Item Recommendation on Monotonic Behavior Chains

Mengting Wan, Julian McAuley
University of California, San Diego

Oct. 2018

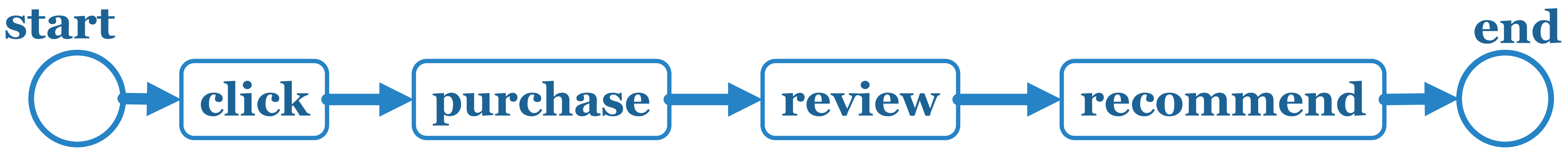
A spectrum of user-item interactions

- Explicit Feedback
  - numeric rating scores, etc.
  - scarce

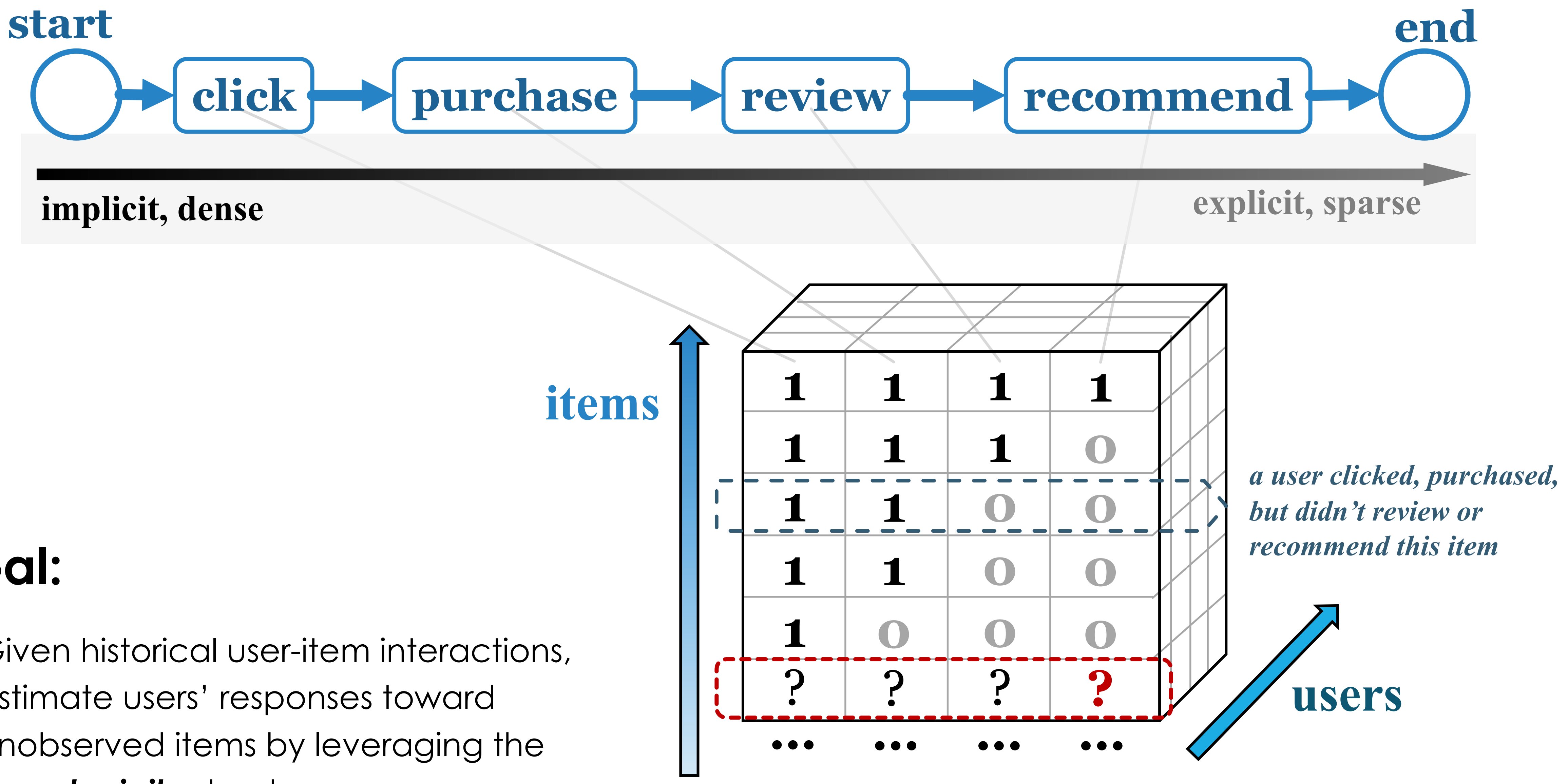
- Implicit Feedback
  - click, purchase, etc.
  - abundant

- **Item Recommendation:**
  - **How to properly represent a spectrum of users' responses?**
  - **How to efficiently leverage the connections among them?**

A spectrum of user-item interactions



- Monotonic Behavior Chains
  - any signal necessarily implies the presence of a weaker (or more implicit) signal
  - a ‘review’ action implies a ‘purchase’ action, which implies a ‘click’ action, etc.



$$y_{ui,1} \geq y_{ui,2} \geq y_{ui,3} \geq y_{ui,4}$$

Goal:

- Given historical user-item interactions, estimate users' responses toward unobserved items by leveraging the ***monotonicity*** structure

Preliminary Learning Strategies

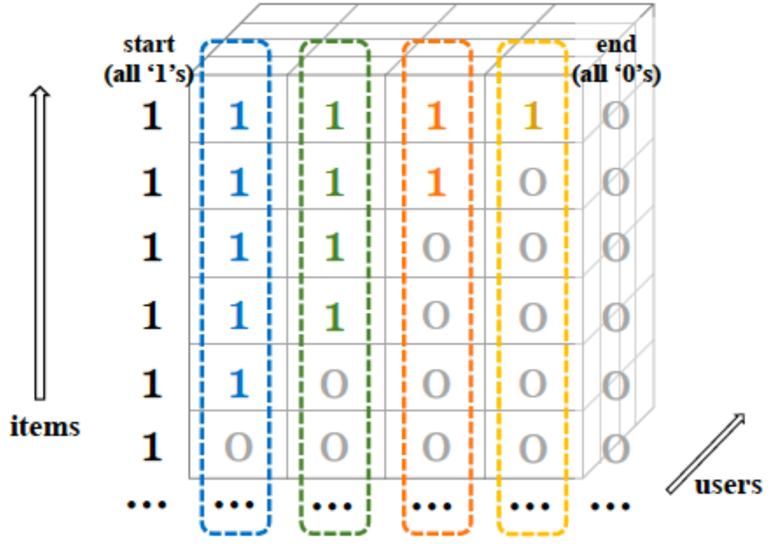
- Learning User Preferences
  - Independent of Stages

    Ignores stage dependency
    - train different models for different types of feedback signals
  - Jointly on Different Stages **(sliceOpt)**

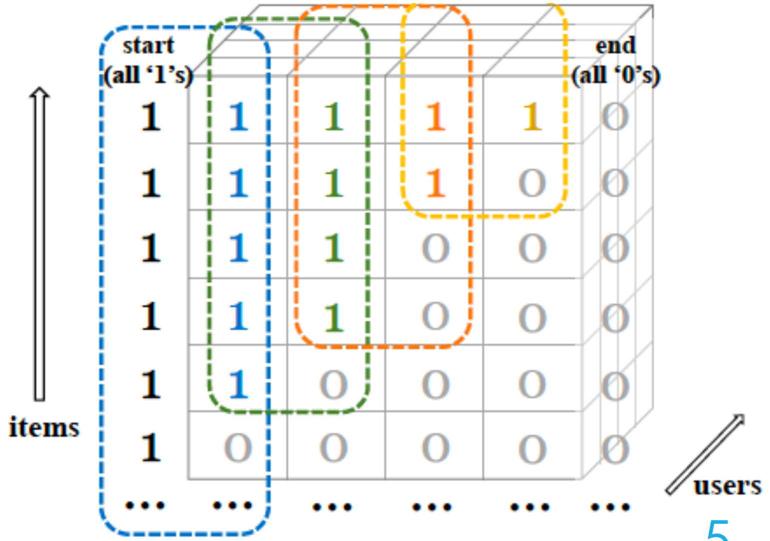
    Doesn't model the 'monotonicity' directly
    - Share user and item embeddings, train on all stages together
  - Conditioned on Previous Stages **(condOpt)**

    Scarce data at later stages
    - Train on stage 'escalation', use the jointly probability as the preference score

$$s_{ui,l} \coloneqq P(y_{ui,1} = \ldots = y_{ui,l} = 1) = \prod_{l'=1}^{l} p_{ui,l'|l'-1}$$



(a) sliceOpt



(b) condOpt

The Proposed Algorithm: chainRec

- **Monotonic Scoring Function**
  - Preserve the monotonicity of the interaction matrix
- **Edgewise Optimization Criterion**
  - Prune the redundant information and improve optimization efficiency

Monotonic Scoring Function

Monotonicity of user feedback

$$\boldsymbol{y_{ui,1} \geq y_{ui,2} \geq y_{ui,3} \geq y_{ui,4}}$$

approximate

Estimated preference scores

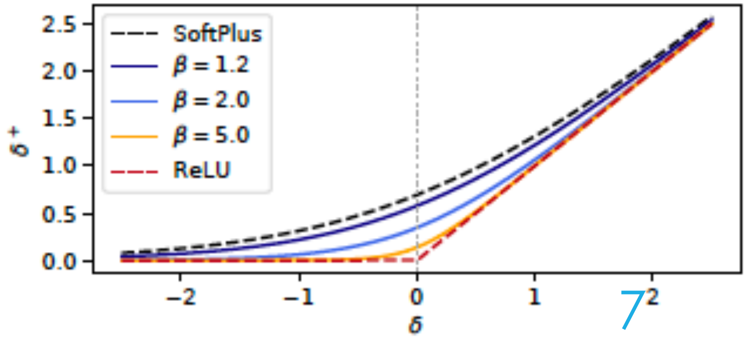
$$\boldsymbol{s_{ui,1} \geq s_{ui,2} \geq s_{ui,3} \geq s_{ui,4}}$$

Make sure the difference is non-negative

$$\boldsymbol{s_{ui,l} - s_{ui,l-1} = \delta^{+}_{ui,l} \geq 0}$$

Tensor decomposition + Rectifier

$$\boldsymbol{\delta_{ui,l}} = <\boldsymbol{\gamma_l}, \boldsymbol{\gamma_i} \circ \boldsymbol{\gamma_u}> \quad \Rightarrow \quad \boldsymbol{\delta^{+}_{ui,l}} = \frac{\mathbf{1}}{\boldsymbol{\beta}} \log(\mathbf{1} + \exp(\boldsymbol{\beta\delta_{ui,l}}))$$



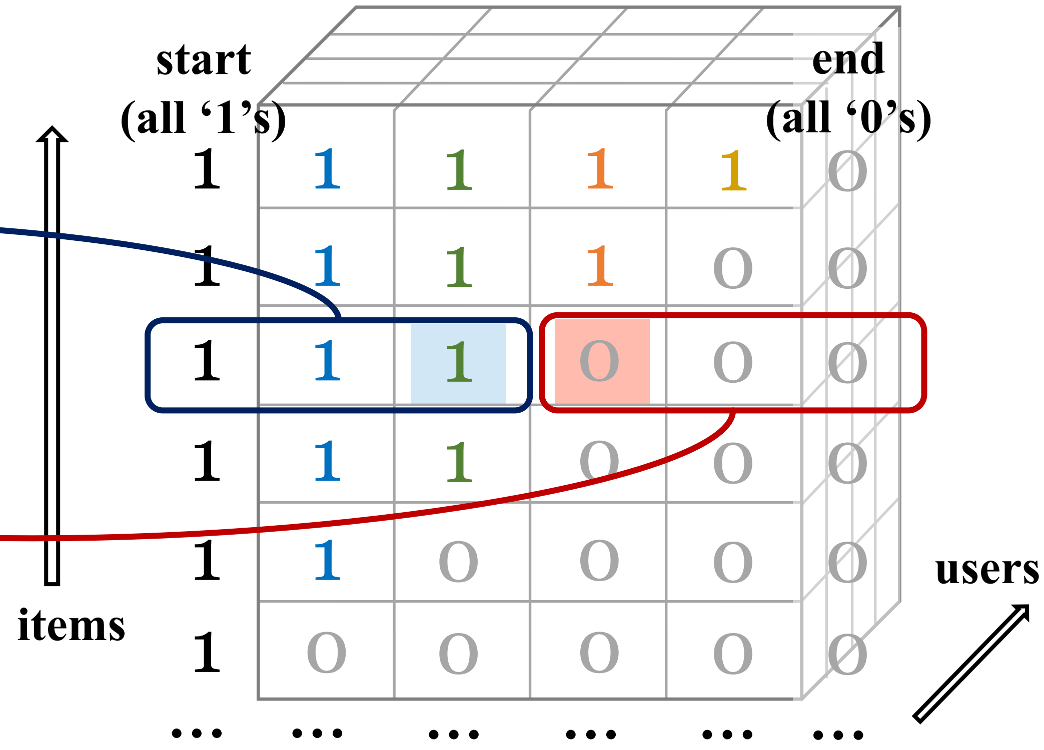
Monotonic Scoring Function

$$s_{ui,l} = b_0 + b_i + b_u + \sum_{l'=l}^{L} \delta^{+}_{ui,l'}$$

Edgewise Optimization Criterion (edgeOpt)

- Probabilistic implications from the monotonicity:

$$P(y_{ui,1} = 1, \ldots, y_{ui,l} = 1) = P(y_{ui,l} = 1), \quad \forall l;$$
$$P(y_{ui,l+1} = 0, \ldots, y_{ui,L} = 0) = P(y_{ui,l+1} = 0), \quad \forall l.$$



Edgewise Optimization Criterion (edgeOpt)

$$
\begin{aligned}
P(y_{ui,1}=1,\ldots,y_{ui,l}=1) &= P(y_{ui,l}=1), && \forall l;\\
P(y_{ui,l+1}=0,\ldots,y_{ui,L}=0) &= P(y_{ui,l+1}=0), && \forall l.
\end{aligned}
$$

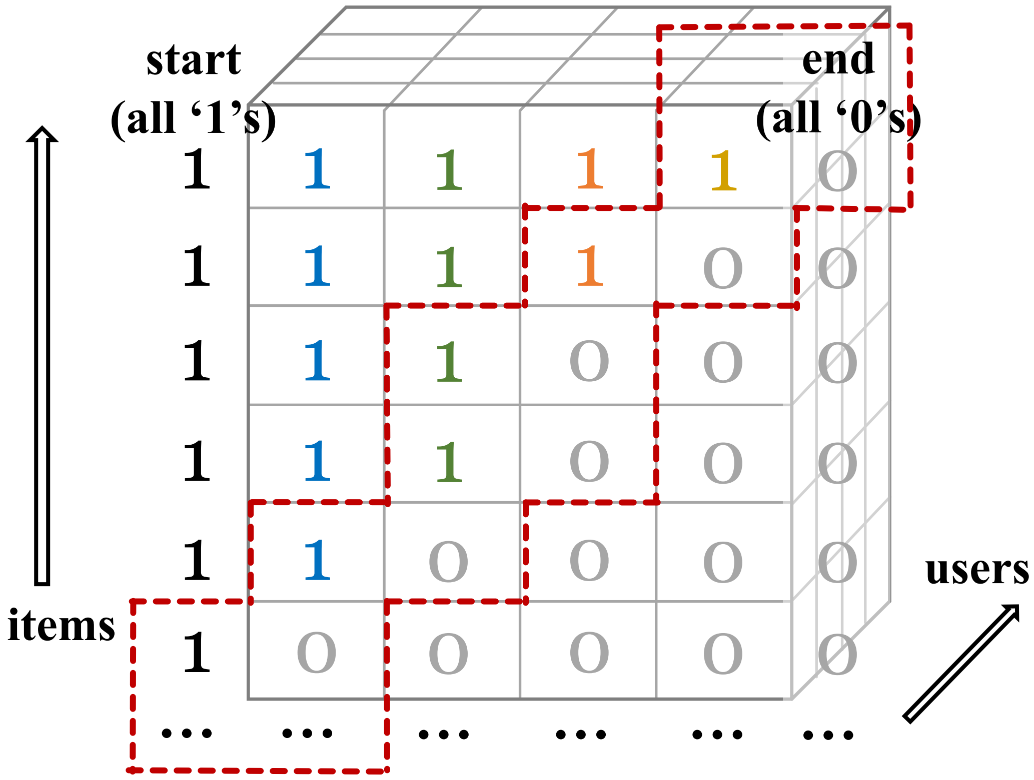
- What does it imply in the objective function?
  - Critical information is included in the ‘edges’ only

$$
\begin{aligned}
\log P(y_{ui,1},\ldots,y_{ui,L}) &= \log P(y_{ui,l^*_{ui}}=1, y_{ui,l^*_{ui}+1}=0)\\
&= \log \boxed{p_{ui,l^*}}\,\boxed{(1-p_{ui,l^*+1})}\,\boxed{p_{ui,\cap}},
\end{aligned}
$$

The very last positive stage

The very first negative stage

Mutual information on the edge



edgeOpt

Algorithm: chainRec

Algorithm 1 chainRec

for each user u, and each item $i \in I_u^+$ do

Locate the last positively interacted stage l_{ui}^*

Update the associated parameters Θ_{ui} based on the gradients

monotonic scoring function: $s_{ui,l}$

$$p_{ui,l} = \sigma(s_{ui,l}) = \frac{1}{1 + exp(-s_{ui,l})}$$

$$\frac{\partial}{\partial \Theta_{ui}} \log p_{ui,l_{ui}^*}$$

Update positive example

end for

Algorithm: chainRec

Algorithm 1 chainRec

---

for each user u, and each item $i \in \mathcal{I}_u^+$ **do**

Locate the last positively interacted stage l^*_{ui}

Update the associated parameters Θ_{ui} based on the gradients

$$\frac{\partial}{\partial \Theta_{ui}} \log p_{ui, l^*_{ui}}$$

Update positive example

Sample N contrastive items based on the given sampling scheme

for each contrastive item i' **do**

Locate the last positively interacted stage $l^*_{ui'}$

Update the associated parameters $\Theta_{ui'}$ based on the gradients

$$\frac{\partial}{\partial \Theta_{ui'}} \left(\log \left(1 - p_{ui', l^*_{ui'}+1} \right) + \log p_{ui', \cap} \right)$$

Update contrastive example

The first non-interacted stage

end for

end for

Experiments (Datasets)

- Steam
  - purchase (100%) – play (64%) – review (2.2%) – recommend (2%)
- YooChoose
  - click (100%) – purchase (45.7%)
- Yelp
  - review (100%) – recommend (rating>3, 71.1%)
- GoogleLocal
  - review (100%) – recommend (rating>3, 85%)
- Goodreads (new dataset)
  - shelve (100%) – read (49.1%) – rate (45.9%) – recommend (rating>3, 32%)

Experiments (Results)

Recommendation Results on the Most Explicit Interaction Stage

Dataset	Metric	(a) itemPop	(a) bprMF	(a) WRMF	(a) logMF	(b) condMF	(b) condTF	(b) sliceTF	(b) sliceTF (m.)	(c) chainRec (uniform)	(c) chainRec (stage.)	(c) %impr. vs. (a)	(c) %impr. vs. (b)
Steam	AUC	0.955	0.963	0.963	0.962	0.961	0.959	0.967	0.957	0.964	0.968	0.44%	0.06%
	NDCG	0.318	0.318	0.314	0.319	0.298	0.310	0.278	0.266	0.319	0.323	1.21%	4.23%
YooChoose	AUC	0.914	0.924	0.920	0.922	0.929	0.920	0.940	0.928	0.951	0.950	2.90%	1.13%
	NDCG	0.140	0.152	0.154	0.150	0.124	0.133	0.185	0.154	0.199	0.176	28.73%	7.09%
Yelp	AUC	0.838	0.921	0.912	0.903	0.900	0.838	0.928	0.918	0.937	0.927	1.71%	0.91%
	NDCG	0.093	0.105	0.096	0.100	0.090	0.088	0.107	0.096	0.108	0.102	3.05%	0.60%
GoogleLocal	AUC	0.597	0.661	0.625	0.661	0.679	0.616	0.684	0.667	0.695	0.722	9.31%	5.69%
	NDCG	0.064	0.067	0.064	0.066	0.064	0.063	0.070	0.065	0.072	0.072	8.36%	2.92%
Goodreads	AUC	0.938	0.971	0.963	0.971	0.904	0.933	0.984	0.934	0.982	0.978	1.17%	-0.17%
	NDCG	0.124	0.125	0.098	0.127	0.072	0.104	0.132	0.121	0.132	0.113	3.94%	0.00%

Ignore stage dependency — Incorporate stage dependency — The proposed method

Conclusions and Future

- Monotonic Behavior Chain
- chainRec
  - Monotonic scoring function
  - Edgewise Optimization Strategy

- chainRec -> treeRec
- Incorporating features and counts of interactions
- Other areas
  - medical diagnosis: dependencies exist between progressive symptoms

Thanks!

Mengting Wan (m5wan@ucsd.edu)

Data: https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home

Source Code: https://github.com/MengtingWan/chainRec