Simulation for Recommendations in Dynamic and Interactive Environments

SIGIR-AP 2023

Maarten de Rijke

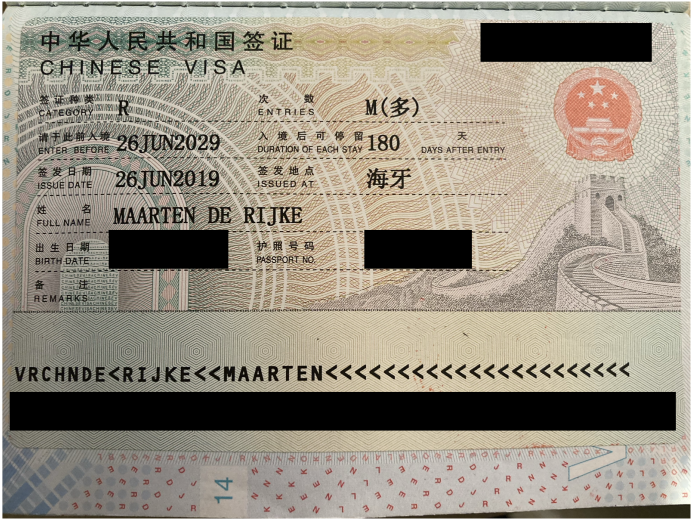
November 27, 2023

University of Amsterdam

Congratulations!

SIGIR-AP 2023

Thank you!



中华人民共和国签证
CHINESE VISA

[redacted]

签证种类 CATEGORY R

次数 ENTRIES M(多)

请于此前入境 ENTER BEFORE 26JUN2029

入境后可停留 DURATION OF EACH STAY 180 天 DAYS AFTER ENTRY

签发日期 ISSUE DATE 26JUN2019

签发地点 ISSUED AT 海牙

姓名 FULL NAME MAARTEN DE RIJKE

出生日期 BIRTH DATE [redacted]

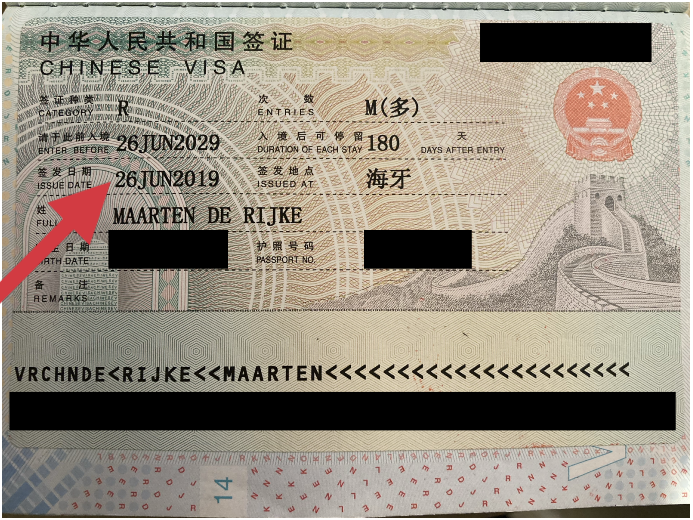
护照号码 PASSPORT NO. [redacted]

备注 REMARKS

VRCHNDE<RIJKE<<MAARTEN<<<<<<<<<<<<<<<<<<<<<<

[redacted]

14



中华人民共和国签证
CHINESE VISA

[redacted]

签证种类 CATEGORY R

次数 ENTRIES M(多)

请于此前入境 ENTER BEFORE 26JUN2029

入境后可停留 DURATION OF EACH STAY 180 天 DAYS AFTER ENTRY

签发日期 ISSUE DATE 26JUN2019

签发地点 ISSUED AT 海牙

姓 FULL MAARTEN DE RIJKE

生日期 IRTH DATE [redacted]

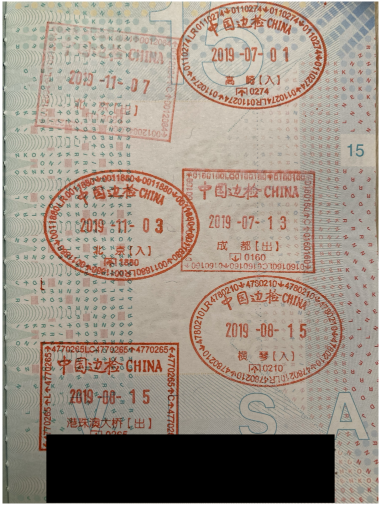
护照号码 PASSPORT NO. [redacted]

备注 REMARKS

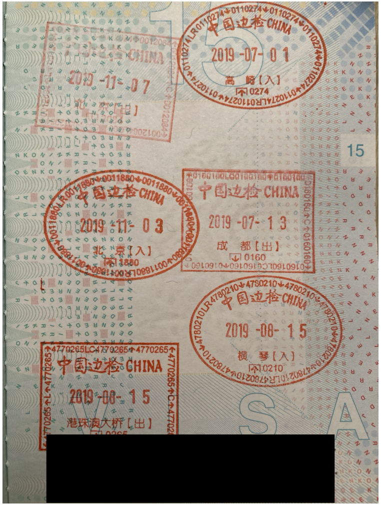
VRCHNDE<RIJKE<<MAARTEN<<<<<<<<<<<<<<<<<<<<<<

[redacted]

14


中国边检 CHINA
2019-11-07
中国边检 CHINA
2019-07-01
高崎【入】
0274
15
中国边检 CHINA
2019-11-03
北京【入】
1880
中国边检 CHINA
2019-07-13
成都【出】
0160
中国边检 CHINA
2019-08-15
横琴【入】
0210
中国边检 CHINA
2019-08-15
港珠澳大桥【出】
0265
VISA

[redacted]



中国边检 CHINA
2019-11-07
北京［出］

中国边检 CHINA
2019-07-01
高鸣［入］
0274

中国边检 CHINA
2019-11-03
北京［入］
1880

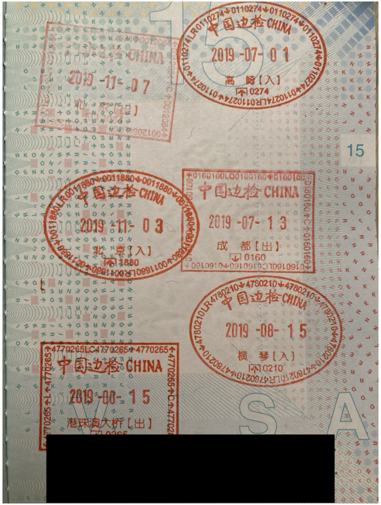
中国边检 CHINA
2019-07-13
成都［出］
0160

中国边检 CHINA
2019-08-15
横琴［入］
0210

中国边检 CHINA
2019-08-15
港珠澳大桥［出］
0265

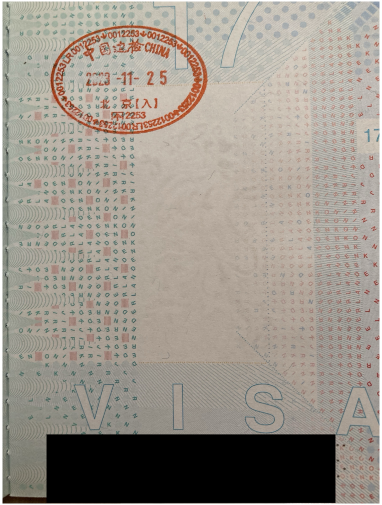
[redacted]

[redacted]



[redacted]

[redacted]



[redacted]

Acknowledgements

Talk based on recent and ongoing joint work with Dongyoon Hwang, Jean-Michel Renders, Onno Zoeter, Philipp Hager, Romain Deffayet, Thibaut Thonet, and Vassilissa Lehoux

Outline

1. Background
2. Evaluation
3. Simulators
4. Fish
5. Experimental results
6. Conclusion

Part 1. Background

Semantic aspects of recommendation

Recommender systems

- Goal: Estimate user preference and item value based on features about users (interacted items, device and user features, . . .) and items (text, thumbnail, . . .)

Semantic aspects of recommendation

Recommender systems

- Goal: Estimate user preference and item value based on features about users (interacted items, device and user features, . . .) and items (text, thumbnail, . . .)

- Learn semantic information that explains why user is attracted to item, usually leveraging user features, item content, logged interactions
- But there's more than the semantic aspects

The dynamic aspect of recommendation

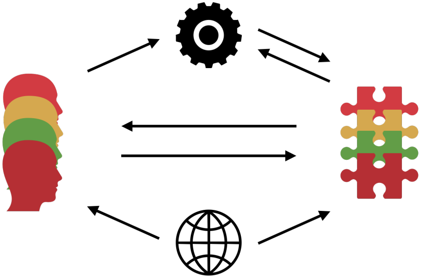
Converting semantic understanding of users and items into increased value for user, content providers, and other potential stakeholders

- E.g., click-through rate, user satisfaction, retention rate, fairness, . . .

The dynamic aspect of recommendation

Converting semantic understanding of users and items into increased value for user, content providers, and other potential stakeholders

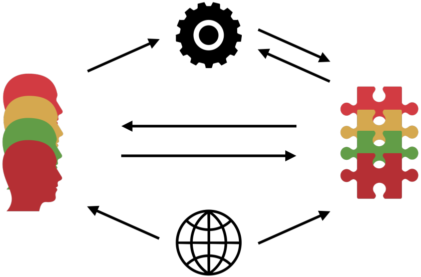
- E.g., click-through rate, user satisfaction, retention rate, fairness, . . .
- Users, items, system, external factors



The dynamic aspect of recommendation

Converting semantic understanding of users and items into increased value for user, content providers, and other potential stakeholders

- E.g., click-through rate, user satisfaction, retention rate, fairness, . . .
- Users, items, system, external factors

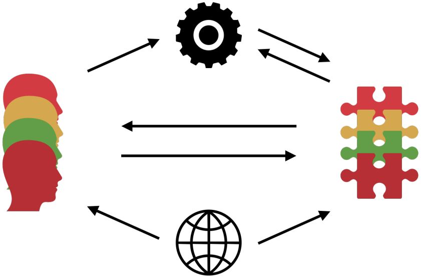


- Beyond-accuracy goals

The dynamic aspect of recommendation

Converting semantic understanding of users and items into increased value for user, content providers, and other potential stakeholders

- E.g., click-through rate, user satisfaction, retention rate, fairness, . . .
- Users, items, system, external factors



- Beyond-accuracy goals
- Recommendation as a dynamic and interactive task

Dynamic and interactive aspects (1)

Recommender systems often trained from user **interaction data**, online or offline

- Recommender systems must learn to deal with noisy user feedback, limited knowledge about new users in cold start scenario, plus potential biases in user behavior that may impact the training data

Dynamic and interactive aspects (2)

Items consumed by a user may have an effect on the user state

- Potentially alters user preferences: by developing a user's interest, by educating users, or by changing their perspective
- Items may temporarily affect user behavior, e.g., by causing boredom, which reduces user interest and engagement in the platform

Dynamic and interactive aspects (3)

Exogenous factors may change value of items and preferences of users

- Yields an ever-changing dynamic environment

Wrap-up for Part 1

Long-term optimization and control requires multi-step thinking, because recommendations are performative

Wrap-up for Part 1

Long-term optimization and control requires multi-step thinking, because recommendations are performative

Accounting for dynamic and interactive aspects of recommendation

- Contextual bandits, active learning, counterfactual learning to rank, click modeling, . . .

Wrap-up for Part 1

Long-term optimization and control requires multi-step thinking, because recommendations are performative

Accounting for dynamic and interactive aspects of recommendation

- Contextual bandits, active learning, counterfactual learning to rank, click modeling, . . .

Approaches often trained from user data

Wrap-up for Part 1

Long-term optimization and control requires multi-step thinking, because recommendations are performative

Accounting for dynamic and interactive aspects of recommendation

- Contextual bandits, active learning, counterfactual learning to rank, click modeling, . . .

Approaches often trained from user data

Should not be evaluated solely on accuracy-centric benchmarks [Deffayet et al., 2022, Jannach et al., 2016, Sun, 2023] as these miss potential benefits brought by beyond-accuracy methods

Part 2. Evaluation

Problem definition (1)

Slate recommendation in a dynamic environment

- User interacts with recommender system of session of L steps
- At each step, recommender system presents slate with multiple items from catalog

Problem definition (1)

Slate recommendation in a dynamic environment

- User interacts with recommender system of session of L steps
- At each step, recommender system presents slate with multiple items from catalog

Naturally modeled as Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R)$

- States: represents user state, summarizes past interactions
- Actions: possible slates
- Transition probabilities: define dynamics in the process
- Reward function (potentially stochastic): for us, sum of clicks over recommended slate

Problem definition (2)

Possibly stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, decides what slate *a* recommender system should return in given state *s*

Problem definition (2)

Possibly stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, decides what slate *a* recommender system should return in given state *s*

Trajectory τ: set of successive states, actions and rewards collected in session of interactions between user and recommender

Problem definition (3)

Slate recommendation in a dynamic environment: identify policy π^* that maximizes cumulated reward in expectation over possible trajectories

- $\pi^* \in \arg\max_\pi \mathbb{E}_{\tau \sim \pi} \left[\sum_{(s,a) \in \tau} R(s, a) \right]$

Problem definition (3)

Slate recommendation in a dynamic environment: identify policy π^* that maximizes cumulated reward in expectation over possible trajectories

- $\pi^* \in \arg\max_\pi \mathbb{E}_{\tau \sim \pi}\left[\sum_{(s,a)\in\tau} R(s,a)\right]$
- Want best decision vs. most likely prediction – reinforcement learning vs. supervised learning
- Contrast with $\arg\min_{\hat{y}} \mathbb{E}_{y\sim\mathcal{D}}\left[\mathcal{L}(y,\hat{y})\right]$

Different evaluation setups for recommenders (1)

Online evaluation

- Still a gold standard
- Rare resource
- May negatively impact user satisfaction, revenue, . . .

⋮

Different evaluation setups for recommenders (2)

⋮

Off-policy evaluation: Evaluate (new) target recommender system using data collected with (old) logging recommender system

⋮

Different evaluation setups for recommenders (3)

⋮

Conduct experiments in simulated environment

- See [Balog and Zhai, 2023] for comprehensive picture on simulation
- Good performance obtained in a simulator is no guarantee of success in live system

Different evaluation setups for recommenders (3)

⋮

Conduct experiments in simulated environment

- See [Balog and Zhai, 2023] for comprehensive picture on simulation
- Good performance obtained in a simulator is no guarantee of success in live system
- Value lies in ability to control relevant parameters in a way that spans potential dynamics encountered in real environment

Different evaluation setups for recommenders (3)

⋮

Conduct experiments in simulated environment

- See [Balog and Zhai, 2023] for comprehensive picture on simulation
- Good performance obtained in a simulator is no guarantee of success in live system
- Value lies in ability to control relevant parameters in a way that spans potential dynamics encountered in real environment
- Tweaking parameters and observing their effect on candidate methods allows one to identify general trends and study important research topics
  - Regimes of success and failure (e.g., low data, high bias), robustness to environment features that may be observed in real world (e.g., noise, distribution shifts), generalizability of results, etc.

Pluses and minuses

Simulated evaluation can be less opaque than off-policy evaluation and online evaluation

Observing variables that are normally not accessible to practitioner can help better interpret observed performance of candidate systems

Wrap-up for Part 2

Reinforcement learning allows us to train agents in dynamic and interactive environments, in a way that recovers novel policies

For the ambitions of using reinforcement learning, next-item prediction is not a sufficient evaluation framework

Combination of online, off-policy and simulated evaluation can help research and understanding of new recommender systems

Part 3. Simulators

Wishlist for simulators: Enablers

Four **long-term research topics** to be addressed, typically with simulators:

(RT1) How to enable multi-step reasoning and control user-related metrics in the long run?

Wishlist for simulators: Enablers

Four **long-term research topics** to be addressed, typically with simulators:

(RT1) How to enable multi-step reasoning and control user-related metrics in the long run?

(RT2) How to learn meaningful and reliable information from biased data?

Wishlist for simulators: Enablers

Four **long-term research topics** to be addressed, typically with simulators:

(RT1) How to enable multi-step reasoning and control user-related metrics in the long run?

(RT2) How to learn meaningful and reliable information from biased data?

(RT3) How to make sure that interactive recommender systems are robust to uncertainties of the real-world?

Wishlist for simulators: Enablers

Four **long-term research topics** to be addressed, typically with simulators:

(RT1) How to enable multi-step reasoning and control user-related metrics in the long run?

(RT2) How to learn meaningful and reliable information from biased data?

(RT3) How to make sure that interactive recommender systems are robust to uncertainties of the real-world?

(RT4) How to effectively and efficiently recommend slates of items to users in a dynamic and interactive environment?

Wishlist for simulators: Properties

Comprehensiveness: Most of the important research questions for interactive recommender systems can be studied in one core simulated engine

Wishlist for simulators: Properties

Comprehensiveness: Most of the important research questions for interactive recommender systems can be studied in one core simulated engine

Interpretability: One or a few well-defined parameters control specific aspect of interest in recommender system research, i.e., simulator should be interpretable and controllable

Wishlist for simulators: Properties

Comprehensiveness: Most of the important research questions for interactive recommender systems can be studied in one core simulated engine

Interpretability: One or a few well-defined parameters control specific aspect of interest in recommender system research, i.e., simulator should be interpretable and controllable

Effect isolation: Effect of individual parameters or algorithmic modules can be singled out, to allow focused study of one aspect of environment (e.g., noise, user drift, etc.) or method (e.g., user and item representation, decision-making module, etc.)

Wishlist for simulators: Properties

Comprehensiveness: Most of the important research questions for interactive recommender systems can be studied in one core simulated engine

Interpretability: One or a few well-defined parameters control specific aspect of interest in recommender system research, i.e., simulator should be interpretable and controllable

Effect isolation: Effect of individual parameters or algorithmic modules can be singled out, to allow focused study of one aspect of environment (e.g., noise, user drift, etc.) or method (e.g., user and item representation, decision-making module, etc.)

Non-triviality: Simulated task should not be trivially solved by off-the-shelf baselines

Wishlist for simulators: Properties

Comprehensiveness: Most of the important research questions for interactive recommender systems can be studied in one core simulated engine

Interpretability: One or a few well-defined parameters control specific aspect of interest in recommender system research, i.e., simulator should be interpretable and controllable

Effect isolation: Effect of individual parameters or algorithmic modules can be singled out, to allow focused study of one aspect of environment (e.g., noise, user drift, etc.) or method (e.g., user and item representation, decision-making module, etc.)

Non-triviality: Simulated task should not be trivially solved by off-the-shelf baselines

Configurability: Additions and changes to existing simulator should be easy enough to enable deeper studies or new research questions

Simulators (1)

RecoGym [Rohde et al., 2018]

- E-commerce and advertising simulator where the agent aims to display attractive ads

Simulators (1)

RecoGym [Rohde et al., 2018]

- E-commerce and advertising simulator where the agent aims to display attractive ads

MARS-Gym [Santana et al., 2020]

- Aims to simulate online marketplaces, and is based on real data from such platforms

Simulators (1)

RecoGym [Rohde et al., 2018]

- E-commerce and advertising simulator where the agent aims to display attractive ads

MARS-Gym [Santana et al., 2020]

- Aims to simulate online marketplaces, and is based on real data from such platforms

RL4RS [Wang et al., 2023]

- E-commerce, slate recommendation simulator based on real purchase data, and where the reward function is a black-box sequential recommendation model

Simulators (2)

RecSim [Ie et al., 2019]

- Configurable simulator and three environment instantiations that cover, at least partially, all long-term research topics of interest to us

Simulators (2)

RecSim [Ie et al., 2019]

- Configurable simulator and three environment instantiations that cover, at least partially, all long-term research topics of interest to us

Virtual-TaoBao [Shi et al., 2019]

- Online retail simulator trained from real data, where generative adversarial networks are trained via multi-agent imitation learning in order to approximate the user response to recommendations

Simulators (2)

RecSim [Ie et al., 2019]

- Configurable simulator and three environment instantiations that cover, at least partially, all long-term research topics of interest to us

Virtual-TaoBao [Shi et al., 2019]

- Online retail simulator trained from real data, where generative adversarial networks are trained via multi-agent imitation learning in order to approximate the user response to recommendations

SOFA [Huang et al., 2020]

- Uses an intermediate re-weighting step in order to remove popularity and positivity biases in the resulting simulator

Simulators (3)

OBP [Saito et al., 2021]

- Semi-synthetic, research-oriented simulator for off-policy training evaluation of bandit agents

Summarizing

	Research topic				**Properties**		
Simulator	Multi-step	Bias	Uncertainty	Slates	Interpret.	Effect isol.	Configur.
RecoGym		±	+		±	+	+
MARS-Gym							±
RL4RS		±		+			
RecSim	+	+	±	±	±		±
Virtual-TB	+		+				
SOFA		+			+	±	+
OBP		+			+	+	+

+/±: topic is addressed/partially addressed or specification is fully/partially addressed

Summarizing

	Research topic				Properties		
Simulator	Multi-step	Bias	Uncertainty	Slates	Interpret.	Effect isol.	Configur.
RecoGym		±	+		±	+	+
MARS-Gym							±
RL4RS		±		+			
RecSim	+	+	±	±	±		±
Virtual-TB	+		+				
SOFA		+			+	±	+
OBP		+			+	+	+
SARDINE	+	+	+	+	+	+	+

+/±: topic is addressed/partially addressed or specification is fully/partially addressed

Wrap-up for Part 3

Many simulators for recommender systems available already

Research topics not fully addressed yet by current proposals

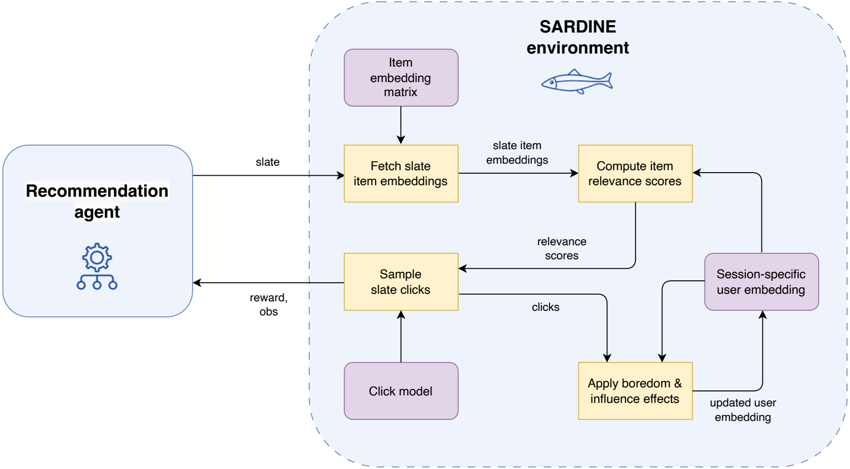
Desirable properties not fully satisfied yet by current proposals

Part 4. Fish

SARDINE

Simulator for Automated Recommendation in Dynamic and INteractive Environments

SARDINE



Simulator for Automated Recommendation in Dynamic and INteractive Environments

In words . . .

Initialize by forming synthetic embeddings for the set of recommendable items

Each user session is generated by following these successive steps:

1. Sample user embedding for current session's user
2. Provide initial recommendation or prompt agent to recommend slate to user
3. Compute relevance of items in slate with respect to user
4. Sample clicks on slate based on items' relevance and rank
5. Update user embedding to account for effects of boredom and clicked item influence, if those mechanisms are included in simulator
6. Repeat steps 1. to 5. until the number of interaction steps reaches session length

Basic choices made in SARDINE (1)

1. Item and user embeddings

    - Randomly-generated sparse embeddings for users and items

Basic choices made in SARDINE (1)

1. Item and user embeddings
   - Randomly-generated sparse embeddings for users and items
2. Initial recommendation
   - First recommendation is a slate containing random issues

Basic choices made in SARDINE (2)

3. Relevance computation

- Dot-product of item embedding and user embedding, followed by application of sigmoid

Basic choices made in SARDINE (2)

3. Relevance computation
    - Dot-product of item embedding and user embedding, followed by application of sigmoid
4. Click model
    - Position-based click model
    - Probability of click is defined as product of item-specific attractiveness and rank-specific examination probability

Basic choices made in SARDINE (3)

5. Boredom and influence mechanisms

    - Penalize myopic strategies that require agent to consider consequences of its actions

Basic choices made in SARDINE (3)

5. Boredom and influence mechanisms

- Penalize myopic strategies that require agent to consider consequences of its actions
- *Boredom*: user may become less interested in consuming content (i.e., clicking on items) when items recommended in successive slates are too similar
  - Temporary loss-of-interest boredom
  - Churn-and-return boredom

Basic choices made in SARDINE (3)

5. Boredom and influence mechanisms

- Penalize myopic strategies that require agent to consider consequences of its actions
- *Boredom*: user may become less interested in consuming content (i.e., clicking on items) when items recommended in successive slates are too similar
    - Temporary loss-of-interest boredom
    - Churn-and-return boredom
- *Influence of the clicked items*: when user consumes item, this may shift user's interest towards the item's topics

Full vs. partial observability (1)

Full observability: access to entire information about user state

- (i) Current user embedding, (ii) Histogram of number of times each topic was the main topic among last clicked items, (iii) Boredom timeout duration

Full vs. partial observability (1)

Full observability: access to entire information about user state

- (i) Current user embedding, (ii) Histogram of number of times each topic was the main topic among last clicked items, (iii) Boredom timeout duration

In the state, the current user embedding is used to keep track of the dynamic user preferences, while the histogram and timeout vectors maintain the information about recent item consumption and boredom

Full vs. partial observability (2)

Partial observability: agent only provided observations about the interaction

- (i) Slate that was recommended, (ii) Clicks on the recommended slate, (iii) History of recently clicked topics (e.g., think of item categories)

Full vs. partial observability (2)

Partial observability: agent only provided observations about the interaction

- (i) Slate that was recommended, (ii) Clicks on the recommended slate, (iii) History of recently clicked topics (e.g., think of item categories)

Agent is able to identify which recommended items led to a click and exploit recently clicked topics to better infer user preferences.

Full vs. partial observability (2)

Partial observability: agent only provided observations about the interaction

- (i) Slate that was recommended, (ii) Clicks on the recommended slate, (iii) History of recently clicked topics (e.g., think of item categories)

Agent is able to identify which recommended items led to a click and exploit recently clicked topics to better infer user preferences.

Items (i)–(iii) not enough to perfectly determine user state

- Agent may need to incorporate history of observations in same session in order to improve its estimation of user state (usually done through state encoders)

Hyperparameters of SARDINE

Hyperparameter	Description
L	Session length (in time steps).
S	Slate size (in number of items).
$n_{\mathcal{I}}$	Number of items.
$n_{\mathcal{T}}$	Number of topics (and user/item embedding dimension).
λ	Scale hyperparameter for the relevance function.
μ	Shift hyperparameter for the relevance function.
α	Scale hyperparameter for item attractiveness.
ϵ	Click propensity for examination probability.
n_{b}	Number of items considered for boredom computation.
t_{b}	Click recency (in time steps) for boredom computation.
τ_{b}	Threshold on topic occurrence for boredom computation.
ω	Weight controlling the influence of clicked items on user.
$\mathcal{O}$	Hyperparameter indicating full or partial state observability.

Wrap-up for Part 4

Introduced SARDINE

Enables study of long-term research topics we care about (multi-step reasoning, biased data, dealing with uncertainty, slate recommendation), while satisfying key properties (interpretability, effect isolation, configurability)

Available at https://github.com/RomDeffayet/SARDINE

Part 5. Experimental results

Motivation

Showcase SARDINE to

- Provide guidance for its usage
- Define a testbed for studying methods w.r.t. research topics mentioned
- Demonstrate SARDINE's utility for recommendation research

Motivation

Showcase SARDINE to

- Provide guidance for its usage
- Define a testbed for studying methods w.r.t. research topics mentioned
- Demonstrate SARDINE's utility for recommendation research

Need to specify

- Simulated environments
- Recommendation methods
- Hyperparameters (simulator, methods)

Simulated environments (1)

Environment name	Rec. type	Boredom	Influence	Click uncertainty	Observability	Reranking
`SingleItem-Static`	Single item	No	No	Low	Full	No
`SingleItem-BoredInf`	Single item	Yes	Yes	Low	Full	No
`SingleItem-PartialObs`	Single item	No	No	Low	Partial	No
`SlateTopK-Bored`	Slate	Yes	No	Low	Full	No
`SlateTopK-BoredInf`	Slate	Yes	Yes	Low	Full	No
`SlateTopK-PartialObs`	Slate	Yes	Yes	Low	Partial	No
`SlateTopK-Uncertain`	Slate	Yes	Yes	Medium to v. high	Partial	No
`SlateRerank-Static`	Slate	No	No	High	Full	Yes
`SlateRerank-Bored`	Slate	Yes	No	High	Full	Yes

Simulated environments (2)

Example specifications

⋮

- `SlateTopK-Bored`: Includes slate recommendation (as opposed to single-item recommendation) and boredom mechanism, with full state observability; suitable to evaluate RL-based slate recommendation methods in MDP setting

Simulated environments (2)

Example specifications

⋮

- `SlateTopK-Bored`: Includes slate recommendation (as opposed to single-item recommendation) and boredom mechanism, with full state observability; suitable to evaluate RL-based slate recommendation methods in MDP setting
- `SlateTopK-BoredInf`: Based on `SlateTopK-Bored` with an additional influence mechanism, making dynamics more complex as clicked items' influence causes a drift in user interests

⋮

Simulated environments (2)

Example specifications

⋮

- `SlateTopK-Bored`: Includes slate recommendation (as opposed to single-item recommendation) and boredom mechanism, with full state observability; suitable to evaluate RL-based slate recommendation methods in MDP setting
- `SlateTopK-BoredInf`: Based on `SlateTopK-Bored` with an additional influence mechanism, making dynamics more complex as clicked items' influence causes a drift in user interests

⋮

- `SlateRerank-Bored`: Testbed for presentation biases; adds boredom mechanism so that greedy agents, even with perfectly alleviated position bias, are not optimal; enables research on effect of data biases on, e.g., RL agents

Recommendation methods (1)

Random

- Recommend a random slate

Recommendation methods (1)

Random

- Recommend a random slate

Greedy oracle

- At each step, recommend optimal slate, based on current user embedding

Recommendation methods (1)

Random

- Recommend a random slate

Greedy oracle

- At each step, recommend optimal slate, based on current user embedding

REINFORCE + Top-K

- Extend REINFORCE policy-gradient to slate recommendation

Recommendation methods

SAC + Top-K [Deffayet et al., 2023]

- Soft-actor critic that takes actions in item embedding space

Recommendation methods

SAC + Top-K [Deffayet et al., 2023]

- Soft-actor critic that takes actions in item embedding space

SAC + GEMS [Deffayet et al., 2023]

- Use VAE to embed high-dimensional slate space into low-dimensional latent space

Recommendation methods

SAC + Top-K [Deffayet et al., 2023]

- Soft-actor critic that takes actions in item embedding space

SAC + GEMS [Deffayet et al., 2023]

- Use VAE to embed high-dimensional slate space into low-dimensional latent space

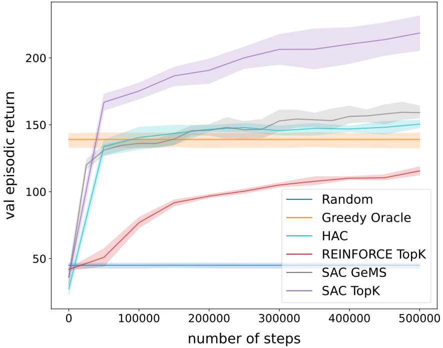
HAC [Liu et al., 2023]

- Hyper-actor critic that uses RL agent with actions in latent space (+ translation into slates)

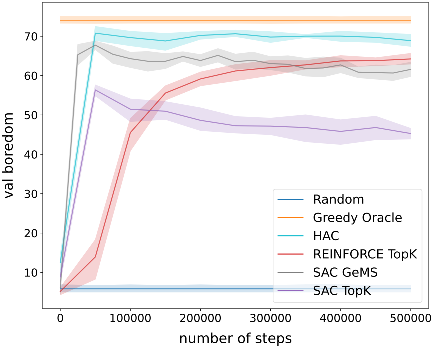
Hyperparameters

	Hyperparameter value												
Environment name	L	S	$n_{\mathcal{I}}$	$n_{\mathcal{T}}$	λ	μ	α	ϵ	n_{b}	t_{b}	τ_{b}	ω	$\mathcal{O}$
`SingleItem-Static`	100	1	1000	10	100	0.65	1.0	0.85	N/A	N/A	N/A	N/A	full
`SingleItem-PartialObs`	100	1	1000	10	100	0.65	1.0	0.85	N/A	N/A	N/A	N/A	partial
`SingleItem-BoredInf`	100	1	1000	10	100	0.65	1.0	0.85	10	5	5	0.95	full
`SlateTopK-Bored`	100	10	1000	10	100	0.65	1.0	0.85	10	5	5	N/A	full
`SlateTopK-BoredInf`	100	10	1000	10	100	0.65	1.0	0.85	10	5	5	0.95	full
`SlateTopK-PartialObs`	100	10	1000	10	100	0.65	1.0	0.85	10	5	5	0.95	partial
`SlateTopK-Uncertain`	100	10	1000	10	$\{2, 5, 10\}$	0.65	1.0	0.85	10	5	5	0.95	partial
`SlateRerank-Static`	10	10	10	10	5	0.30	1.0	0.85	N/A	N/A	N/A	N/A	full
`SlateRerank-Bored`	10	10	10	10	5	0.30	1.0	0.85	10	5	5	N/A	full

Example results



Return (↑) on `SlateTopK-Bored`



Boredom (↓) on `SlateTopK-Bored`

Some insights (1)

- SAC+Top-K stable high performance across different environments
- Results seem to depend on high quality of item embeddings used
- When replacing ideal item embeddings with sub-optimal, MF embeddings, performance drops

Some insights (2)

Transformer state encoder vs. GRU state encoder in PO environments

- Transformer beats GRU
- Impact of click uncertainty not fully understood

Some insights (3)

Experiments on impact of presentation bias in user feedback

- When the environment is dynamic, click models trained offline may be less accurate than on static environments
- May have detrimental effect on downstream tasks, such as counterfactual learning-to-rank or offline reinforcement learning
- Open up possibility of studying end-to-end training of RL agents from biased data, including a click modeling step

Wrap-up for Part 5

Demonstrated usage of SARDINE + sample of findings

Proposed environments at https://github.com/RomDeffayet/SARDINE

Methods compared at https://github.com/RomDeffayet/SARDINE_Experiments

Experiments at https://github.com/RomDeffayet/SARDINE_Experiments

Part 6. Conclusion

A look back (1)

Called attention to recommendation as dynamic and interactive task

A look back (1)

Called attention to recommendation as dynamic and interactive task

Long-term research topics

- Multi-step reasoning capacity of models
- Ability to learn models from biased data
- Robustness to uncertainty
- Challenges associated with recommending slates

A look back (2)

Shortcomings addressed with SARDINE simulator

- Comprehensiveness in the covered research questions, that compels researchers and practioners to scatter their study across several simulators
- Interpretability and controllability, when specific aspects of the simulator depend on the setting of multiple parameters
- Inability to study in isolation the phenomena and effects of interest in the simulator
- Solvability of the simulator through trivial off-the-shelf baselines
- Difficulty for researchers and practitioners to make additions and changes to simulator to go in more depth, or investigate new research questions

A look forward

Still many variants of the simulator to implement to target further research questions

- Performance when environment is non-stationary
- Reaching best possible policy in limited number of deployments ("deployment efficiency")
- Continual learning, deploying agents that keep on learning
- ...

Back matter

References i

K. Balog and C. Zhai. User simulation for evaluating information access systems. *arXiv preprint arXiv:2306.08550*, 2023.

R. Deffayet, T. Thonet, J. Renders, and M. de Rijke. Offline evaluation for reinforcement learning-based recommendation: A critical issue and some alternatives. *ACM SIGIR Forum*, 56(2): Article 3, 2022. doi: https://doi.org/10.1145/3582900.3582905.

R. Deffayet, T. Thonet, J.-M. Renders, and M. de Rijke. Generative slate recommendation with reinforcement learning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, pages 580–588, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394079. doi: 10.1145/3539597.3570412. URL https://doi.org/10.1145/3539597.3570412.

R. Deffayet, T. Thonet, D. Hwang, V. Lehoux, J.-M. Renders, and M. de Rijke. SARDINE: A simulator for automated recommendation in dynamic and interactive environments. *ACM Transactions on Recommender Systems*, Under review.

References ii

J. Huang, H. Oosterhuis, M. de Rijke, and H. van Hoof. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 190–199, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3412252. URL https://doi.org/10.1145/3383313.3412252.

E. Ie, C. wei Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, and C. Boutilier. RecSim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*, 2019.

D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker. Recommender systems — beyond matrix completion. *Commun. ACM*, 59(11):94–102, oct 2016. doi: https://doi.org/10.1145/2891406.

S. Liu, Q. Cai, B. Sun, Y. Wang, J. Jiang, D. Zheng, P. Jiang, K. Gai, X. Zhao, and Y. Zhang. Exploration and regularization of the latent action space in recommendation. In *WWW*, pages 833–844, 2023. doi: https://doi.org/10.1145/3543507.3583244.

D. Rohde, S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou. RecoGym: A reinforcement learning environment for the problem of product recommendation in online advertising. *arXiv preprint arXiv:1808.00720*, 2018.

References iii

Y. Saito, S. Aihara, M. Matsutani, and Y. Narita. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/33e75ff09dd601bbe69f351039152189-Paper-round2.pdf.

M. R. O. Santana, L. C. Melo, F. H. F. Camargo, B. Brandão, A. Soares, R. M. Oliveira, and S. Caetano. MARS-Gym: A gym framework to model, train, and evaluate recommender systems for marketplaces. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 189–197, 2020. doi: 10.1109/ICDMW51313.2020.00035.

J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4902–4909, Jul. 2019. doi: 10.1609/aaai.v33i01.33014902. URL https://ojs.aaai.org/index.php/AAAI/article/view/4419.

References iv

A. Sun. Take a fresh look at recommender systems from an evaluation standpoint. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 2629–2638, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591931. URL https://doi.org/10.1145/3539618.3591931.

K. Wang, Z. Zou, M. Zhao, Q. Deng, Y. Shang, Y. Liang, R. Wu, X. Shen, T. Lyu, and C. Fan. RL4RS: A real-world dataset for reinforcement learning based recommender system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 2935–2944, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591899. URL https://doi.org/10.1145/3539618.3591899.