论文｜AGREE-基于注意力机制的群组推荐（附代码）

原创 Thinkgamer 搜索与推荐Wiki 2020-04-24

收录于话题
#论文笔记

19个

本文内容的目录结构如下：

这篇文章主要分享的论文是2018年被CCF收录的一篇论文：Attentive Group Recommendation（基于注意力机制的群组推荐），第一作者是湖南大学的曹达老师，二作是论文Neural Collaborative Filtering的作者何老师。当然也会结合小编的工作来进行一些补充说明，写的不好，欢迎拍砖！

这篇文章结合了注意力机制和NCF框架，使得在群组推荐中有良好的效果。

关于群组推荐的概述可以参考：论文｜组推荐系统及其应用研究

关于Neural Collaborative Filtering可以参考：论文｜被“玩烂”了的协同过滤加上神经网络怎么搞

关于注意力的解释可以参考：深度学习中的注意力机制

1、聊一聊群组推荐

群组推荐在工业推荐系统中应用十分广泛，其主要作用在新老用户上。

对于新用户来讲， 则是冷启动，常规的冷启动解决办法是，根据新用户的注册信息（性别、地域等，这个一般和产品设计进行结合）找到其所属的群组，将群组下的Top信息返回推荐给当前来访的新用户。

对于老用户来讲，可以作为一路召回使用。比如对所有用户进行群组划分，再统计群组下的Top内容作为群组下用户的一路召回内容。当然也可以按照某种属性进行群组划分。

所以群组推荐的实际应用中是很广泛和有效的，上面举的例子只是冰山一角，实际中要比这个复杂和广泛。但往往我们在得到群组下Top内容的时候采用的方法简单粗暴，这一点论文中也提到了群组推荐中的一个基本问题是如何汇总小组成员的偏好以推断小组的决定，因为群组中不同的成员扮演的角色是不一样的， 常见的计算方法有：

- 均值策略，即认为群组中每个人贡献是等价的，计算群组内top内容时，取对某内容贡献的均值。
- 最小化策略，即取群组内对某内容的最低贡献作为该内容的排序值
- 最大化策略，和最小化策略相反，即取群组内对某内容的最高贡献作为该内容的排序值
- 专家策略，根据用户在项目上的专业知识为其赋予了个性化的权重

但是，这些预定义的策略和数据无关，因为这些策略很难灵活的捕获群组成员的动态权重。因此提出了self-Attention Mechanism + NCF的思路，即Attentive Group Recommendation（AGREE）。

AGREE的主要贡献点有：

- 第一次将神经网络中的注意力机制用于群组推荐，基于输入数据动态地学习融合策略。
- 引入用户和内容之间的交互行为，提升群组推荐的效果，同时在一定程度上能够缓解冷启动问题（首先根据某种策略得到群组，新用户来访时匹配其对应的群组，根据群组内其他成员选取的top内容作为新用户的冷启动内容）。
- 基于数据集（爬取的和公开的）进行了大量的实验，并进行公开，促进学术研究。

2、模型介绍

AGREE模型包含两部：

- （1）：基于小组成员聚集和偏好表示的群组表示学习
- （2）：通过NCF进行交互学习，为用户和群组推荐项目

2.1、符号表示含义和要解决的问题

符号表示的规则

- 粗体大写字母（eg：$\mathbf{X}$）表示矩阵
- 粗体小写字母（eg：$\mathbf{x}$）表示向量
- 粗体小写字母（eg：x）表示标量
- 花体字母（eg：$\mathcal{X}$）表示集合

如果未特别说明，所有的向量都是列向量。

下图说明了群组推荐任务中输入数据形态：

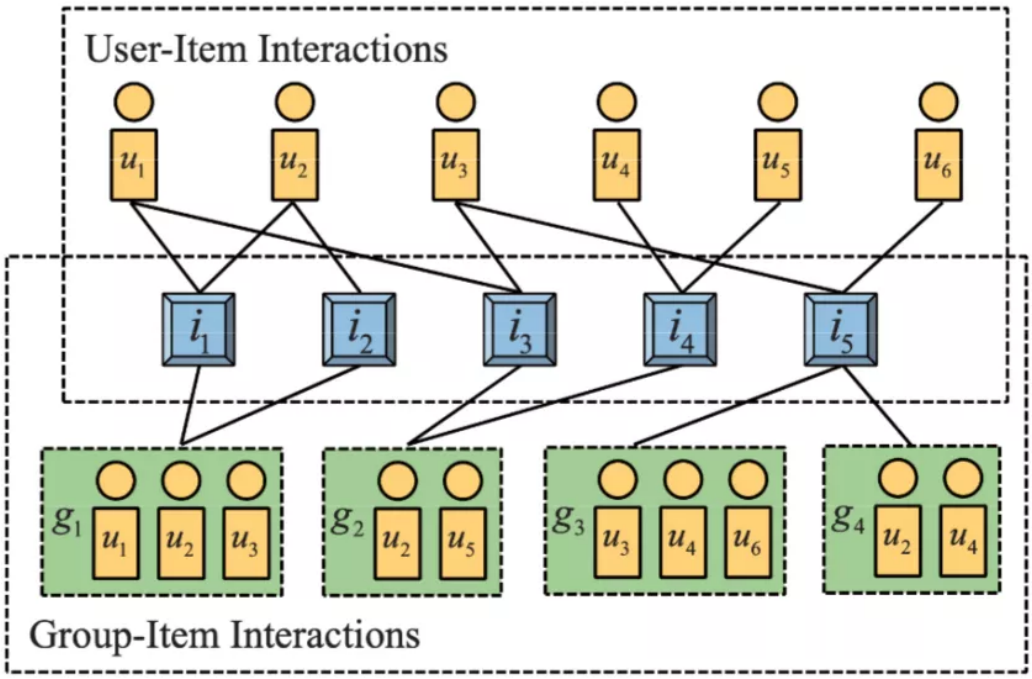


Figure 1: Illustration of the input data of attentive group recommendation task, which contains user-item interactions and group-item interactions.

群组推荐任务中输入数据形态

论文中给出的群组推荐问题定义的是：对于给定的群组，输出为该群组推荐的 Top-K物品。

整个模型训练的

- 输入是：用户集合$\{u_1, u_2, ...\}$，群组集合$\{g_1, g_2, g_3, ...\}$，物品集合$\{v_1, v_2, v_3, ...\}$，群组和物品的交互矩阵Y，用户和物品的交互矩阵R
- 输出是：两个个性化的排序函数，f_g（将一个物品映射给一个群组的分值）、f_u（将一个物品映射给一个用户的分值）

2.2、基于注意力的群组表征学习

论文建议基于表征学习解决群组推荐问题，在该框架下，每个实体都被表示为一个向量，该向量对实体的固有属性（eg：单词的语义，用户的兴趣等）进行编码，且可以从数据中学习。物品推荐中一个众所周知的矩阵分解方法就是一个表征学习模型，该模型将用户和物品通过嵌入向量联系起来。

u_i表示用户向量，v_j表示物品向量，我们的目标是通过群组成员的兴趣偏好为每个群组学习到一个向量，要从数据中动态学习聚合策略，可以将群组向量定义成依赖用户和物品的向量，抽象表示为：

$$g_l(j) = f_a(v_j, \{u_t\})$$

u_t 表示 l群组中的用户。$g_l(j)$表示群组g_l对物品v_j的偏好向量。在AGREE模型中，$g_l(j)$包括两部分：用户向量聚合和群组偏好向量。形式化表示为：

$$g_l(j) = \sum_{u_t \in g_l} a(j,t)u_t + q_l$$

上面公式中加号左侧为用户向量，加号右侧为群组偏好向量。

用户向量聚合

对群组中的成员进行加权求和，系数$a(j,t)$是一个可学习的参数，表示用户u_t选择物品v_j的影响力，很显然，如果一个用户在一个物品上的影响力比较大，则其对应的$a(j,t)$也会比较大。例如在决定一个群组该去哪里旅行的决定中，如果一个用户去过中国很多次，那么这个用户对是否去中国旅行的决定权就会更大一些，这也比较符合人的期望。

在表征学习框架下，u_t表示用户的历史偏好，v_j表示物品的属性，我们将$a(j,t)$设置为注意力神经网络，输入为u_t和v_j：

$$o(j,t) = h^T ReLU(P_v v_t + P_u u_t + b)$$
$$a(j,t) = softmax(o(j,t)) = \frac{exp(o(j,t))}{\sum_{t \in g_l} exp(o(j,t))}$$

其中：

- $t \in g_l$表示群组g_l 中用户的下标
- P_v 和 P_u表示物品v_j和用户u_t对应的权重矩阵

- b 表示偏置
- $ReLU(.)$ 表示激活函数
- h^T 表示将结果映射为$o(j,t)$的权重向量
- $softmax(.)$ 表示将结果归一化的概率函数

下图说明了用户向量聚合部分的模型设计。

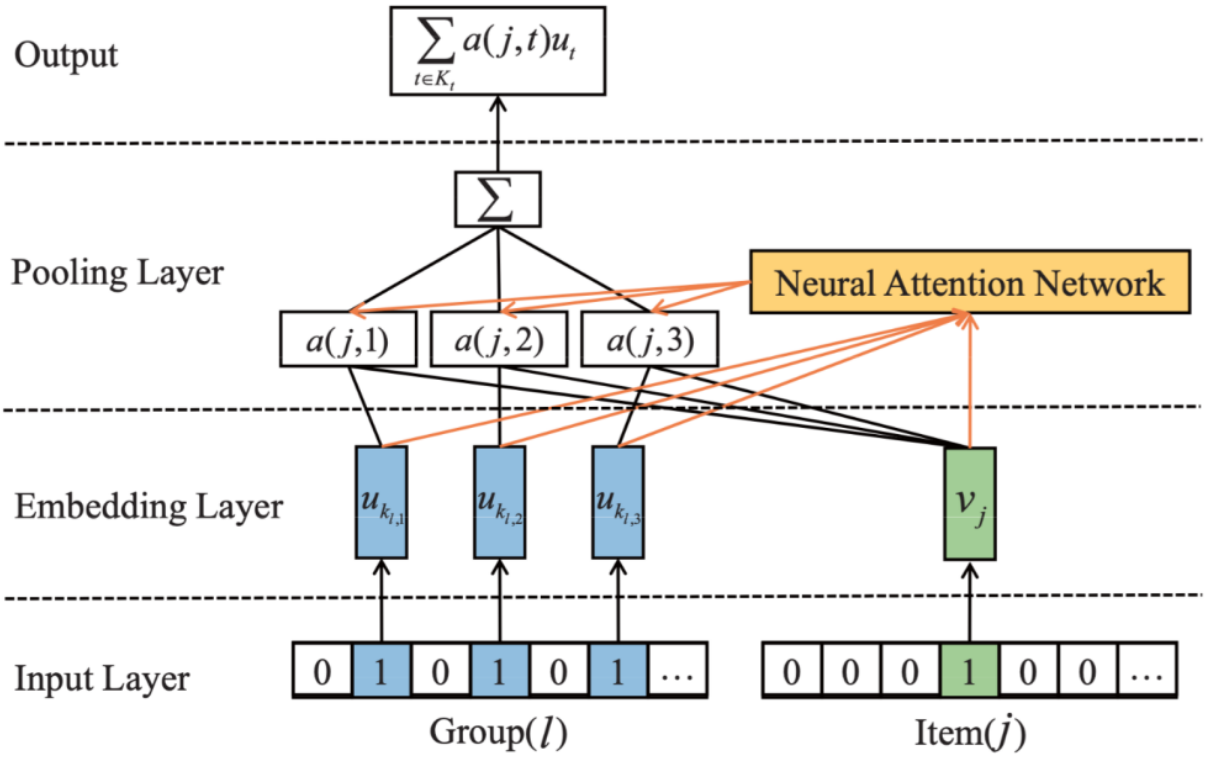


Figure 2: Illustration of the user embedding aggregation component based on neural attention network.

群组偏好向量

除了对用户向量进行聚合，还对群组偏好向量进行了表示，目的是考虑群组的整体偏好。这里需要考虑的是当用户汇聚成一个小组时，他们追求的目标可能会与每个小组成员的偏好不同。例如：一个三口之家去看电影，小朋友喜欢看动画片，父母喜欢看浪漫电影，那么他们最终可能会考虑看一部教育电影。因此用一个通用的向量表示群组的普遍偏好是很必要的。这里直接使用群组成员的向量求和作为群组的向量表示，在后续的实验中也证明这是有效的。

2.3、基于NCF的交互学习

NCF是一个关于物品推荐的多层神经网络框架，可以参见之前的文章：**Neural Collaborative Filtering**。其主要思路是将用户和物品的向量，输入到一个定制的神经网络中，基于历史数据学校用户和物品之间的交互，由于神经网络具有很强的表达能力，NCF比传统的矩阵分解模型

效果要好很多。因此我们选择了NCF框架对用户、物品的表达向量和用户-物品之间的交互进行端到端的学习。

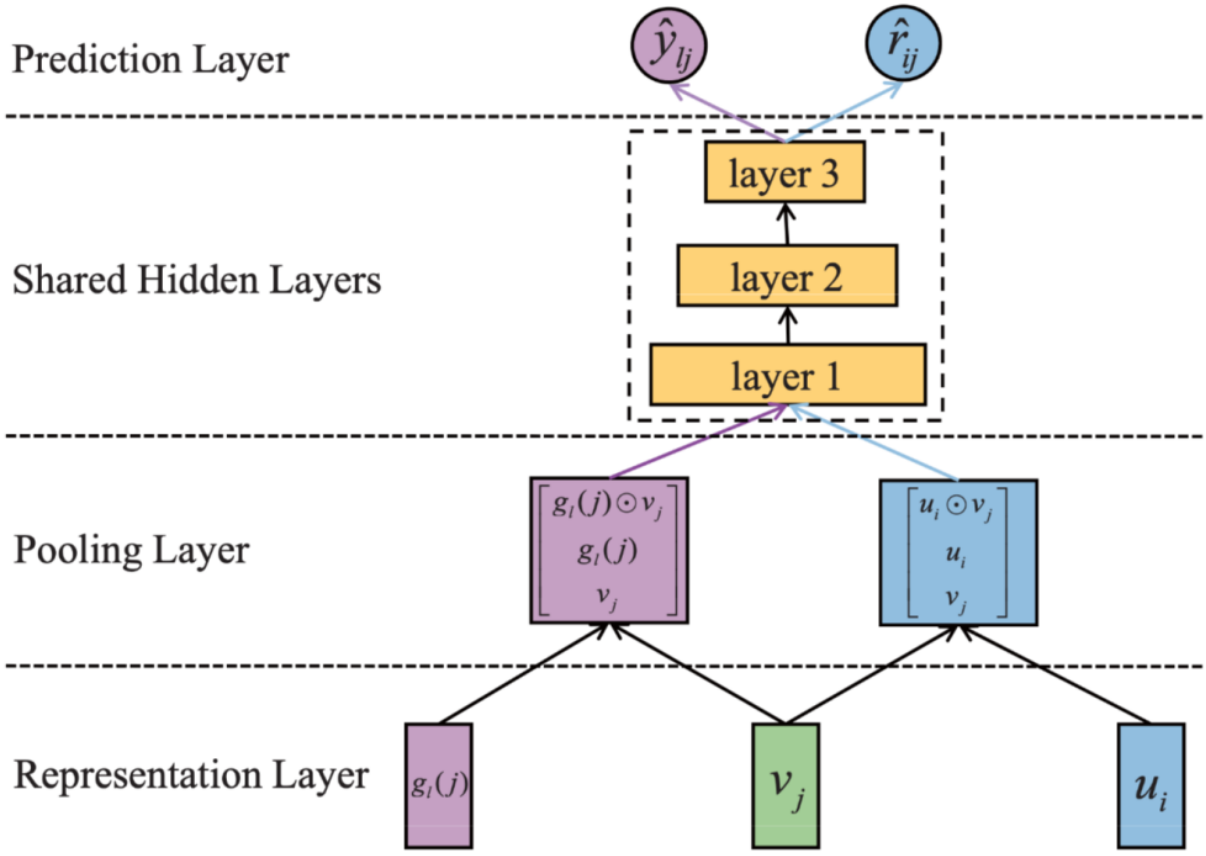


Figure 3: Illustration of interaction learning based on NCF.

NCF框架的解决方案

上图展示了论文使用的NCF解决方案，因为需要同时为群组和用户进行推荐，所以在输入的时候既有用户-物品对(u_i, v_j)，还有群组项对(g_l, v_j)，经过 Representation Layer 返回每个实体的表征向量（具体的生成方法参考：2.2、参考基于注意力的群组表征学习），然后输入到Pooling Layer（池化层）和Hidden Layer （隐藏层），（hidden layer被两个任务进行共享），最后获得预测分数。

pooling layer

假设输入的是群组-项目对(g_l, v_j)，pooling layer（池化层）首先对输入的向量进行g_l, v_j进行逐元素乘积，然后再与g_j, v_j进行连接，表示如下：

$$e_0 = \varphi_{pooling}(g_l(j), v(j)) = \begin{bmatrix} g_l(j) \odot v(j) \\ g_l(j) \\ v(j) \end{bmatrix}$$

之所以用这种方式表示，是因为内积能够有效的捕获到群组和物品之间的交互，同时连接群组和物品本身的信息能够避免信息丢失。同样用户-物品项之间的池化也采用这种方式。

hidden layer

池化层之上这是一个多层的隐藏层结构，能够捕获群组、用户、物品之间的非线性、高维的交互。

$$\begin{cases} e_1 = ReLU(W_1 e_0 + b_1) \\ e_2 = ReLU(W_2 e_1 + b_2) \\ \cdots \\ e_h = ReLU(W_h e_{h-1} + b_h) \end{cases}$$

其中：

- W_h 表示第h层的权重矩阵
- b_h 表示第h层的偏置
- e_h 表示第h层的输出
- $ReLU$ 表示激活函数

最后一层输出e_h可以经过下面的公式转化为预测分：

$$\begin{cases} \hat{r}_{ij} = w^T e_h, if e_0 = \varphi_{pooling}(u_i, v_j) \\ \hat{y}_{ij} = w^T e_h, if e_0 = \varphi_{pooling}(g_l(j), v_j) \end{cases}$$

其中

- w 表示预测层的权重
- $\hat{r}_{ij}$ 表示 用户对物品的预测分
- $\hat{y}_{ij}$ 表示 群组对物品的预测分

这里需要强调的是，我们针对用户-物品和群组-物品两个预测任务共享了隐藏层，因为群组的embedding向量是通过用户的embedding向量汇聚而成的，他们在同一个向量空间中，且有利于两个任务的相互补充。

2.4、模型优化

目标函数

群组推荐本质也是做排序，所以这里使用的pairwise方法（关于pairwise、pointwise、listwise的区别可以参考：怎么理解基于机器学习“四大支柱”划分的学习排序方法）。

用户-物品：

$$L_{user} = \sum_{(i,j,s) \in O} (r_{i,j,s} - \hat{r}_{i,j,s})^2 = \sum_{(i,j,s) \in O} (\hat{r}_{i,j} - \hat{r}_{i,s} - 1)^2$$

其中：

- O 表示训练集
- (i,j,s) 表示用户u_i 对物品v_j有行为，但对物品v_s没有行为 （v_s 是负采样产生）
- $\hat{r}_{i,j,s} = \hat{r}_{i,j} - \hat{r}_{i,s}$ 表示观察到的交互(u_i, v_j)和为观察到的交互(u_i, v_s)之间的差值分数

这里认为有行为交互分值为1，没有交互分值为0，所以$r_{i,j,s} = r_{i,j} - r_{i,s} = 1$，又因为用户-物品的损失函数中为平方和，所以最后形式转为了$(\hat{r}_{i,j} - \hat{r}_{i,s} - 1)$。

同样，群组-物品的损失函数为：

$$L_{group} = \sum_{(l,j,s) \in O'} (y_{l,j,s} - \hat{y}_{l,j,s})^2 = \sum_{(l,j,s) \in O'} (\hat{y}_{l,j} - \hat{y}_{l,s} - 1)^2$$

优化技巧

主要是在模型设计和训练上，有以下几点：

1、mini-batch。需要注意的首先需要将训练集进行shuffle，即混洗，保证数据之间是无序的。

2、pre-training。神经网络对于相关的初始化值是比较敏感的，因此在AGREE模型中，对网络的初始值进行了初始化，初始化的方法是去除了AGREE中的Attention机制，将得到的相关参数复用在AGREE中，而且在预训练模型中使用的是Adma优化方法，因为速度较快，而在AGREE模型中使用的则是SGD（关于SGD可参考：梯度算法之批量梯度下降，随机梯度下降和小批量梯度下降）。

3、droupout。由于神经网络的拟合能力强，为了提升模型的泛化能力，这里采用了深度学习中比较常见的Droupout机制，不仅在第一层中去除了p，而且隐藏层也进行了剔除。另外需要注意的是在训练的时候需要进行dropout，但是在预测的时候不需要dropout。

3、实验效果

实验采用的数据集有两个：

- 自己爬取和整理的马蜂窝数据
- CAMAa2011

这两个数据集都只有正样本，因此从缺失的数据中进行随机负采样构建负样本。而且适当的调整正负样本的比例能够增强实验的效果，所以论文中选择的负样本的采样率在4～6之间。

实验采用的方法是：留一法（即交叉验证法），为了评估算法的效果，使用了两种指标来进行评估：

- Hit Ratio（HR）
- Normalized Discounted Cumulative Gain (NDCG)

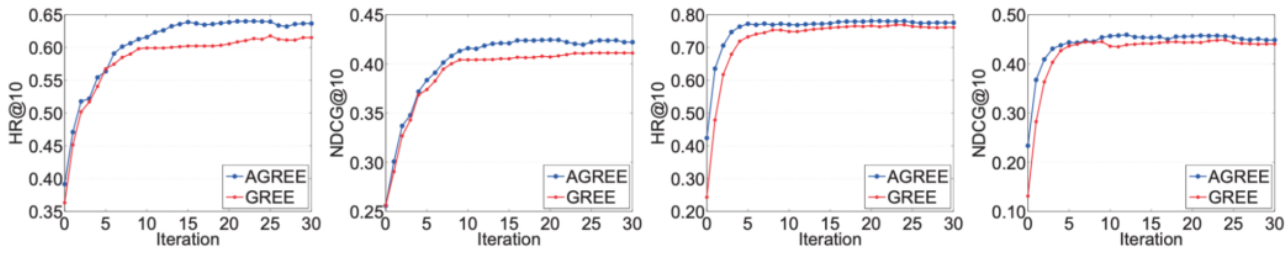
实验中超参数的设置：

- 负采样率：4
- 嵌入层使用Glorot初始化权重
- 隐藏层使用均值为0，标准差为0.1的高斯分布初始化权重
- 基于梯度优化的算法使用Adma优化，mini-batch设置为[128, 256, 512, 1024]，学习率设置为[0.001, 0.005, 0.01, 0.05, 0.1]
- 隐藏成第一层的维度和嵌入层维度相同，为32
- 激活函数：ReLU
- 每组实验参数进行5次实验，选取平均值

为了验证算法的效果，和以下几种方法进行了对比：

- NCF
- Popularity（根据物品的流行度进行推荐）
- COM（基于概率的方法生成群组推荐）
- GREE（AGREE去除注意力机制）

AGREE和GREE的实验效果对比结果如下：



(a) Mafengwo — HR@10 (b) Mafengwo — NDCG@10 (c) CAMRa2011 — HR@10 (d) CAMRa2011 — NDCG@10

Figure 4: Performance of AGREE and GREE in each training iteration on Mafengwo and CAMRa2011 datasets (Section 3.2).

同时为了证明从数据集中学习聚合策略的有效性，和以下实验做了对比：

- NCF+avg
- NCF+ the least misery strategy
- NCF + The maximum satisfaction strategy
- NCF + The expertise strategy

AGREE算法的有效性实验对比结果如下：

Table 2: Top-K performance of both recommendation tasks for users and groups on Mafengwo (Section 3.3).

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**	—

Table 3: Top-K performance of both recommendation tasks for users and groups on CAMRa2011 (Section 3.3).

Overall 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
NCF	0.6119	0.4018	1.03e-06	0.5803	0.3896	9.02e-06	0.7894	0.4535	1.89e-07	0.7593	0.4448	3.92e-07
Popularity	0.4624	0.3104	9.15e-11	0.4324	0.2825	5.92e-11	0.6026	0.3560	5.99e-11	0.5793	0.3302	3.67e-11
COM	—	—	—	0.5793	0.3762	7.20e-08	—	—	—	0.7682	0.4368	1.46e-17
GREE	0.6163	0.4079	5.02e-05	0.5883	0.3871	2.11e-06	0.7841	0.4571	5.67e-07	0.7690	0.4479	5.43e-08
NCF+avg	—	—	—	0.5683	0.3819	2.97e-07	—	—	—	0.7641	0.4452	4.47e-07
NCF+lm	—	—	—	0.5593	0.3788	1.29e-07	—	—	—	0.7648	0.4455	4.94e-07
NCF+ms	—	—	—	0.5434	0.3710	2.75e-08	—	—	—	0.7607	0.4348	3.74e-08
NCF+exp	—	—	—	0.5648	0.3787	1.26e-07	—	—	—	2[illegible]	[illegible]	[illegible]
AGREE	**0.6223**	**0.4118**	—	**0.5883**	**0.3955**	—	**0.7967**	**0.4687**	—	**0.7807**	**0.4575**	—

同时也拿AGREE和其变体（AGREE-U、AGREE-G）做了实验对比：

- AGREE-U：AGREE算法中只利用了用户的embedding向量
- AGREE-G：AGREE算法中只利用了群组的embedding向量

AGREE算法和其变体的实验效果对比结果如下：

Table 4: Top-K performance of AGREE and its two simplified variants on the Mafengwo dataset (Section 3.4).

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	2.80e-07	0.4141	0.3322	2.99e-09	0.7309	0.5716	9.02e-06	0.5709	0.3832	3.39e-09
AGREE-G	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
AGREE	**0.6383**	**0.5502**	—	**0.4814**	**0.3747**	—	**0.7491**	**0.5775**	—	**0.6400**	**0.4244**	—

Table 5: Top-K performance of AGREE and its two simplified variants on the CAMRa2011 dataset (Section 3.4).

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	1.12e-07	0.5793	0.3832	4.47e-07	0.7601	0.4465	4.09e-08	0.7441	0.4376	6.36e-08
AGREE-G	0.6119	0.4018	1.03e-06	0.5803	0.3896	9.02e-06	0.7894	0.4535	1.89e-07	0.7593	0.4448	3.92e-07
AGREE	**0.6223**	**0.4118**	—	**0.5883**	**0.3955**	—	**0.7967**	**0.4687**	—	**0.7807**	**0.4575**	—

同时也拿AGREE和冷启动的中一些算法进行了结果对比，这里冷启动的前提是用户已经加入了某群组，但是没有产生行为，实验结果如下：

Table 6: Top-K performance of the new-user cold-start scenario on Mafengwo (Section 3.5).

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**	**0.5763**	

AGREE和冷启动的中一些算法的结果对比

综合来看，AGREE算法的效果是要比其他群组推荐算法好很多的。

4、代码解析

论文中也给出了代码的github 地址：https://github.com/LianHaiMiao/Attentive-Group-Recommendation

代码的环境是：

```
pytorch version: '0.3.0'
python version: '3.5'
```

由于目前pytorch已经更新到1.4版本了，所以对代码进行了更新，更新后的repo库为：https://github.com/Thinkgamer/Attentive-Group-Recommendation

运行代码，输出为：

```
num_group is: 290
num_users is: 602
num_items is: 7710
AGREE 的Embedding 维度为: 32, 迭代次数为: 30, NDCG、HR评估选择topK: 5
Iteration 0, loss is [1.0019 ]
Iteration 0, loss is [0.9933 ]
user and group training time is: [1201.2 s]
User Iteration 0 [1203.3 s]: HR = 0.2133, NDCG = 0.1389, [2.2 s]
Group Iteration 0 [1246.6 s]: HR = 0.1897, NDCG = 0.1229, [45.5 s]
Iteration 1, loss is [0.9112 ]
...
```