Session-Based Recommendation

Anthony Huang, Jasper Liu, Ping Zhang, Shubham Singh, Vinay Pahwa

Table of Contents

Introduction

Main Objective: To propose a new neural network that incorporates time and assigns corresponding weights to distant time periods

Why Session-Based RS?

- User's preference is not static and may change over time
- Reflects short-term and long-term preferences within a session
- Ensures dynamic preferences
- Better accuracy on predictions

Drawbacks of Traditional RSs

- User/Item cold start
- Learn each user's long-term and static preferences
- Assume all of the historical interactions of a user are equally important

Dataset - Retailrocket

Source:

Kaggle

Industry:

E-Commerce

Dataset Type:

Website Visitor Actions (Clicks, Add to Cart, Transaction)

4.5 months
Time Span
(Whole dataset)

7
Features
- Visitor ID
- Item ID
- Events
- Transaction ID
- Timestamp
- Item Property
- Property Value

131,533
Unique Items

2,756,101
Events:
- Views: 2 664 312
- AddtoCart: 69 332
- Transaction: 22 457

How to define 'Session'?

01

Session Length

Dynamic Session Length

Max time between actions: 30 min

Max number of actions: 30 actions

Max time of a session: 900 min

Minimizes static session length drawbacks: long sessions include noisy information, short sessions contains limited information

02

Internal Order

Flexible Ordered Sessions

Order between actions is not strict, not unordered, but flexible

03

Action Type

Multi-type Action Sessions

Three types of actions:
view, addtocart, transaction

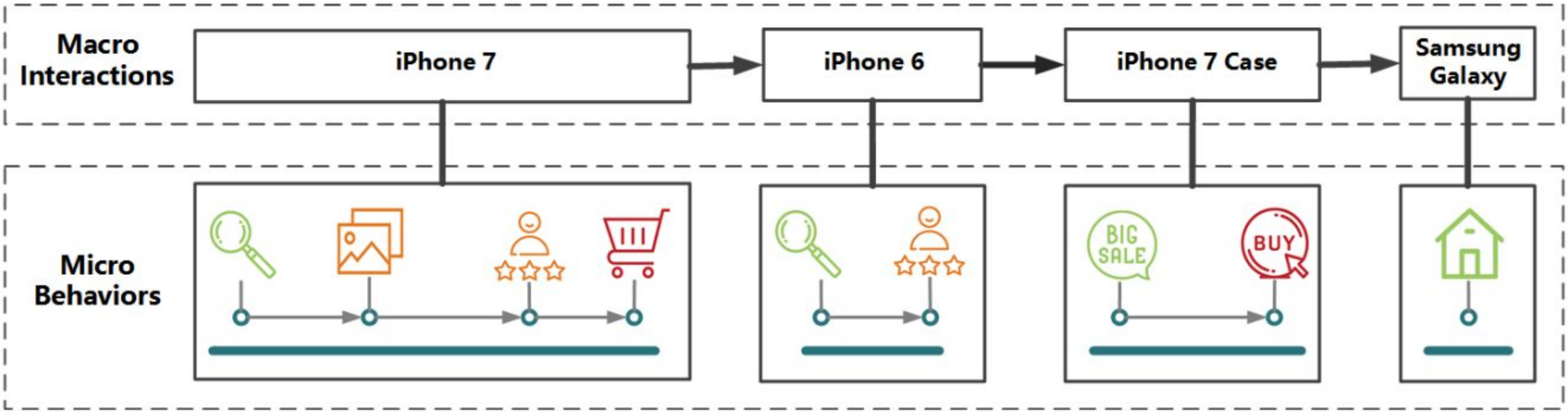
04

User Information Anonymity

Anonymous Sessions

Once sessions are created, user information is not necessary for predictions

Macro-behavior vs. Micro-behavior



Micro-behavior Session-based

Recommendation System

Key Steps

Items: GGNN

#1

Convert item sequences into directed graphs and feed into GGNN model.

Operations: GRU

#2

Feed operation sequences into GRU model.

Micro-behavior

#3

Concatenate output from GGNN and GRU to form a micro-behavior embeddings

Sessions: Attention

#4

Combine global preference and local preference into session representations.

Prediction

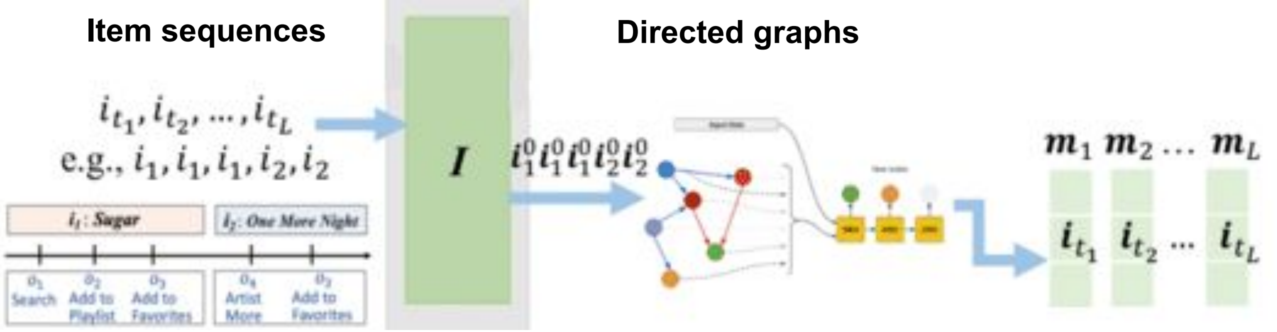
#5

Train the model and evaluate the prediction accuracy

Step 1: Item Embeddings Learning

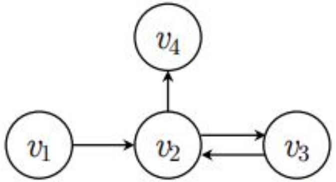
Item sequences

Directed graphs



GGNN

Item embeddings

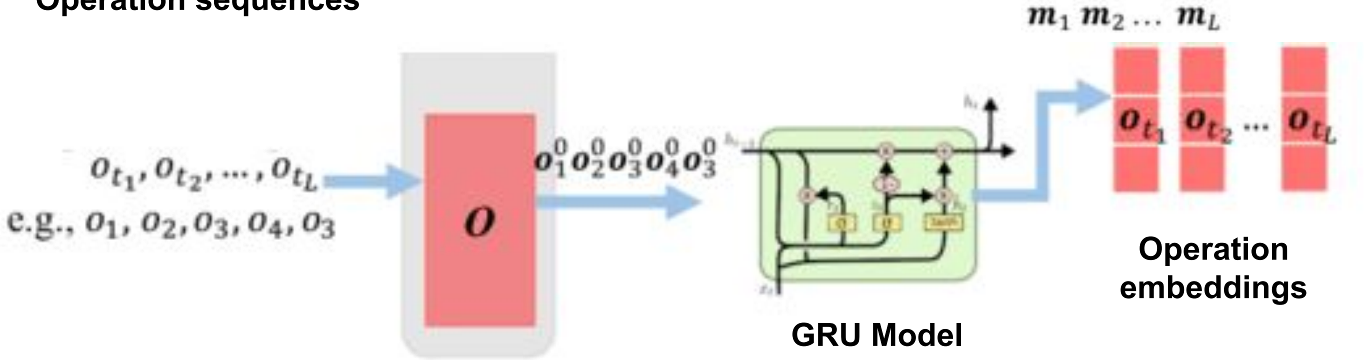


	Outgoing edges				Incoming edges			
	1	2	3	4	1	2	3	4
1	0	1	0	0	0	0	0	0
2	0	0	1/2	1/2	1/2	0	1/2	0
3	0	1	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0

An example of a session graph and the connection matrix As

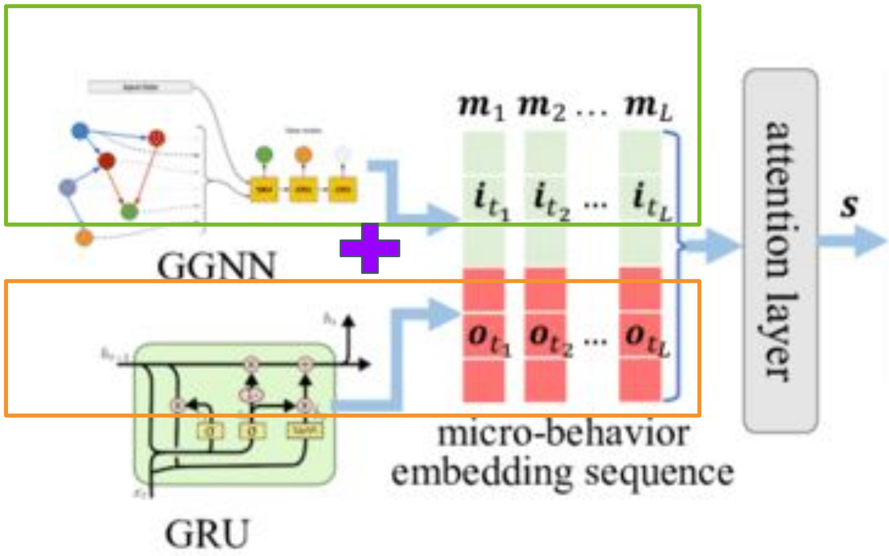
Step 2: Operation Embeddings Learning

Operation sequences



Step 3-4: Session Representations

Item Embeddings



Operation Embeddings

Micro-behavior Embeddings

Soft-Attention Mechanism

Local preference: $\mathbf{m}_L$

Global preference:

1. Attention Weight

$$\alpha_t = \boldsymbol{\beta}^{\top} \sigma(\mathbf{W}_1 \mathbf{m}_L + \mathbf{W}_2 \mathbf{m}_t + \mathbf{b}_\alpha)$$

2. Global Representation of a Session

$$\mathbf{s}_g = \sum_{t=1}^{L} \alpha_i \mathbf{m}_t$$

Final Representation of a Session

$$\mathbf{s} = \mathbf{W}_3[\mathbf{m}_L; \mathbf{s}_g] \in \mathbb{R}^d$$

Step 5: Training and Prediction

Softmax Operation

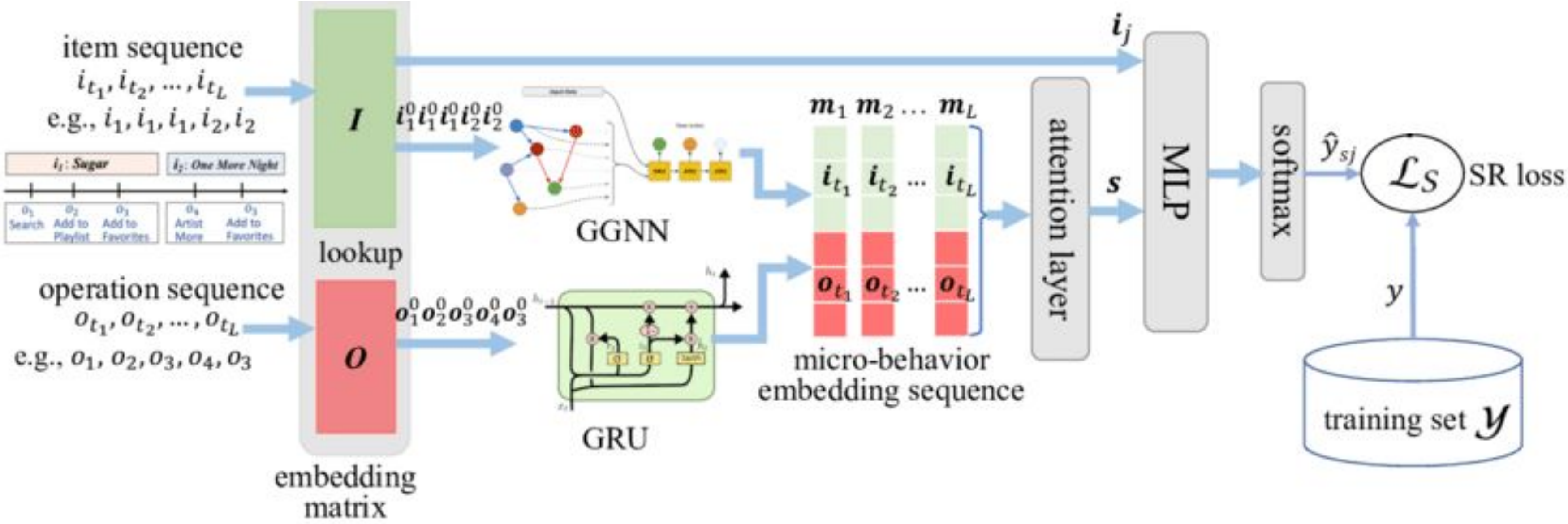
$$\hat{y}_{sj} = softmax\left(MLP(\mathbf{s} \oplus \mathbf{i}_j)\right)$$

Loss Function

$$\mathcal{L}_S = -\sum_{s \in \mathcal{S}} \sum_{j \in I} \{y_{sj} \log(\hat{y}_{sj}) + (1 - y_{sj}) \log(1 - \hat{y}_{sj})\}$$

Full Model

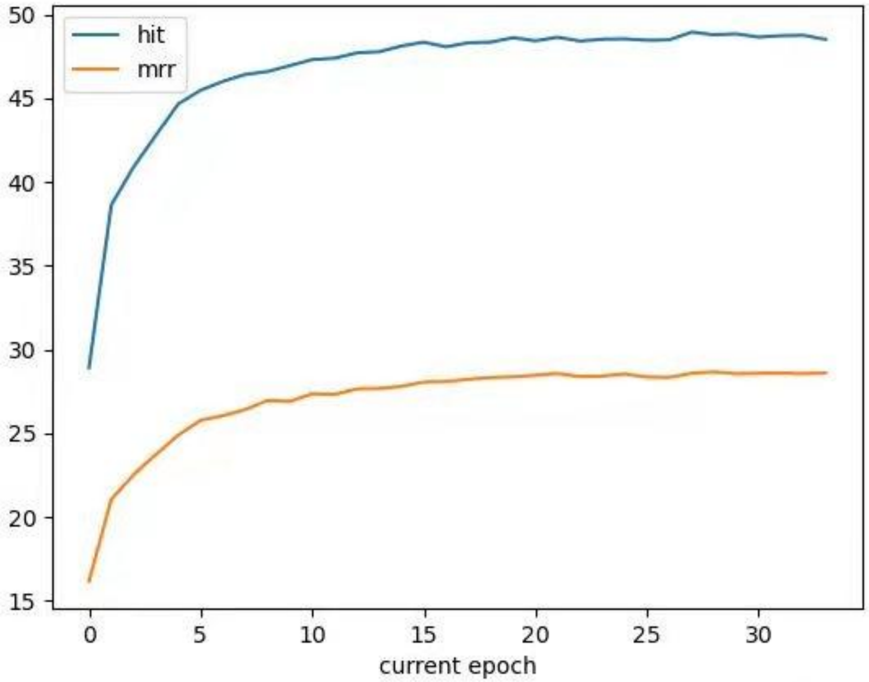
Sequence Inputs → ①Item Embeddings ②Operation Embeddings → ③~④Session Representations → ⑤Model Training & Recommendation



Performance

Hit@k

It is the proportion of hit samples to all samples that have the correct next interacted item in the top-k ranking lists.



Hit@20

49.04%

MRR@20

28.83%

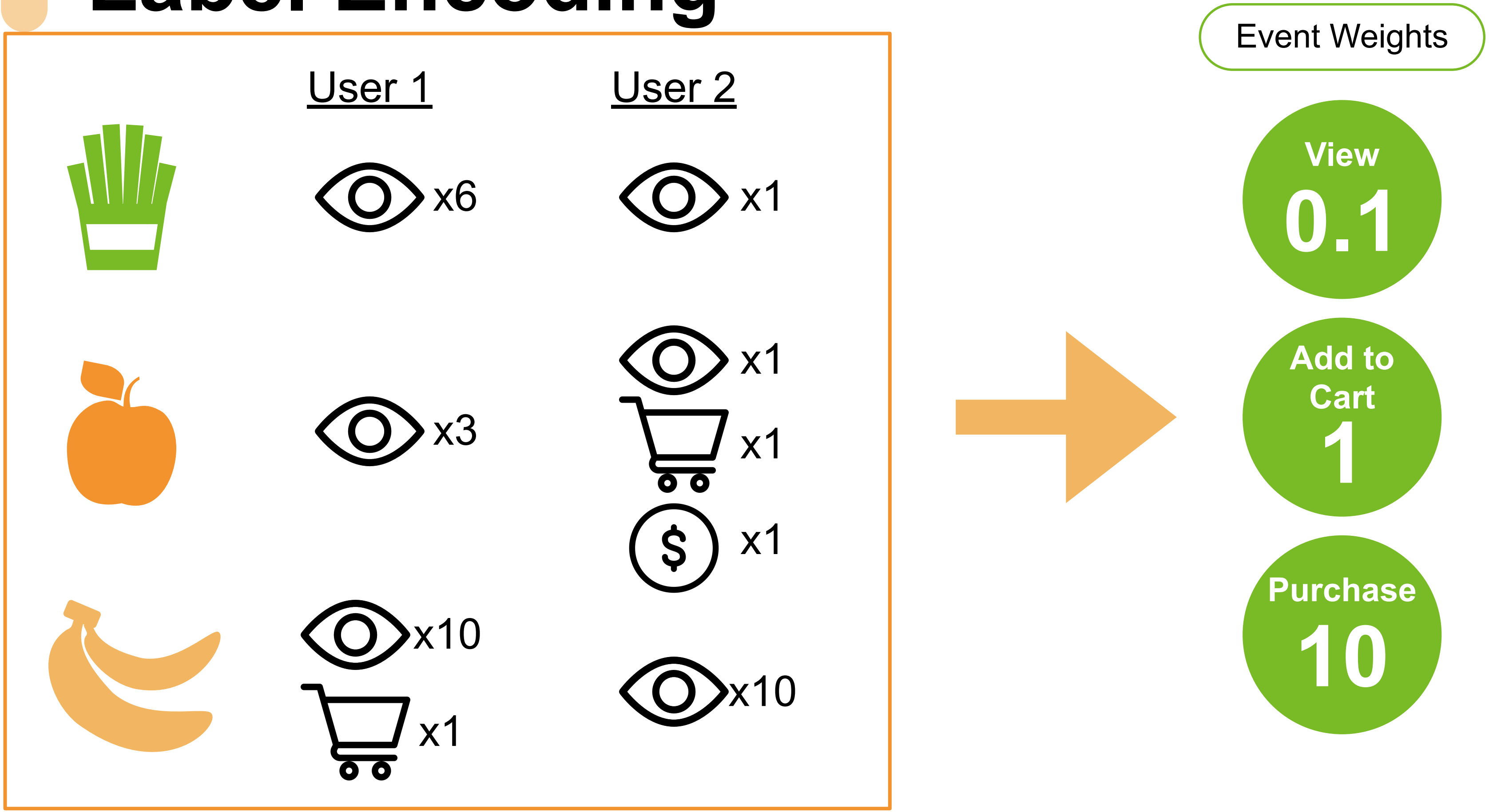
MRR@k

The average reciprocal rank of the correct next interacted item in the top-k ranking list, set to zero if the correct item is ranked behind top-k.

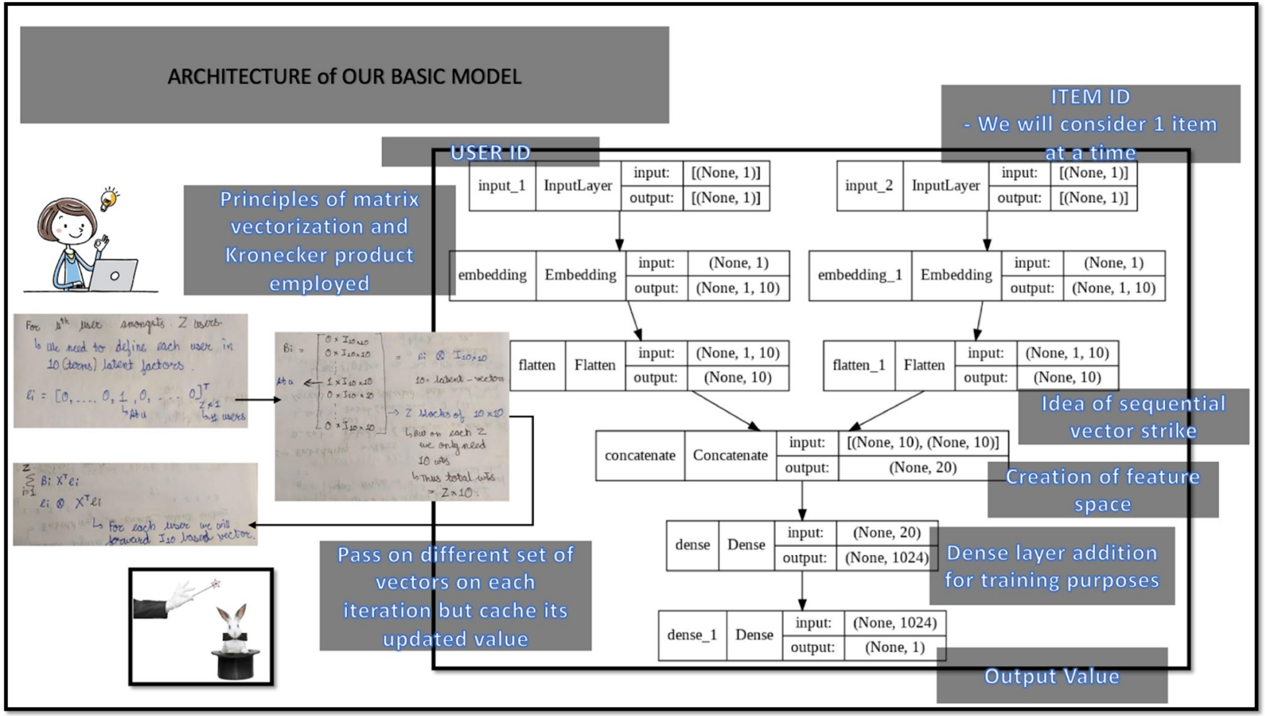
NN Based Matrix Factorization Model

BASELINE MODEL

Label Encoding



Model Formulation



IDEATION STEPS

1) So we wish to capture the dynamic operations of items and users using our customized encoded labels

2) Next we intended to encode the non-linear association of item and user

3) And we wish to prepare vectors that forms a good representation of user-item entities

$$L = \sum_{i,j \in \text{train}} (r_{ij} - w_i^T u_j)^2.$$

Experimentation

EVENT ENCODE-
VIEW:0.1,ADDtoCART:1,Trasaction:10

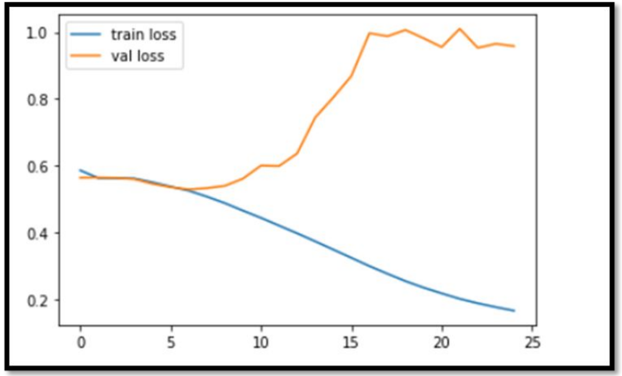
Summation of events for all combination of user-items

Bin based rating →
[0.1-0.2):1, [0.2-1):2, [1-10):3, [10-Large value):4

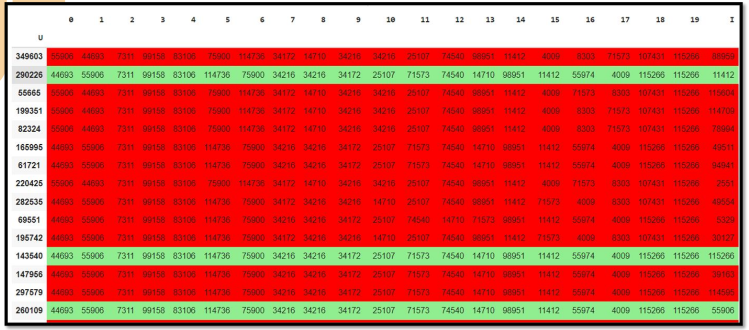
```
events_df['event_encode']=events_df['event'].map({
    'view':0.1,'addtocart':1,'transaction':10
})
df_events=events_df.groupby(['visitorid','itemid'])['event_encode'].sum().to_frame('rating').reset_index()

df_events.loc[:,'cum_rating']=pd.cut(df_events['rating'],bins=[0.1,0.2,1,10,1000],right=False,
                    labels=[1,2,3,4],
                    include_lowest=True)
```

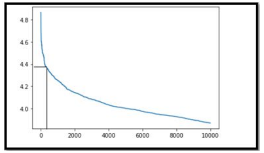
Fraction of 0.1 encode based users to be assigned pseudo user label
Parallelly fraction of 1 rating users that we wish to train on
Number of epochs → between 5 and 10 for majority of experiments
Momentum based SGD→
LR:0.8,momentum =0.9



Prediction Methods



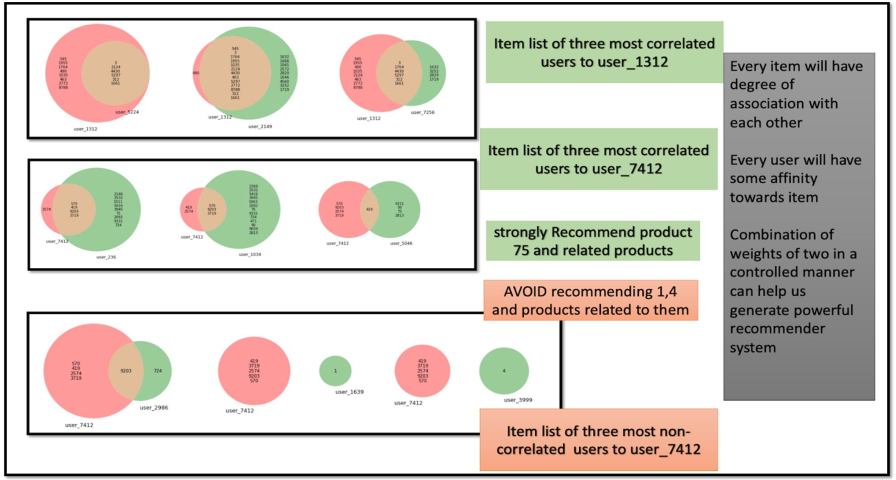
U	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	I
349603	55906	44693	7311	99158	83106	75900	114736	34172	14710	34216	34216	25107	74540	98951	11412	4009	8303	71573	107431	115266	88959
290226	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	11412
55665	55906	44693	7311	99158	83106	75900	114736	34172	14710	34216	34216	25107	74540	98951	11412	4009	71573	8303	107431	115266	115604
199351	55906	44693	7311	99158	83106	75900	114736	34172	14710	34216	34216	25107	74540	98951	11412	4009	8303	71573	107431	115266	114709
82324	55906	44693	7311	99158	83106	75900	114736	34172	14710	34216	34216	25107	74540	98951	11412	4009	8303	71573	107431	115266	78994
165995	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	49511
61721	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	94941
220425	55906	44693	7311	99158	83106	75900	114736	34172	14710	34216	34216	25107	74540	98951	11412	4009	71573	8303	107431	115266	2551
282535	44693	55906	7311	99158	83106	114736	75900	34172	34216	34216	14710	25107	74540	98951	11412	71573	4009	8303	107431	115266	49554
69551	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	74540	14710	71573	98951	11412	55974	4009	115266	115266	5329
195742	44693	55906	7311	99158	83106	114736	75900	34172	34216	34216	14710	25107	74540	98951	11412	71573	4009	8303	107431	115266	30127
143540	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	115266
147956	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	39163
297579	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	114595
260109	44693	55906	7311	99158	83106	114736	75900	34216	34216	34172	25107	71573	74540	14710	98951	11412	55974	4009	115266	115266	55906



DIRECT USE of RATING Scores

One straightforward way of user based prediction is to evaluate ratings for each item and then select from the sorted values

Hit rate@20 - Quite low ~5%



MANIPULATION OF EMBEDDER MATRIX

-Sample set of 50 most active users along with 30 most used products

-Can use more robust selection criterion

-Each entity is explained in terms of 10 latent factors

-A magnitude invariant cosine similarity score is computed for all pairs

-Weighted multiplication of user-user and user-item similarity can help to formulate basic recommender system

Cosine similarity between users

-Determine un/associated users

-Encode similarity in behaviour to capture information about more indirect products

-Helps in determination of weights for upcoming item selection

Food for Thought

- Limitations & Solutions
- Business Insights
- Key takeaways

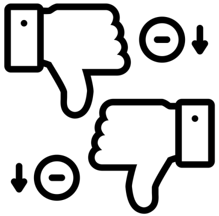
Limitations & Solutions

Limitations

1. User cold start is eliminated through sessions, but item cold start is still present

2. Model results might only be applicable to e-commerce Industry

3. Only used Hit@20 and MRR@20 scores to evaluate performance. Limitations applies to both metrics



Solutions

1. By implementing a knowledge graph to our model, item-based cold start may be minimized

2. Training our model with datasets of other industries such as movies, books, music would improve the robustness of our model

3. Accuracy is not the only relevant quality factor in practice. Additional measurements includes coverage and popularity bias.

Business Insights & Applications

Incorporating user micro-behaviors in session-based recommendation systems increases the accuracy of prediction

Wide scope of applications: Netflix Recommendation, Trip Planner, Route Suggestions

01
Improve Customer Retention

Ex: User receive tailored recommendation to renew their subscription

02
Increase Sales

Ex: Learn what sells what does not according to customer's preference

03
Easily Analyze Market

Businesses can utilize it to ensure they offer similarity competitive products

04
Enhance User Experience

Feed recommendation to customer that suits their tastes to improve business reputation

Key Takeaways

Pros and Cons of session-based recommendation

- Pros: short-term, dynamic, timely
- Cons: ignore users' long-term general preferences

Challenges Related to Operation Types

- There are complex dependencies inside operations.
- How to learn about intra- and inter-operation type dependencies.

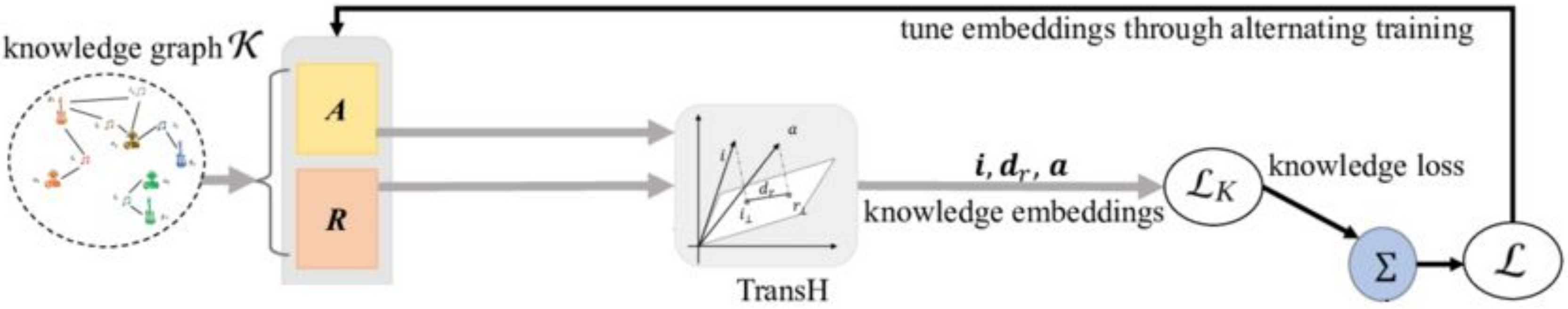
Improvement on Metrics

- Accuracy is not the only relevant quality factor in practice
- Multi-metric evaluation approach helps

Thank you!

Any questions?

Appendix: Knowledge Graph



Trains an item-properties network to reduce item cold-start

References

- https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems)
- https://en.wikipedia.org/wiki/Kronecker_product
- https://en.wikipedia.org/wiki/Vectorization_(mathematics)
- Incorporating User Micro-behaviors and Item Knowledge into Multi-task Learning for Session-based Recommendation (Wenjing Meng, Deqing Yang, Yanghua Xiao)
- Session-based Recommendation with Graph Neural Networks (Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, Tianniu Tan)
- A Survey on Session-based Recommender Systems (Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, Defu Lian)