
A. Lohachab, S. Garg, B. Kang, M. B. Amin, J. Lee, S. Chen, and X. Xu

Article 136 (37 pages) More than Privacy: Adopting Differential Privacy in Game-theoretic Mechanism Design
L. Zhang, T. Zhu, P. Xiong, W. Zhou, and P. S. Yu

Article 137 (26 pages) Affective Automotive User Interfaces–Reviewing the State of Driver Affect Research and Emotion Regulation in the Car
M. Braun, F. Weber, and F. Alt

Article 138 (34 pages) Centralized, Distributed, and Everything in between: Reviewing Access Control Solutions for the IoT
S. Dramé-Maigné, M. Laurent, L. Castillo, and H. Ganem

Article 139 (36 pages) Machine Learning–based Cyber Attacks Targeting on Controlled Information: A Survey
Y. Miao, C. Chen, L. Pan, Q.-L. Han, J. Zhang, and Y. Xiang

Article 140 (36 pages) Text Mining in Cybersecurity: A Systematic Literature Review
L. Ignaczak, G. Goldschmidt, C. Andre da Costa, and R. Da Rosa Righi

Article 141 (37 pages) Deep Learning for Medical Anomaly Detection – A Survey
T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes

continued on back cover

ACM Computing Surveys

Vol. 54 · No. 7 · 2022

Articles 135–156

Association for Computing Machinery

Advancing Computing as a Science & Profession

The goal of this tutorial

- Provides a comprehensive and systematic overview and review of the fields of sequential recommender systems (SRSs) and session-based recommender systems (SBRSs);

- Provide a comprehensive research landscape in the aforementioned areas with an emphasis on key aspects, main challenges, notable progress, applications and future directions in these areas.

Main contributions of this tutorial

- A unified framework to categorize the studies on SBRSs, which can reduce the confusions and inconsistent views;
- A unified problem statement of SBRSs;
- A comprehensive overview of the unique characteristics of session data as well as the challenges of SBRSs incurred by them;
- A systematic classification and comparison of SBRS approaches;
- SBRS applications, algorithms and datasets;
- Open issues and prospects.

Outline

SIGIR 2022
Madrid

Sect. 1 Introduction

30mins, by Shoujin

- Introduction to RS
- Introduction to SBRS
- Classification of SBR
- Sequential RS vs. Session-based RS

Sect. 2 Problem Statement & Challenges

30mins, by Qi + Break (15mins)

- Problem statement
- Characteristics and challenges

Sect. 3 Approaches

80mins, by Zhongyuan & Shoujin + Break (15mins)

- Conventional approaches
- Latent represent-tation
- Deep learning

Sect. 4 Applications & Opportunities

30mins, by Liang + QA (10mins)

- Applications
- Algorithms and datasets
- Future directions
- Conclusions

Outline: Section 1

Section 1

Introduction

- An introduction to recommender systems
- An introduction to session-based recommender systems
- A classification of session-based recommendations
- Sequential recommender systems vs. session-based recommender systems

What are recommender systems

- Recommender systems (push information) are the evolution of information retrieval systems (pull information).

Information Age → **Recommendation Age**

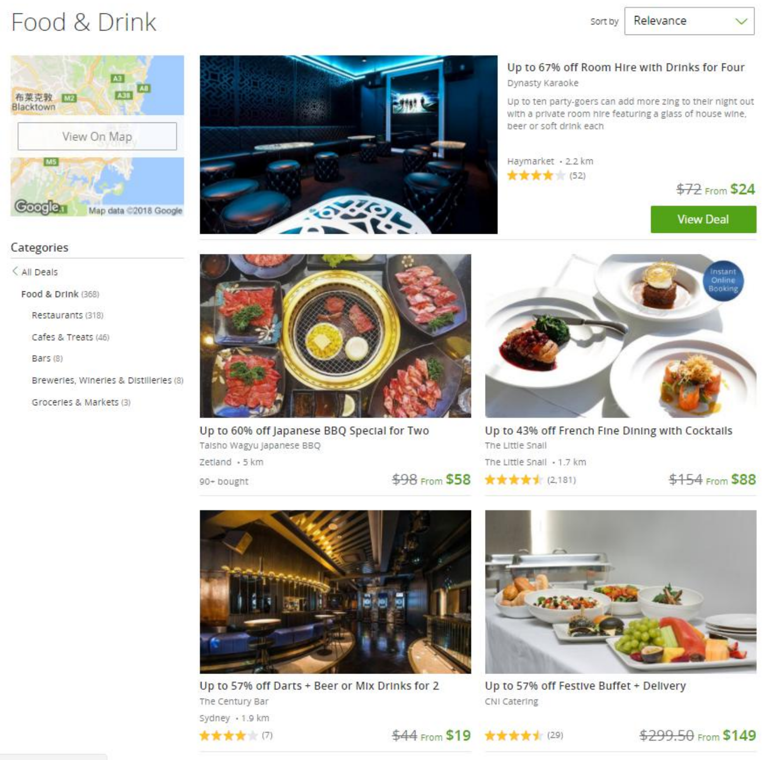
Pull mode (IRS):
Query → Matched Results → Manual Filtering

Push mode (RS):
Potential Requirement → Machine Filtering → Recommendation

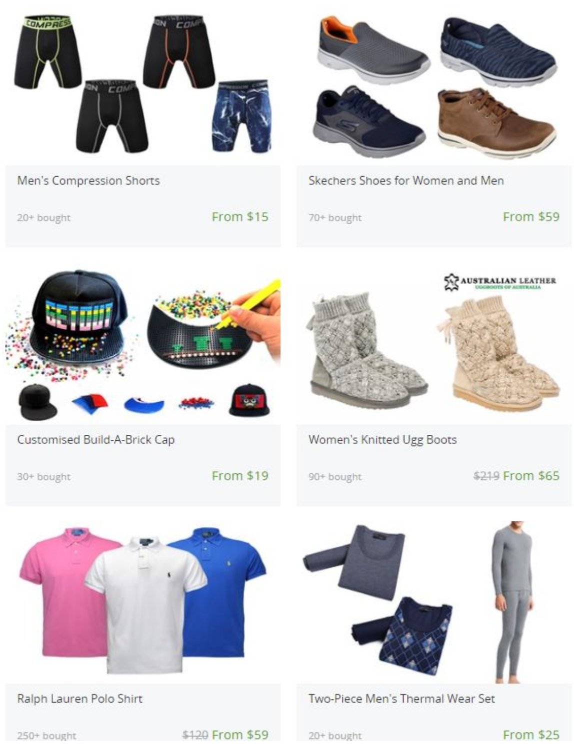
Anderson, C. (2006). *The long tail: Why the future of business is selling less of more*

Recommender systems have occupied our life

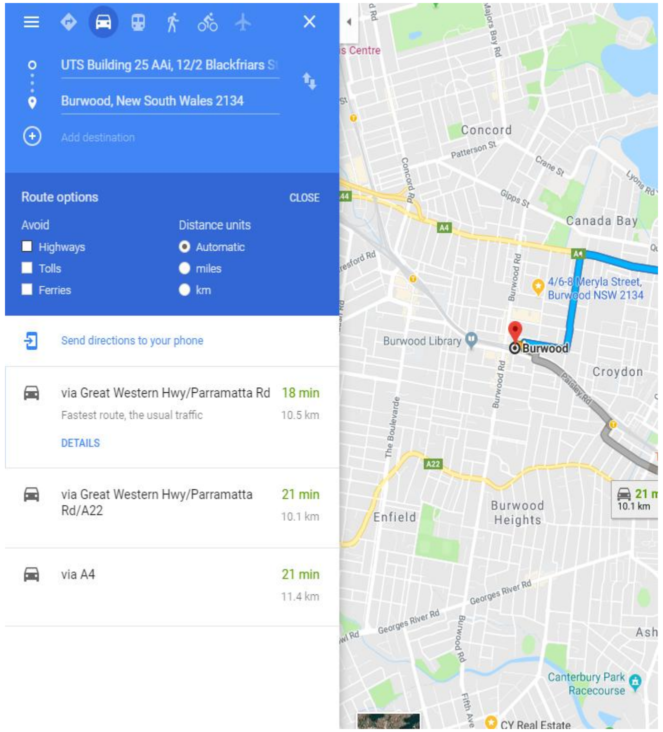
What to eat



Which to dress

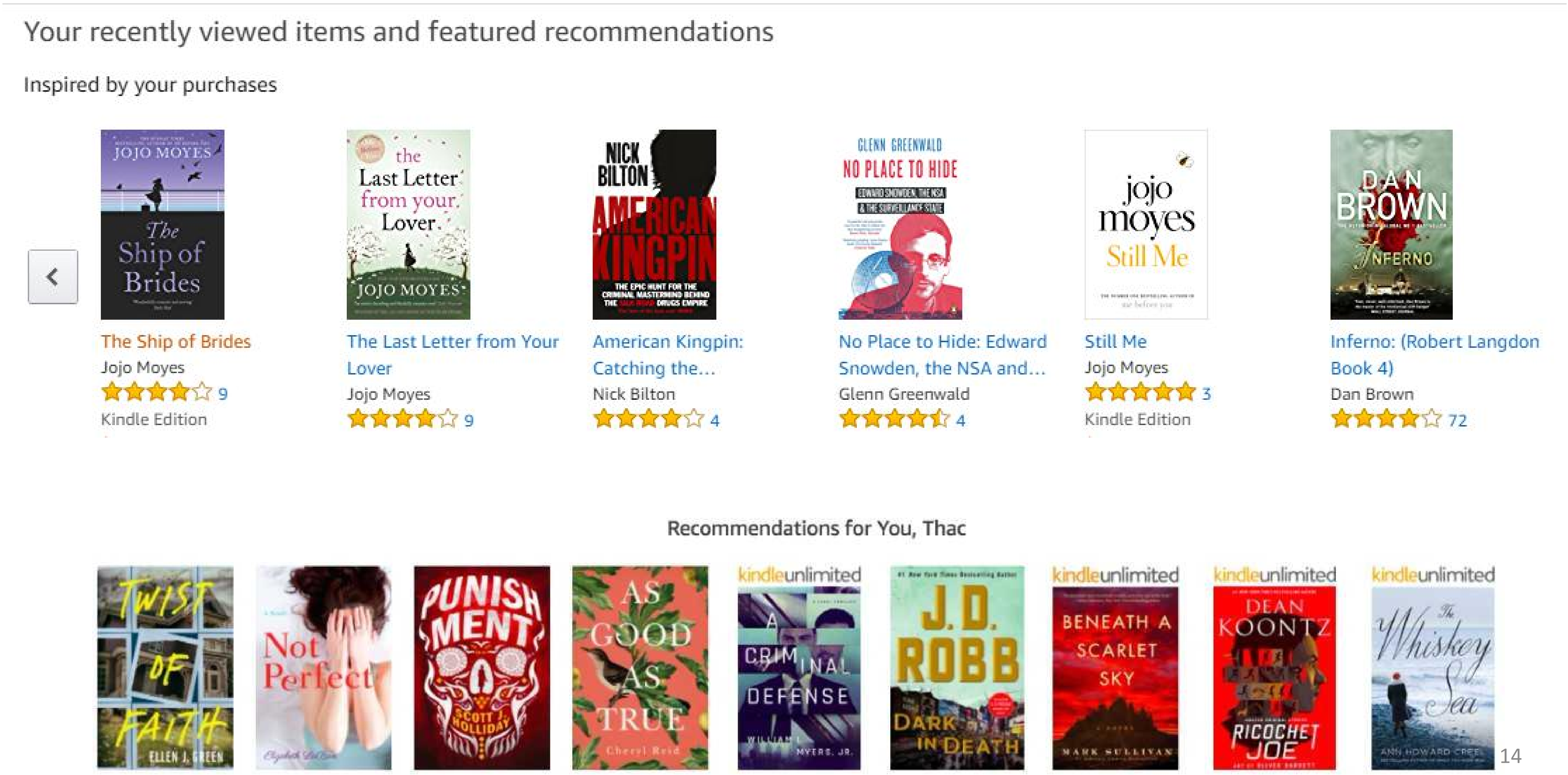


Where to go

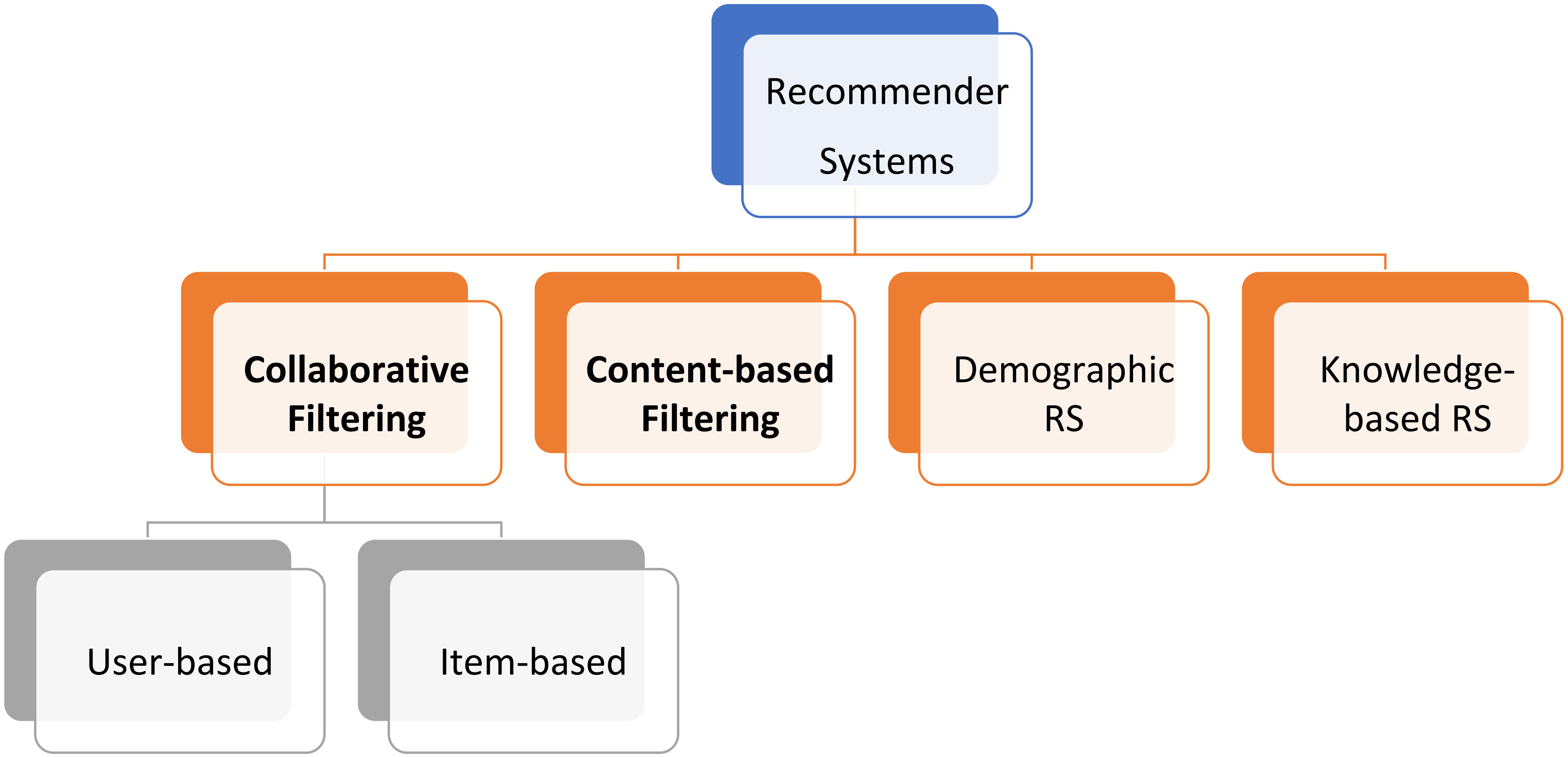


SIGIR 2022 Madrid

Personalization: the core of recommender systems



Classical recommender systems



An general classification of classical recommender systems

Outline: Section 1

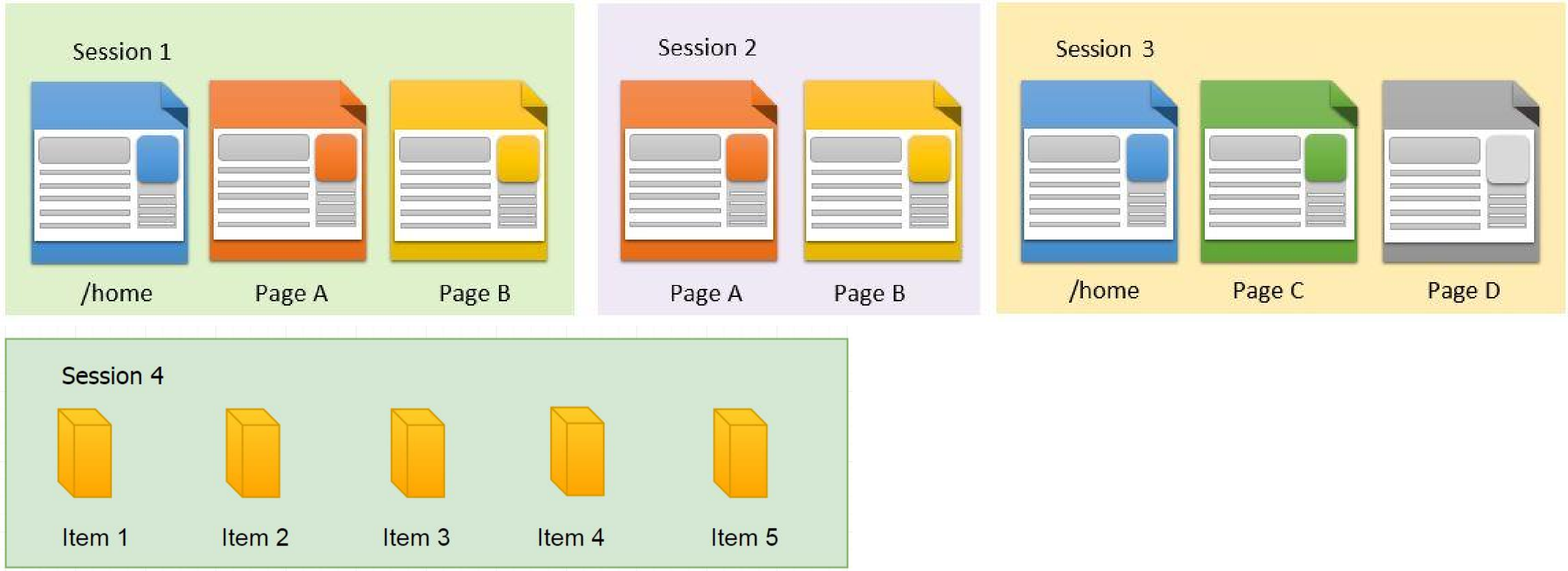
Section 1

Introduction

- An introduction to recommender systems
- An introduction to session-based recommender systems
- A classification of session-based recommendations
- Sequential recommender systems vs. session-based recommender systems

What is a session?

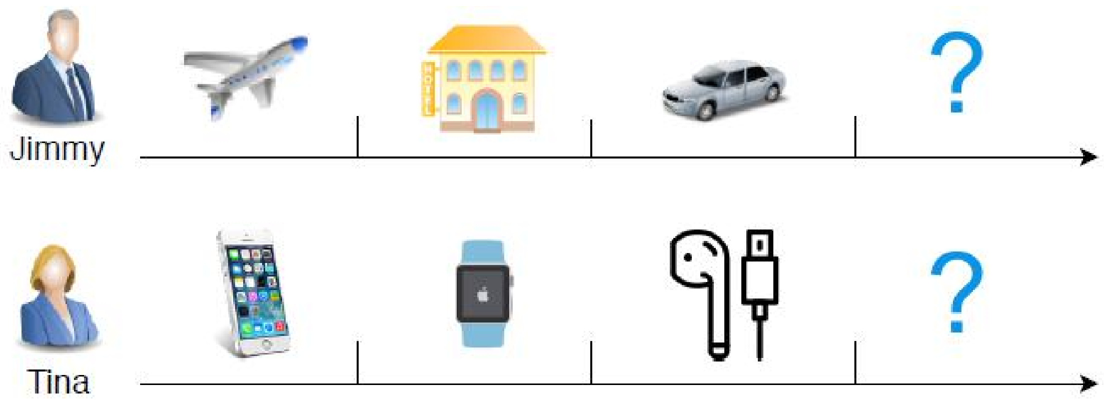
- A session is a list of items, actions or events with a bound, in most cases, there is an order between them.
- There is dependency between the objects within a session.



Examples of four sessions

What is a sequence?

- A sequence is a list of **ordered** symbols, items or actions that happened successively.
- There are **strong sequential** dependencies between the objects in a sequence.

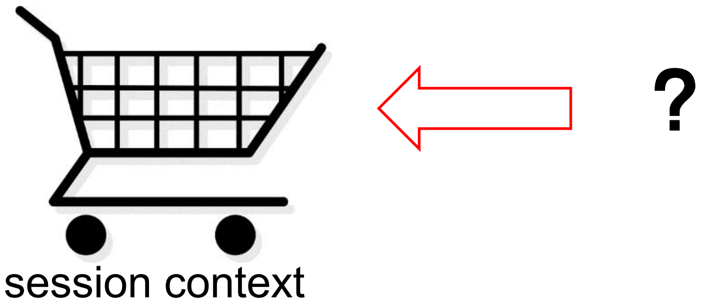


Examples of two sequences

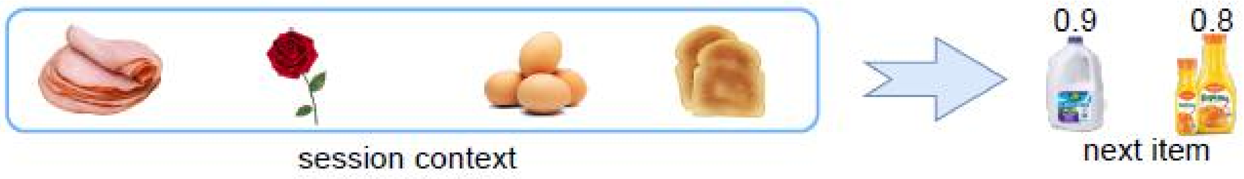
Shoujin Wang, Liang Hu , Yan Wang, et al., Sequential Recommender Systems: Challenges, Progress and Prospects.

What is a session-based recommender system (SBRS)?

Given a list of chosen items as the session context, a session-based recommender system (SBRS) try to predict the next probable item conditioned on it.



A toy example of an SBRS



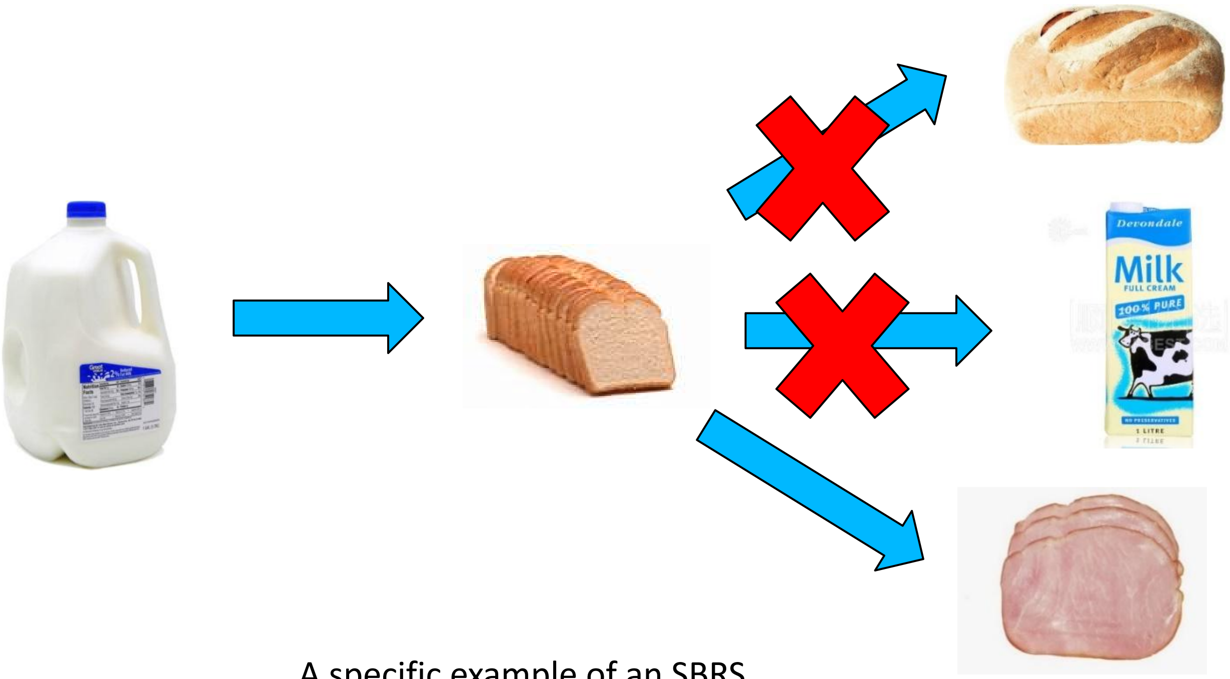
A specific example of an SBRS

Why modeling sessions?

- Deficiencies of non-session based RS
  - RSs built on historical profile are often **repeatedly recommended similar items**.
    - E.g. neighborhood-based methods, matrix factorization methods
  - In most real-world scenarios, we prefer to find items that are **related** to our recent context instead of only **similar** items.
- A system makes more sensible and relevant recommendations if the session context was taken into consideration.

Diversifying recommendations

- Users prefer **more diversified options** than those they have had.
  - It is unlikely that a customer will purchase another loaf of bread if he/she has purchased one, whereas butter or ham may be a more appealing recommendation.



A specific example of an SBRS

Outline: Section 1

Section 1

Introduction

- An introduction to recommender systems
- An introduction to session-based recommender systems
- A classification of session-based recommendations
- Sequential recommender systems vs. session-based recommender systems

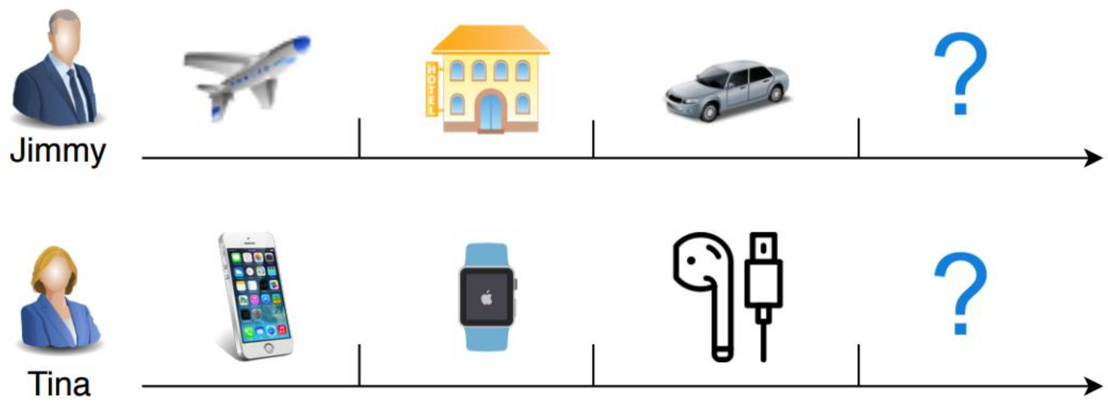
Sub-areas of SBRSs

Session-based Recommendation (SBR)

- **Next interaction recommendtion**
- **Next partial-session recommendtion**
- **Next session recommendtion**

Next interaction recommendation

- Input
  - Mainly known part of the current session
- Output
  - Next interaction(item).
- Typical research topic
  - Next item recommendation
  - Next song/movie recommendation
  - Next POI recommendation



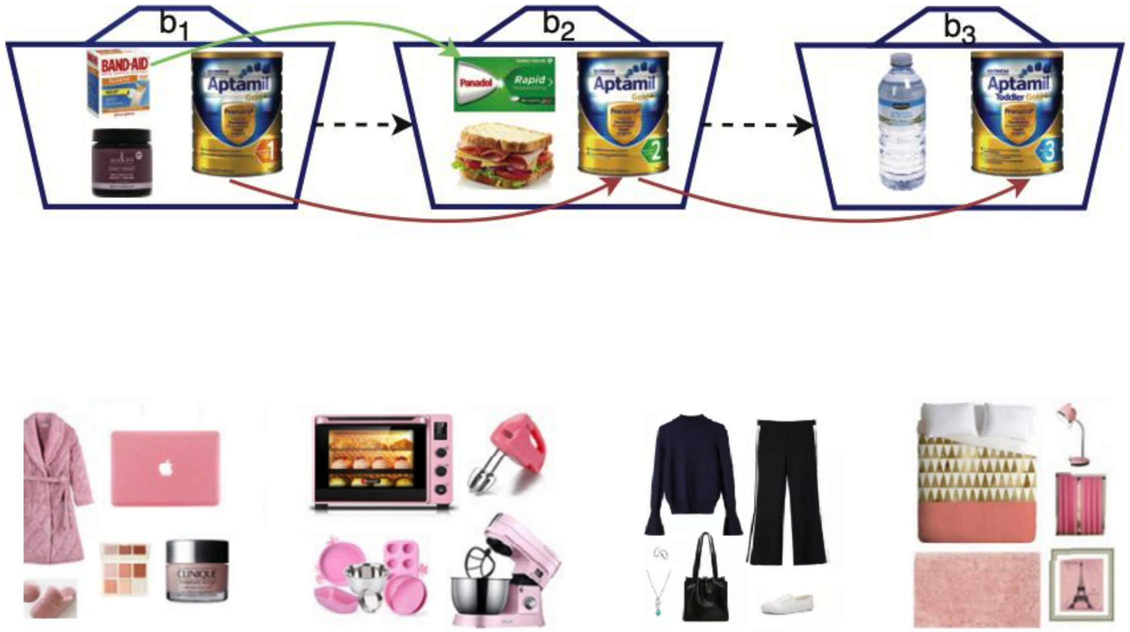
Examples of next interaction recommendation

Next partial-session recommendation

- Input
  - Mainly known part of the current session
- Output
  - Subsequent part of the session
- Typical research topic
  - Next item recommendation
  - Session/Basket completion

Next session recommendation

- Input
  - Historical sessions
- Output
  - Next session
- Typical research topic
  - Next basket recommendation
  - Next bundle recommendation



Examples of next session recommendation

Shoujin Wang, and et al. “Intention Nets: Psychology-Inspired User Choice Behavior Modeling for Next-Basket Prediction .” In: AAAI. 2020. Jinze Bai, and et al. “Personalized Bundle List Recommendation.” In: WWW. 2019.

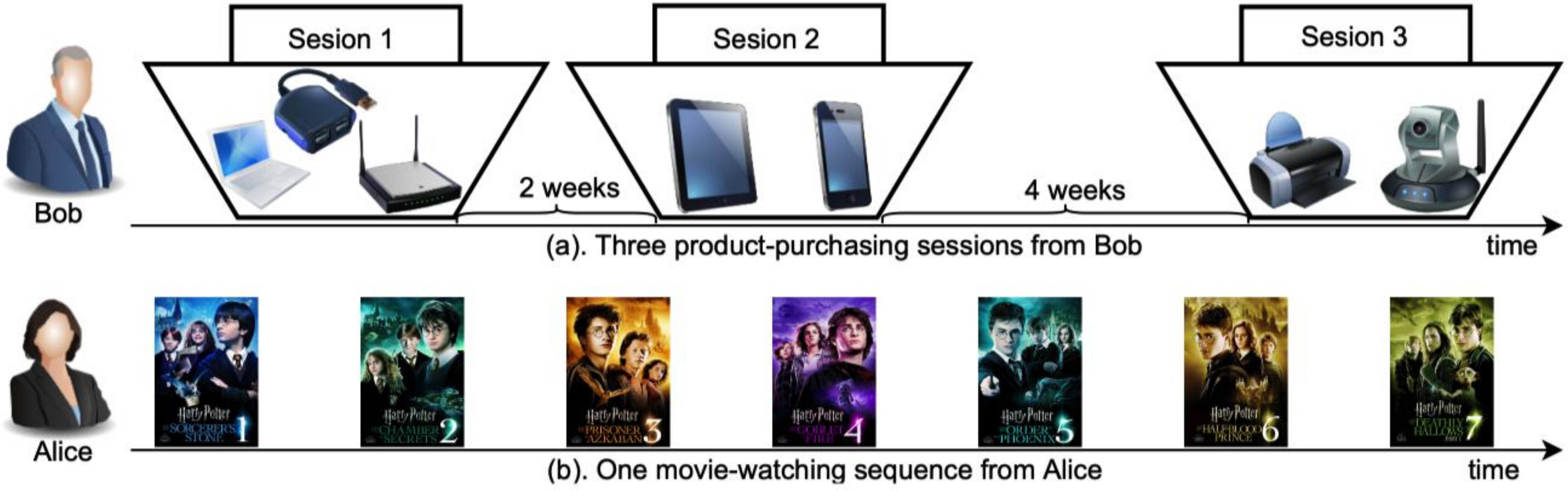
Outline: Section 1

Section 1

Introduction

- An introduction to recommender systems
- An introduction to session-based recommender systems
- A classification of session-based recommendations
- Sequential recommender systems vs. session-based recommender systems

Session data vs. sequence data



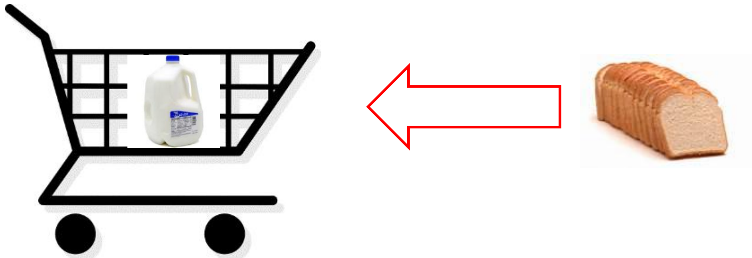
Examples of session data and sequence data

Table 1. A comparison between session data and sequence data

Data type		Boundary	Order	Time interval	Main relations embedded
Session data	Unordered session	Multiple	No	Non-identical	Co-occurrence-based dependencies
	Ordered session	Multiple	Yes	Non-identical	Co-occurrence-based dependencies and sequential dependencies
Sequence data		Single	Yes	Not included	Sequential dependencies

Differences between SBRSs and SRSs

- An SBRS aims to predict either the unknown part (e.g., an item or a batch of items) of a session given the known part, or the future session (e.g., the next-basket) given the historical sessions via learning the intra- or inter-session dependencies based on co-occurrence. The dependencies are not necessarily to be sequential.



session context

A toy example of an SBRS

- An SRS predicts the successive elements given a sequence of historical ones by learning the sequential dependencies among them.

Time line

A toy example of an SRS

Wang S and et al. "A survey on session-based recommender systems". In: CSUR. 2021.

- The end of Section 1!

Outline

Sect. 1 Introduction	Sect. 2 Problem Statement & Challenges	Sect. 3 Approaches	Sect. 4 Applications & Opportunities
30mins, by Shoujin	30mins, by Qi + Break (15mins)	80mins, by Shoujin, Zhongyuan + Break (15mins)	30mins, by Liang + QA (10mins)
• Introduction to RS • Introduction to SBRS • Classification of SBR • Sequential RS vs. Session-based RS	• Problem statement • Characteristics and challenges	• Conventional approaches • Latent representation • Deep learning	• Applications • Algorithms and datasets • Future directions • Conclusions

Outline: Section 2

Section 2

Problem Statement & Challenges

- Problem statement
  - User and User Properties
  - Item and Item Properties
  - Interaction and Interaction Properties
  - Session and Session Properties
  - SBRS Problem
- Data characteristics and challenges

User and user properties

User: a user in an SBRS is the subject who takes actions on items and receives the recommendation results.

User Properties: each user is associated with a unique ID and a set of attributes to describe him/her, e.g., the gender and age.

Bob

← **ID: 77**

Gender	Age	…
male	30	…

Note!

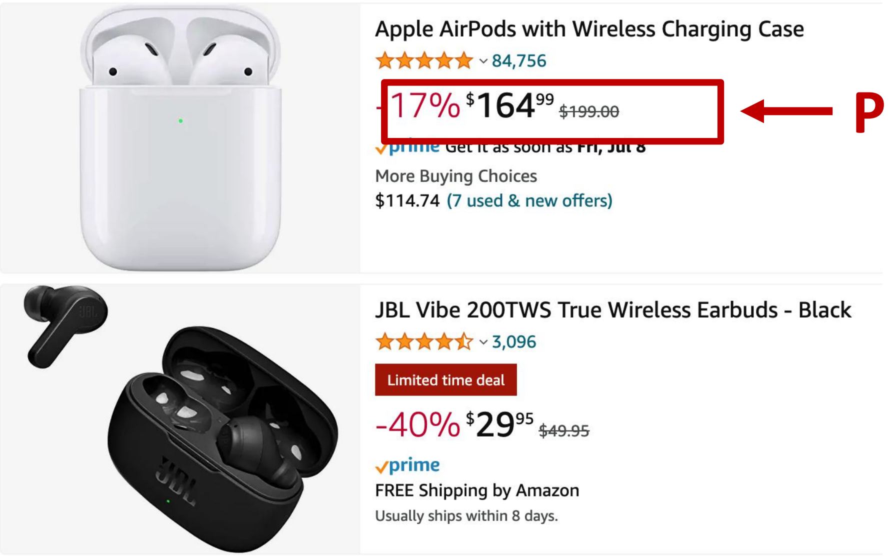
The user information of a session may not be always available.

1. it is not recorded due to the privacy protection.
2. some users do not log in when interacting with online platforms like amazon.com.

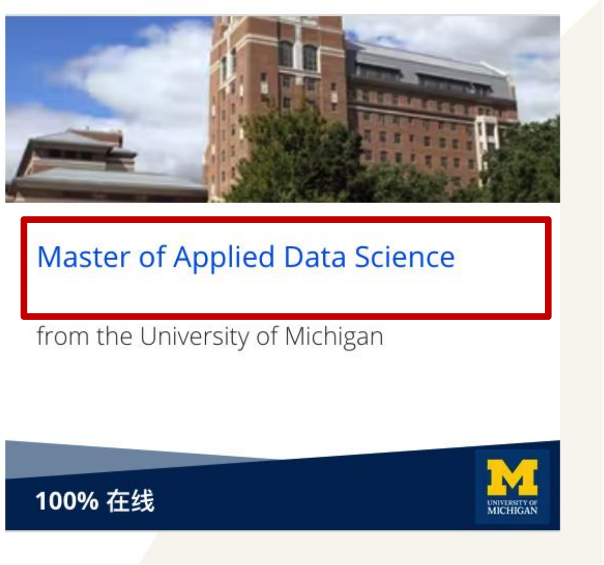
Item and item properties

Item: An item in an SBRS is an entity to be recommended, such as a product, e.g., earphones, or a service, e.g., graduate or undergraduate courses.

Item properties: a set of attributes to provide the description information of the item, such as the price of the earphones and the category of the courses.



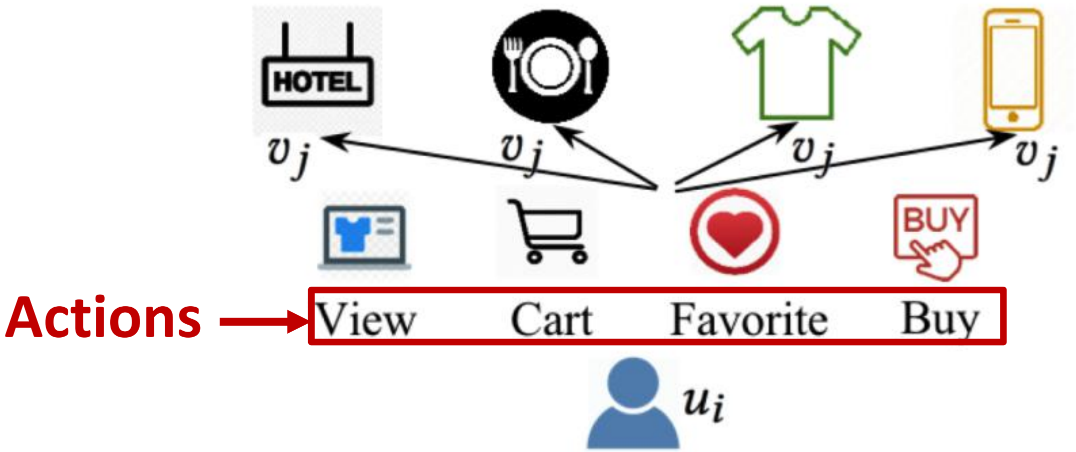
Price



SIGIR 2022 Madrid

Interaction and interaction properties

Action: An action is taken by a user on an item in a session, e.g., view or buy an item.



Interaction is the most basic unit in sessions, consisting of a user, an item and an action

$$o = \langle u, v, a \rangle$$

user �, item �
action � taken by � on
�

- When the user information is not available, the interaction become anonymous

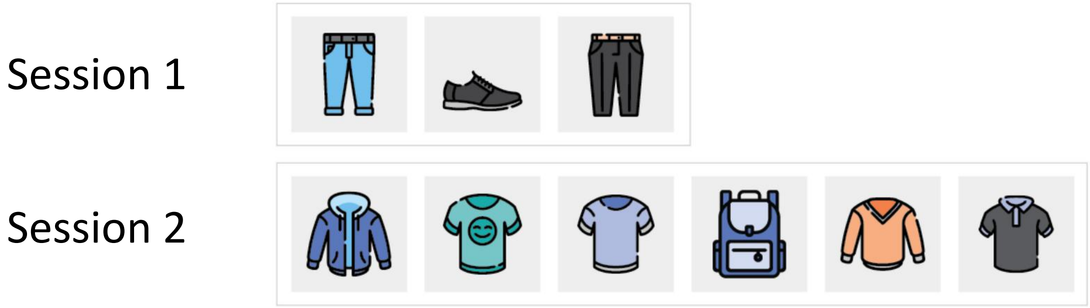
$$o = \langle u, v, a \rangle \quad \Rightarrow \quad o = \langle v, a \rangle$$

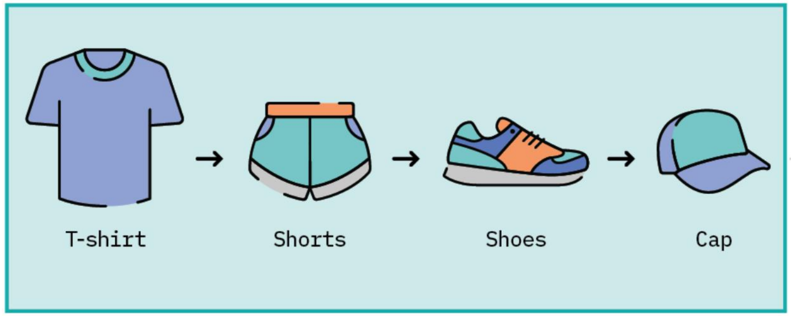
- When there is only one type of actions, e.g., clicks, the interaction � can be further simplified:

$$o = \langle v, a \rangle \quad \Rightarrow \quad o = \langle v \rangle$$

Session and session properties

A session is a non-empty bounded list of interactions generated in a period of continuous time.





In the next part, we will discuss the following five important characteristics of sessions and present their corresponding challenges.

- Length
- Internal Order
- Action Type
- User Information
- Data Structure

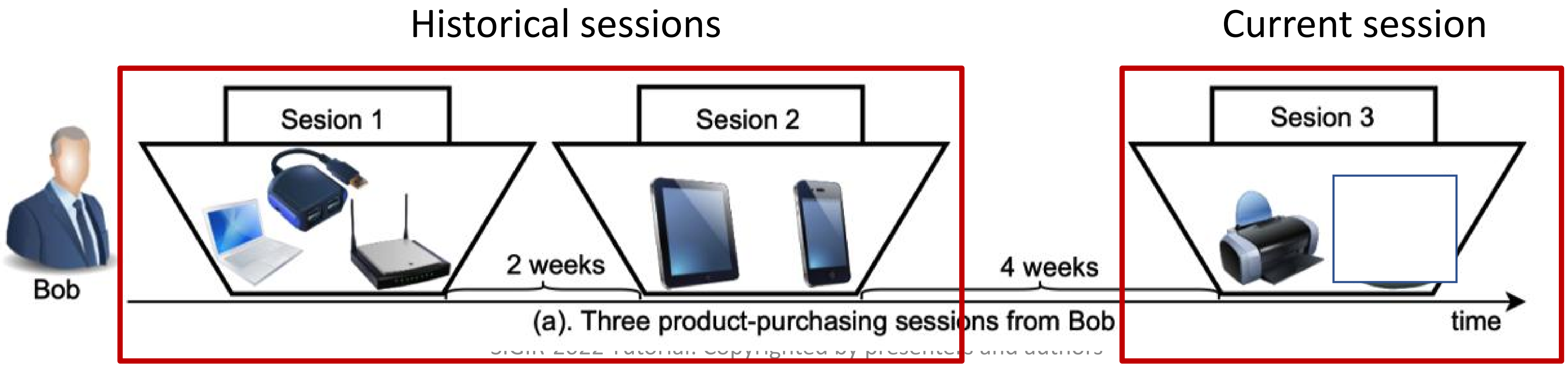
The SBRS Problem

Input

- The basic input of an SBRS is the partially known session information that is used for recommendations.

- According to the specific scenarios, the basic input has three cases:

  1. the known part of the current session
  2. the list of known historical sessions
  3. the combination of the first two



(a). Three product-purchasing sessions from Bob

The SBRS problem (Cont')

Output

- The goal of an SBRS is to make recommendations according to a given session context.

- According to the specific sub-areas, there are three cases for the output
  1. Next interaction recommendation
  2. Next partial-session recommendation
  3. Next session recommendation

The SBRS problem (Cont')

Work mechanism

1. Learn the comprehensive dependencies among interactions within or/and between sessions.
2. Utilize the learned dependencies to guide the prediction of the subsequent interactions or sessions to accomplish the recommendation task.

Problem formalization.

- An SBRS is to select the recommended interaction list $\hat{l} \in L$ by maximizing the utility score conditioned on a given session context �:

$$\hat{l} = arg\ max\ f(c, l), c \in C, l \in L,$$

- �
- � is the session context set.
- $l = \{o_1, ..., o_j, ..., o_n\}$ is a list of � interactions

� is a utility function that

Outline: Section 2

Section 2

Problem Statement & Challenges

- Problem statement
- Data characteristics and challenges
  - Session length
  - Internal order
  - Action type
  - User information
  - Session data structure

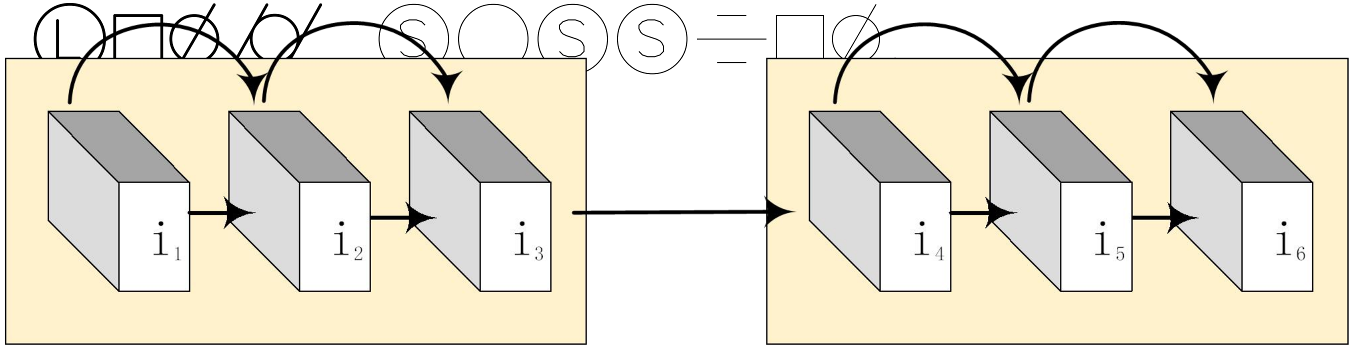
Data characteristics and challenges

Data characteristics

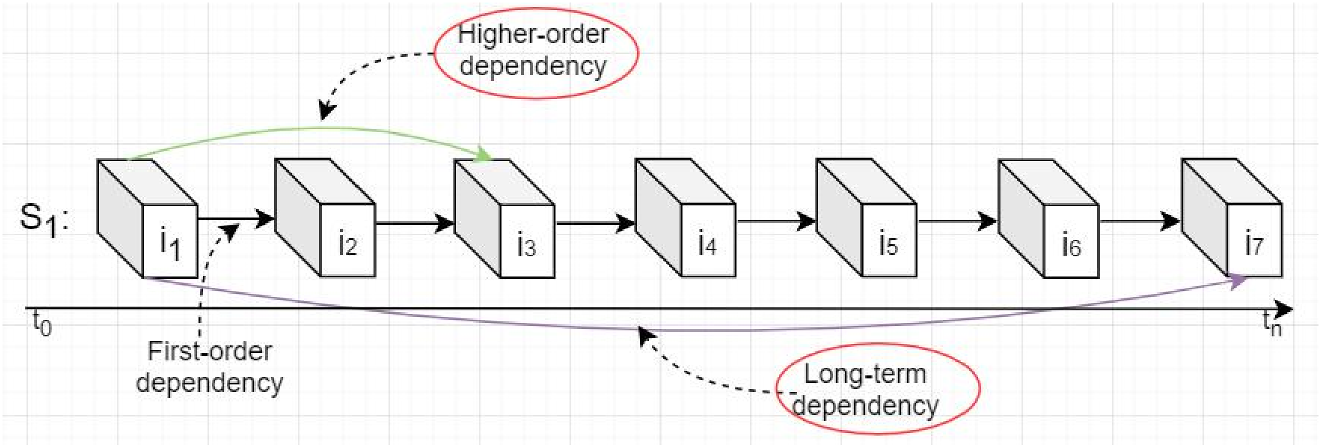
- Length
  - **session length**
- Internal order
  - **the internal order within sessions**
- Action type
  - **the type of actions within sessions**
- User information
  - **user information**
- Data structure
  - **session data structure**

Session length

- The length of a session is defined as the total number of interactions contained in it.



An example of short session



An example of long session

- Quite limited interactions, usually less than 4.
- Leading to limited information available for recommendation.
- **Long-range** dependencies between two interactions that are far from each other in a session.
- **High-order** dependencies across multiple interactions in a session.

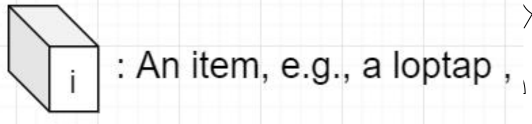
Session length (Cont')

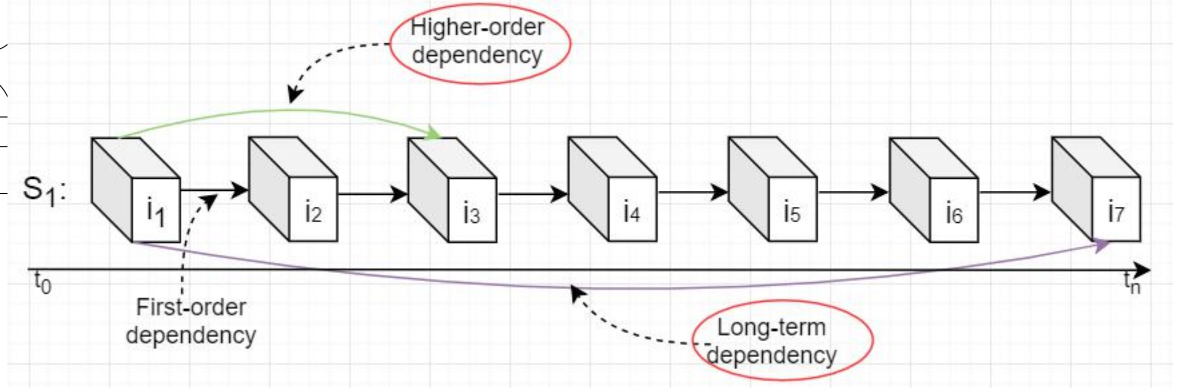
SIGIR 2022 Madrid

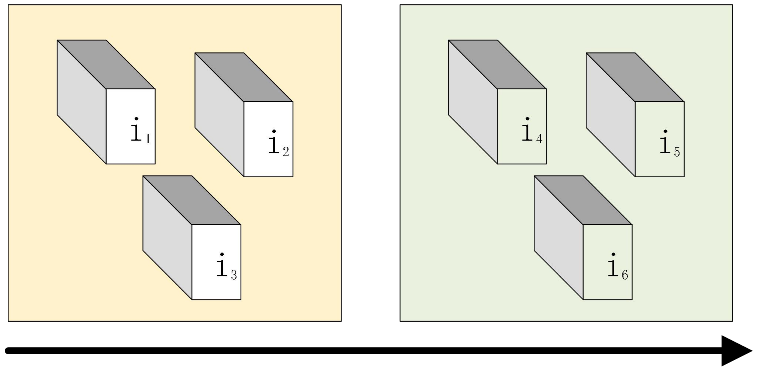
Characteristics	Challenges
Short	How to obtain **enough** dependency information with **limited** interactions.
Long	How to learn **long-range/high-order** dependencies

The internal order

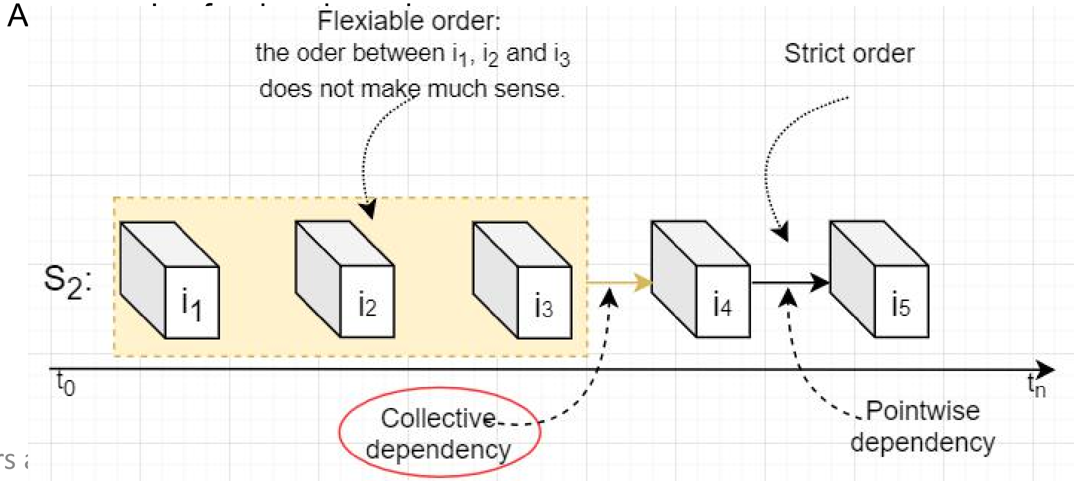
- The internal order of a session refers to the order over interactions within it.







An example of unordered session



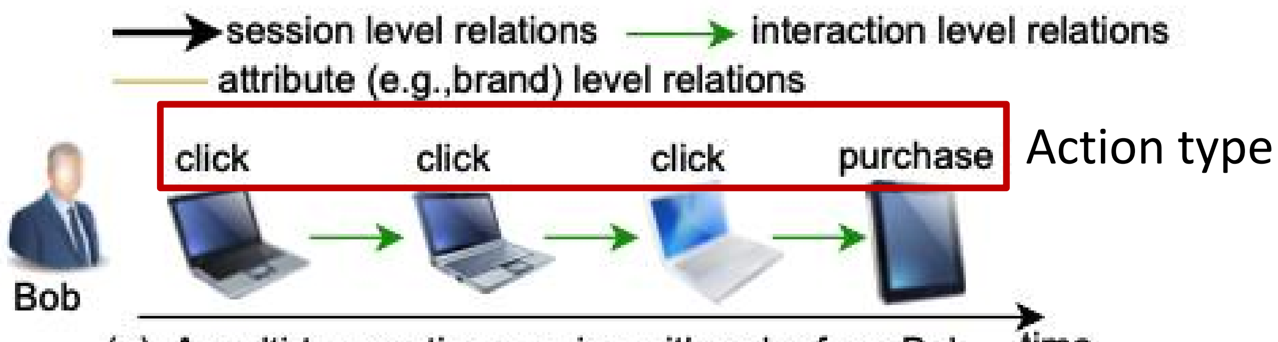
An example of flexibly-ordered session

The internal order (Cont')

Characteristics	Challenges
Unordered	How to learn the relatively **weak and fuzzy** dependencies
Ordered	How to learn the **long-term sequential** dependencies which may decay gradually over time
Flexibly-ordered	How to learn the **complex and mixed** dependencies

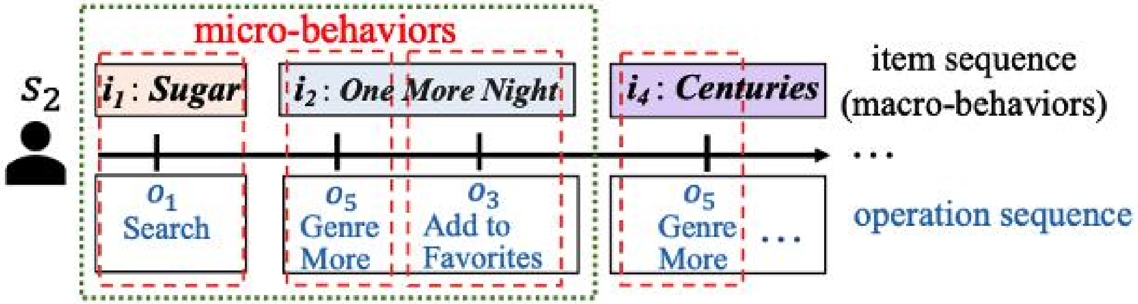
The type of actions

According to the number of action types(. , purchase, click, view, and add to cart) included in a session, sessions can be divided into single-type-action sessions and multi-type-action sessions.



(a). A multi-type-action session with order from Bob

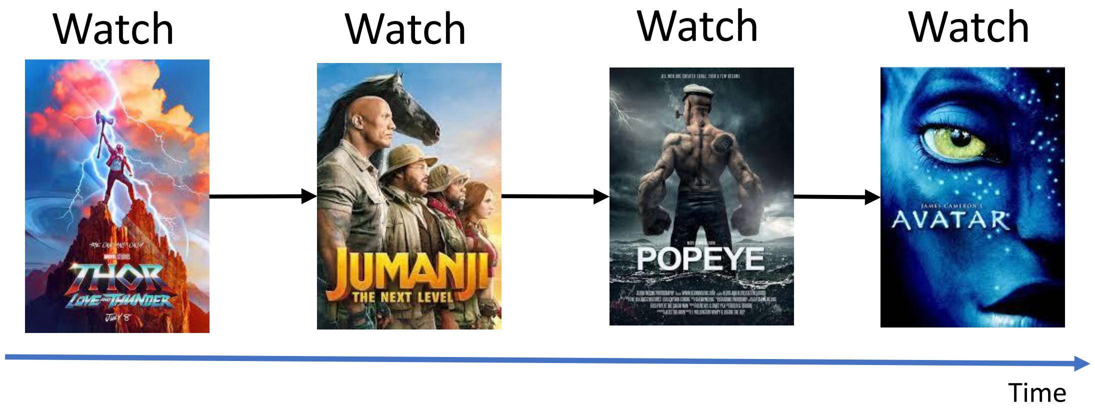
- A multi-type-action session includes more than one types of actions.

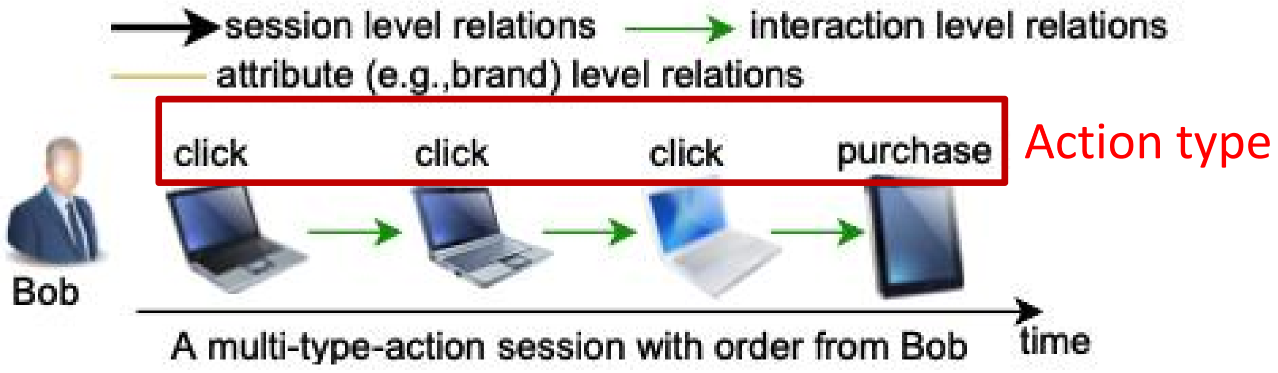


- A multi-type-action session with complex dependencies.

The type of actions

According to the number of action types(e.g., purchase, click, view, and add to cart) included in a session, sessions can be divided into single-type-action sessions and multi-type-action sessions.





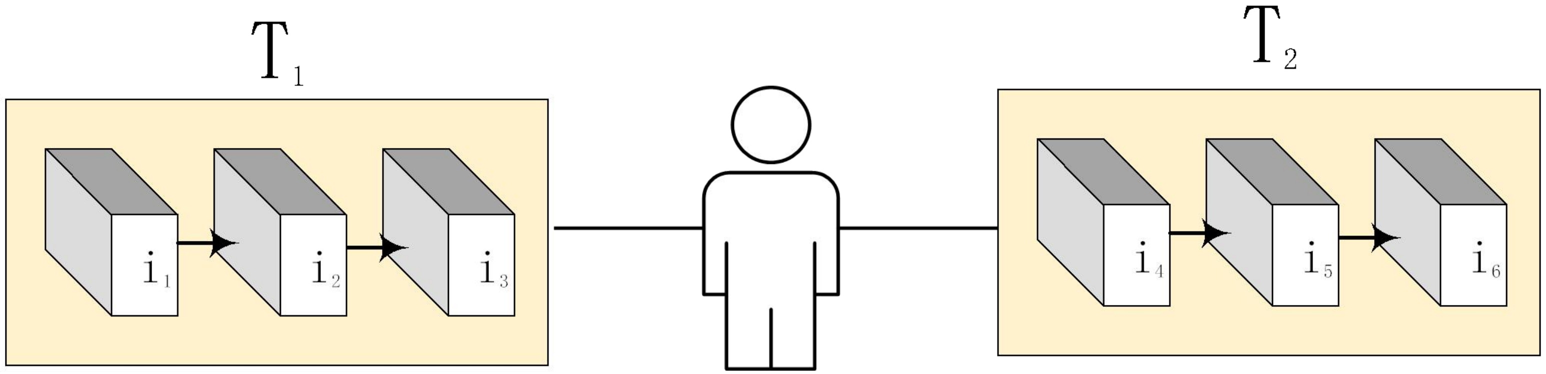
- A single-type-action session of watched movies
- A multi-type-action session includes more than one types of actions.

The type of actions (Cont')

Characteristics	Challenges
Single-type-action	Limited type of dependencies
Multi-type-action	How to learn both the intra- and inter-action type dependencies

User information

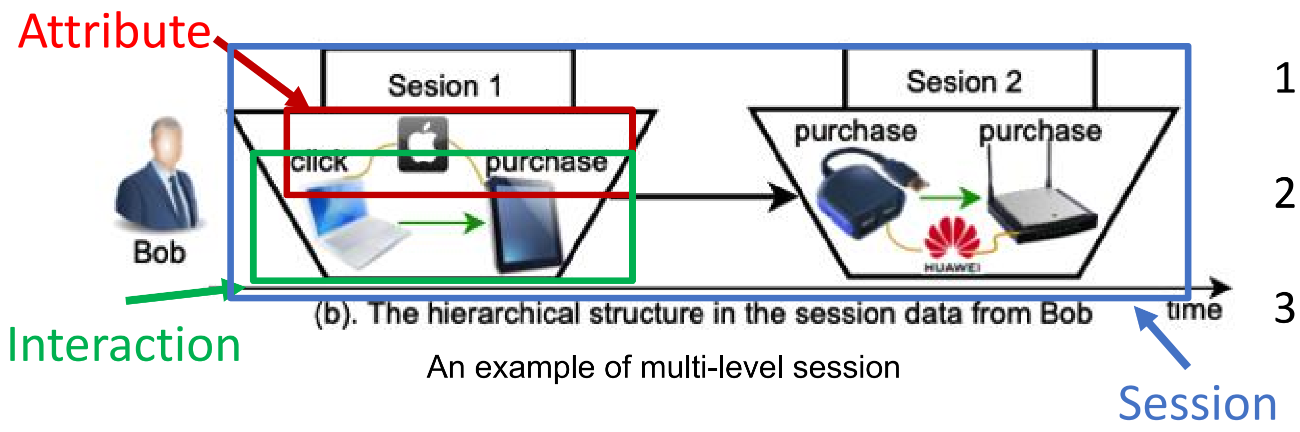
- According to whether the user information is available or not, sessions can be divided into non-anonymous sessions and anonymous sessions.



Characteristics	Challenges
Non-anonymous	How to learn the personalized long-term preference over multiple non-anonymous sessions.
Anonymous	How to capture the user's personalized preference with limited contextual information

Session data structure

- According to the number of levels of structures, session data can be roughly divided into multi-level session data and single-level session data.



An example of multi-level session

1. The attribute level consists of the attributes of entities (e.g., users, items) in an interaction.
2. The interaction level consists of interactions in each session.
3. The session level consists of multiple historical sessions from the current user (Session 1, Session 2).

Characteristics	Challenges
Multi-level session	How to learn the intra- and inter-level dependencies
Single-level session	How to overcome the cold-start and sparsity issues

A summary of characteristics and challenges

SIGIR 2022 Madrid

Table 4. Comparison of representative works regarding targeted session type, basic model and application domain

Work	Session type	Model	Domain	Work	Session type	Model	Domain
An SBRS based on association rules [77]	FO, ST, A, SL	ARD	Web page	Attention-gated recurrent network (IARN) [85]	L, O, ST, NA, SL	RNN, ATT	Video, movie
Access Pattern Approach (APA) [101]	UO, ST, NA, SL	FPM	Music	KNN-GRU4Rec [51]	O, ST, A, SL	KNN, RNN	Item
Personalized sequential pattern [143]	L,[1] O, ST, NA, SL	SPM	Item	Temporal deep semantic structured model (TDSSM) [106]	O, ST, NA, ML	MLP, RNN	News
Item/session KNN [51, 70]	O/UO, ST, A, SL	NN	Item	List-wise deep neural network [136]	UO, MT, A, SL	MLP	Item
Sequence and Time Aware Neighbourhood (STAN) [32]	O, ST, A, SL	NN	Item	DeepPredict [52]	UO, MT, NA, ML	MLP	Fashion
Temporal-Item-Frequency-based User (TIFU)-KNN [46]	UO, ST, NA, SL	NN	Item	ConvolutionAl Sequence Embedding Recommendation Model (CASER) [109]	FO, ST, NA, SL	CNN	Movie, POI
Page rank and Markov model [23]	O, ST, A, SL	MC	Web page	3D Convolutional Neural Network (3D CNN) [112]	O, ST, A, ML	CNN	Item
Factorized Personalized Markov Chain (FPMC) [95]	O, ST, NA, ML	MC, MF	Item	Hierarchical Temporal Convolutional Networks (HierTCN) [146]	L, O, MT, NA, ML	TCN	Item
Dynamic emission and transition model [56]	O, ST, NA, SL	HMM	Music, tweet	Session-based Recommendation with Graph Neural Network (SR-GNN) [138]	O, ST, A, SL	GNN	Item
Personalized Ranking Metric Embedding (PRME) [28]	O, ST, NA, SL	MC, ME	POI	Graph Contextualized Self-Attention Network (GC-SAN) [141]	O, ST, A, SL	GNN, ATT	Item
Personalized Markov Embedding (PME) [139]	O, ST, NA, SL	MC, ME	Music	Target Attentive Graph Neural Network (TAGNN) [147]	O, ST, A, SL	GNN, ATT	Item

Ordered (O), Unordered (UO), Flexible-Ordered (FO), Single-Type-action (ST), Anonymous (A), Single-Level (SL), Multi-Level (ML), Non-Anonymous (NA)

- The end of Section 2
- 15mins break

Outline

SIGIR 2022
Madrid

Sect. 1 Introduction	Sect. 2 Problem Statement & Challenges	**Sect. 3 Approaches**	Sect. 4 Applications & Opportunities
30mins, by Shoujin	30mins, by Qi + Break (15mins)	80mins, by Shoujin, Zhongyuan + Break (15mins)	30mins, by Liang + QA (10mins)
• Introduction to RS • Introduction to SBRS • Classification of SBR • Sequential RS vs. Session-based RS	• Problem statement • Characteristics and challenges	• Conventional approaches • Latent representation • Deep learning	• Applications • Algorithms and datasets • Future directions • Conclusions

Outline: Section 3

Section 3

Approaches

- An Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- Latent Representation Approaches
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

The categorization of SBRS approaches

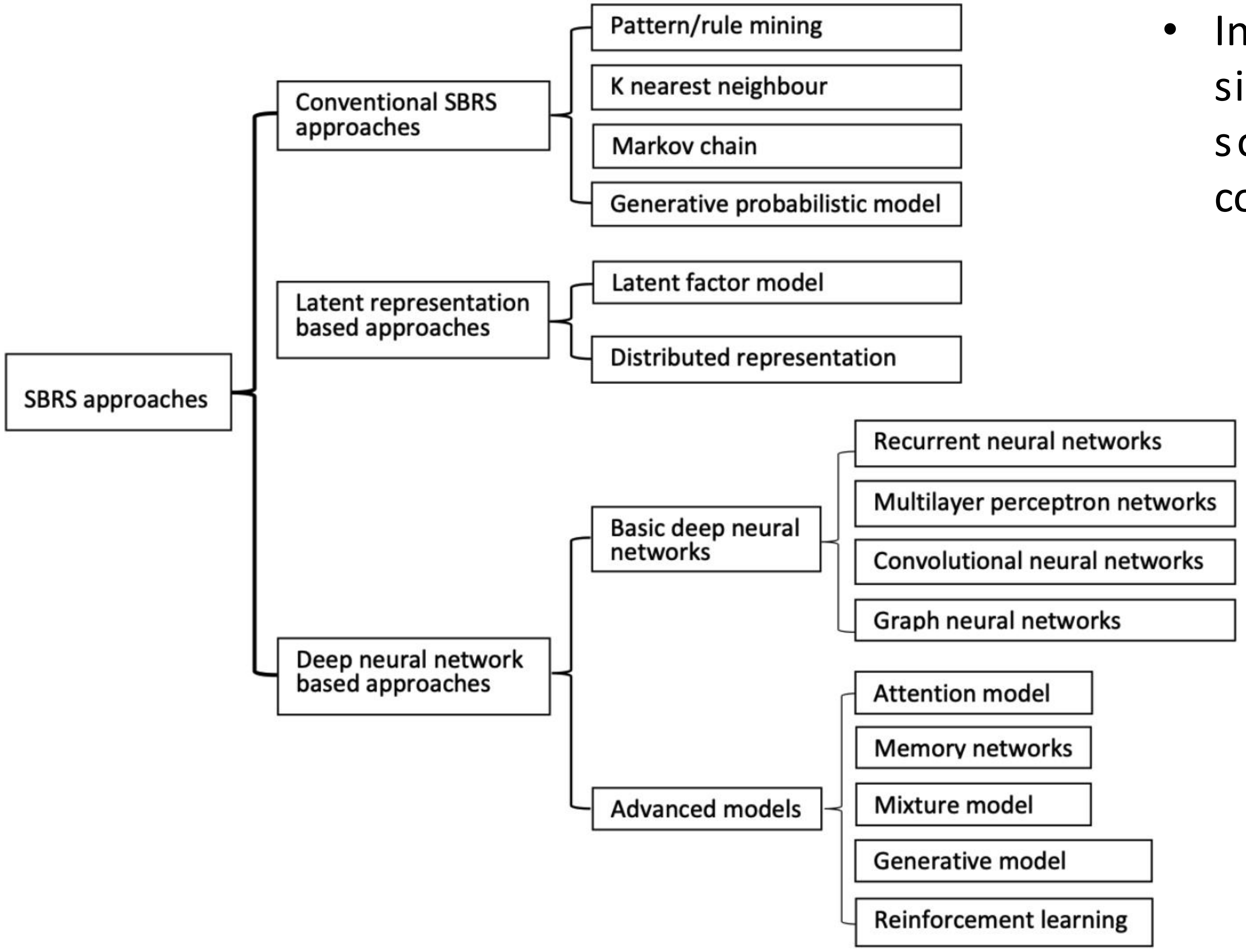


Fig. 3. The categorization of SBRS approaches

- In addition to approaches based on a single technique/model, there are some hybrid approaches which combine multiple techniques/models.

A comparison of different classes of approaches

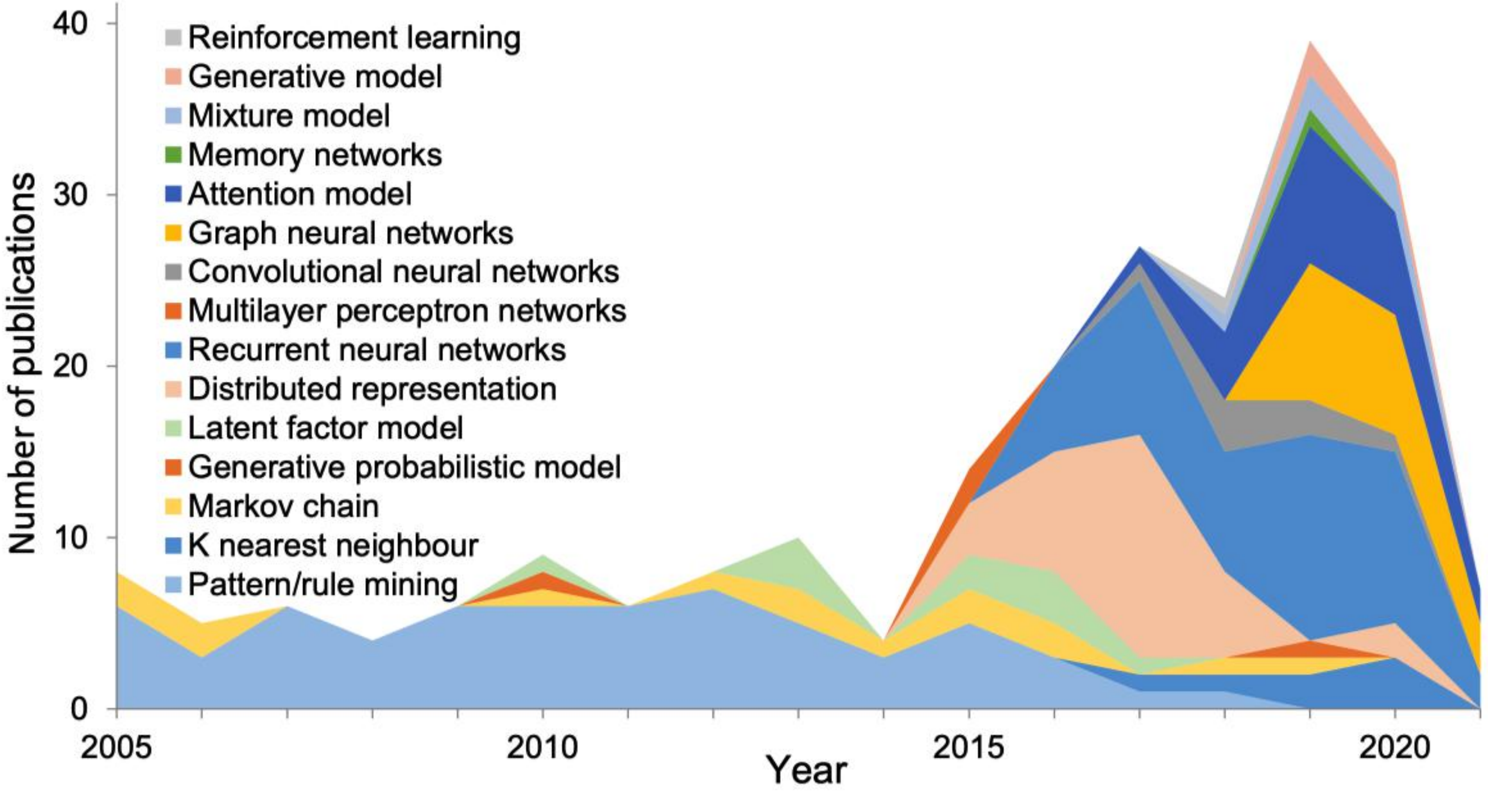
Table 5. A comparison of learned dependencies by different classes of approaches

Approach	Sequential or non-sequential	Short- or long-term	First or high-order	Pointwise or collective
Pattern/rule mining	Both[1]	Both	Both	Both
K nearest neighbour	Mainly non-sequential	Both	Mainly first-order	Both[2]
Markov chain	Sequential	Short-term	First-order	Pointwise
Generative probabilistic model	Sequential	Long-term	Higher-order	Collective
Latent factor model	Sequential	Short-term	First-order	Pointwise
Distributed representation	Mainly non-sequential	Both	Mainly first-order	Collective
Recurrent neural networks	Sequential	Long-term	High-order	Pointwise
Multilayer perceptron networks	Non-sequential	Both	First-order	Collective
Convolutional neural networks	Mainly sequential	Both	Mainly first-order	Collective
Graph neural networks	Both	Both	High-order	Pointwise
Attention models	Mainly non-sequential	Both	First-order	Mainly pointwise
Memory networks	Non-sequential	Both	First-order	Pointwise
Mixture models	Both	Both	Both	Both
Generative models[3]	Either	Either	Either	Either
Reinforcement learning	Sequential	Both	High-order	Pointwise

[1] Non-sequential and sequential dependencies are learned by frequent pattern mining and sequential pattern mining respectively.
[2] Item-KNN and session-KNN mainly models pointwise and collective dependencies respectively.
[3] The learned dependencies mainly depend on the employed encoder for encoding the input of the generation model.

A comparison of different classes of approaches

- Use the typical keywords “session, recommendation”, ”next item/basket/POI/song/news/ video recommendation" for searching and then manually counted those relevant publications only on 20 March, 2021.



Wang S and et al. “A survey on session-based recommender systems”. In: CSUR. 2021.

Outline: Section 3

Section 3

Approaches

- A Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- Latent Representation Approaches
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

Pattern/rule mining based SRSs/SBRSs

There are two types of pattern/rule mining based approaches for SBRS

1. Frequent pattern/association rule mining based approaches

   Mining the association rules over different interactions within **unordered sessions.**

2. Sequential pattern mining based approaches

   Mining the sequential patterns over sequences of sessions or interactions within **ordered sessions.**

- **Disadvantage**: only can handle single-type-action sessions in which all the actions are the same in a dataset, so each interaction in a given session is simplified into an item.

Frequent pattern/association rule mining based approaches

General framework

1. Frequent pattern or association rule mining

   A set of frequent patterns $FP = \{p_1, p_2, \ldots, p_{|FP|}\}$ are mined by using pattern mining algorithms.

2. Session matching

   Given a partial session $\hat{s}$ over full item list �, if an item $\hat{v} \in$ � \ $\hat{s}$ exists so that $\hat{s} \cup \{ \hat{v} \} \in FP$, then $\hat{v}$ is a candidate item for recommendation.

3. Recommendation generation

   If the conditional probability � $(\hat{v} \mid \hat{s})$ is greater than a predefined confidence threshold, then $\hat{v}$ is added into the recommendation list.

T1: $\{ABDE\}$
T2: $\{ABECD\}$
T3: $\{ABEC\}$
T4: $\{BEBAC\}$
T5: $\{DABEC\}$

Size 1	Size 2	Size 3	Size 4
$\{A\}(5)$	$\{A,B\}(5)$	$\{A,B,C\}(4)$	$\{A,B,C,E\}(4)$
$\{B\}(6)$	$\{A,C\}(4)$	$\{A,B,E\}(5)$	
$\{C\}(4)$	$\{A,E\}(5)$	$\{A,C,E\}(4)$	
$\{E\}(5)$	$\{B,C\}(4)$	$\{B,C,E\}(4)$	
	$\{B,E\}(5)$		
	$\{C,E\}(4)$		

Table 2: Frequent Itemsets generated by the Apriori algorithm

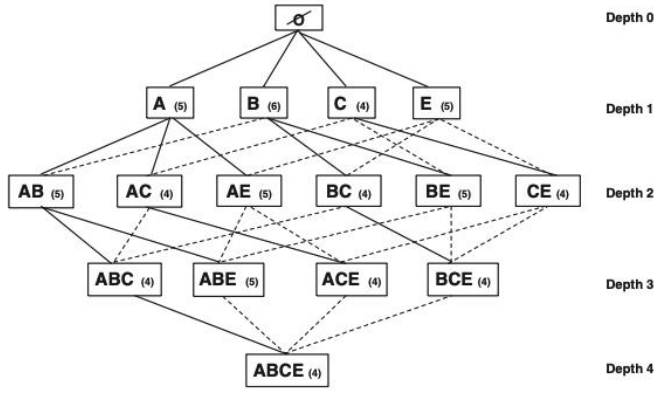


Figure 1: The Frequent Itemsets Graph for the example

Bamshad Mobasher and et al. "Effective personalization based on association rule discovery from web usage data." In: WIDM. 2001.

Sequential pattern mining based approaches

General framework

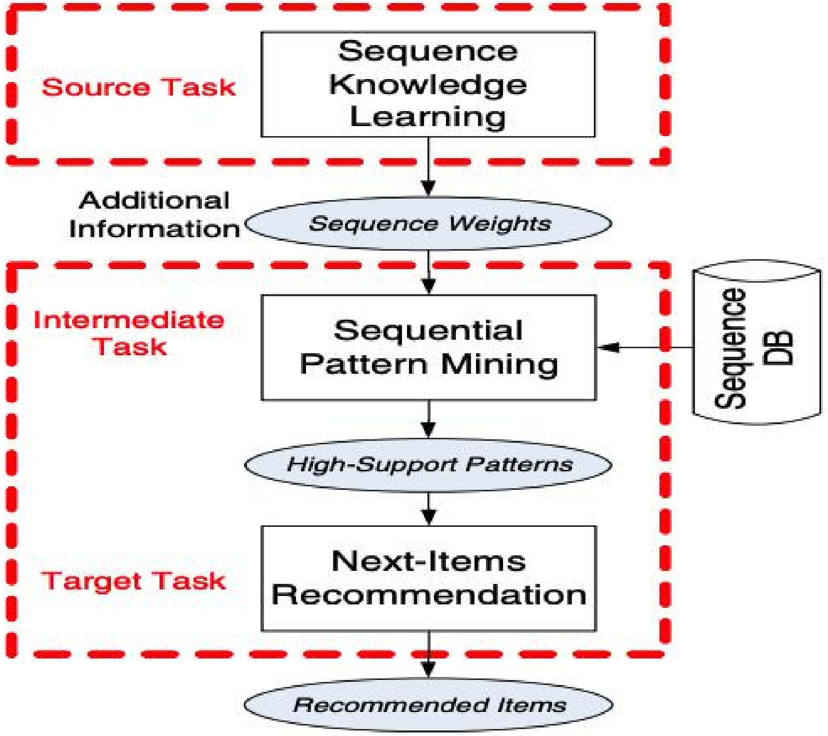
Given a sequence set $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$ where $q_i = \{s_{i1}, s_{i2}, \ldots, s_{i|q|}\}$ is a sequence of sessions from the same user i ordered according to time

1. Sequential pattern mining: a set of sequential patterns $SP = \{p_1, p_2, \ldots, p_{|SP|}\}$ is mined on �;
2. Sequence matching: given a user �'s sequence $q_u = \{s_{u1}, s_{u2}, \ldots, s_{ug}\}$, for any sequential pattern $p \in SP$, if the last session s_{ug} of q_u belongs to p, i.e., $p = \{s_1, s_2, \ldots, s_{ug}, s_r, \ldots\}$, then p is a relevant pattern for this specific recommendation
3. Recommendation generation

   From results of sequence matching (p was matched in the previous step) and items after s_{ug} (s_r) are candidate items. For each candidate items $\hat{v}$ we compute its support in terms of support of patterns

$$supp(\hat{v}) = \sum_{s_g \in q_u, s_g \in p, \hat{v} \in s_r, s_r \in p, p \in SP} supp(p).$$

Recommend candidate items with the top support values to user �.



Ghim-Eng Yap and et al. "Effective next-items recommendation via personalized sequential pattern mining." In: DASFAA. 2012.

Summary of pattern/rule mining based SRSs/SBRSs

Applicable scenario

- Simple, balanced and dense, ordered or unordered sessions

Pros

- Intuitive, simple and effective on session data where dependencies are easy to learn

Cons

- Information loss, cannot handle complex data (e.g., imbalanced or sparse data)

Outline: Section 3

Section 3

Approaches

- A Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- Latent Representation Approaches
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

K nearest neighbor (KNN) based approaches

Main idea

K Nearest Neighbour (KNN) based approaches for SBRS are proven to be simple but effective.

General framework

1. Find out the K most similar interactions or sessions
2. Calculates a score
   - For each candidate interaction, calculates a score based on the similarity to indicate its relevance to the current interaction.

According to whether the similarity is actually calculated between items or sessions, KNN-based approaches for SBRSs can be divided into **item-KNN** and **session-KNN**.

Item-KNN

- Recommends the K items most similar to current item in terms of co-occurence in other sessions;
- Each item is encoded into a binary vector where each element indicates whether the item occurs (set to "1") in a specific session or not (set to "0").
- Consequently, the similarity between items can be calculated on their vectors with a certain similarity measure, like cosine similarity.

Session-KNN

- Unlike item-KNN (only last item in session), session-KNN compares whole sessions with the current session.
- Given a session c, first determine k most similar past sessions $N(c)$ by applying suitable session similarity measure (Jaccard index or cosine similarity on binary vectors over item space).
- The score of each candidate item is computed by summing over similarity scores:

$$score(\hat{v}) = \sum_{s_{nb} \in N(c)} sim(c, s_{nb}) * 1_{sb}(\hat{v})$$

The indicator function $1_{\mathrm{sb}}(\hat{\mathrm{v}})$ returns 1 if session s_{nb} contains $\hat{v}$ and zero otherwise

- Compared with item-KNN, session-KNN can capture more information.

Summary

Applicable scenarios

- Simple sessions and ordered or un-ordered sessions

Pros

- Intuitive, simple and effective, quick response.

Cons

- Information loss, hard to select �, limited ability for complex sessions (e.g., noisy sessions).

Outline: Section 3

Section 3

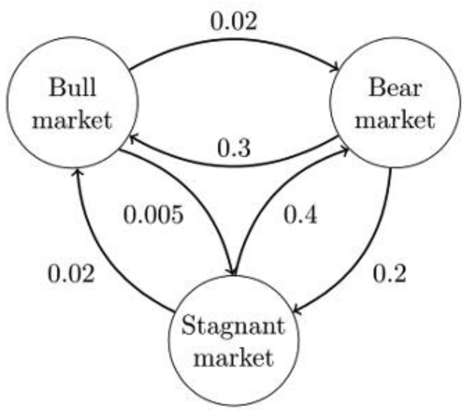
Approaches

- A Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- Latent Representation Approaches
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

Markov chain based approaches

Main idea

Markov chains to model the transitions over interactions within or between sessions to predict the probable next interaction(s) or session given a session context.



An example of Markov chain

Markov chain based approaches can be divided into

- basic Markov chain based approaches.
- latent Markov embedding based approaches.

Basic Markov chain based approaches

General framework

1. Calculating the transition probabilities: the first-order transitional probability from interaction o_i to o_j is defined as:

$$P_t(i,j) = P(o_i -> o_j) = \frac{freq(o_i -> o_j)}{\sum_{o_t} freq(o_i -> o_t)}$$

2. Predicting the transition paths: using the chain rule we construct transition paths

$$P(o_1 -> o_2 -> o_3) = P(o_i) * P(o_2|o_1) * P(o_3|o_2)$$

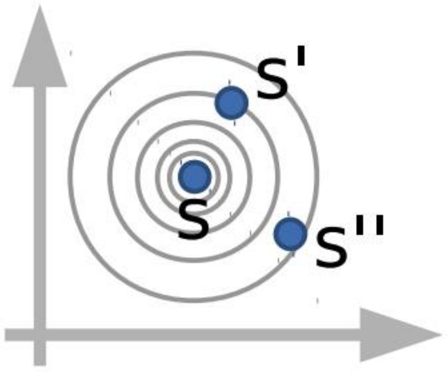
3. Select reference paths
   - Given a session context, the paths with high probabilities are chosen as the reference paths

4. Making recommendations
   - If the session context occurs in a reference path, items occuring after it in this path are put in the recommendation list

Magdalini Eirinaki, Michalis Vazirgiannis, and et al. 2005. Web path recommendations based on page ranking and Markov models. In WIDM. ACM, 2–9

Latent Markov embedding based approaches

- LME bypasses need to adjust different similarity metrics based on different types of content (e.g., song playlists)
- embedding in Euclidean space enable derivation of unobserved transitions and thus solve the data sparsity issue in limited observed data.
- $P(o_i -> o_j)$ is assumed to be negatively related to the Euclidean distance $||o_i - o_j||$ between o_i and o_j.

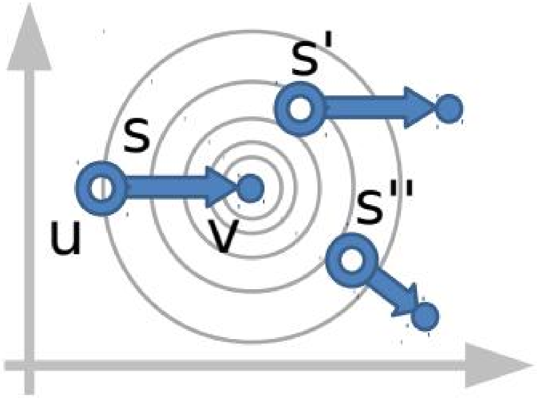


Single Point Model

General framework

1. Embed the Markov chains into an Euclidean space
2. Calculate the transition probabilities $P(\{o_1 -> o_2 -> \ldots -> o_{pa}\})$is defined based on Markov model

$$P(\{o_1 \rightarrow o_2 \rightarrow, \ldots, \rightarrow o_{|pa|}\}) = \prod_{i=2}^{|pa|} P(o_{i-1} \rightarrow o_i) = \prod_{i=2}^{|pa|} \frac{e^{-||o_i - o_{i-1}||_2^2}}{\sum_{o_t} e^{-||o_t - o_{i-1}||_2^2}}.$$



Dual Point Model (how to create playlists)

Shuo Chen and et al. “Playlist prediction via metric embedding.” In: SIGKDD. 2012.

Summary

Applicable scenario

- Short and ordered sessions with short-term and low-order dependencies

Pros

- Good at modelling short-term and low-order sequential dependencies

Cons

- Usually ignore long-term and higher-order dependencies, the rigid order assumption is too strong

Comparison of conventional SBRS approaches

Table 6. A comparison of different classes of conventional approaches for SBRSs

Approach	Applicable scenario	Pros	Cons	Typical work
Pattern/rule mining based SBRSs	Simple, balanced and dense, ordered or unordered sessions	Intuitive, simple and effective on session data where dependencies are easy to learn	Information loss, cannot handle complex data (e.g., imbalanced or sparse data)	[30],[77],[78], [79],[101],[143]
KNN based SBRSs	Simple, ordered or unordered sessions	Intuitive, simple and effective, quick response	Information loss, hard to select *K*, limited ability for complex sessions (e.g., noisy sessions)	[32],[46],[51], [70]
Markov chain based SBRSs	Short and ordered sessions with short-term and low-order dependencies	Good at modelling short-term and low-order sequential dependencies	Usually ignore long-term and higher-order dependencies, the rigid order assumption is too strong	[13],[23],[28], [56],[95],[139], [154]

After comparison, the result shows KNN-based approaches especially the session-KNN achieve superior performance.

Outline: Section 3

Section 3

Approaches

- A Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- **Latent Representation Approaches**
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

Latent factor model based approaches

Main idea

Adopt factorization models to factorize the observed transition matrix over interactions (items) into their latent representations, and then utilize the resultant latent representations to estimate the unobserved transitions.

General framework

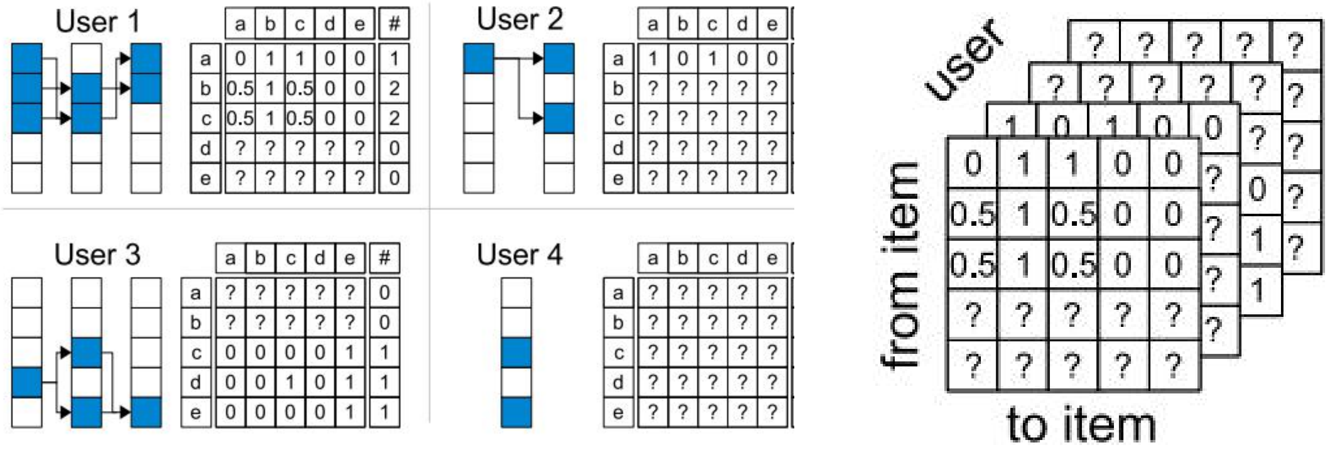
1. Factorize transition matrix
2. Utilize the resultant latent representations to estimate the unobserved transitions

An example of latent factor model based approaches

1. Build a transition tensor B

$$\hat{\mathcal{B}} = \mathcal{C}o \times U \times O_i \times O_j,$$

C� : The core tensor;
� : Latent factor matrix of users;
�� : Latent factor matrix of last items
�� : Latent factor matrix of current items



2. The observed transitions for cube B are very sparse, a special case of **Canonical Decomposition** that models pairwise interactions is used:

$$\hat{b}_{k,i,j} =< \boldsymbol{u}_k, \boldsymbol{o}_i > + < \boldsymbol{o}_i, \boldsymbol{o}_j > + < \boldsymbol{u}_k, \boldsymbol{o}_j >,$$

o_i (last interaction) and o_j (current interaction) are the latent representation vector of user u_k

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In WWW. ACM, 811–820

Outline: Section 3

Section 3

Approaches

- A Overview of Different Classes of Approaches
- Conventional SBRS Approaches
  - Pattern / Rule Mining Approaches
  - K Nearest Neighbour Approaches
  - Markov Chain Approaches
  - Generative Probabilistic Models
- Latent Representation Approaches
  - Latent Factor Models
  - Distributed Representations
- Deep Neural Network Approaches
  - Basic Deep Neural Network Models
  - Advanced Deep Models

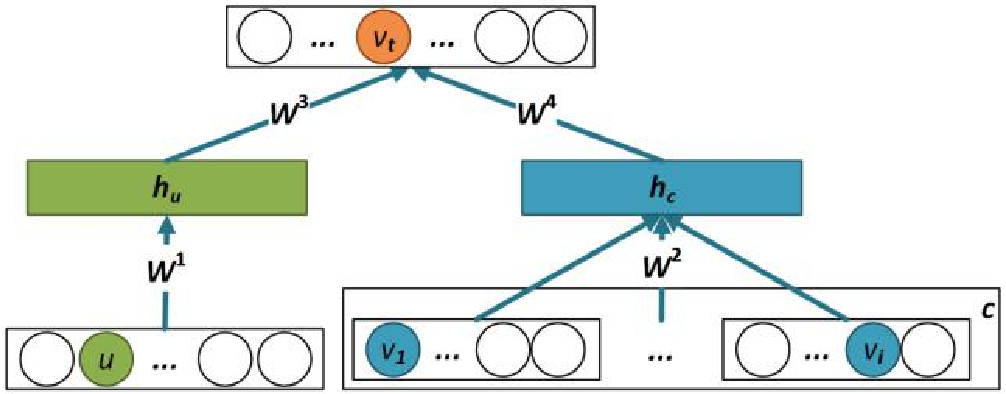
Distributed representation based approaches

Main idea

- inspired by natural language modelling approaches, this method utilizes the mapping words <--> items and sentences <--> sessions;
- conventional (word2vec) NLP models not suitable for SBRS due to lack of users;
- this approach design shallow network with double wide-in vectors (items + users) and wide-out vector for items relevant to specific user-session context.

General framework

- a shallow neural network embeds a user $u_k$and an item v_i into a latent distributional vector respectively using the logistic function $\delta(\cdot)$for nonlinear transformation;
- use representation to generate $P(v_t|u,\boldsymbol{c})$, where $\boldsymbol{c}$ is context information.



Liang Hu, Longbing Cao, Shoujin Wang, and et al. "Diversifying personalized recommendation with user-session context." In: IJCAI. 2017.

Comparison of latent representation based SBRS approaches

Table 7. A comparison of different classes of latent representation approaches for SBRSs

Approach	Applicable scenario	Pros	Cons	Typical work
Latent factor model	Dense, ordered session data	Relatively simple and effective	Suffer from data sparsity, cannot capture higher-order and long-term dependencies	[17],[62],[63],[66],[95],[100]
Distributed representation	Unordered session data	Simple and efficient, strong encoding capability	Hard to model ordered or heterogeneous sessions (e.g., noisy sessions)	[33],[49],[60],[118],[123],[126],[127]

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
    - Recurrent neural networks (RNN)
    - Multi layer perceptron (MLP)
    - Convolutional neural networks (CNN)
    - Graph neural networks (GNN)
  - Advanced deep model based approaches

Comparison of latent representation based SBRS approaches

Table 7. A comparison of different classes of latent representation approaches for SBRSs

Approach	Applicable scenario	Pros	Cons	Typical work
Latent factor model	Dense, ordered session data	Relatively simple and effective	Suffer from data sparsity, cannot capture higher-order and long-term dependencies	[17],[62],[63], [66],[95],[100]
Distributed representation	Unordered session data	Simple and efficient, strong encoding capability	Hard to model ordered or heterogeneous sessions (e.g., noisy sessions)	[33],[49],[60],[118], [123],[126],[127]

- 8mins break

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
    - Recurrent neural networks (RNN)
    - Multi layer perceptron (MLP)
    - Convolutional neural networks (CNN)
    - Graph neural networks (GNN)
  - Advanced deep model based approaches

Recurrent Neural Networks (RNN) based approaches

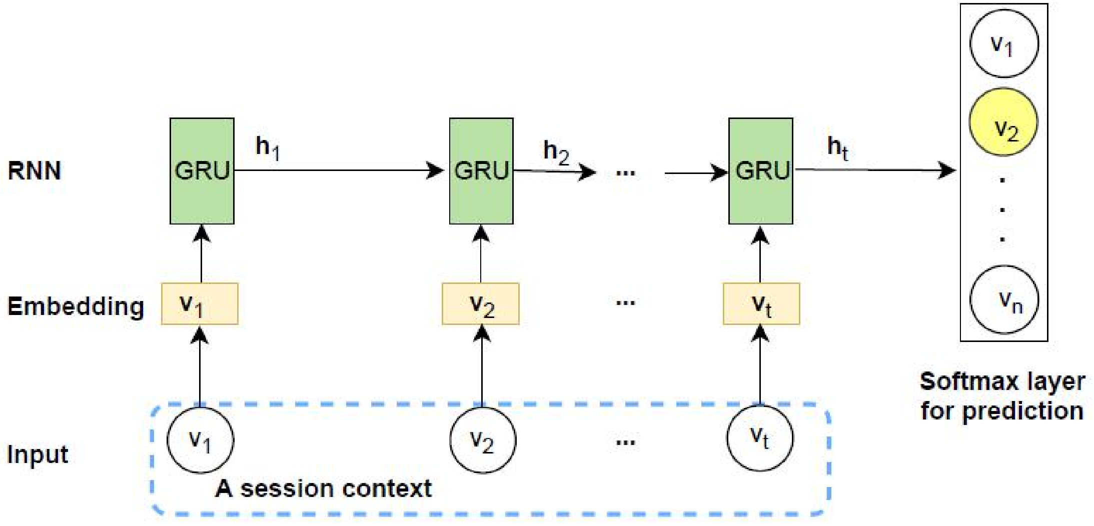
SIGIR 2022
Madrid

Main idea

- Benefiting from their intrinsic advantages for **modeling sequential dependencies**, RNN-based approaches dominate deep neural network approaches for SBRSs.

General framework

- An RNN is built to model the session context consisting of a sequence of interactions.



A general framework of RNN-based SBRS

An example of RNN-based approaches – GRU4Rec

The Model

1. The embedding �$_{t}$ of the �th interaction �$_{t}$ in the context is taken as the input of the �th time step of the RNN.

2. The RNN unit, i.e., GRU, is used to update the hidden state �$_{t}$ at the �th time step by absorbing information from both the last hidden state �$_{t-1}$ and the current candidate state �$_{t}$ by using an update gate �$_{t}$.

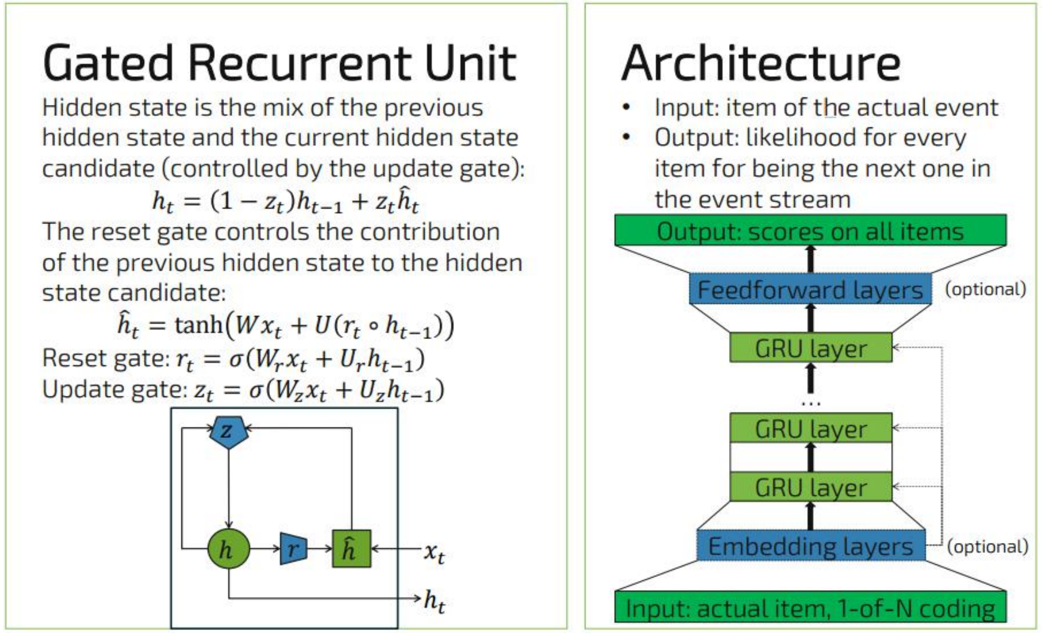
$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\hat{\boldsymbol{h}}_t,$$

$$z_t = \sigma(\boldsymbol{W}_z\boldsymbol{o}_t + \boldsymbol{X}_z\boldsymbol{h}_{t-1}),$$

$$\hat{\boldsymbol{h}}_t = tanh(\boldsymbol{W}_h\boldsymbol{o}_t + \boldsymbol{X}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})),$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{o}_t + \boldsymbol{X}_r\boldsymbol{h}_{t-1}),$$

3. The hidden state �$_{|�|}$ from the last time step is used as the representation �� of a session context � for the prediction of the next interaction.



The model architecture of GRU4Rec

Hidasi, B and et al. "Session-based Recommendations with Recurrent Neural Networks." In: ICLR. 2016.

Summary of RNN-based approaches

Applicable scenario

- Long and rigidly ordered sessions.

Pros

- Model long-term and high- order sequential dependencies.

Cons

- The rigid order assumption is too strong for session data.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
    - Recurrent neural networks (RNN)
    - Multi layer perceptron (MLP)
    - Convolutional neural networks (CNN)
    - Graph neural networks (GNN)
  - Advanced deep model based approaches

Multi Layer Perceptron (MLP) networks based approaches

SIGIR 2022 Madrid

Main idea

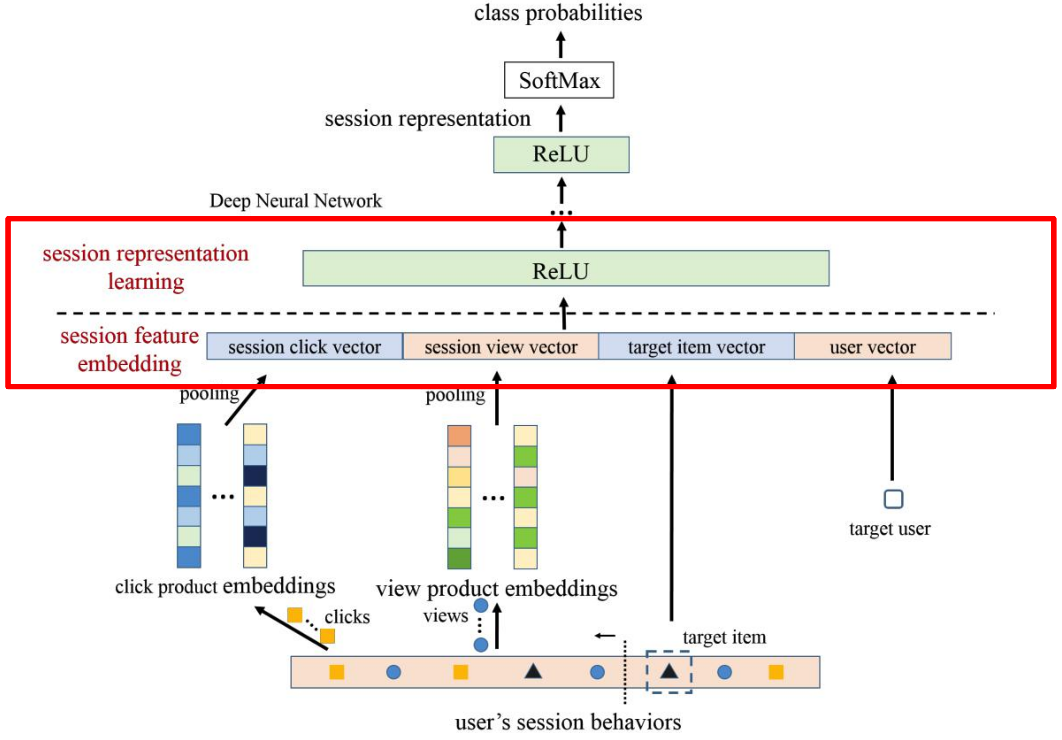
- MLP-based approaches are usually applied to learn an optimized combination of different representations to form a compound representation of session context for the subsequent recommendations.

General framework

- MLP layer is utilized to connect the representations of different parts of a session context to export a unified and compound representation e_c for context c.

$$e_c = \sigma(W_c e_{c_c} + W_v e_{c_v}),$$

where e_{c_c}, e_{c_v} are the representation of the sub-session context; like: click, view.



The model architecture of an MLP based SBRS

Wu C and et al. "Session-aware information embedding for e-commerce product recommendation." In: CIKM. 2017.

Summary of MLP-based approaches

Applicable scenario

- Unordered sessions, sessions with multi-aspects (e.g., static and dynamic features) to be combined.

Pros

- A simple structure, project sparse features to dense ones, learn the combination of different parts.

Cons

- Cannot model complex sessions, e.g., ordered, heterogeneous sessions.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
    - Recurrent neural networks (RNN)
    - Multi layer perceptron (MLP)
    - Convolutional neural networks (CNN)
    - Graph neural networks (GNN)
  - Advanced deep model based approaches

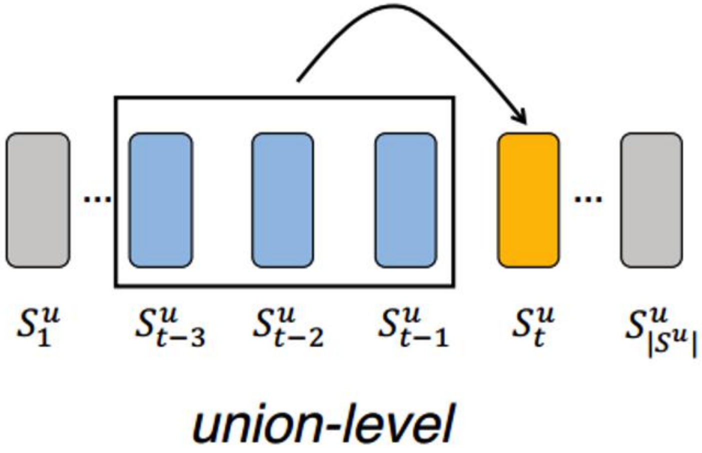
Convolutional Neural Networks (CNN) based approaches

Main idea

1. CNN relax the rigid order assumption over interactions within sessions, which makes the model more robust and flexible;
2. They have high capabilities in learning local features from a certain area and relationships between different areas in a session to effectively capture the union-level collective dependencies.

General framework

1. Utilize the filtering and pooling operations to better learn an informative representation for each session context.
2. Use the learned representation for the subsequent recommendations.



Union-level collective dependencies in session data

Tang J and et al. "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding." In: WSDM. 2018.

An example of CNN-based approaches – Caser

The model

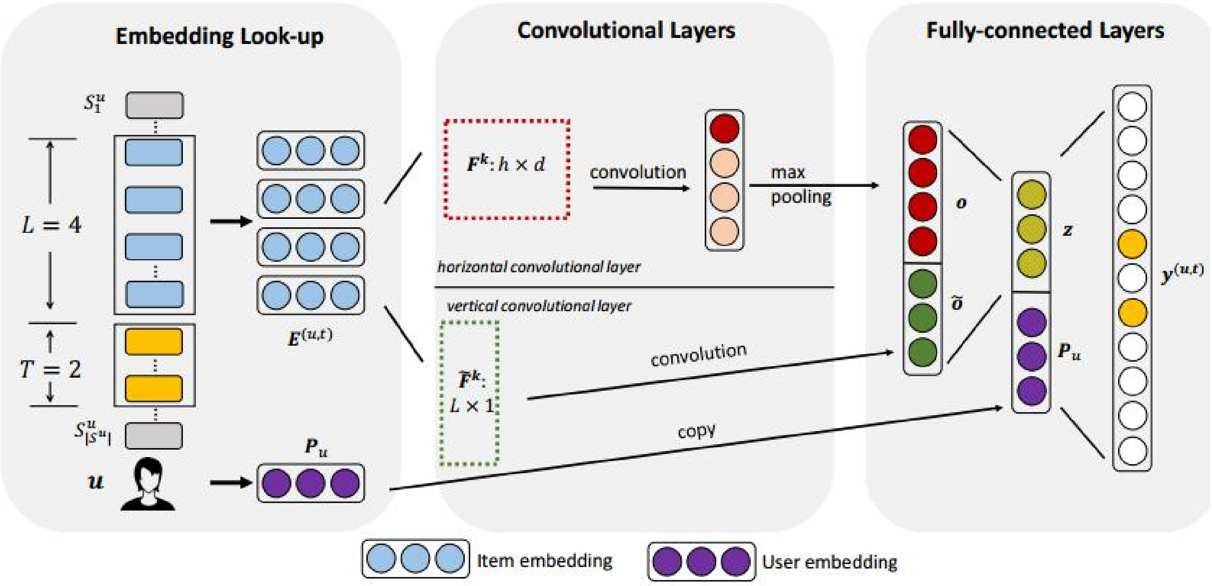
1. Construct embedding matrix $\boldsymbol{E}$:

   First mapping each interaction in context � into a �-dimensional latent vector and then putting all the vectors together into a matrix.

2. Horizontal convolutional layers:

   The � th convolution value � � is achieved by sliding the � th filter � � from the top to the bottom on � to interact with its horizontal dimensions:

$$\alpha_m^x = \phi_\alpha(E_{m:m+h-1} \odot F^x),$$



The model architecture of Caser

Tang J and et al. "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding." In: WSDM. 2018.

An example of CNN-based approaches – Caser (Cont')

The model

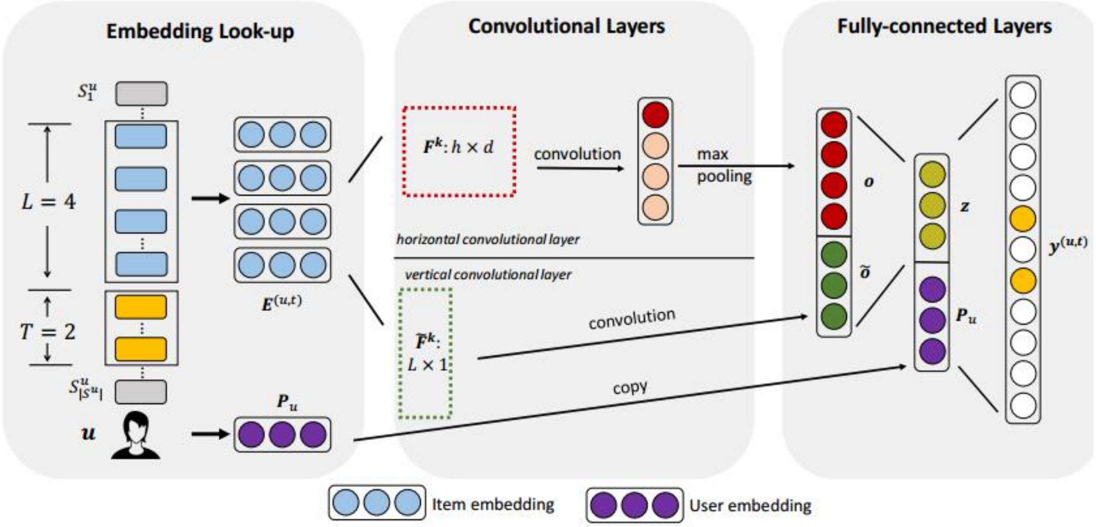
3. Final output

�� is obtained by performing the max pooling operation on the convolution result to capture the most significant features $\boldsymbol{\alpha}^x = [\alpha_1^x, \alpha_2^x, \ldots, \alpha_{|c|-h+1}^x]$

$$\boldsymbol{e}_c = max\{max(\boldsymbol{\alpha}^1), max(\boldsymbol{\alpha}^2), \ldots, max(\boldsymbol{\alpha}^z)\}.$$

4. Recommendation

�� is treated as the representation of the session context � and is used for subsequent recommendations.



The model architecture of Caser

Tang J and et al. "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding." In: WSDM. 2018.

Summary of MLP-based approaches

Applicable scenario

- Flexible-ordered, heterogeneous or noisy sessions.

Pros

- Robust, no rigid order assumption, capture the union- level collective dependency.

Cons

- Relatively high complexity.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
    - Recurrent neural networks (RNN)
    - Multi layer perceptron (MLP)
    - Convolutional neural networks (CNN)
    - Graph neural networks (GNN)
  - Advanced deep model based approaches

Graph Neural Networks (GNN) based approaches

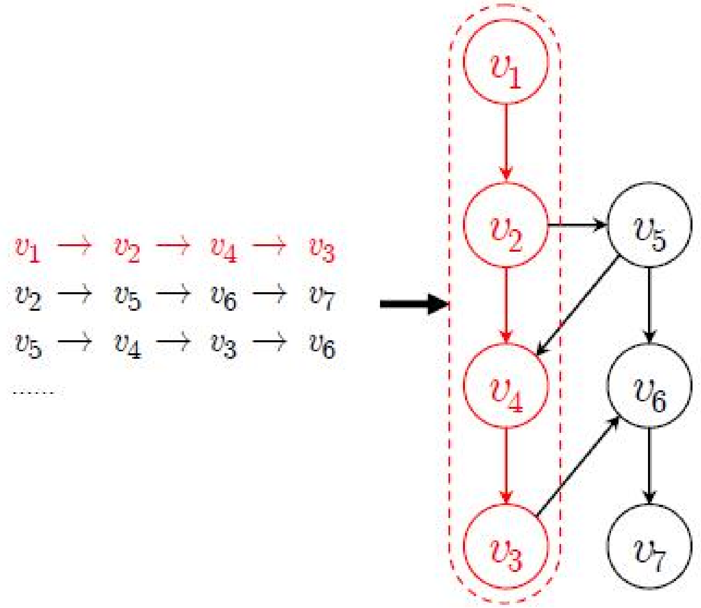
Main idea

- GNN have shown great expressive power in modeling the complex relations embedded in graph structured data by introducing deep neural networks into graph data.

Taxonomy

GNN approaches can be divided into three classes

1. Gated Graph Neural Networks (GGNN)
2. Graph Convolutional Networks (GCN)
3. Graph Attention networks (GAT)



An example of a session graph

Gated Graph Neural Networks (GGNN) for SBRSs

General framework

1. Construct session graph

   A directed graph is constructed based on all the historical ordered sessions, where the direction of each edge indicates the order of adjacent interactions within sessions.

2. Learn item embeddings on session graphs with GNN

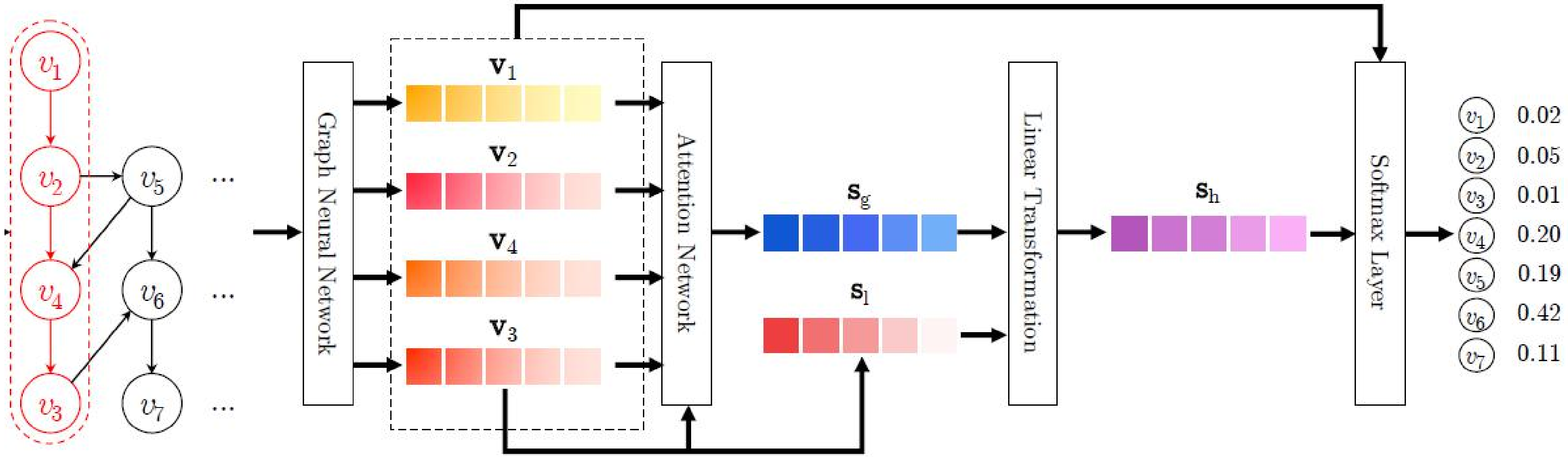
   Use gated recurrent unit (GRU) to update the embedding of each item.

$$\boldsymbol{h}_i^t = GRU(\boldsymbol{h}_i^{(t-1)}, \sum_{n_j \in N(n_i)} \boldsymbol{h}_j^{(t-1)}, \boldsymbol{A}),$$

� (��) is the set of neighbourhood nodes of �� in the session graph, and � is the adjacency matrix built on the session graph.

An example of GGNN - SRGNN

- Input the built graph into a GNN to learn the embedding of each item, and then built session context embedding based on item embedding.



The model structure of SRGNN

Wu et al. "Session-based Recommendation with Graph Neural Networks." In: AAAI. 2019.

An example of GGNN – SRGNN (Cont')

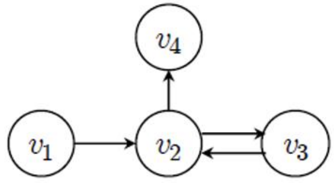
The model

1. Constructing session graphs: $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$
2. Learning item embeddings on session graphs with Gate-GNN to capture sequential dependencies over items;
3. Generating session context embeddings: $\mathbf{s}_\mathrm{h} = \mathbf{W}_3 \left[\mathbf{s}_\mathrm{l}; \mathbf{s}_\mathrm{g}\right],$
4. Making prediction and recommendation on the next item:

$$\alpha_i = \mathbf{q}^\top \sigma(\mathbf{W}_1 \mathbf{v}_n + \mathbf{W}_2 \mathbf{v}_i + \mathbf{c}),$$

$$\mathbf{s}_\mathrm{g} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i,$$

s: v_1-> v_2 -> v_3 -> v_2 -> v_4



	Outgoing edges				Incoming edges			
	1	2	3	4	1	2	3	4
1	0	1	0	0	0	0	0	0
2	0	0	1/2	1/2	1/2	0	1/2	0
3	0	1	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0

An example of a session graph and the connection matrix **A**s

Wu et al. "Session-based Recommendation with Graph Neural Networks." In: AAAI. 2019.

Graph Convolutional Networks (GCN) for SBRSs

Main idea

- GCN-based SBRSs mainly utilize the pooling operation to integrate information from node �� 's neighbourhood node �� in the graph to help with the update of the hidden state of �� for better learning item embeddings.

General framework

1. Utilize the pooling operation to combine information from neighborhood items:

$$\hat{h}_i^t = pooling(\{h_j^{(t-1)}, n_j \in N(n_i)\}),$$

�(��) is the set of neighbourhood nodes of node �i

2. Integrated neighbourhood information:

$$h_i^t = h_i^{(t-1)} + \hat{h}_i^t.$$

3. The last hidden state of node �� is taken as its embedding ��.

Graph ATtention networks (GAT) for SBRSs

Main idea

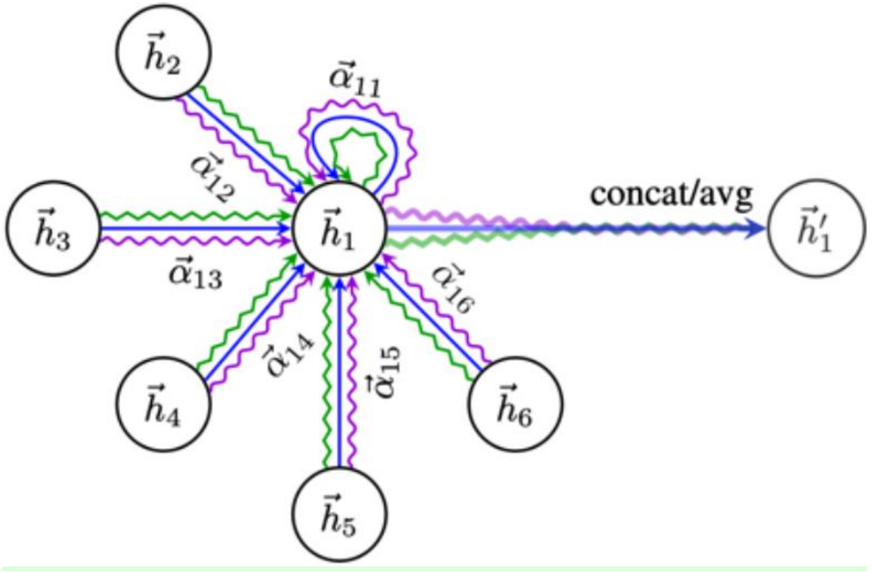
- GAT-based SBRSs mainly utilize attention mechanism to attentively integrate the information from the neighbourhood nodes of node �� in a session graph to update its hidden state in each layer for better learning item embeddings.

General framework

1. Calculating the importance weights of each neighborhood node.

2. Aggregating the hidden states of neighbourhood nodes according to their importance weights:

$$\boldsymbol{h}_i^t = attention(\{\boldsymbol{h}_j^{(t-1)}, n_j \in N(n_i)\}),$$



An example of GAT, figure from https://next-nus.github.io/

Summary of GNN-based approaches

Applicable scenario

- Complex sessions with complex transitions, e.g., repeat transactions.

Pros

- Model the complex transitions among interactions.

Cons

- Complex and costly.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
  - Advanced deep model based approaches
    - Attention model
    - Memory network
    - Mixture model
    - Reinforcement learning
    - Contrastive learning

Attention model based approaches

Main idea

Attention-based SBRSs introduce the attention mechanism to discriminatively exploit different elements in a session context to build an informative session context representation for accurate recommendations.

General framework

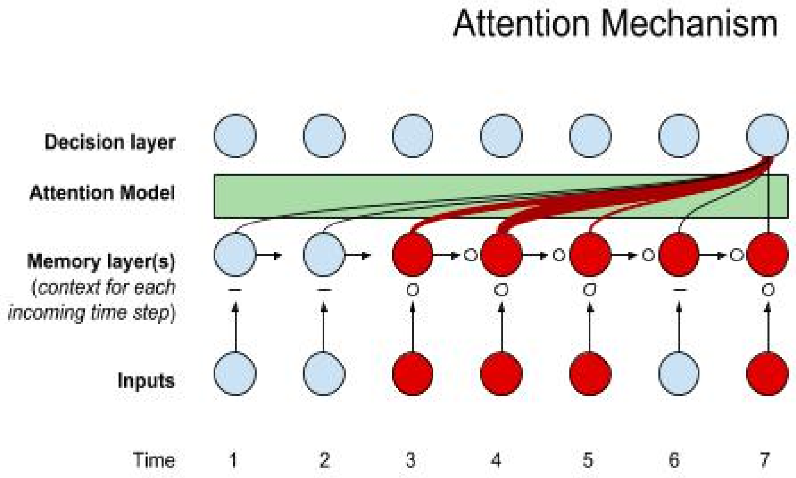
1. Attention weight calculation

$$\beta_{tg,i} = \frac{exp(e(o_i))}{\sum_{o_j \in c} exp(e(o_j))},$$

2. Aggregation

$$\mathbf{e}_c = aggregate(\{\mathbf{o}_i, \beta_{tg,i},\ o_i \in c\}),$$

���������� is an aggregation function which is often specified as a weighted sum



An example of a attention model based SBRS

An example of attention model

Intuition

- Usually no **rigid order** exist over items within one transaction.
- Different chosen items contribute differently to the choice of next items. Those **relevant** items must be **emphasized** to make correct predictions.

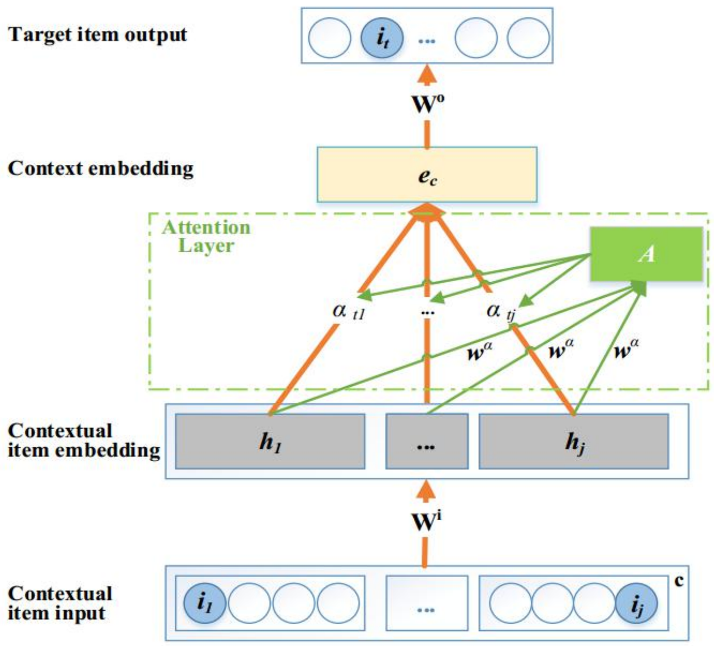
The Model

1. Learns **embeddings of items** in each session.
2. The **attention weights** are calculated using softmax function.

$$\alpha_{tj} = \frac{exp(e(\mathbf{h}_j))}{\sum_{s \in \mathbf{c}_t} exp(e(\mathbf{h}_s))} \qquad e(\mathbf{h}_j) = \mathbf{w}^{\alpha}\mathbf{h}_j^T$$

3. Weighted sum of the embeddings of all items to form the **session context embedding**.

$$\mathbf{e}_c = \sum_{i_j \in \boldsymbol{c}} \alpha_{tj}\mathbf{h}_j, \quad s.t. \sum_{i_j \in \boldsymbol{c}} \alpha_{tj} = 1$$



An example of attention model based SBRS

Shoujin Wang, et al. Attention-Based Transactional Context Embedding for Next-Item Recommendation. In: AAAI 2018.

Summary of attention-based approaches

Applicable scenario

- Heterogeneous, noisy, or long sessions.

Pros

- Identify and highlight important information.

Cons

- Cannot capture sequential information.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
  - Advanced deep model based approaches
    - Attention model
    - Memory network
    - Mixture model
    - Reinforcement learning
    - Contrastive learning

Memory networks based approaches

Main idea

- Memory network can capture the dependency between any historical interaction in a session and the next interaction directly by introducing an external memory matrix.

General framework

1. A memory matrix that maintains the embeddings of interactions in a session context �, and it will be updated along with the incoming interactions by "write" operations to maintain the latest information.

$$\mathbf{M}^c \leftarrow write(\mathbf{M}^c, \mathbf{o}_i),$$

2. During the prediction step, the relevant information is carefully "read" from the maintained memory matrix to form an informative session context embedding.

$$\mathbf{e}_c = read(\mathbf{M}^c, \mathbf{o}_{tg}),$$

An example of memory network based approaches

Intuition

- Enhance **model representation** with memory network to maintain user interaction history.

The Model

1. Memory enhanced user embedding

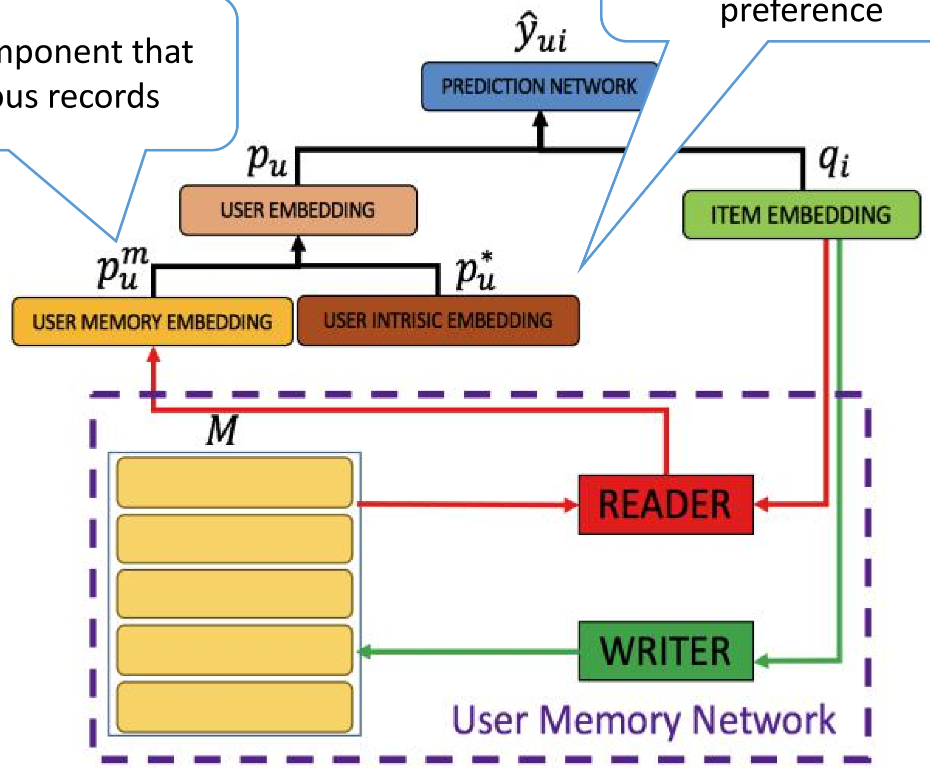
$$\boldsymbol{p}_u^m = READ(\boldsymbol{M}^u,\ \boldsymbol{q}_i) \quad \boldsymbol{p}_u = MERGE(\boldsymbol{p}_u^*,\ \boldsymbol{p}_u^m)$$

$$MERGE(x, y) \doteq x + \alpha y = \boldsymbol{p}_u^* + \alpha \boldsymbol{p}_u^m$$

2. Prediction function $\hat{y}_{ui} = PREDICT(\boldsymbol{p}_u,\ \boldsymbol{q}_i)$
3. Memory updating $\boldsymbol{M}^u \leftarrow WRITE(\boldsymbol{M}^u,\ \boldsymbol{q}_i)$

user's memory component that encodes her previous records

a free vector used to encode her intrinsic preference



A memory network based SBRS

Chen X and et al. Sequential recommendation with user memory networks. In: WSDM 2018.

Summary of memory network based approaches

Applicable scenario

- Long, incremental or noisy sessions

Pros

- Dynamically store the latest information

Cons

- Cannot capture sequential information

Outline: Section 3

Section 3

Approaches

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
  - Advanced deep model based approaches
    - Attention model
    - Memory network
    - Mixture model
    - Reinforcement learning
    - Contrastive learning

Mixture model based approaches

Main idea

A mixture model based SBRS mainly builds a compound model containing multiple sub-models to take the advantage of different sub-models.

General framework

1. Learn different types of dependencies using different sub-models.
2. Carefully integrate the learned different dependencies for accurate session-based recommendations.

An example of mixture models

The Model

1. Employ item embeddings and context features (optional) as inputs: $x_\tau = [Q_\tau \oplus c_\tau^{\text{in}}]$,
2. A feed-forward layer F^{in} is used to map the raw context features and item embeddings to the same high-dimensional

$$Z_\tau^{\text{in}} = \{z_i^{\text{in}}\}_{i=1\cdots\tau} \text{ where } z_\tau^{\text{in}} = F^{\text{in}}(x_\tau)$$

3. Employ three different sequence models (encoders) in conjunction

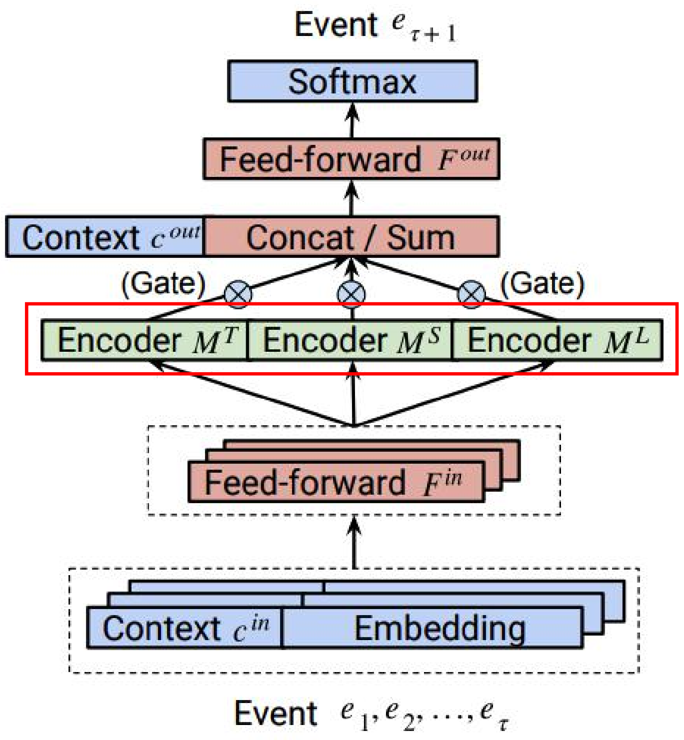
$$SE_\tau^T = M^T(Z_\tau^{\text{in}}),\ SE_\tau^S = M^S(Z_\tau^{\text{in}}),\ SE_\tau^L = M^L(Z_\tau^{\text{in}}),$$

4. Aggregate all sequence encoders' results by weighted-concatenate

$$SE_\tau = (G_\tau^T \times SE_\tau^T) \oplus (G_\tau^S \times SE_\tau^S) \oplus (G_\tau^L \times SE_\tau^L),$$

5. Fuse it with the annotation's context features (optional)

$$z_\tau^{\text{out}} = F^{\text{out}}([SE_\tau \oplus c_\tau^{\text{out}}])$$



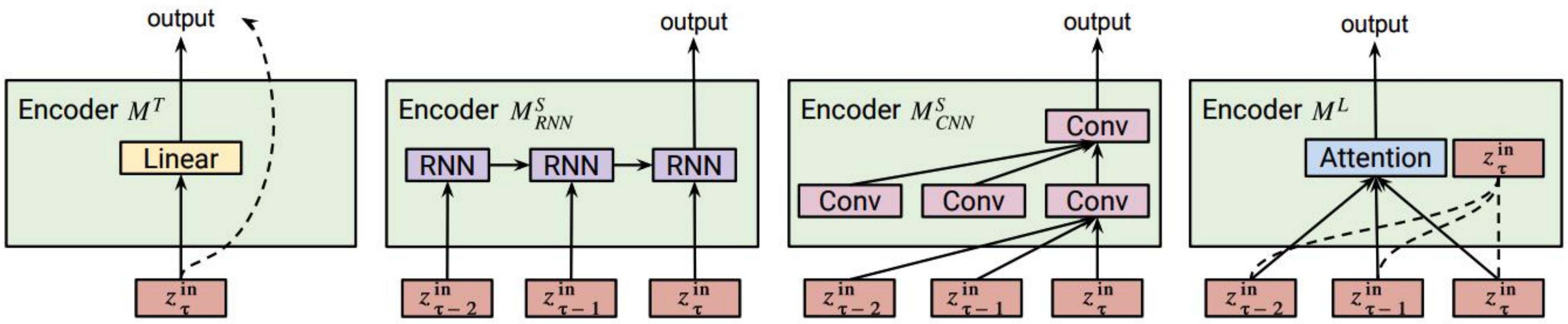
A mixture model based SBRS

Tang J and et al. Towards neural mixture recommender for long range dependent user sequences. In: WWW 2019.

An example of mixture models (Cont')

Intuition

Four different sub-models are used to capture **different ranges of temporal dependencies** in a user-item interaction sequence to provide an informative representation for the sequence.



(a) Tiny-range encoder **(b) Short-range encoder (RNN)** **(c) Short-range encoder (CNN)** **(d) Long-range encoder**

Different sub-models to learn different types of dependencies in a session/sequence

Tang J and et al. Towards neural mixture recommender for long range dependent user sequences. In: WWW 2019.

Summary of mixture model based approaches

Applicable scenario

- Heterogeneous, noisy sessions/sequences

Pros

- Model different types of dependencies, e.g., long and short term dependencies.

Cons

- Relatively complex and costly.

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
  - Advanced deep model based approaches
    - Attention model
    - Memory network
    - Mixture model
    - Reinforcement learning
    - Contrastive learning

Reinforcement learning (RL) based approaches

Main idea

Reinforcement learning approaches for SBRSs generally model the interactions between a user and an RS in a session as a Markov Decision Process (MDP).

General framework

1. Calculate the state-specific weight parameters by mapping state sa_t to a weight matrix W_t

$$f_t : sa_t \rightarrow \mathbf{W}_t .$$

2. Calculate the score of each candidate item using the score function f_s and then select items with the highest score for recommendations

$$score(v_i) = f_s(v_i, \mathbf{W}_t).$$

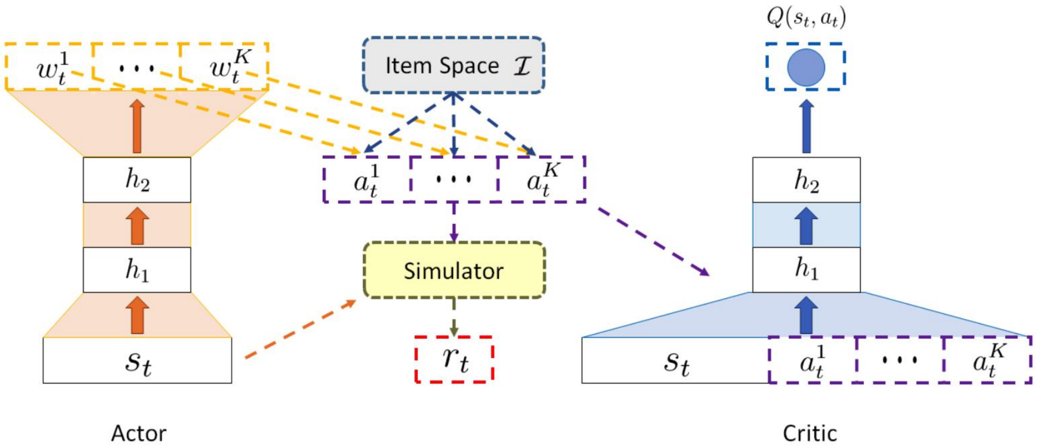
3. Calculate the action value E (sa_t , ac_t) of the potential action ac_t to judge whether ac_t matches the current state sa_t or not

$$E^*(sa_t, ac_t) = \mathbb{E}_{sa_{t+1}}[Re_t + df\ max_{ac_{t+1}} E^*(sa_{t+1}, ac_{t+1}) | sa_t, ac_t].$$

A example of reinforce learning based approaches

The Model

1. Build an online user-agent **interact-tion environment** simulator.
2. Propose an **Actor-Critic** based reinf-orcement learning framework under this setting.
3. Utilize the framework for **listwise recommendations.**



The model structure of an RL based approach

Zhao X and et al. Deep reinforcement learning for list-wise recommendations. In: Arxiv 2018.

Summary of reinforcement learning approaches

Applicable scenario

- Dynamic, incremental sessions

Pros

- Interactive process, consider the future effect of actions

Cons

- Hard to simulate the interactive environment

Outline: Section 3

Section 3

Approache
s

- An Overview of Different Classes of Approaches
- Conventional approaches
- Latent representation based approaches
- Deep learning based approaches
  - Basic deep neural network based approaches
  - Advanced deep model based approaches
    - Attention model
    - Memory network
    - Mixture model
    - Reinforcement learning
    - Contrastive learning

Contrastive learning based approaches

Main idea

- Utilizing the contrastive learning framework to derive **self-supervision signals** from the original user behavior sequences and extract **more meaningful user behavior patterns**.

Model

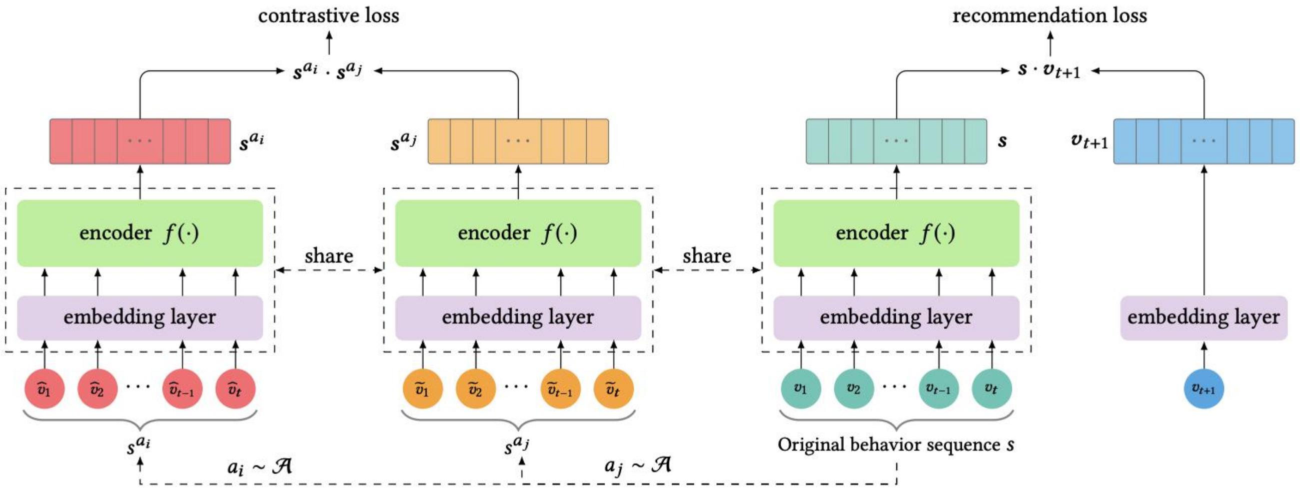
1. **Data augmentation module**

   Transform each data sample randomly into two correlated **instances** (positive or negative pair) to **augment** the original sequences.

2. **User representation encoder**

   Utilize a neural network to extract information from the **augmented sequences**.

$$s_u^a = f(s_u^a)$$



The model structure of an contrastive learning based approach

Xie X and et al. Contrastive learning for sequential recommendation. In: SIGIR 2021.

Contrastive learning based approaches (Cont')

Model

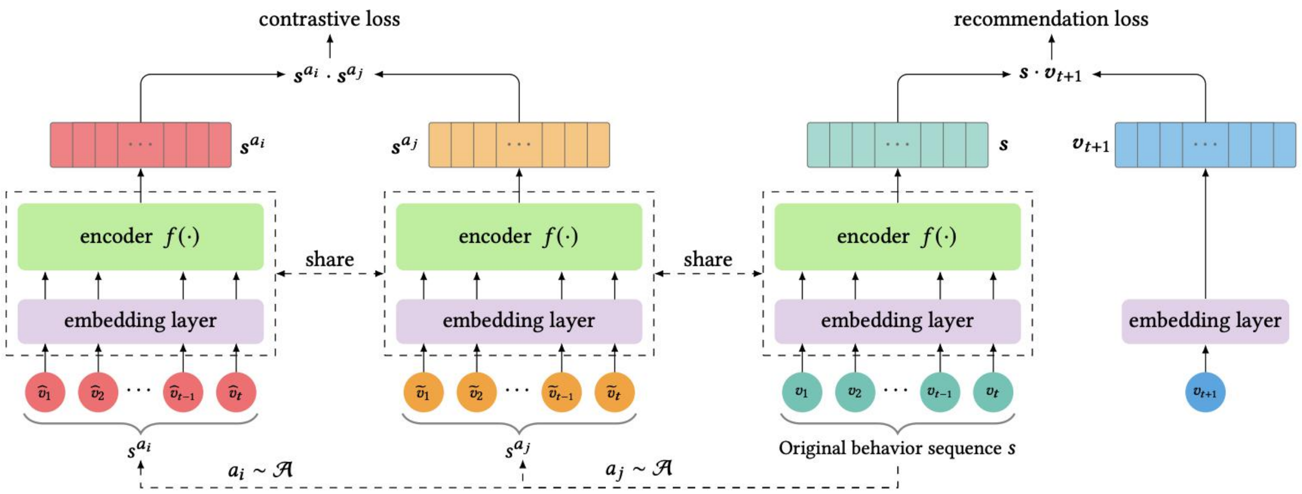
3. Contrastive loss function

Contrastive loss function is applied to distinguish whether the two representations are derived from the same user historical sequence.

$$\mathcal{L}_{\mathrm{cl}}(s_u^{a_i}, s_u^{a_j}) = -\log \frac{\exp\left(\mathrm{sim}(s_u^{a_i}, s_u^{a_j})\right)}{\exp\left(\mathrm{sim}(s_u^{a_i}, s_u^{a_j})\right) + \sum_{s^- \in S^-} \exp\left(\mathrm{sim}(s_u^{a_i}, s^-)\right)}.$$

$(s_u^{a_i}, s_u^{a_j})$ is the positive pair

S^- is the negative samples



The model structure of an contrastive learning based approach

Xie X and et al. Contrastive learning for sequential recommendation. In: SIGIR 2021.

Comparison of deep neural network based SBRS approaches

Table 8. A comparison of different classes of deep neural network approaches for SBRSs

Approach		Applicable scenario	Pros	Cons	Typical work
Basic deep neural networks	RNN	Long and rigidly ordered sessions	Model long-term and high-order sequential dependencies	The rigid order assumption is too strong for session data	[7],[12],[43], [44],[85],[92], [96],[107],[148]
	MLP	Unordered sessions, sessions with multi-aspects (e.g., static and dynamic features) to be combined	A simple structure, project sparse features to dense ones, learn the combination of different parts	Cannot model complex sessions, e.g., ordered, heterogeneous sessions	[18],[52],[106], [136]
	CNN	Flexible-ordered, heterogeneous or noisy sessions	Robust, no rigid order assumption, capture the union-level collective dependency	Relatively high complexity	[83],[109],[112], [146],[149]
	GNN	Complex sessions with complex transitions, e.g., repeat interactions	Model the complex transitions among interactions	Complex and costly	[88],[89],[134], [138],[141],[147]
Advanced models	Attention	Heterogeneous, noisy, or long sessions	Identify and highlight important information	Cannot capture sequential information	[35],[59],[65], [69],[125],[127], [145],[152]
	Memory	Long, incremental or noisy sessions	Dynamically store the latest information	Cannot capture sequential information	[16],[74],[97], [104],[121]
	Mixture	Heterogeneous, noisy sessions	Model different types of dependencies, e.g., long and short term dependencies	Relatively complex and costly	[108],[132]
	Generative	Dynamic, incremental sessions	Close to the practical session formation	Complex	[130],[135],[150]
	RL	Dynamic, incremental sessions	Interactive process, consider the future effect of actions	Hard to simulate the interactive environment	[45],[156],[157]

- The end of Section 3
- 7mins break

Outline

SIGIR 2022
Madrid

Sect. 1 Introduction	Sect. 2 Problem Statement & Challenges	Sect. 3 Approaches	**Sect. 4 Applications & Opportunities**
30mins, by Shoujin	30mins, by Qi + Break (15mins)	80mins, by Shoujin, Zhongyuan + Break (15mins)	30mins, by Liang + QA (10mins)
• Introduction to RS • Introduction to SBRS • Classification of SBR • Sequential RS vs. Session-based RS	• Problem statement • Characteristics and challenges	• Conventional approaches • Latent representation • Deep learning	• Applications • Algorithms and datasets • Future directions • Conclusions

Outline: Section 4

Section 4

Applications and opportunitie
s

- Applications
- Algorithms and datasets
- Future directions
- Summary

SBRS applications

SIGIR 2022 Madrid

Table 9. A summary of SBRS applications

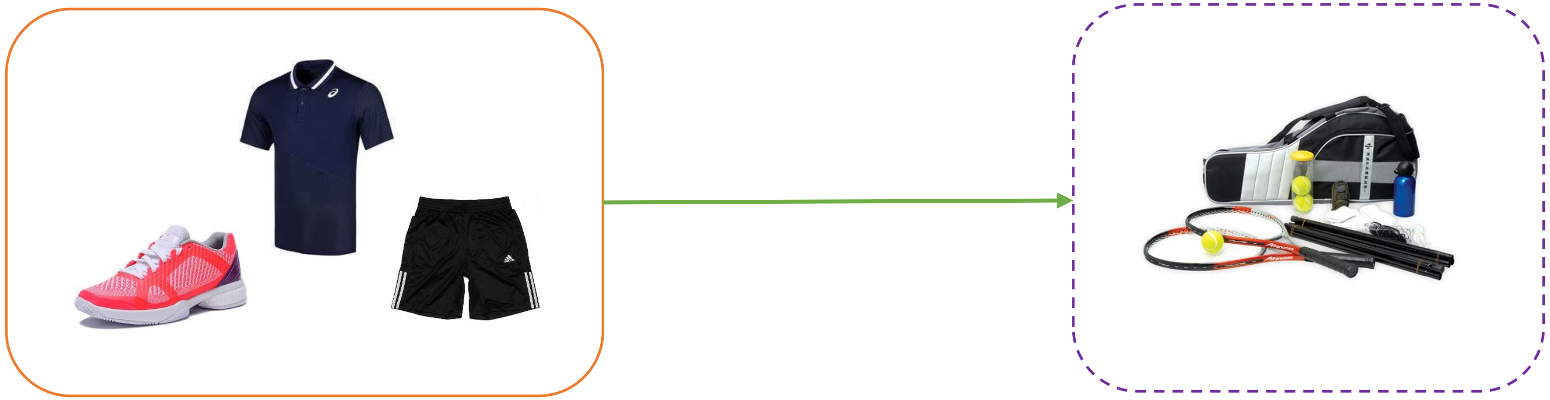
Category		Application domain	Application scenario	Typical work
Conventional application	Product recommendation	E-commerce	Next-item/basket recommendation	[43],[49],[51],[70], [123],[130],[138],[143]
	Content recommendation	Media, entertainment	Next news/web-page/song/movie /video recommendation	[23],[55],[77],[85], [106],[108],[116],[157]
	Service recommendation	Tourism	Next-POI recommendation	[17],[66]
Emerging application	Service recommendation	Finance	Next-trading recommendation	[29],[140]
		Healthcare	Next-treatment recommendation	[36]

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR)

SBRS for product recommendation

- Most SBRS focus on the conventional applications, especially in E-commerce
  - E.g., recommending the next item or next basket of items on online shopping platforms

Next item recommendation

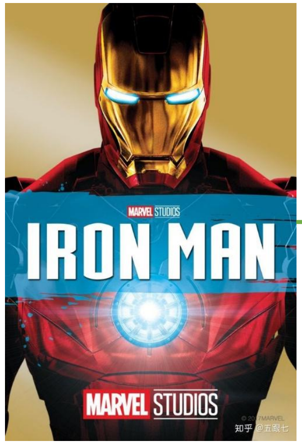


Next basket recommendation

SBRS for content recommendation

Application domain

- Media, entertainment



Next movie recommendation

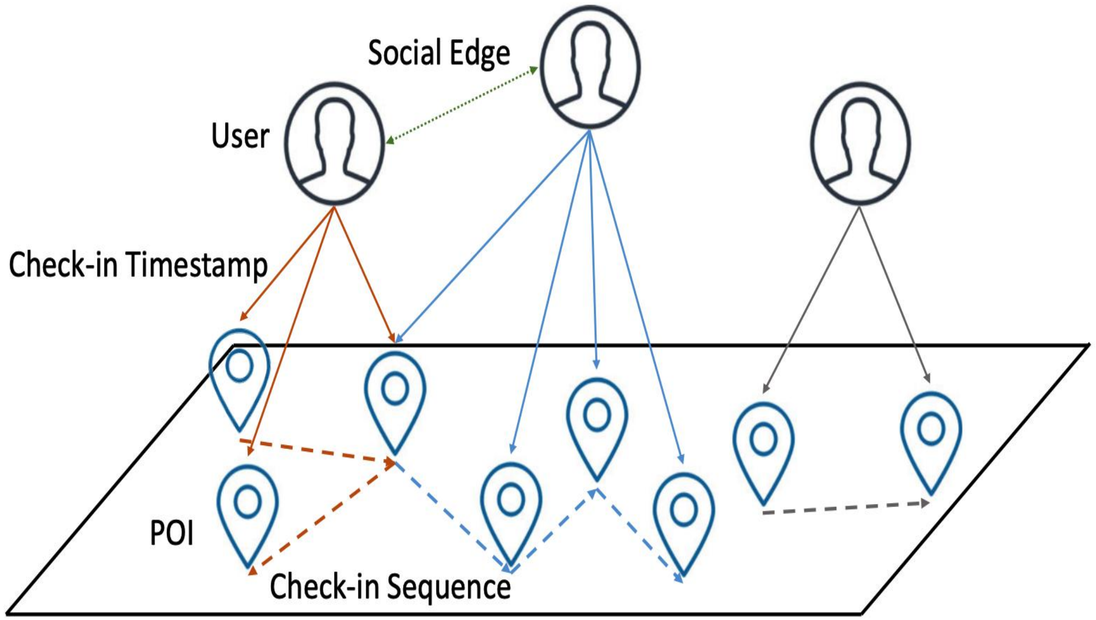
In series

Similar topics

SBRS for service recommendation

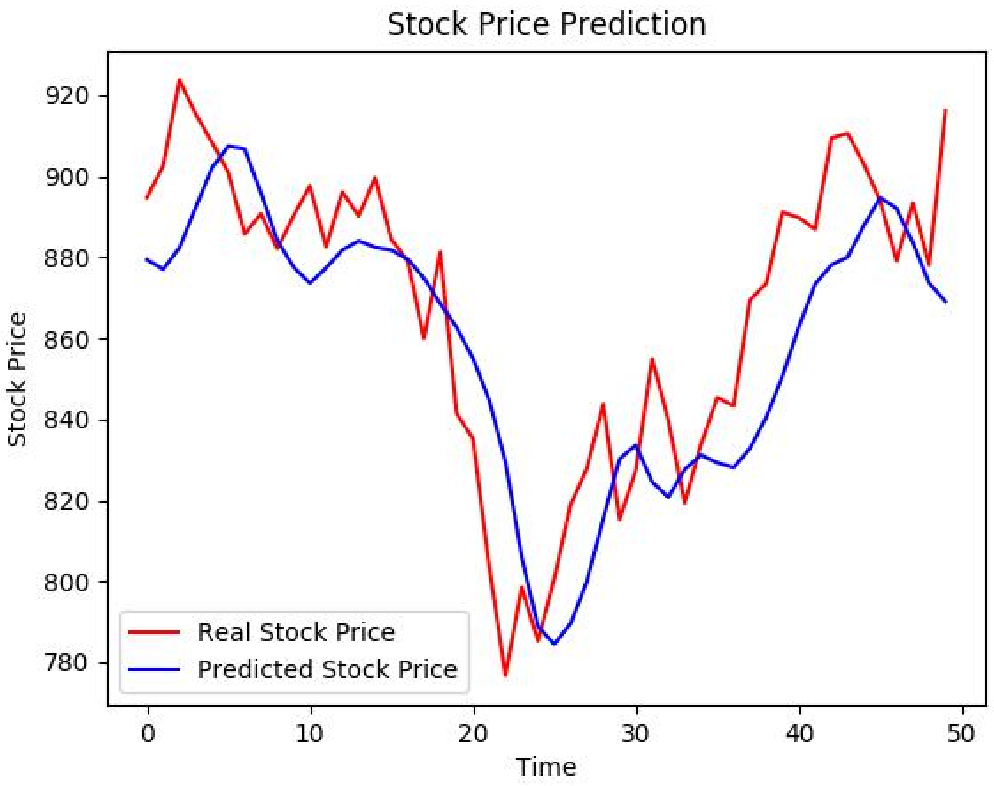
Application

- Next-POI recommendation in tourism domain

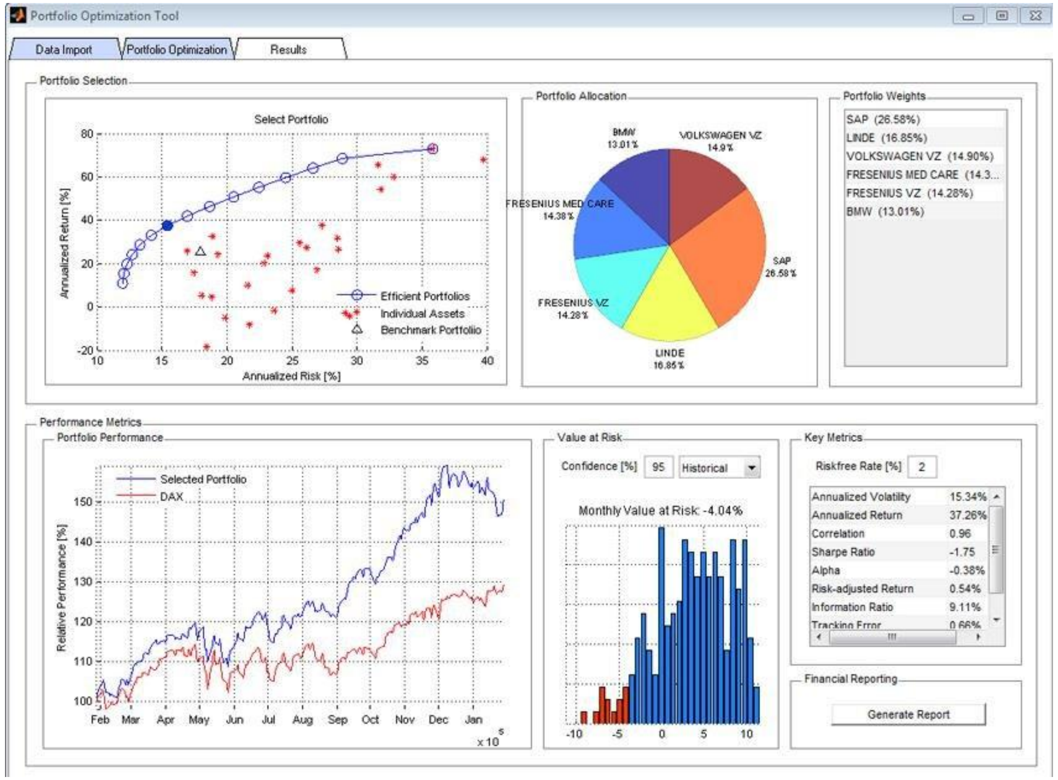


Emerging application — Finance

- Next trading strategy recommendation
- Optimal portfolio recommendation



“Next-item recommendation” problem

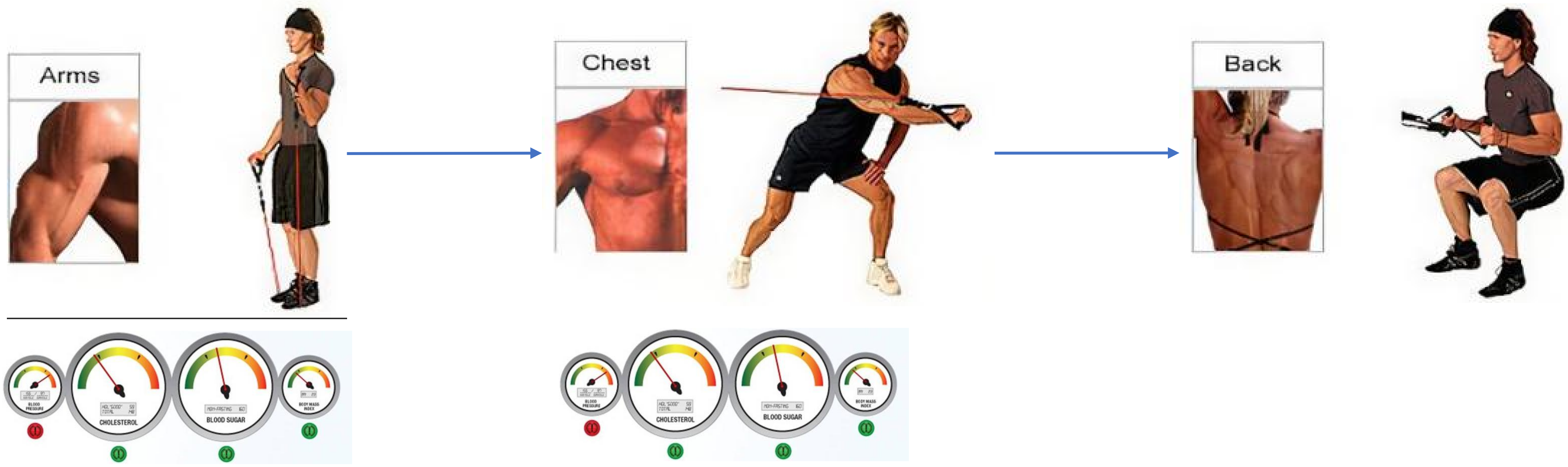


“Next-basket recommendation” problem

Emerging application — Healthcare

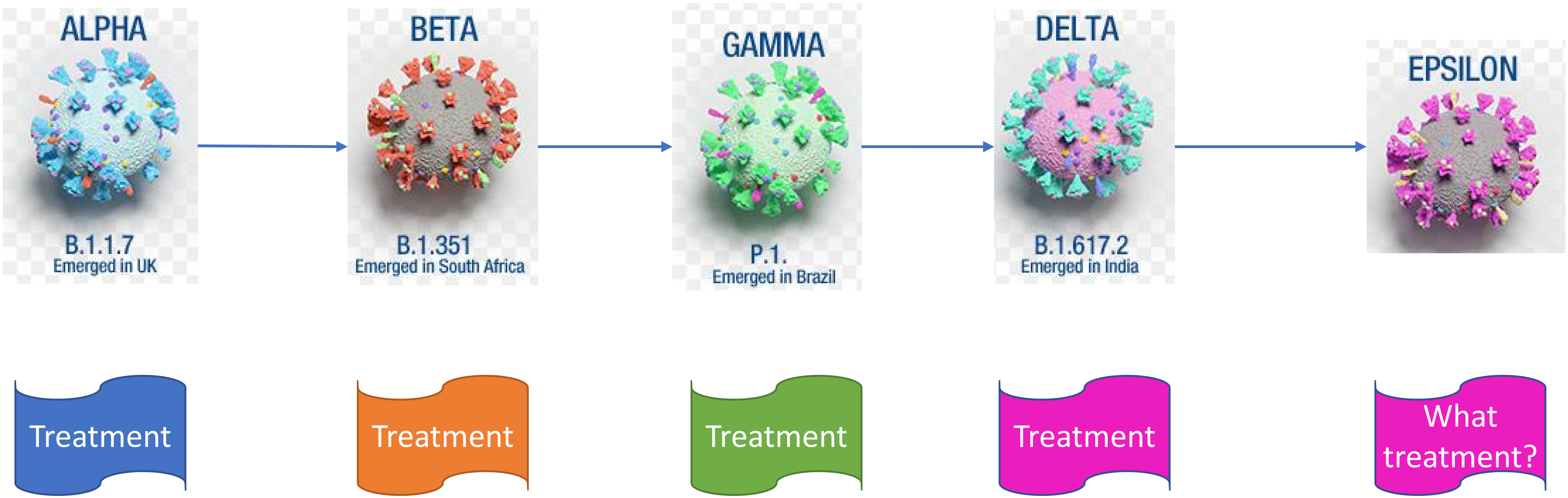
- Fitness program recommendations

Next training item



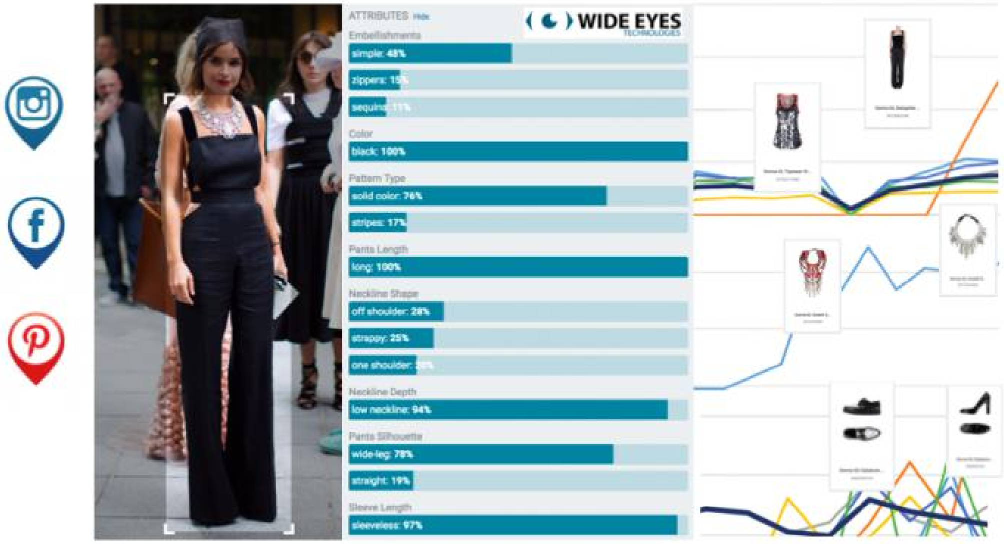
Emerging application — Healthcare

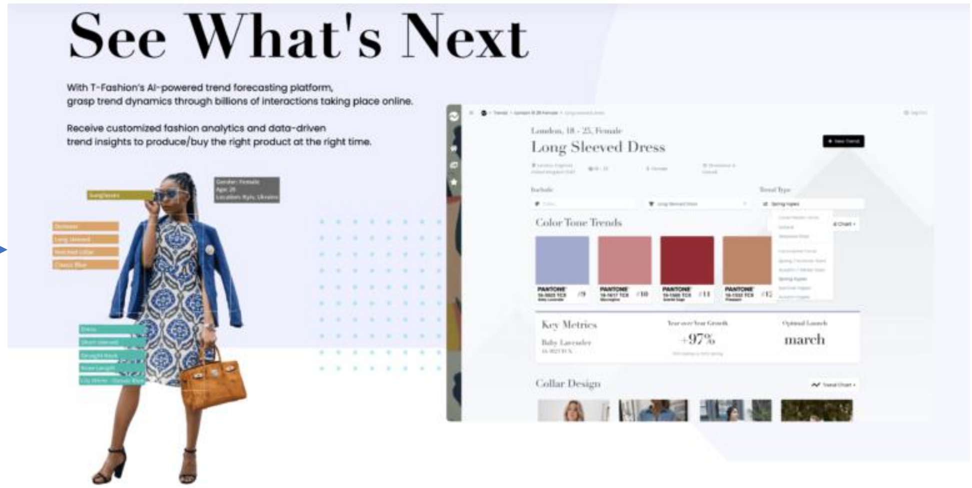
- Covid-19 treatment recommendations



Emerging application — Fashion

- Forecasting the fashion trend



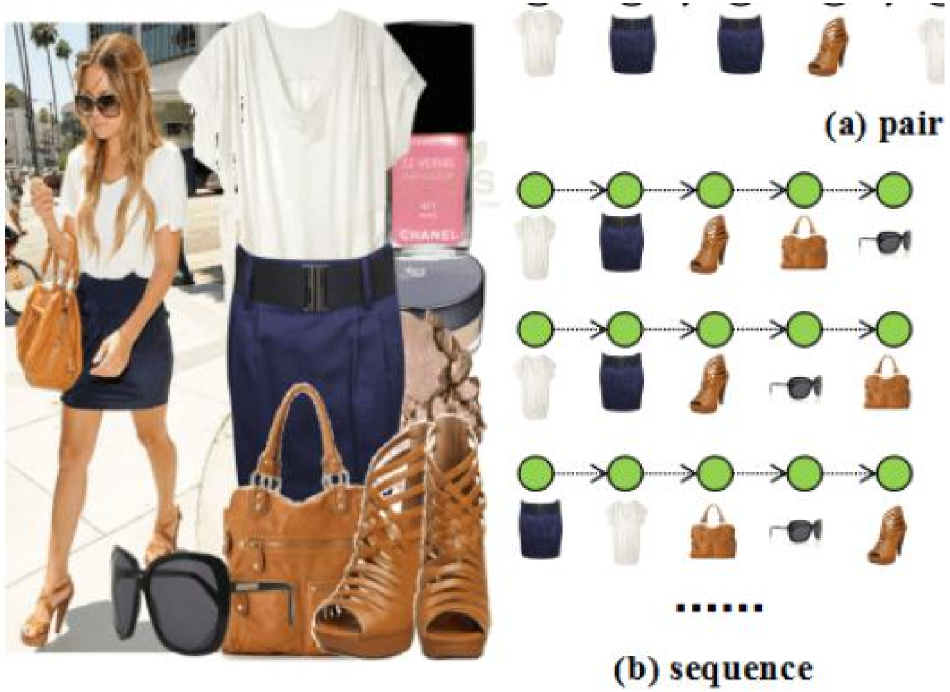


https://blog.wideeyes.ai/2017/10/09/artificial-intelligence-fashion-industry/

https://www.tucmag.net/fashion/fashion-trend-forecasting-with-ai/

Emerging application — Fashion

- Personalized apparel



Outline: Section 4

Section 4

Applications and opportunitie s

- Applications
- Algorithms and datasets
- Future directions
- Summary

SBRS algorithms

Algorithm	Task	Utilized model	Venue	Link
TBP [34]	Next basket	Pattern mining	ICDM 2017	https://github.com/GiulioRossetti/tbp-next-basket
UP-CF [26]	Next basket	KNN	UMAP 2020	https://github.com/MayloIFERR/RACF
FPMC [95]	Next basket	Markov chain	WWW 2010	https://github.com/khesui/FPMC
HRM [123]	Next basket	Distributed representation	SIGIR 2015	https://github.com/chenghu17/Sequential_Recommendation
DERAM [148]	Next basket	RNN	SIGIR 2016	https://github.com/yihong-chen/DREAM
Beacon [57]	Next basket	RNN	IJCAI 2019	https://github.com/PreferredAI/beacon
TIFUKNN [47]	Next basket	KNN	SIGIR 2020	https://github.com/HaojiHu/TIFUKNN
AR [70]	Next item	Association rule	UMUAI 2018	https://github.com/rn5l/session-rec
BPR-MF [70, 94]	Next item	Latent factor	UAI 2009	https://github.com/rn5l/session-rec
IKNN [51]	Next item	KNN	RecSys 2017	https://github.com/rn5l/session-rec
SKNN [51]	Next item	KNN	RecSys 2017	https://github.com/rn5l/session-rec
FOSSIL [39]	Next item	Latent factor	ICDM 2016	https://github.com/rn5l/session-rec
SMF [70]	Next item	Latent factor	UMUAI 2018	https://github.com/rn5l/session-rec
GRU4Rec [42, 43]	Next item	RNN	ICLR 2016	https://github.com/rn5l/session-rec
STAMP [65]	Next item	Attention	KDD 2018	https://github.com/rn5l/session-rec
NARM [59]	Next item	Attention, RNN	CIKM 2017	https://github.com/rn5l/session-rec
SR-GNN [138]	Next item	GNN	AAAI 2019	https://github.com/CRIPAC-DIG/SR-GNN
CSRM [121]	Next item	Memory network	SIGIR 2019	https://github.com/wmeirui/CSRM_SIGIR2019
RepeatNet [93]	Next item	RNN, Attention	AAAI 2019	https://github.com/PengjieRen/RepeatNet
DGRec [104]	Next item	GNN	WSDM 2019	https://github.com/DeepGraphLearning/RecommenderSystems/tree/master/socialRec
FGNN [89]	Next item	GNN	CIKM 2019	https://github.com/RuihongQiu/FGNN
TAGNN [147]	Next item	GNN	SIGIR 2020	https://github.com/CRIPAC-DIG/TAGNN
LESSR [14]	Next item	GNN	KDD 2020	https://github.com/twchen/lessr
MKM-SR [73]	Next item	RNN, GNN	SIGIR 2020	https://github.com/ciecus/MKM-SR

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR)

SBRS datasets

- [illegible]

Domain	Dataset	# sessions	# interactions	# items	Avg. session length	Reference
E-commerce	RSC 2015[1]	1,375,128	5,426,961	28,582	3.95	[5],[70],[132]
	Tmall[2]	1,774,729	13,418,695	425,348	7.56	[70],[132]
	Tafeng[3]	19,538	144,777	5,263	7.41	[127],[132]
	Diginetica[4]	780,328	982,961	43,097	5.12	[138]
	RetailRocket[5]	59,962	212,182	31,968	3.54	[70]
News	CLEF 2017[6]	1,644,442	5,540,486	742	3.37	[70]
	Globo[7]	1,031,167	2,930,849	13,092	2.84	[21]
	Adressa 16G[8]	2,215	62,908	6,765	28.4	[151]
Music	Last.FM[9]	169,576	2,887,349	449,037	17.03	[21]
	30Music[10]	31,351,954	2,764,474	210,633	11	[70],[113]
	NowPlaying[11]	27,005	271,177	75,169	10.04	[70]
POI	Gowalla[12]	-[0]	245,157	6,871	-	[28]
	Foursquare[13]	-	155,365	2,675	-	[28]

[0] Raw POI data does not have a session structure, researchers often manually build sessions by treating a user's check-ins in a single day as a session [35].

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR)

Outline: Section 4

Section 4

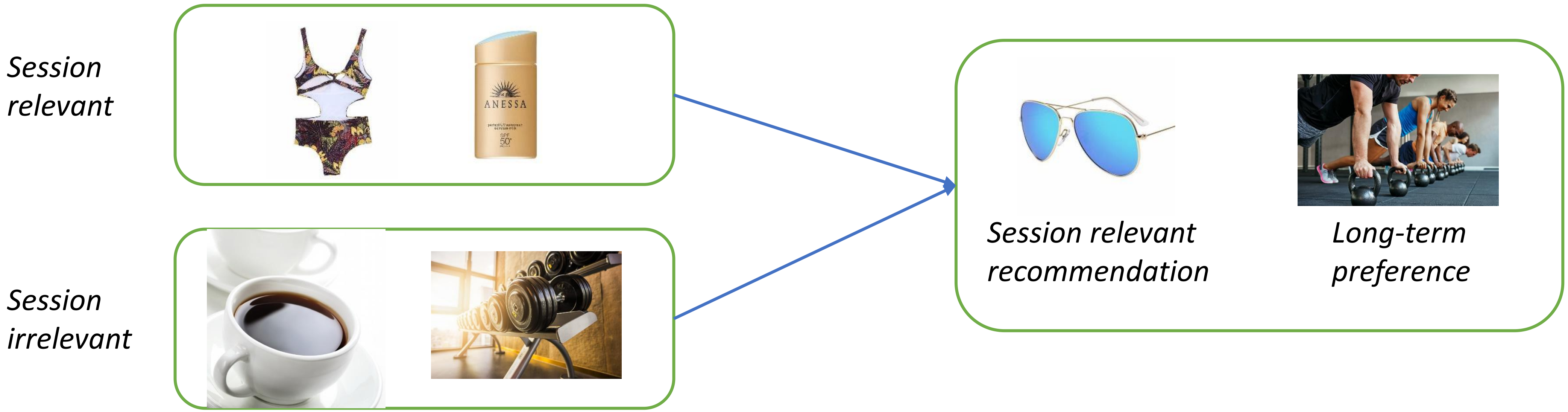
Applications and opportunities

- Applications
- Algorithms and datasets
- Future directions
- Summary

Session-based recommendations with session-irrelevant user preference

Current SBRSs usually ignore users' long-term preferences and consumption habits.

- How to incorporate users' explicit/implicit long-term preference into SBRS?



Session-based recommendations with contextual factors

SIGIR 2022
Madrid

Context refers to the specific internal and external environment when a user makes choice, such as weather, location, time and recent trend.

- How to incorporate more contextual factors into SBRSs?

Recent transaction

Travel package

Breakfast

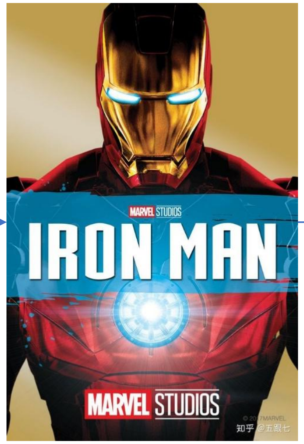
Dinner

Session-based recommendations with cross-domain information

In the real life, users always take activities over multiple domains, e.g., book domain, movie domain and fashion domain.

- How to perform SBR over multiple domains?

Comics



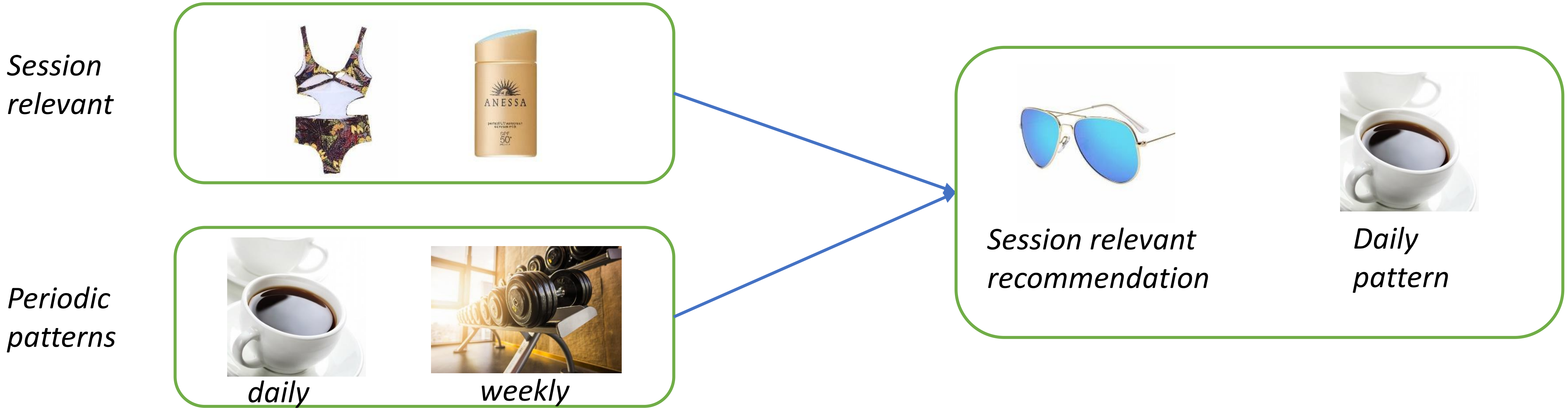
Movies

Accessories

Session-based recommendations by considering user behavior patterns

More user behaviour patterns, such as repeat consumption and periodic consumption, may be considered in SBRS modeling.

- How to incorporate user behaviour patterns to improve SBRS?



Session-based recommendations with constraints

In real world, there are always some constraints needed to be considered when users make choices.

- How to make recommendations under some constraints according to user preferences?

Recent transaction

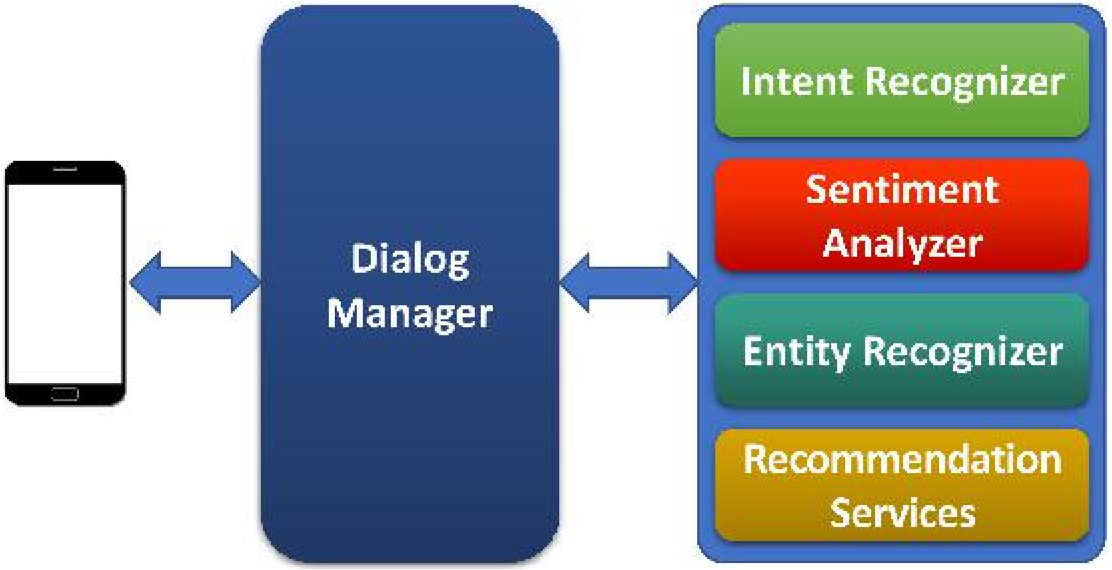
Travel package

s.t. Budget < \$3000, Days = 6, Comfortable > 3

Interactive session-based recommendations

Interactive RSs enable the user to steer the received recommendations in the desired direction through explicit interaction with the systems.

- How to model SBRS with an efficient interactive mechanism?



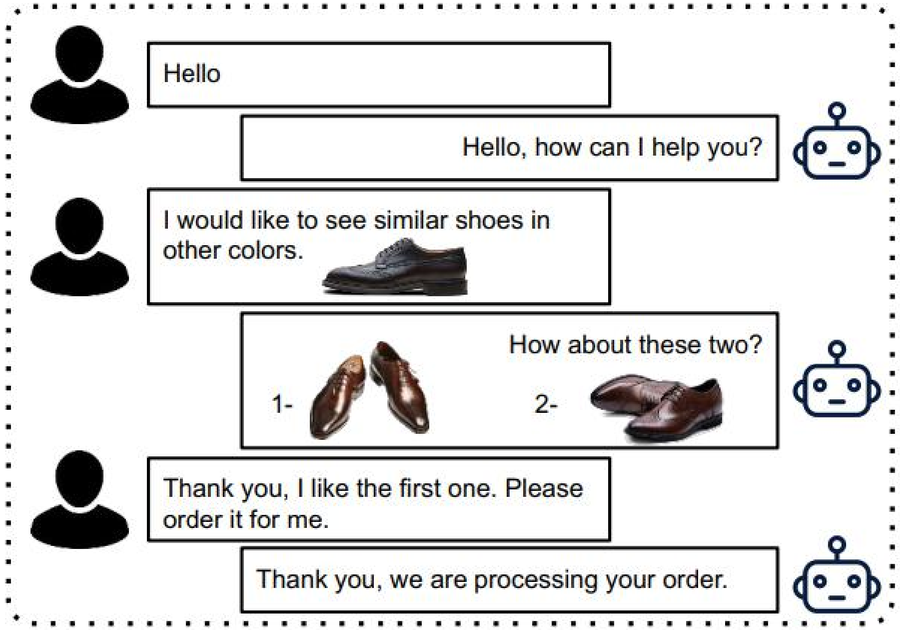
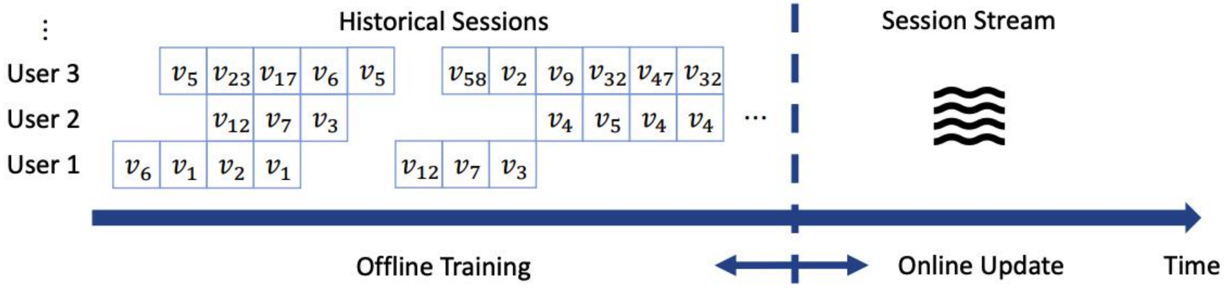


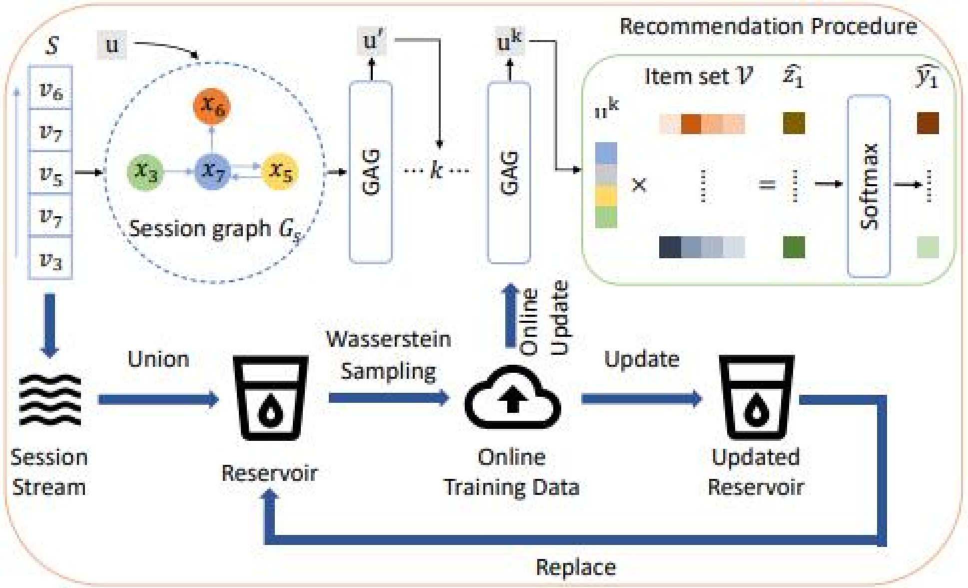
Figure 1: An example conversation between a user and an agent in a Conversational Recommender System in e-Commerce domain.

Online/streaming session-based Recommendations

Session data usually comes incrementally in a streaming scenario.

- How to effectively learn users' dynamic preferences in an online and streaming scenario for better recommendation?





Qiu R and et al. "Gag: Global attributed graph neural network for streaming session-based recommendation." In: SIGIR. 2020.

Outline: Section 4

Section 4

Applications and opportunitie s

- Applications
- Algorithms and datasets
- Future directions
- Summary

Summary

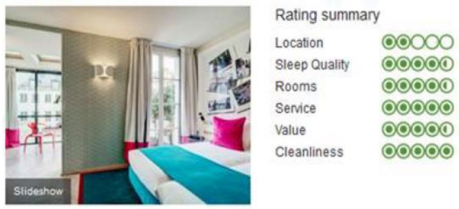
Apps: Tourism | Media | Entertainment | E-Commerce | … …

Systems: Next-item recommendation | Next-basket recommendation

Models: Factorization Model | MLP | CNN | GNN | RNN | Transformer

Areas: Regression analysis | Natural language processing | Sentiment anaysis | Computer vision | Graph learning | Time series analysis

Data:




282 Reviews Ordered By: Helpfulness

10/10

Gracias Pixar

mryohual 27 October 2017

Im Mexican and all i can say is Thanks you Pizax,I just saw this movie and i just remembered all my childhood with my grandparents, Mexico was represented beautifully,the music, the colors. This movie touched me in my soul and i cried a lot, i created an account just to say how good pixar made everything, again gracias pixar por tan hermosa pelicula.

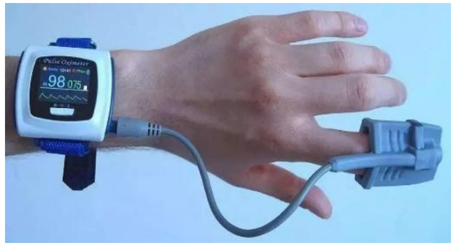
Vision

Healthcare | Education | Manufacturing | Economy | Fashion

Omniscient sequential recommendation services

Interdisciplinary learning

psychology | biology | sociology | finance | economics | neuroscience | ...



Thanks for your attention (Q&A)

- Tutorial link:
  https://neurec22.github.io/SRS&SBRS/

- WeChat group

  Scan to join before July 14

The end!

- We are open to any kinds of collaborations!

- Dr. **Shoujin Wang**, Research Fellow, RMIT University, Australia
- Homepage: https://shoujinwang1.github.io/
- E-mail: shoujin.wang@mq.edu.au

Dr. **Liang Hu**, Professor, Tongji University, China

Homepage: https://sites.google.com/view/lianghu/home

E-mail: lianghu@tongji.edu.cn