# Online Movie Group Recommendation

電機所 N26090693 吳承霖

# Outline

1 **Introduction**

Motivation, Problem Definition, Challenges, Related Works

2 **Dataset**

Dataset Introduction, Data Preprocessing

3 **Algorithm**

Network Model, Evaluation Metrics

4 **Experiments**

Effect of Attention, Parameter Test

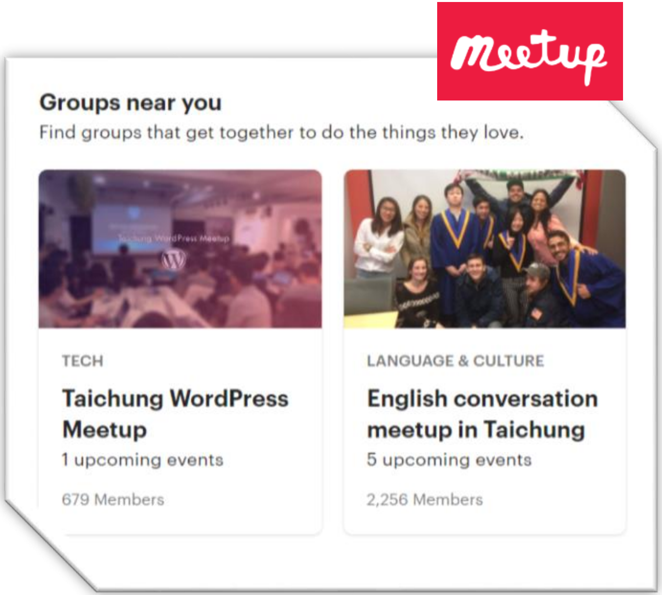
# 1. Introduction

- Motivation and Goal
- Challenge
- Problem Definition

# Motivation and Goal

## Motivation

- Owing to prevalence of social media, online group activities have become common.
    - Group of teenagers can organize a social party on *Meetup*
    - Group of researchers can discuss a paper on *Mendeley*




## Goal

- Build Algorithm to provide movie recommendation for a **group of users**

# Challenges

**The decision making process of a group is complicated and dynamic**

- A member has different influence in choosing items of different types due to his specialty.

- Traditional aggregation strategies are fixed and predefined --> insufficient the capture the dynamic process of group decision making
    - *<u>Average</u>* member aggregation
    - *<u>Maximum Satisfaction</u>* member aggregation
    - … etc

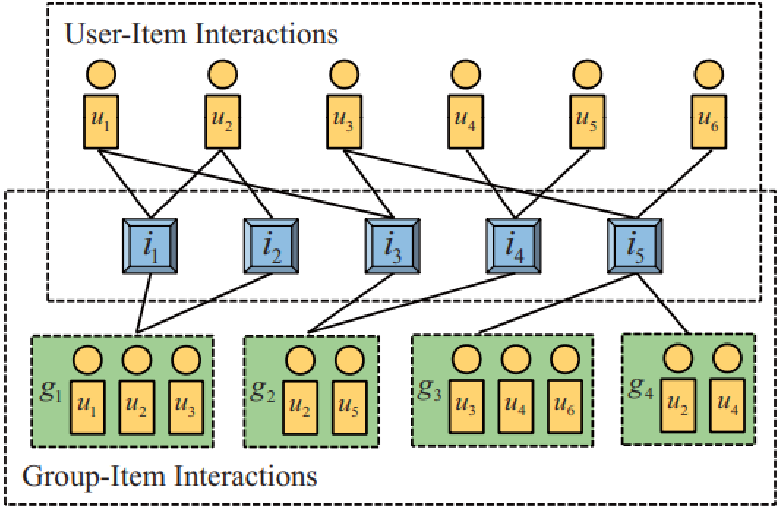
# Problem Definition

**Given**

- User set $\mathrm{U} = \{u_1, u_2, \cdots, u_n\}$, $n$ users in total
- Group set $\mathrm{G} = \{g_1, g_2, \cdots, g_s\}$, $s$ groups in total
- Item set $\mathrm{I} = \{i_1, i_2, \cdots, i_m\}$, $m$ items in total
- group-item interaction $\mathrm{Y} = [y_{lj}]_{s \times m}$
- user-item interaction $\mathrm{R} = [r_{lj}]_{n \times m}$

**Find**

- Item score from group $g$, $f_g: I \to \mathbb{R}$
- Item score from user $u$, $f_u: I \to \mathbb{R}$



# 2. Dataset

- Dataset Introduction
- Data Preprocessing

# Dataset Introduction

## CAMRa2011 Dataset

- a real-world dataset containing the movie rating records of individual users and households
- 602 users, 290 groups, 7710 items, 116344 user-item interactions, and 145068 group-item interactions

**Individual User Ratings**

| user_id | movie_id | value | Created_at |
|---|---|---|---|
| 40426 | 30682 | 80 | 2009-07-16 18:25:09 |
| 40426 | 8132 | 75 | 2009-07-16 18:21:26 |
| ⋮ | ⋮ | ⋮ | ⋮ |

**Group-User mapping**

| household_id | user_id |
|---|---|
| 1 | 40426 |
| 1 | 311738 |
| 2 | 693545 |
| 2 | 894682 |
| ⋮ | ⋮ |

**Group Ratings**

| household_id | movie_id | value | Created_at |
|---|---|---|---|
| 1 | 14816 | 85 | 2009-07-16 18:10:49 |
| 1 | 27935 | 80 | 2009-07-16 18:12:02 |
| ⋮ | ⋮ | ⋮ | ⋮ |

# Data Preprocessing

## Rating Values

Transform rating values of all rows to **1** => treat observed interactions as positive instances

## Negative Instances

Randomly sampled from missing data as negative instances (rating value = **0**) to pair with each positive instance

## Training/Testing Splitting

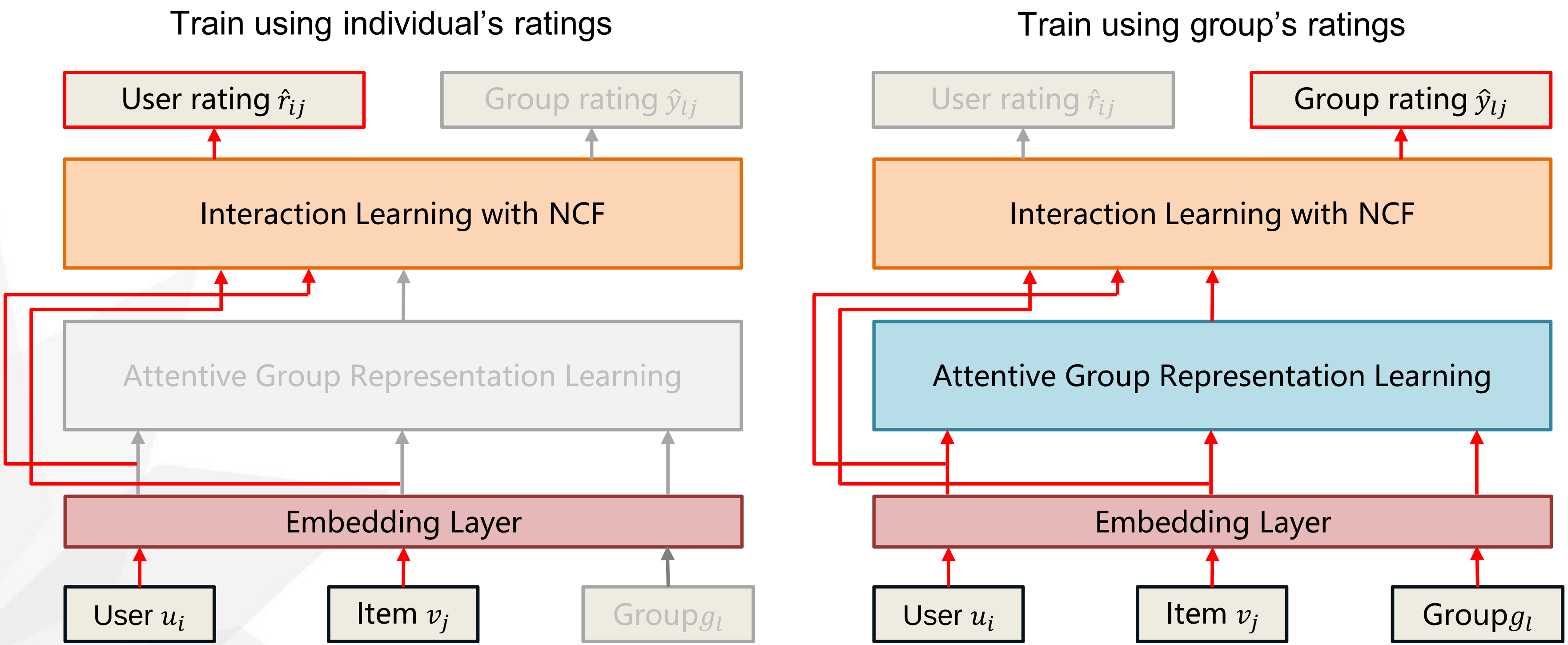
***Leave-one-out,*** i.e. for each user (group), randomly removed one of his(its) interactions for testing

# 3. Algorithm

- Model Architecture
- Evaluation Metrics

# Model Architecture

Implemented model: **AGREE** ( **A**ttentive **G**roup **RE**comm**E**ndation )



# Attentive Group Representation Learning

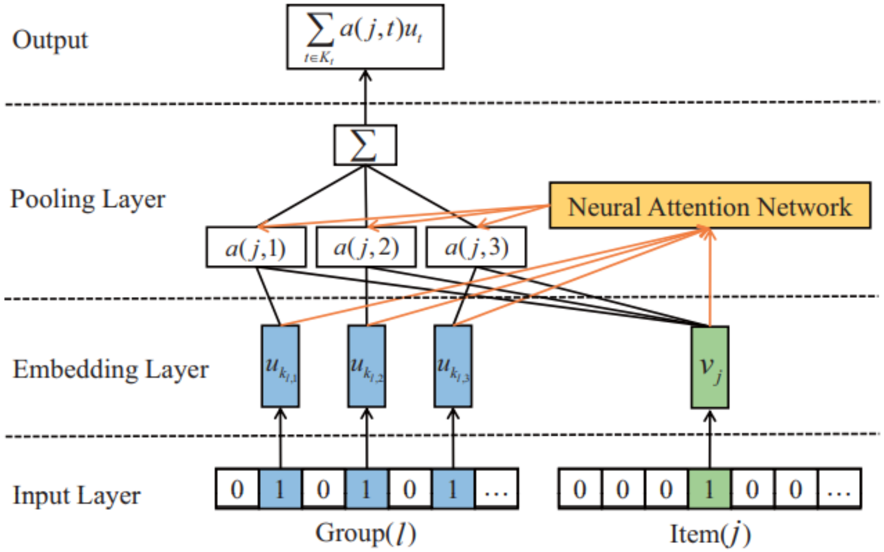
**User embedding Aggregation**

- group embedding is dependent of the embedding vectors of its member users and the target items:

$$g_l(j) = \underbrace{\sum_{t \in K_l} \alpha(j,t) u_t}_{\text{user embedding aggregation}} + \underbrace{q_l}_{\text{group preference embedding}} \qquad (1)$$

- $\alpha(j,t)$ is weight for user embedding aggregation, it's learned at attention network:

$$o(j,t) = h^T \, ReLU\left(P_v v_j + P_u u_t + b\right)$$
$$\alpha(j,t) = softmax(o(j,t))$$




$u_t$ : embedding for user $t$, $\alpha(j,t)$: aggregation weight for user $t$ with item $j$, $P_v$ & $P_u$: weight matrices of the attention network that convert item & user embedding to hidden layer

# Interaction Learning with NCF

**Pooling Layer**

performs element-wise product on their embeddings, then concatenates it with the original embeddings

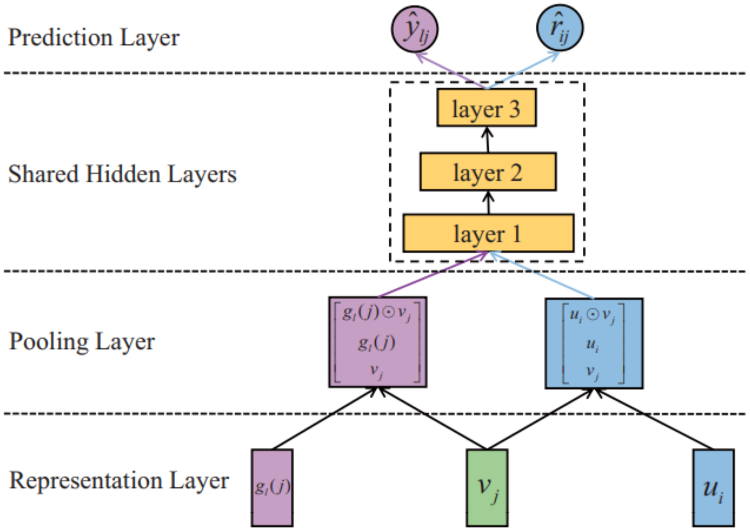
$$e_0 = \varphi_{pooling}(g_l(j), v_j) = \begin{bmatrix} g_l(j) \odot v_j \\ g_l(j) \\ v_j \end{bmatrix}$$

$$e_0 = \varphi_{pooling}(u_i(j), v_j) = \begin{bmatrix} u_i(j) \odot v_j \\ gu_i(j) \\ v_j \end{bmatrix}$$

**Shared Hidden Layers**

at layer $h$: $$e_{h+1} = ReLU(W_{h+1} e_h)$$

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



# Evaluation Metrics

**Row Format of Testset**

uid (or gid), positive iid, negative iid 1, negative iid 2, …, negative iid 100

**Evaluation Procedure**

1. **predict** rating scores for each user-item pair (or group-item pair), then **sort** it from high to low

2. **Top-K** : pick highest 5 predicted, calculate

- Hit Rate: percentage of predicted rating of positive item being in the top 5 ratings
- NDCG: wow high the predicted rating of positive item is

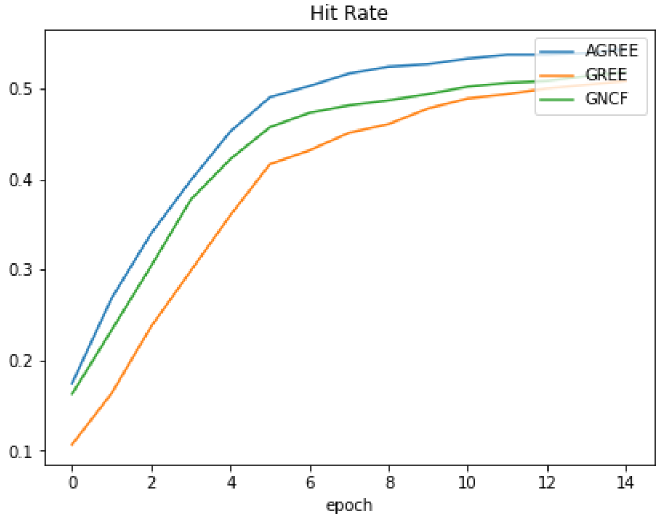
# 4. Experiments

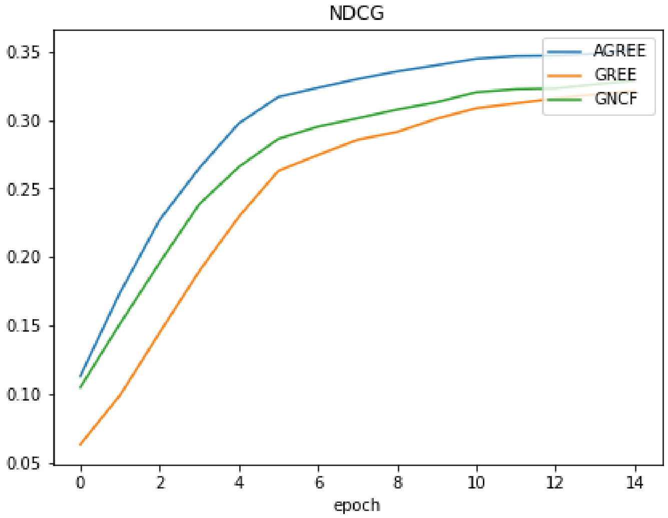
- Effectiveness of Attention
- Parameter Test

# Effectiveness of Attention

To examine the effectiveness of Attention mechanism, we compared our implemented model with:

- **GREE** – AGREE model without attention (average member aggregation)
- **Group NCF** – Take member aggregation away, treats group as a virtual individual

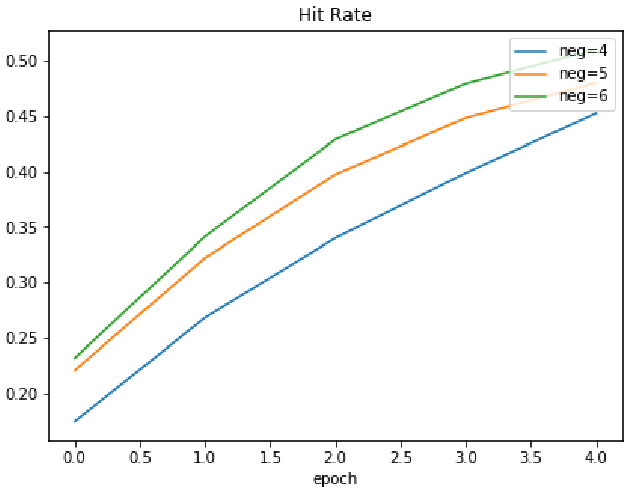


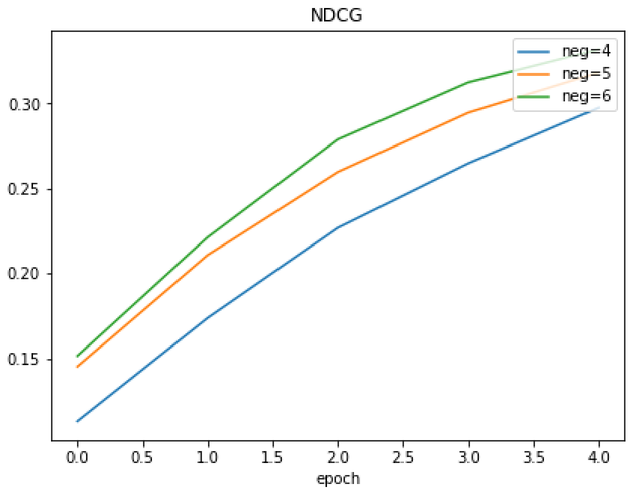



# Parameter Test

To investigate the number of sampled negative instances for each observed interaction:

- For each interaction between user(/group) and item, sample 4, 5, 6 negative instances



# 5. Conclusion

# Conclusion

## What we might be able to conclude

- Attention mechanism is helpful for aggregated group representation
- The number of negative instance sampling is relevant to accuracy of evaluation

## What we can further improve

- Values of Hit Rate@5 and NDCG@5 are not high in CARMa2011 dataset, try on other dataset
- Train model using un-transformed rating values (0-100), instead of [0, 1]
- Alter method of interaction between user & item in Attention network, use product-based ways instead

# Thank you

Any Question?