Redefining Neighbor for Improved Knowledge Graph Based Recommendations

Jingbo Yang, Ruge Zhao, Meixian Zhu
jingboy@stanford.edu, rugezhao@stanford.edu, mxzhu@stanford.edu

CS224W
2019 Autumn

Motivation

Traditionally, recommendation system has been modelled as a collaborative filtering problem.
Inclusion of knowledge graphs (KG) introduces a set of new relations and millions more entities.
Current state-of-the art recommender models with KG retain existing neighborhood structure. This leaves out potential correlations among similar items and structures in the network.

Our contribution:

Experimented with neighborhood expansion to improve KG based approach:

- Clustering
- Including K closest nodes
- Pooling 2-hop neighbors

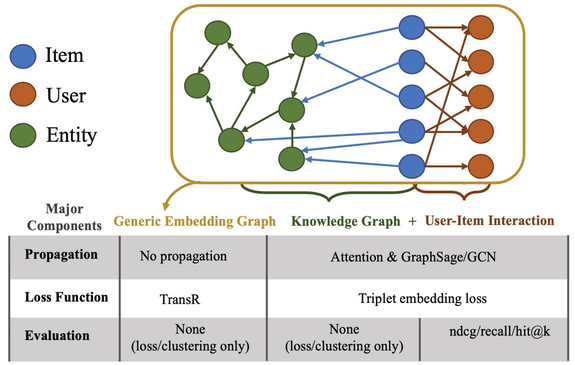
Dataset

We use the same datasets as the original KGAT implementation: Amazon Book, Last-FM and Yelp2018

	Type	Amazon Book	Last-FM	Yelp2018
User-Item Interaction	#Users #Items #Interactions	70,679 24,915 84,733	23,566 48,123 3,034,796	45,919 45,538 1,185,068
Knowledge Graph	#Entities #Relation Types #Triplets	88,572 39 2,557,746	58,266 9 464,567	90,961 42 1,853,704
Modified Clustering Coefficients		0.250	**0.433**	0.251

Yelp2018 and **Amazon Book** datasets have similar sparsity; **Last-FM** is the least sparse in terms of clustering coefficients

Approaches & Model



Propagation	No propagation	Attention & GraphSage/GCN	
Loss Function	TransR	Triplet embedding loss	
Evaluation	None (loss/clustering only)	None (loss/clustering only)	ndcg/recall/hit@k

3-phase alternating training algorithm:
1) node neighbor definition phase
2) embedding learning phase (TransR)
3) collaborative attentive propagation phase (GNN)

We proposed 3 methods for neighbor definition:

- **Clustering**: K-means clustering with k = 25, randomly add edges between items and current user in the same cluster
- Including **K closest** nodes: Add edges between the current user and his top K most similar users' purchased items in Euclidean distance
- Pooling **2-hop neighbors:** reduce 2-hop neighbors to 1-hop (direct) neighbors

Evaluation Metrics: averaged across all users

- **recall**@K
- **ndcg**@K: top K predicted items
  → discounted cumulative gain (dcg): $r_1 + \sum_{i=2}^{K} \frac{r_i}{log_2(i)}$
  → normalize by max possible dcg
- **hit**@K: at least one predicted in top K is correct

Results & Analysis

Model Performance with pretrained embeddings

	Amazon Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
SVD[1]	0.0918	0.0686	0.0538	0.0942	0.0533	0.0622
Vanilla KGAT	0.1489	0.1006	**0.087**	0.1325	**0.0712**	**0.0867**
Clustering	0.1328	0.0913	0.0851	0.1312	0.0645	0.0806
Closet K	0.0381	0.0315	0.0154	0.046	0.0144	0.0252
2-hop	**0.1495**	**0.1011**	0.086	**0.1327**	0.0684	0.0839

- 2-hop performs better than baseline on Amazon-Book for both recall@20 and ndcg@20
- 2-hop outperforms on ndcg@20 for Last-FM
- No improvement from other neighbor definitions
- Improvement is small using pretrained embeddings
- Results obtained by training from scratch show much larger improvement
- Fine-tuning on pretrained embeddings conflicts with our custom neighborhood definition.
- Pretrained embeddings originate from vanilla neighbor definition

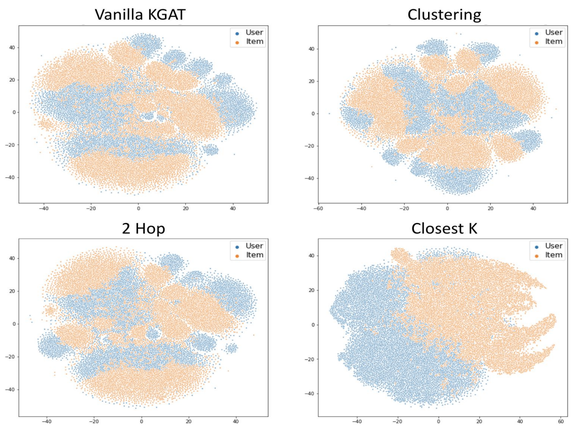
Pooling 2-hop neighbors consistently perform well for ndcg. Weighting predictions at different ranks associated with ndcg plays a role. See Analysis of metrics at different K below.

Comparison of 2-hop and Vanilla KGAT at different K on Last-FM dataset (without pretrained embeddings)

	recall@k		hit@k		ndcg@k	
	k=20	k=100	k=20	k=100	k=20	k=100
Vanilla KGAT	0.0296	**0.0892**	0.1907	**0.4264**	0.0602	**0.1265**
2-hop	**0.0314**	0.0772	**0.1955**	0.4137	**0.0631**	0.1207

- ndcg put higher weight on correctly predicted items at higher ranks while recall treats all predictions equally
- Possible that pooling 2-hop neighbors increases similarity across users → prediction on popular items are more accurate than those specific to individual users
- 2-hop neighborhood is better suited for identifying top ranking items
- Related to high clustering coefficient of Last-FM

Analysis of Trained Embeddings



- User and item embeddings overlap, differ by the "interacted" relation vector
- Embeddings from Vanilla KGAT, Clustering and 2-hop neighbors are similar
- Closest-K neighbors form a contiguous large cluster, destroys node similarities → worst performance

Conclusion

We experimented on multiple neighborhood definition methods.

- ndcg@20 increases 0.001 ~ 0.007 for Amazon-Book and Last-FM dataset by pooling 2-hop neighbors
- Alignment for top-matching items improved

Future Work

- Neighbor definition to improve performance at all k, possibly through a hybrid of vanilla KGAT and with custom neighbor definition
- General-purpose graph neural network that extends Pytorch Geometric to support high performance batching within a large graph

References

X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, "KGAT: knowledge graph attention network for recommendation," in KDD, pp. 950–958, 2019