Multi-Graph Convolution Collaborative Filtering

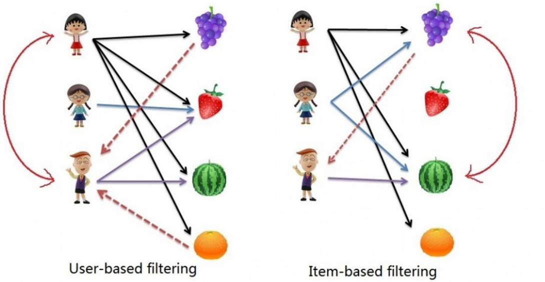
Jianing Sun[1], Yingxue Zhang[1], Chen Ma[2,1], Mark Coates[2], Huifeng Guo[3], Ruiming Tang[3] and Xiuqiang He[3]

[1]Huawei Noah's Ark Lab, Montreal Research Center, QC, Canada

[2]McGill University, Montreal, QC, Canada

[3]Huawei Noah's Ark Lab, Shenzhen, China

Collaborative Filtering: how likely a user will adopt an item based on the historical interactions, i.e. purchase and click



Two key components in modern CF models:

- Embedding: transform users and items to vectorized representations
- Interaction modeling: reconstruct historical interactions based on the embeddings, e.g. cosine similarity

Motivation

- Link prediction is at the core of recommender systems
- User item interaction can be readily modelled as a bipartite graph
- How well is the link prediction depends on how much latent information/user-item relationship that the learned user/item embeddings can represent
- Capture latent similarities & high-order connectivity & model complex user-item relationship

Motivation

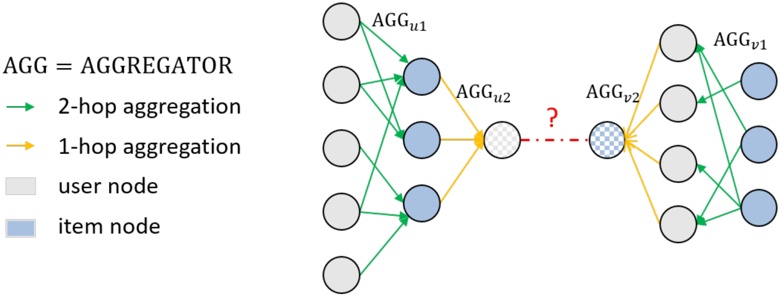
- Link (Edge) prediction is at the core of recommender systems
- User item interaction can be readily modelled as a bipartite graph
- How well is the link prediction depends on how much latent similarities/user-item relationship that the learned user/item (Node) embeddings can represent
- Capture latent similarities & high-order connectivity & model complex user-item relationship (Graph)

Collaborative filtering is a natural scenario for learning on graphs.

Methodology

Component 1: Bipartite Graph Convolutional Networks (Bipar-GCN) between users and items to iteratively aggregate k-hop neighborhood information:

- forward sampling: random sample neighbors from 1 to k search depth
- backward aggregating: train a set of aggregators to extract different information in a convolution manner, with shared parameters across nodes.



- We apply an element-wise weighted mean aggregator with learnable weights $\mathbf{Q}_u^k$:

$$\mathbf{h}_{\mathcal{N}(u)}^{k-1} = \mathrm{AGGREGATOR_u}\left(\left\{\mathbf{h}_v^{k-1}, v \in \mathcal{N}(u)\right\}\right), \tag{1}$$

$$\mathrm{AGGREGATOR_u} = \sigma\left(\mathrm{MEAN}\left(\left\{\mathbf{h}_v^{k-1} \cdot \mathbf{Q}_u^k, v \in \mathcal{N}(u)\right\}\right)\right).$$

  where $\mathcal{N}(u)$ is the neighborhood of users.

- The layer-k embeddings of the target user u can be represented as:

$$\mathbf{h}_u^k = \sigma\left(\mathbf{W}_u^k \cdot [\mathbf{h}_u^{k-1}; \mathbf{h}_{\mathcal{N}(u)}^{k-1}]\right), \;\; \mathbf{h}_u^0 = \mathbf{e}_u \tag{2}$$

- We apply the same operation on item nodes.

Component 2: Multi-Graph Encoding Layer (MGE)

- User-user graph and item-item graph are generated by computing the cosine similarity on rating matrix. Number of neighbors can be adjusted with a threshold.
- Learn on self graphs: for each user/item, we aggregate its adjacent information using a one-hop graph convolution layer with sum aggregator:

$$\begin{aligned} \mathbf{z}_u &= \sigma\left(\textstyle\sum_{i \in \mathcal{N}'(u)} \mathbf{e}_i \cdot \mathbf{M}_u\right), \\ \mathbf{z}_v &= \sigma\left(\textstyle\sum_{j \in \mathcal{N}'(v)} \mathbf{e}_j \cdot \mathbf{M}_v\right). \end{aligned} \tag{3}$$

Here $\mathcal{N}'(u)$ denotes the one-hop neighbourhood of user u in the user-user graph and $\mathcal{N}'(v)$ denotes the one-hop neighbourhood of item v in the item-item graph. $\mathbf{M}_u$ and $\mathbf{M}_v$ are weight matrices.

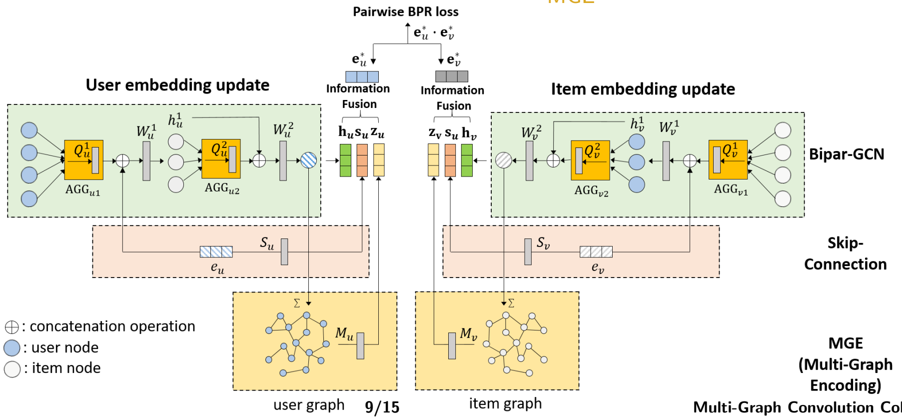
Component 3: Skip connection with input layer: refine the embedding with information passed directly from the input embedding.

Generally speaking,

- Bipar-GCN captures behavioural similarity between user and item.
- MGE captures proximity similarity of user-user and item-item.
- Skip connect captures individual node characteristics.

Overall Structure: $\mathbf{W_u^k}, \mathbf{M_u}, \mathbf{S_u}, \mathbf{W_v^k}, \mathbf{M_v}, \mathbf{S_v}, \mathbf{e}_u, \mathbf{e}_v$ are parameter vectors.

$$\mathbf{e}_u^* = \underbrace{\sigma\left(\mathbf{W}_u^k \cdot [\mathbf{h}_u^{k-1}; \mathbf{h}_{\mathcal{N}(u)}^{k-1}]\right)}_{\text{Bipar-GCN}} + \underbrace{\sigma\left(\sum_{i \in \mathcal{N}'(u)} \mathbf{e}_i \cdot \mathbf{M}_u\right)}_{\text{MGE}} + \underbrace{\sigma(\mathbf{e}_u \cdot \mathbf{S}_u)}_{\text{skip connect}}$$



Objective function

- Triplet loss function: BPR-MF[1]

$$
\begin{aligned}
loss = \sum_{(u,i,j)\in\mathcal{O}} & -\log\sigma(\mathbf{e}_u^* \cdot \mathbf{e}_i^* - \mathbf{e}_u^* \cdot \mathbf{e}_j^*) \\
& + \lambda||\Theta||_2^2 + \beta(||\mathbf{e}_u^*||_2^2 + ||\mathbf{e}_i^*||_2^2 + ||\mathbf{e}_j^*||_2^2)\,,
\end{aligned} \tag{4}
$$

where $\mathcal{O} = \{(u,i,j)|(u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-)\}$ denotes the training batch. Θ is the model parameter set. $\mathbf{e}_u^*$, $\mathbf{e}_i^*$, and $\mathbf{e}_j^*$ are the learned embeddings for user, positive item and negative item.

[1]Rendle et al. BPR: Bayesian Personalized Ranking from Implicit Feedback. UAI'09

	Gowalla		Amazon-Books		Amazon-CDs		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF	0.1291	0.1878	0.0250	0.0518	0.0865	0.0849	0.0494	0.0662
NeuMF	0.1326	0.1985	0.0253	0.0535	0.0913	0.1043	0.0513	0.0719
GC-MC	0.1395	0.1960	0.0288	0.0551	0.1245	0.1158	0.0597	0.0741
PinSage	0.1380	0.1947	0.0283	0.0545	0.1236	0.1118	<u>0.0612</u>	<u>0.0795</u>
NGCF	<u>0.1547</u>	**0.2237**	<u>0.0344</u>	<u>0.0630</u>	<u>0.1239</u>	<u>0.1138</u>	0.0581	0.0719
Multi-GCCF (d=64)	*__0.1595__	*0.2126	*__0.0363__	*__0.0656__	*__0.1390__	*__0.1271__	*__0.0667__	*__0.0810__
Multi-GCCF (d=128)	*__0.1649__	*0.2208	*__0.0391__	*__0.0705__	*__0.1543__	*__0.1350__	*__0.0686__	*__0.0835__

Table: Overall Performance Comparison.

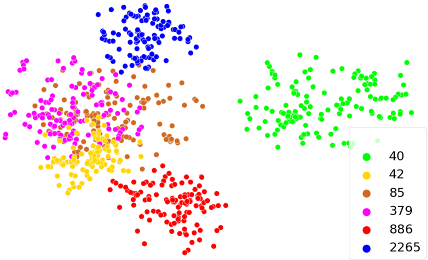
Gowalla: 29,858 user, 40,981 item, 0.084%. Amazon-Books: 52,643 user, 91,599 item, 0.056%. Amazon-CDs: 43,169 user, 35,648 item, 0.051%. Yelp2018: 45,919 user, 45,538 item, 0.056%.

Classic CF methods: BPRMF, NeuMF; Graph-based CF methods: GC-MC, PinSage, NGCF

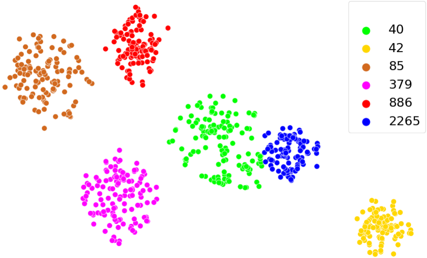
Architecture	Yelp2018	
	Recall@20	NDCG@20
Best baseline (d=64)	0.0612	0.0744
Best baseline (d=128)	0.0527	0.0641
1-hop Bipar-GCN	0.0650	0.0791
2-hop Bipar-GCN	0.0661	0.0804
2-hop Bipar-GCN + skip connect	0.0675	0.0821
2-hop Bipar-GCN + MGE	0.0672	0.0818
Multi-GCCF (d=128)	**0.0686**	**0.0835**

Table: Ablation studies.

Embedding Visualization



BPR-MF

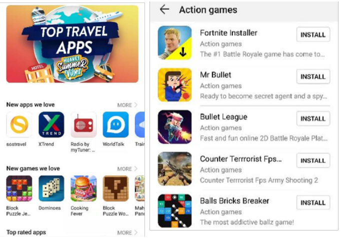


Multi-GCCF[2]

[2]Multi-Graph Convolution Collaborative Filtering, Sun et.al, ICDM'19

Follow-up Work

- Multi-GCCF has been well-supported in MindSpore, a unified training and inference AI framework developed by Huawei.
- Multi-GCCF has been deployed in Huawei App store for a large-scale recommendation task.
- Some research directions: more sophisticated neighborhood aggregator structure, fast and efficient sampling method on graphs, etc.



MindSpore

Thank you!