Dynamic Graph Collaborative Filtering

Xiaohan Li [1*], Mengqi Zhang [2*], Shu Wu [2], Zheng Liu [1], Liang Wang [2], Philip S. Yu [1]

[1] University of Illinois at Chicago

[2] Chinese Academy of Science, Institute of Automation

* Equal Contribution

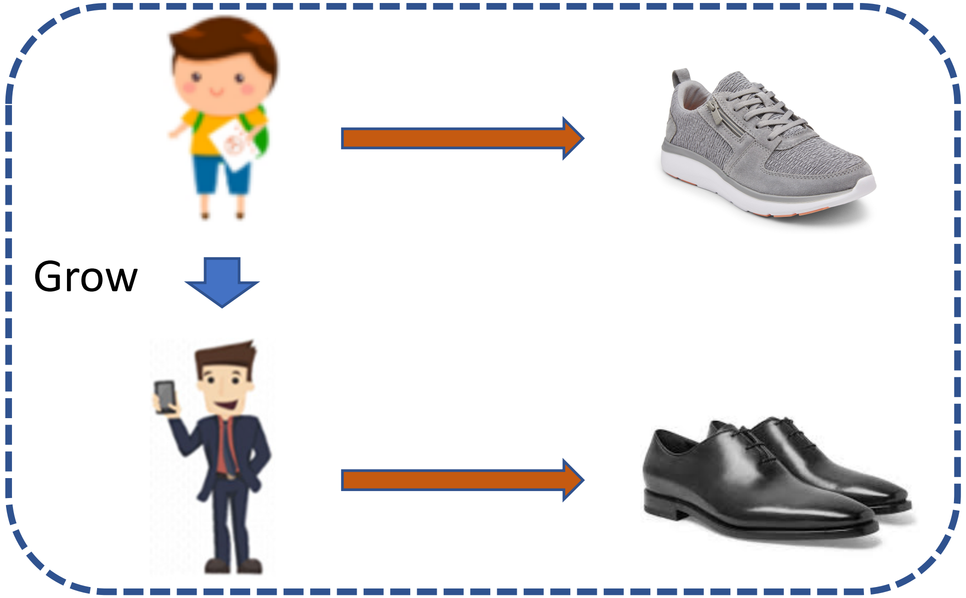
Outline

1. Background
2. Model
3. Experiments
4. Conclusion
5. Q&A

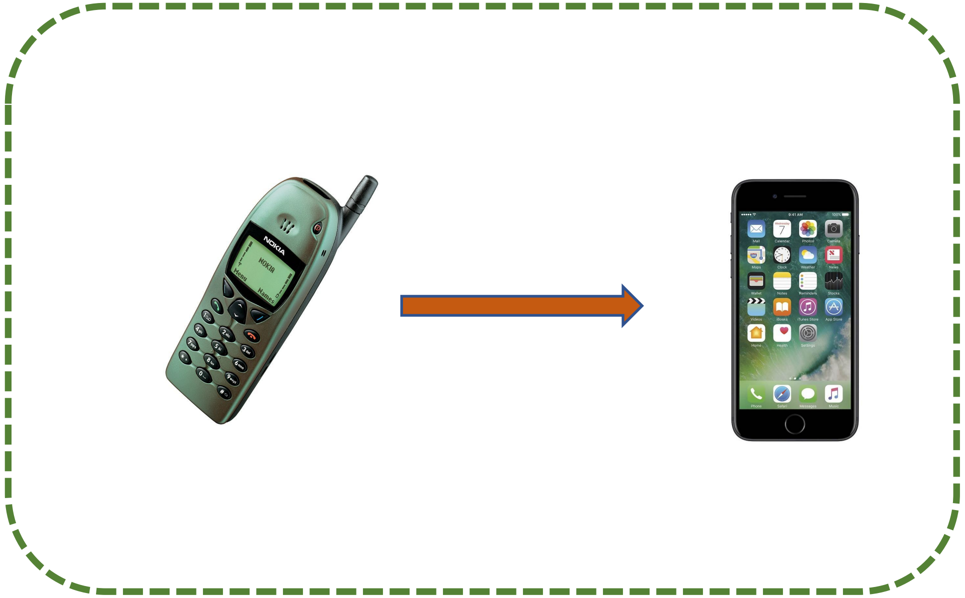
Background

Why dynamic recommender system?

1. User’s interests dynamically shift and evolve over time.
2. Item’s popularity also changes over time.



User’s interests



Item’s popularity

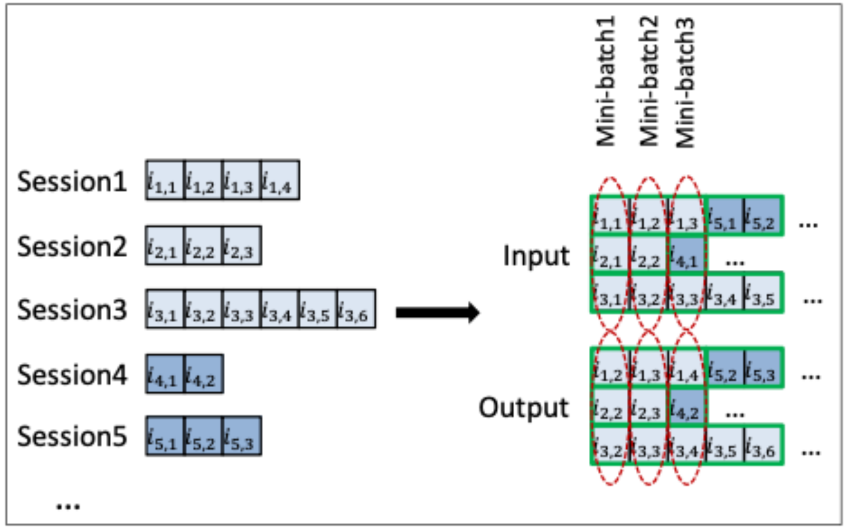
What is dynamic recommender system?

- Dynamic recommender systems consider the changes of users and items over time.

- For example, RNN-based models use item sequences as inputs to capture sequential dependency.

Problem:

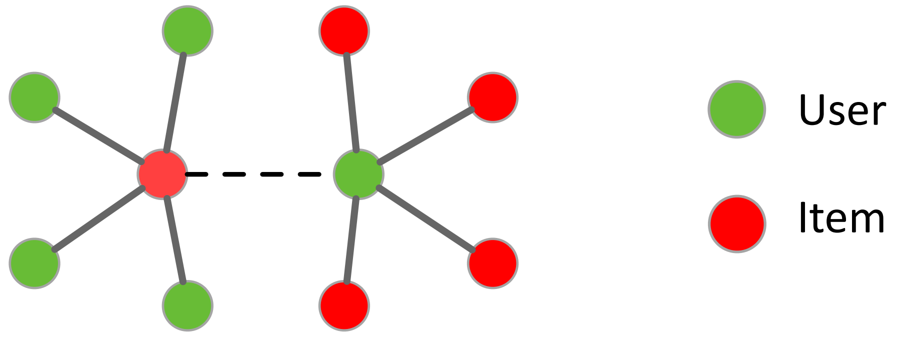
1. Ignore user sequences
2. lack collaborative information!



Session-based recommendations with Recurrent Neural Networks, ICLR 2016

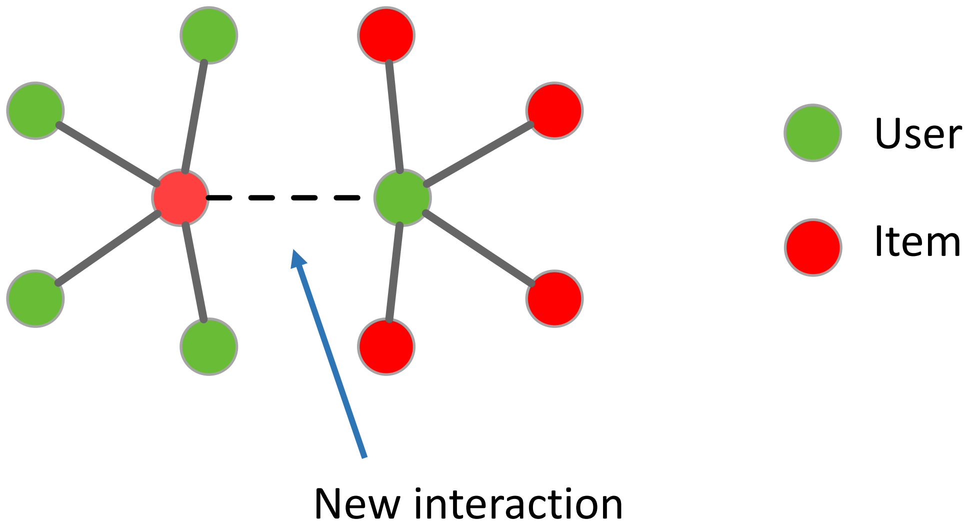
Why graph?

- Graph Neural Networks have been proven to be useful in recommender systems.
- Graph structures can incorporate collaborative information explicitly.
- Graph structures can explore high-order connectivity between users and items.



How to use graph in dynamic recommender system?

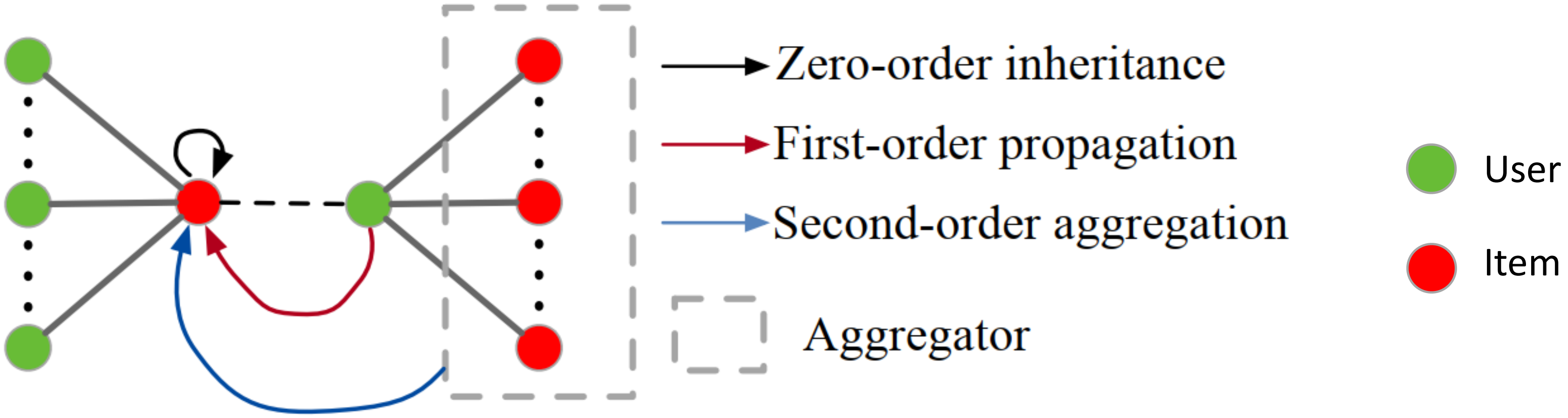
- Dynamic graph is to model the changes of nodes when the graph is evolving.
- When a new interaction join the graph, we need to update the embeddings of users and items.



Model

How to use graph in dynamic recommender system?

- Zero-order ‘inheritance’ is to inherit the node embedding from the last state.
- First-order ‘propagation’ is to propagate user/item embedding to the other side.
- Second-order ‘aggregation’ is to model the collaborative relation between users and items by aggregating second-order neighbors.



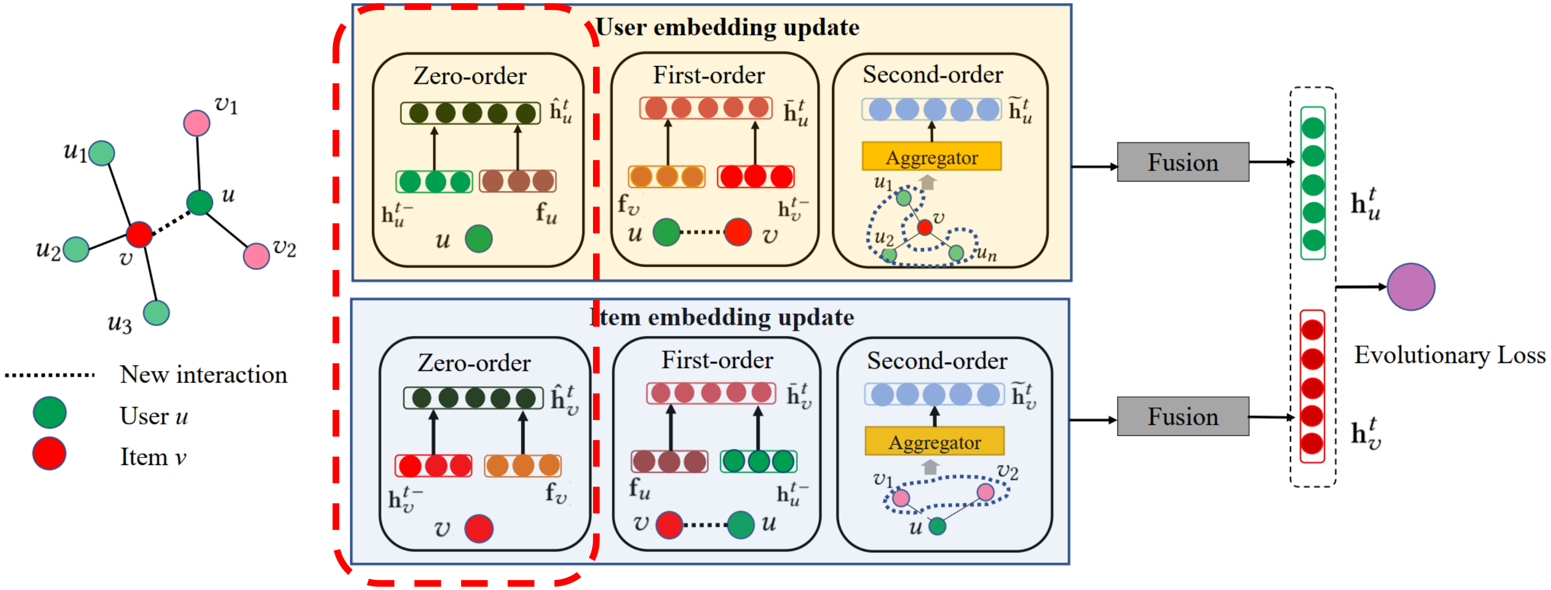
Dynamic Graph Collaborative Filtering (DGCF)

DGCF

- Zero-order relation inherits the user/item embedding from the previous states and time interval.

$$\hat{\mathbf{h}}_u^t = \theta_u(\mathbf{W}_0^u \mathbf{h}_u^{t^-} + \mathbf{w}_0 \Delta t_u + \mathbf{W}_0^f \mathbf{f}_u), \quad (1)$$

$$\hat{\mathbf{h}}_v^t = \theta_v(\mathbf{W}_0^v \mathbf{h}_v^{t^-} + \mathbf{w}_0 \Delta t_v + \mathbf{W}_0^f \mathbf{f}_v), \quad (2)$$

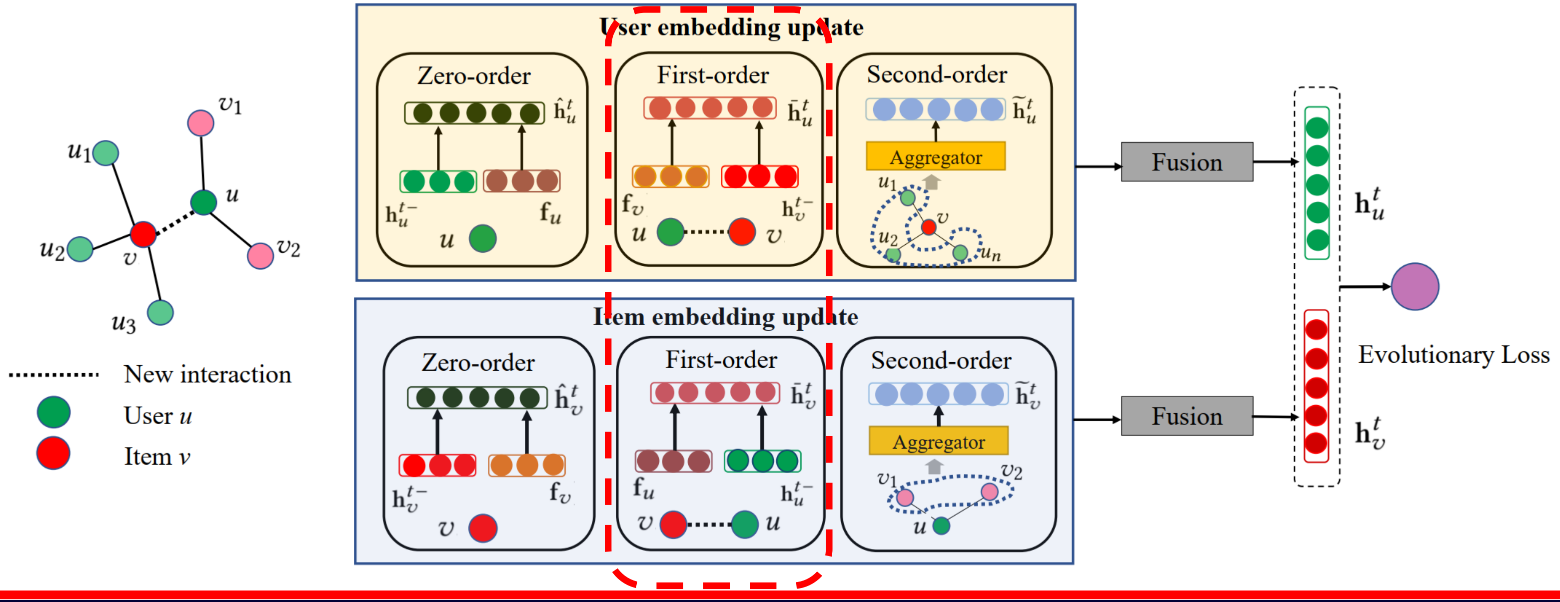


DGCF

- First-order relation propagates the user/item embedding to the other side.

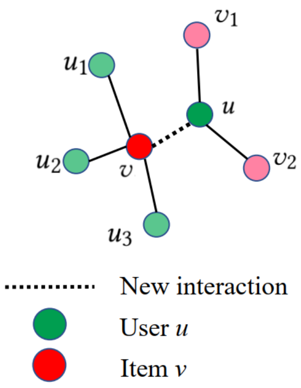
$$\bar{\mathbf{h}}_u^t = \phi_u(\mathbf{W}_1^u \mathbf{h}_v^{t^-} + \mathbf{W}_1^f \mathbf{f}_v), \quad (3)$$

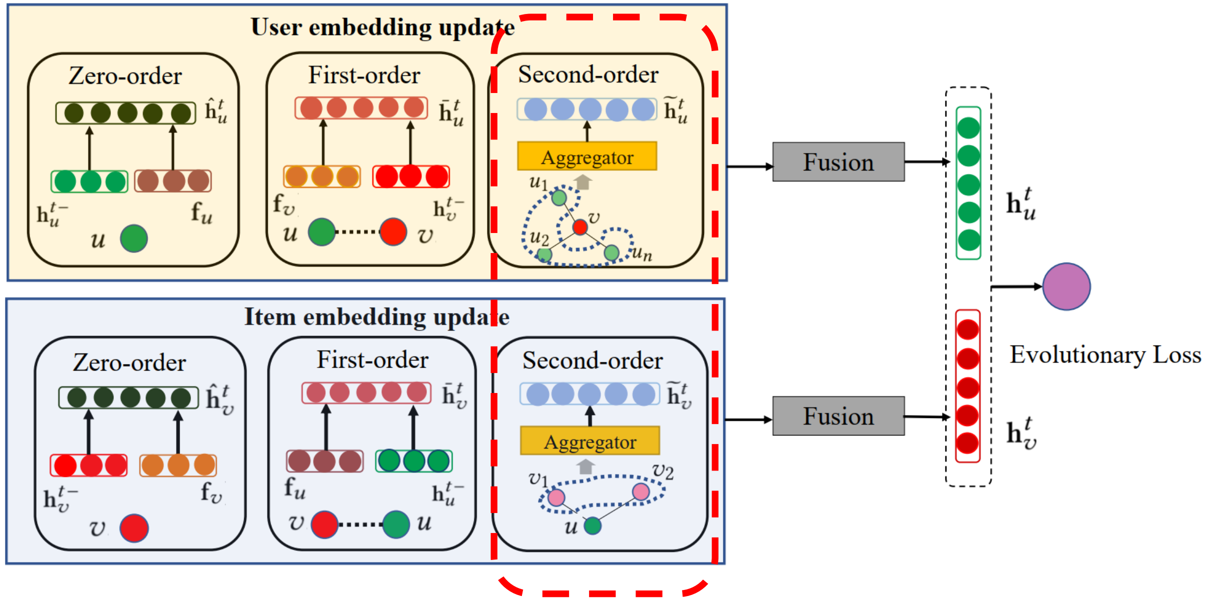
$$\bar{\mathbf{h}}_v^t = \phi_v(\mathbf{W}_1^v \mathbf{h}_u^{t^-} + \mathbf{W}_1^f \mathbf{f}_u), \quad (4)$$



DGCF

- Second-order relation aggregate the neighbors of each side and input them to the other side.
- Node u serves as a bridge passing information from $\{v_1, v_2\}$ to node v so that v receives the aggregated second-order information through u.





DGCF

- Three aggregation functions are tried in our model:

Mean aggregator

$$\tilde{\mathbf{h}}_u^t = \mathbf{h}_u^{t^-} + \frac{1}{|\mathcal{H}_v^u|}\sum_{u_i \in \mathcal{H}_u^v} \mathbf{W}_u^m \mathbf{h}_{u_i}^{t^-},$$

$$\tilde{\mathbf{h}}_v^t = \mathbf{h}_v^{t^-} + \frac{1}{|\mathcal{H}_u^v|}\sum_{v_i \in \mathcal{H}_v^u} \mathbf{W}_v^m \mathbf{h}_{v_i}^{t^-},$$

LSTM aggregator

$$\tilde{\mathbf{h}}_u^t = \mathbf{h}_u^{t^-} + \mathrm{LSTM}_u(\mathbf{h}_{u_1}^{t^-}, \mathbf{h}_{u_2}^{t^-}, \ldots, \mathbf{h}_{u_n}^{t^-}),$$

$$\tilde{\mathbf{h}}_v^t = \mathbf{h}_v^{t^-} + \mathrm{LSTM}_v(\mathbf{h}_{v_1}^{t^-}, \mathbf{h}_{v_2}^{t^-}, \ldots, \mathbf{h}_{v_m}^{t^-}).$$

Graph attention aggregator

$$\tilde{\mathbf{h}}_u^t = \sum_{u_i \in \mathcal{H}_u^v} \alpha_{ui} \mathbf{h}_{u_i}^{t^-}, \qquad \tilde{\mathbf{h}}_v^t = \sum_{u_i \in \mathcal{H}_v^u} \alpha_{vi} \mathbf{h}_{v_i}^{t^-},$$

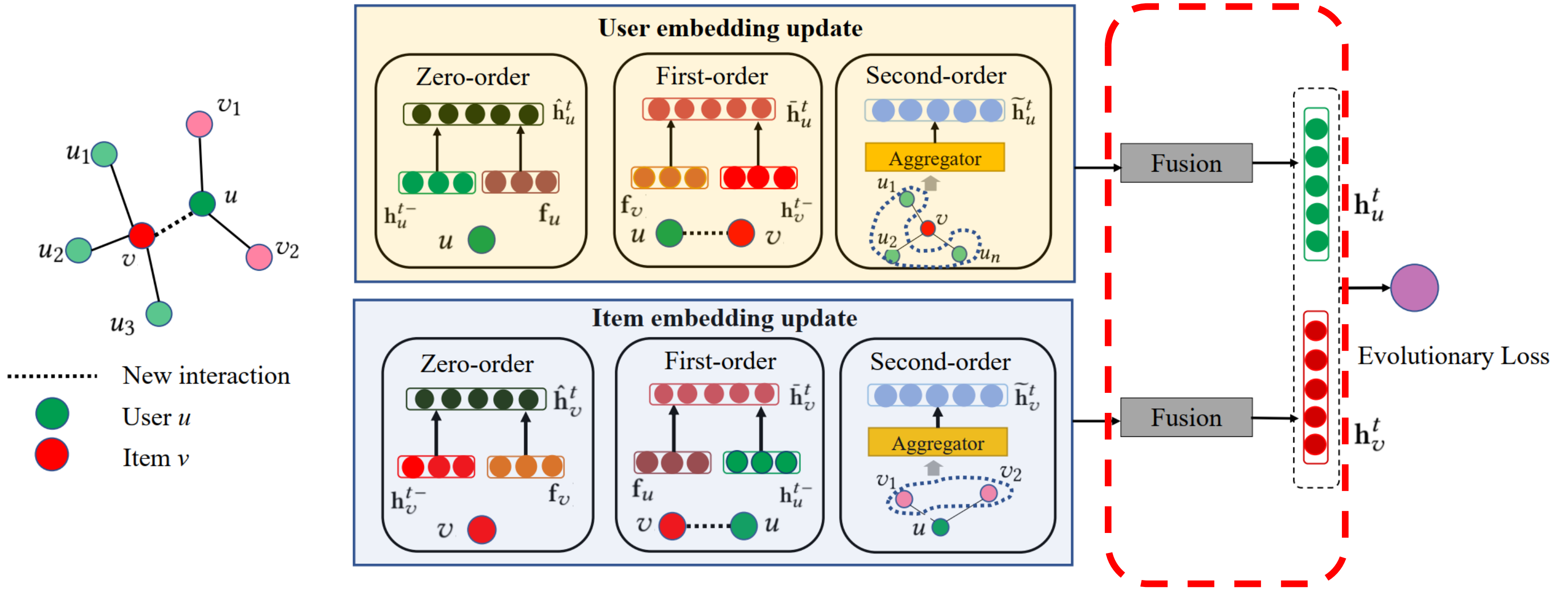
$$\alpha_{ui} = \frac{\exp(\mathrm{LeakyRelu}(\mathbf{W}_w[\mathbf{h}_u^{t^-} \parallel \mathbf{h}_{u_i}^{t^-}]))}{\sum_{u_i \in \mathcal{H}_u^v} \exp(\mathrm{LeakyRelu}(\mathbf{W}_w[h_u^{t^-} \parallel \mathbf{h}_{u_i}^{t^-}]))}, \quad \alpha_{vi} = \frac{\exp(\mathrm{LeakyRelu}(\mathbf{W}_w[\mathbf{h}_v^{t^-} \parallel \mathbf{h}_{v_i}^{t^-}]))}{\sum_{v_i \in \mathcal{H}_v^u} \exp(\mathrm{LeakyRelu}(\mathbf{W}_w[\mathbf{h}_v^{t^-} \parallel \mathbf{h}_{v_i}^{t^-}]))},$$

- The number of second-order neighbors for some node can be very large, so we select a fixed number of neighbors for aggregation.

DGCF

- In the end, we fuse the three relations and get the final user/item embedding.

$$\mathbf{h}_u^t = \mathrm{F}_u(\mathbf{W}_u^{zero}\hat{\mathbf{h}}_u^t + \mathbf{W}_u^{first}\bar{\mathbf{h}}_u^t + \mathbf{W}_u^{second}\widetilde{\mathbf{h}}_{\mathbf{u}}^{\mathbf{t}}),$$
$$\mathbf{h}_v^t = \mathrm{F}_v(\mathbf{W}_v^{zero}\hat{\mathbf{h}}_v^t + \mathbf{W}_v^{first}\bar{\mathbf{h}}_v^t + \mathbf{W}_v^{second}\widetilde{\mathbf{h}}_{\mathbf{v}}^{\mathbf{t}}),$$



Evolutionary loss

- Motivated by Jodie [1], we use the evolutionary loss to predict the item v that the user u is most likely to interact with at time t.

- Given a future time point, we can leverage our model to predict the future embeddings and then make recommendation.

[1] Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks, KDD 2019

Evolutionary loss

User future embedding: $\widehat{\mathbf{h}}_u^{t^+} = \mathrm{MLP}_u(\mathbf{h}_u^t \odot (\mathbf{1} + \mathbf{w}_t(t^+ - t))),$

Item future embedding: $\widehat{\mathbf{h}}_v^{t^+} = \mathrm{MLP}_v(\mathbf{W}_2\widehat{\mathbf{h}}_u^{t^+} + \mathbf{W}_3\mathbf{f}_u + \mathbf{W}_4\mathbf{f}_v)),$

- Loss function

$$\mathcal{L} = \sum_{(u,v,t,f)\in\{S_i\}_{i=0}^{I}} \|\widehat{\mathbf{h}}_v^{t^+} - \mathbf{h}_v^t\|_2 + \lambda_u\|\mathbf{h}_u^t - \mathbf{h}_u^{t^-}\|_2 + \alpha_v\|\mathbf{h}_v^t - \mathbf{h}_v^{t^-}\|_2,$$

Experiments

Datasets

Three datasets are used to evaluate our model, which have different action repetition rate.

TABLE II
THE AMOUNT OF USERS, ITEMS, INTERACTIONS AND ACTION REPETITION RATE IN EACH DATASET.

Data	Users	Items	Interactions	Action Repetition
Reddit	10000	1000	672447	79%
Wikipedia	8227	1000	157474	61%
LastFM	1000	1000	1293103	8.6%

Performance

- We compared our model with 6 state-of-the-art baseline models.
- Our model performs best on LastFM datasets, which has lowest action repetition.

Models	LastFM		Wikipedia		Reddit	
	MRR	R@10	MRR	R@10	MRR	R@10
LSTM	0.081	0.127	0.332	0.459	0.367	0.573
Time-LSTM	0.088	0.146	0.251	0.353	0.398	0.601
RRN	0.093	0.199	0.530	0.628	0.605	0.751
CTDNE	0.010	0.010	0.035	0.056	0.165	0.257
DeepCoevolve	0.021	0.042	0.515	0.563	0.243	0.305
Jodie	0.239	0.387	0.746	0.821	0.724	0.851
DGCF	**0.321**	**0.456**	**0.786**	**0.852**	**0.726**	**0.856**
Improvement	34.3%	27.7%	5.4%	3.6%	0.2%	0.5%

All data and codes available in https://github.com/CRIPAC-DIG/DGCF

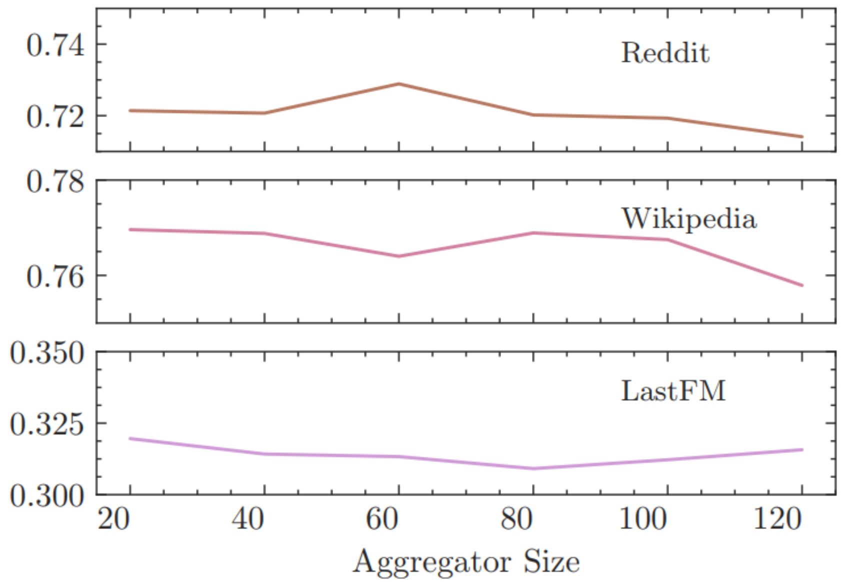
Aggregation function

- Three aggregation functions are tried in DGCF
- Graph attention achieves best performance among them.

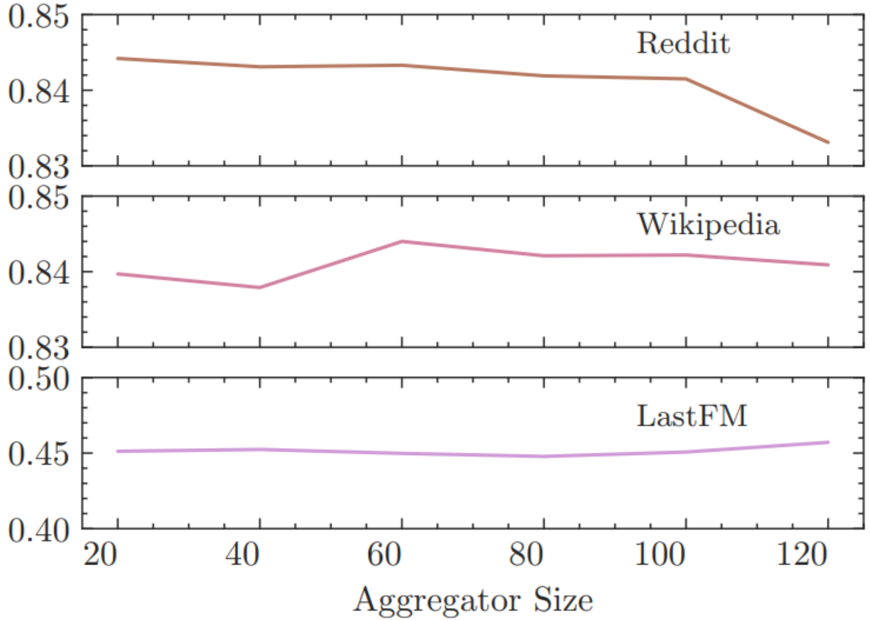
Aggregator	LastFM		Reddit		Wikipedia	
	MRR	R@10	MRR	R@10	MRR	R@10
Mean	0.296	0.419	0.721	0.844	0.770	0.836
LSTM	0.291	0.425	0.721	0.841	0.755	0.815
Attention	0.321	0.456	0.726	0.856	0.786	0.852

Aggregation size

- Generally, a smaller aggregation size can have a better performance, so we tend to choose 20 as the aggregation size.



(a) MRR



(b) Recall@10

Conclusion

Future directions

- We design a model based on dynamic graph to learn collaborative information explicitly in dynamic recommender system.
- In the future, we will also try to extend our model to more complicated graphs, such as knowledge graph, social network, and attributed graph.

[1] Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks, KDD 2019

Thank you!
Q&A