Attentive Group Recommendation

Da Cao[1], Xiangnan He[2], Lianhai Miao[1],
Yahui An[3], Chao Yang[1], Richang Hong[4]

July 11, 2018 @ SIGIR 2018

1 Hunan University
2 National University of Singapore
3 University of Electronic Science and Technology of China
4 Hefei University of Technology

SIGIR 2018

Group Recommendation

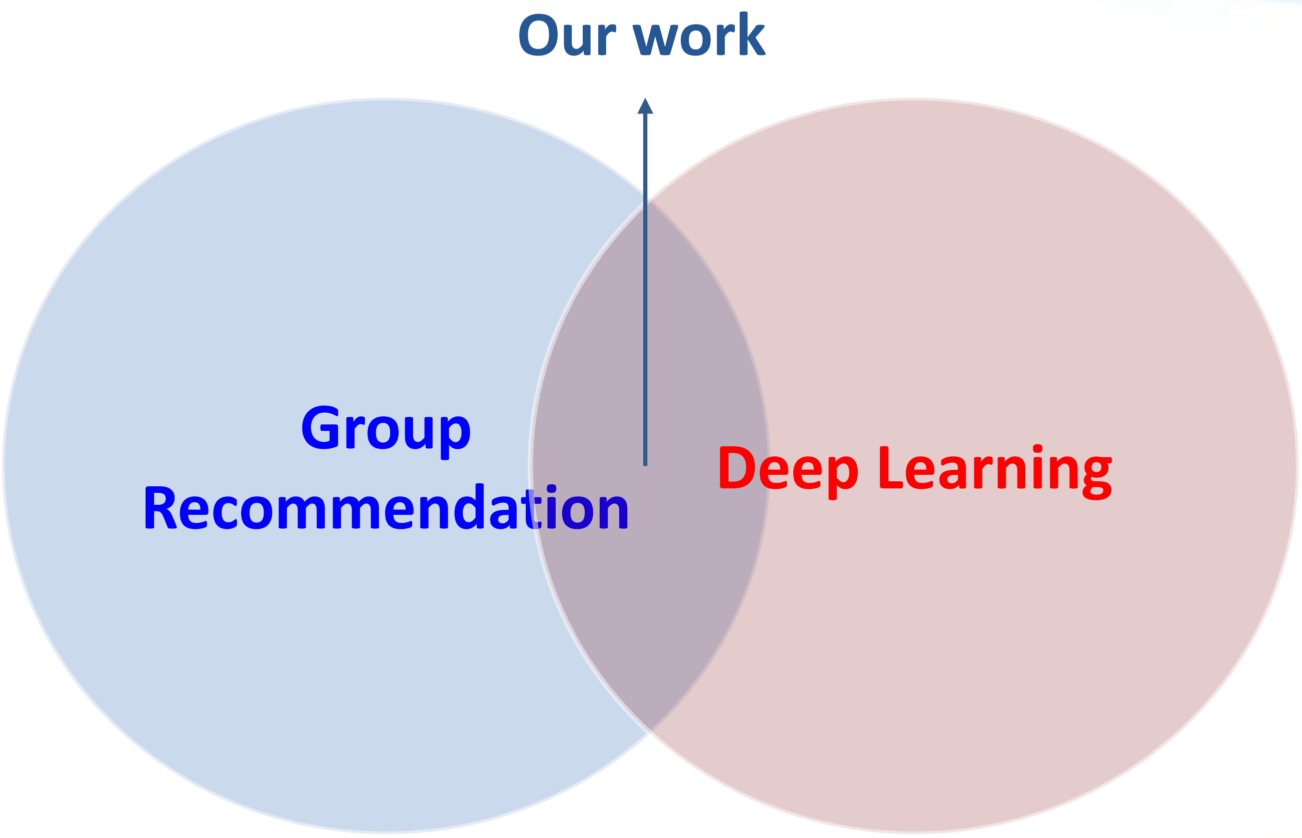
Examples:

- A group of travelers work together to plan a trip on *Mafengwo*.
- A group of teenagers organize a social party on *Meetup*.
- A group of researchers discuss a paper on *Mendeley*.

Limitations:

- Statistic aggregation strategies, such as average, least misery, maximum satisfaction, expertise, and so on.
- Mainly optimize group-item interactions, ignore significant user-item interactions.

Related Work



Related Work

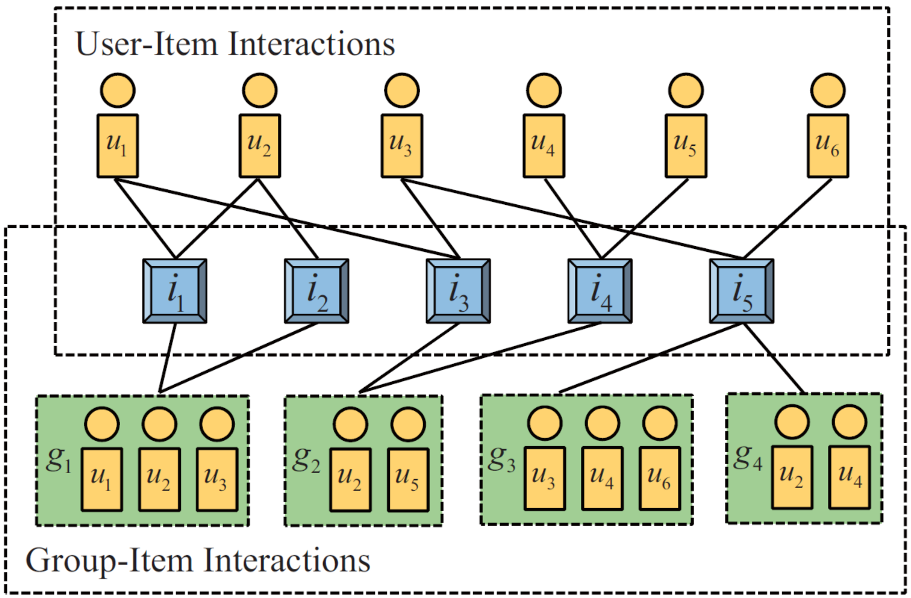
Group Recommendation

- Memory-based (e.g., preference aggregation, average, least misery, maximum satisfaction, expertise, and so on)
- Model-based (e.g., COM [SIGKDD 2014], DLGR [AAAI 2014])

Deep Learning-Based Recommendation

- Deep neural network (NCF [WWW 2017], NFM [SIGIR 2017])
- Attention mechanism (ACF [SIGIR 2017], AFM [IJCAI 2017])

Problem formulation



- Group-item interactions
- User-item interactions
- Group-user interactions

Proposed Methods

- Group representation learning which represents a group based on its members aggregation and its general preference
- Interaction learning with NCF which recommends items for both users and groups.

Group Representation Learning

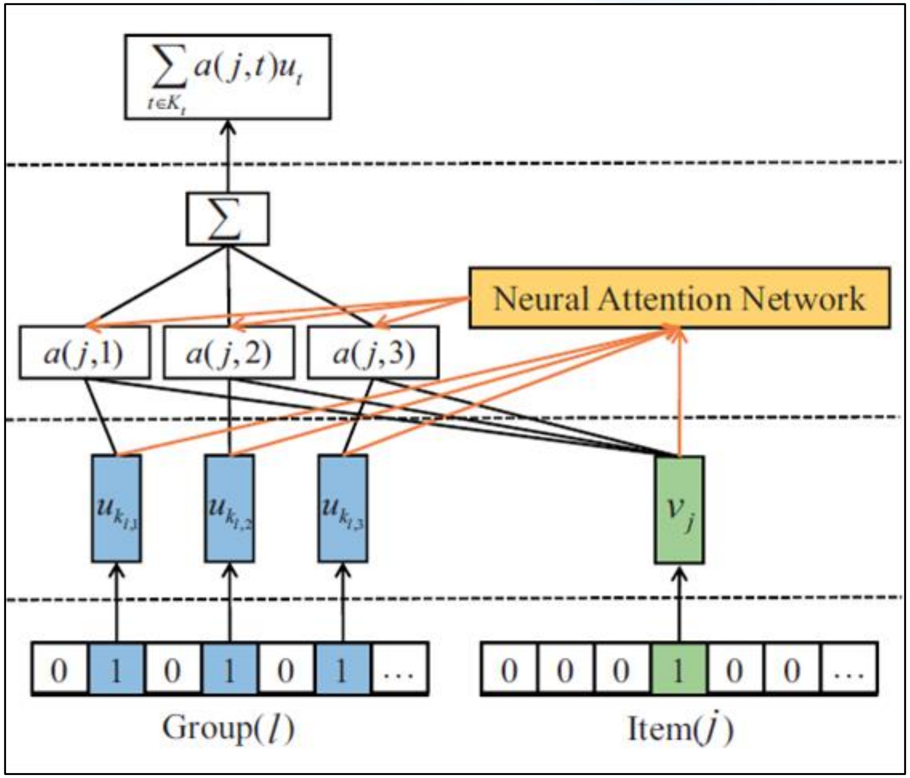
Group representation:

$$\mathbf{g}_l(j) = \underbrace{\sum_{t \in \mathcal{K}_l} \alpha(j,t)\mathbf{u}_t}_{\text{user embedding aggregation}} + \underbrace{\mathbf{q}_l}_{\text{group preference embedding}}$$

Attention network:

$$o(j,t) = \mathbf{h}^T \mathrm{ReLU}(\mathbf{P}_v \mathbf{v}_j + \mathbf{P}_u \mathbf{u}_t + \mathbf{b}),$$

$$\alpha(j,t) = \mathrm{softmax}(o(j,t)) = \frac{\exp o(j,t)}{\sum_{t' \in \mathcal{K}_l} \exp o(j,t')},$$



Interaction Learning with NCF

Pooling layer:

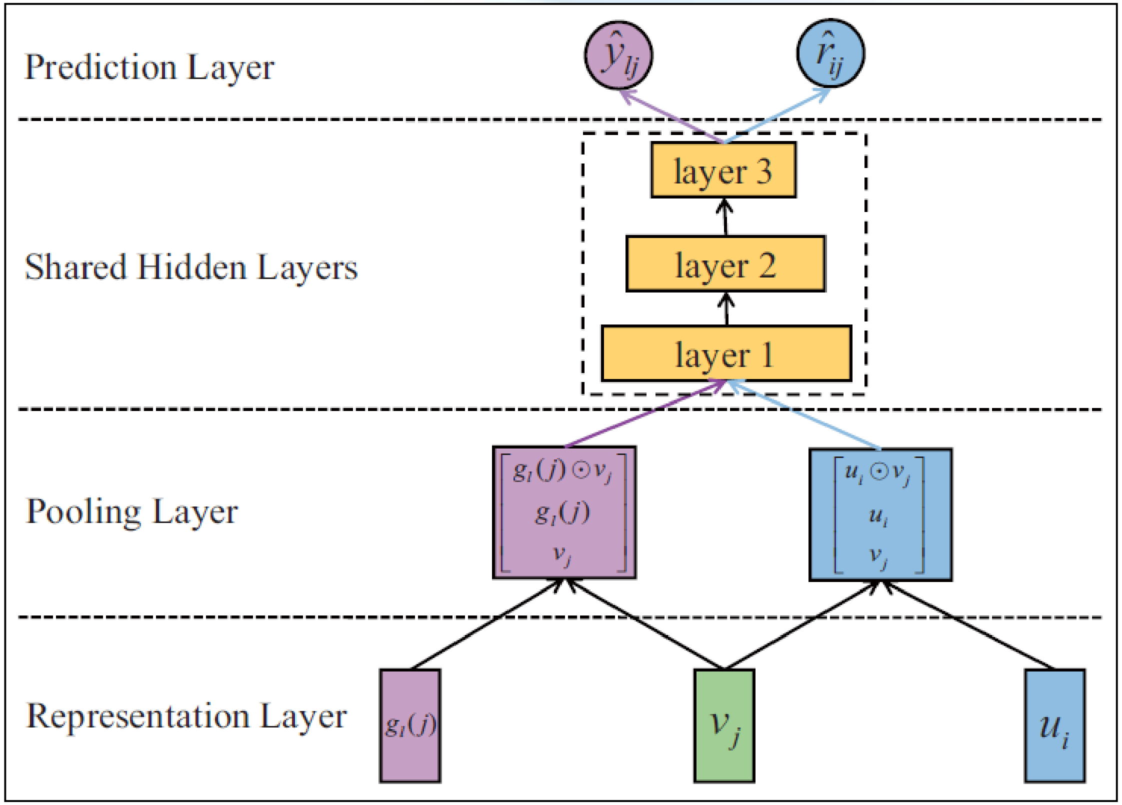
$$\mathbf{e}_0 = \varphi_{pooling}(\mathbf{g}_l(j), \mathbf{v}_j) = \begin{bmatrix} \mathbf{g}_l(j) \odot \mathbf{v}_j \\ \mathbf{g}_l(j) \\ \mathbf{v}_j \end{bmatrix}$$

Shared hidden layer:

$$\begin{cases} \mathbf{e}_1 = \mathrm{ReLU}(\mathbf{W}_1\mathbf{e}_0 + \mathbf{b}_1) \\ \mathbf{e}_2 = \mathrm{ReLU}(\mathbf{W}_2\mathbf{e}_1 + \mathbf{b}_2) \\ \ldots\ldots \\ \mathbf{e}_h = \mathrm{ReLU}(\mathbf{W}_h\mathbf{e}_{h-1} + \mathbf{b}_h) \end{cases},$$

Prediction layer:

$$\begin{cases} \hat{r}_{ij} = \mathbf{w}^T\mathbf{e}_h,\ if\ \mathbf{e}_0 = \varphi_{pooling}(\mathbf{u}_i, \mathbf{v}_j) \\ \hat{y}_{lj} = \mathbf{w}^T\mathbf{e}_h,\ if\ \mathbf{e}_0 = \varphi_{pooling}(\mathbf{g}_l(j), \mathbf{v}_j) \end{cases},$$



Optimization:

$$\mathcal{L}_{user} = \sum_{(i,j,s)\in O} (r_{ijs} - \hat{r}_{ijs})^2 = \sum_{(i,j,s)\in O} (\hat{r}_{ij} - \hat{r}_{is} - 1)^2, \quad \mathcal{L}_{group} = \sum_{(l,j,s)\in O'} (y_{ljs} - \hat{y}_{ljs})^2 = \sum_{(l,j,s)\in O'} (\hat{y}_{lj} - \hat{y}_{ls} - 1)^2,$$

Experimental Setup

- Two real-world datasets from Mafengwo and CAMRa2011:

Statistics of the evaluation datasets

Dataset	User#	Group#	Item#	User-Item Interactions	Group-Item Interactions
Mafengwo	5,275	995	1,513	39,761	3,595
CAMRa2011	602	290	7,710	116,344	145,068

- Evaluation protocols:
  - Leave-one-out: randomly remove one interaction of each user for testing, and pair it with 100 negative samples.
  - Top-N evaluation
  - The ranked list are evaluated by Hit Ratio and NDCG.

Baselines

State-of-the-art:

- NCF [WWW 2017]
- Popularity [RecSys 2010]
- COM [SIGKDD 2014]
- GREE [variance]

Aggregation strategies:

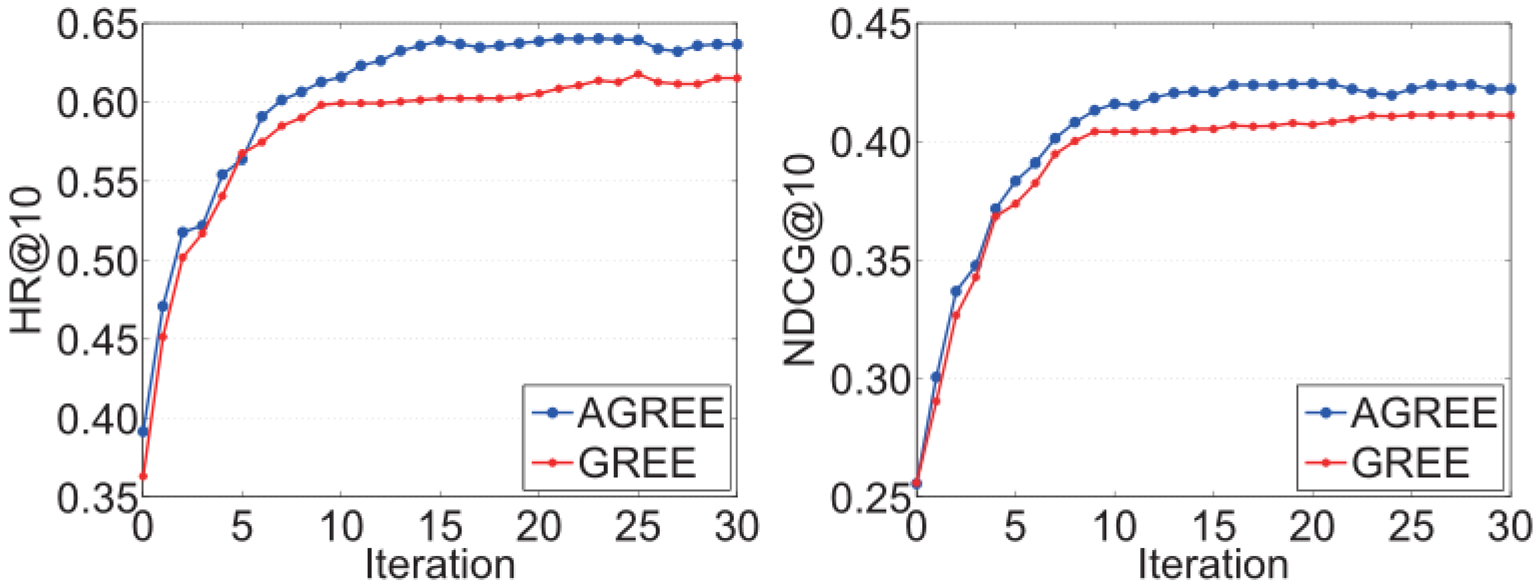
- NCF+avg [average]
- NCF+lm [least misery]
- NCF+ms [maximum satisfaction]
- NCF+exp [expertise]

Overall Performance Comparison

Overall Performance Comparison (Mafengwo)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
NCF	0.6363	0.5432	4.46e-06	0.4291	0.3405	7.18e-09	0.7417	0.5733	3.68e-05	0.6181	0.4020	3.95e-08
Popularity	0.4047	0.2876	2.02e-12	0.3115	0.2169	1.55e-11	0.4971	0.3172	2.09e-12	0.4251	0.2537	1.13e-11
COM	–	–	–	0.4420	0.3297	6.54e-09	–	–	–	0.5434	0.3727	1.36e-09
GREE	0.6306	0.5395	7.87e-07	0.4513	0.3577	1.20e-07	0.7206	0.5687	1.74e-06	0.6151	0.4111	3.25e-07
NCF+avg	–	–	–	0.4774	0.3669	2.86e-06	–	–	–	0.6222	0.4140	8.84e-07
NCF+lm	–	–	–	0.4744	0.3631	5.67e-07	–	–	–	0.6302	0.4152	1.45e-06
NCF+ms	–	–	–	0.4700	0.3616	3.46e-07	–	–	–	0.6281	0.4114	3.57e-07
NCF+exp	–	–	–	0.4724	0.3647	1.03e-06	–	–	–	0.6251	0.4015	3.61e-08
AGREE	**0.6383**	**0.5502**	–	**0.4814**	**0.3747**	–	**0.7491**	**0.5775**	–	**0.6400**	**0.4244**	–

1. AGREE achieves the best performance w.r.t. HR and NDCG when K=5 and K=10.
2. Neural network-based solutions (i.e., NCF, GREE, NCF+avg, NCF+lm, NCF+ms, NCF+exp, and AGREE) beat Popularity and COM.
3. There is no obvious winner among the score aggregation based solutions (i.e., NCF+avg, NCF+lm, NCF+ms, and NCF+exp).

Effect of Attention



(a) Mafengwo — HR@10 **(b) Mafengwo — NDCG@10**

1. AGREE beats GREE w.r.t. both metrics.
2. AGREE and GREE converge rather fast, reach their stable performance around the 20th iteration.

Effect of Attention

	Model	User #805	User #806	User #807	$\hat{y}$
Venue #30	GREE	0.333	0.333	0.333	0.260
	AGREE	0.286	0.302	**0.412**	**0.572**
Venue #32	GREE	0.333	0.333	0.333	0.096
	AGREE	0.222	**0.583**	0.195	**0.370**
Venue #106	GREE	0.333	0.333	0.333	0.192
	AGREE	**0.364**	0.287	0.347	**0.318**
Venue #65	GREE	0.333	0.333	0.333	0.132
	AGREE	**0.408**	0.311	0.281	**0.091**
Venue #121	GREE	0.333	0.333	0.333	0.132
	AGREE	0.335	**0.374**	0.291	**0.053**
Venue #123	GREE	0.333	0.333	0.333	0.109
	AGREE	0.288	**0.411**	0.301	**0.063**

1. The attention weights of group members vary significantly in AGREE.
2. For positive venues (#30, #32, #106), the prediction scores of AGREE are larger than that of GREE and are closer to the target value of 1. For negative venues (#65, #121, #123), ….

Importance of Components

Component Performance Comparison (Mafengwo)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
AGREE-U	0.6220	0.5364	2.80*e*-07	0.4141	0.3322	2.99*e*-09	0.7309	0.5716	9.02*e*-06	0.5709	0.3832	3.39*e*-09
AGREE-G	0.6363	0.5432	4.46*e*-06	0.4291	0.3405	7.18*e*-09	0.7417	0.5733	3.68*e*-05	0.6181	0.4020	3.95*e*-08
AGREE	**0.6383**	**0.5502**	–	**0.4814**	**0.3747**	–	**0.7491**	**0.5775**	–	**0.6400**	**0.4244**	–

Component Performance Comparison (CAMRa2011)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
AGREE-U	0.6043	0.3945	1.12*e*-07	0.5793	0.3832	4.47*e*-07	0.7601	0.4465	4.09*e*-08	0.7441	0.4376	6.36*e*-08
AGREE-G	0.6119	0.4018	1.03*e*-06	0.5803	0.3896	9.02*e*-06	0.7894	0.4535	1.89*e*-07	0.7593	0.4448	3.92*e*-07
AGREE	**0.6223**	**0.4118**	–	**0.5883**	**0.3955**	–	**0.7967**	**0.4687**	–	**0.7807**	**0.4575**	–

1. AGREE consistently and significantly outperforms AGREE-U and AGREE-G on both datasets w.r.t. both metrics.
2. AGREE-G shows better performance that AGREE-U on both datasets w.r.t. both metrics.

New-User Cold-Start

New-User Cold-Start (Mafengwo)						
	K=5			K=10		
	HR	NDCG	p-value	HR	NDCG	p-value
Popularity	0.3115	0.2169	1.22*e*-12	0.4251	0.2537	8.18*e*-13
NCF+group	0.6576	0.4687	2.20*e*-09	0.7716	0.5512	9.62*e*-09
AGREE	**0.6989**	**0.5146**	–	**0.8013**	**0.5830**	–

1. AGREE outperforms Popularity and NCF+group.
2. Personalized methods (AGREE and NCF+group) achieve preferable results as compared with non-personalized method (Popularity).

Conclusion and Future Work

- We addressed the group recommendation problem under the neural network structure :
  - How to obtain a semantic representation for a group.
  - How to reinforce the recommendation performance of both group-item and user-item.
- Experiments show promising results:
  - Overall performance comparison.
  - Effect of attention.
  - Importance of components.
  - New-user cold-start.

- Future work:
  - The utility of social network.
  - Online learning.

Thanks!

Codes: https://github.com/LianHaiMiao/Attentive-Group-Recommendation