Attentive Score Aggregation

Group Recommender Systems

Horia Turcuman

Interactive Recommender Systems Seminar
Technical University Munich

January 25, 2023

Group Recommender Systems

Introduction

Group Recommender Systems are designed to tackle the complex task of recommending to groups of individuals which shows up in various social activities such as watching movies, city breaks, or shopping.

Some challenges:

- **Members' preferences** can vary and even oppose each other
- Factors such as **age**, **relationships**, **expertise**, **personality**, ..., affect the decision making process
- Different goals: mimic the **real decision making process** or create a **fair system**
- Difficult to **gather training data** since groups are often ephemeral and users only rate a small number of items
- Difficult to asses the quality of a system since there is **little to no feedback** (either at group level or member level) to recommendations

Group Recommender Systems

Classification

Preference Aggregation methods aim to model and combine user preference into a group preference which is then used to determine scores or ranks of items. The two models that have been chosen for this paper are the **AGREE** and **MoSAN**.

Score Aggregation aims at building on top of individual user score prediction systems by aggregating the individual scores. Various aggregation functions exist:

- **Minimum Misery** keeps the least satisfied group member as happy as possible
- **Average Satisfaction** maximizes the mean of the scores of group members
- **Maximum Satisfaction** maximizes the satisfaction of the most satisfied member
- **Attention weighing** maximizes the weighted mean of the scores of group members

Group Recommender Systems

Main idea

Currently attention modules are successfully used in preference aggregation but not used at all for group recommendation via score aggregation. Intuitively, these modules could unravel important information as relationships and expertise and use it to determine weights for score aggregation. The attention modules developed in the mentioned papers are ready with little or no change to be used in this sense.

$$s_g(i, G = \{u_1, ..., u_{|G|}\}) = b(G) + \sum_{u \in G} \alpha(u, i, G) \cdot s_u(u, i)$$

The 2 considered models learn embeddings for both users and items together with the attention model's parameters.

Group Recommender Systems

Attentive Group Aggregation (AGREE)

AGREE models expertise of users on items. Intuitively, if users are familiar with an item or with items similar to it, they should have (and do have in real life) a more decisive say on the item. To model this, the following 2-layer network is proposed:

$$
\begin{aligned}
o(u,i) &= \mathbf{h}^T \mathrm{ReLU}(\mathbf{P}_i \mathbf{i} + \mathbf{P}_u \mathbf{u} + \mathbf{b}) \\
\alpha(u,i,G) &= \mathrm{softmax}(o(u,i)) = \frac{\exp o(u,i)}{\sum_{u' \in G} \exp o(u',i)}
\end{aligned}
$$

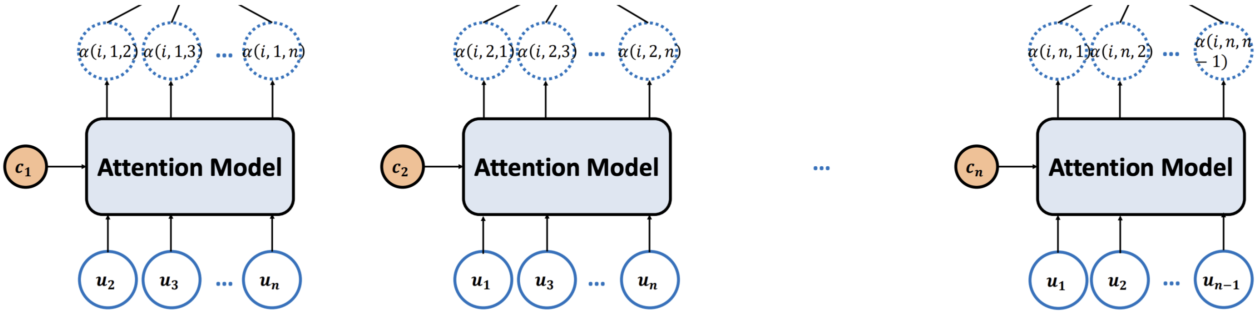
Group Recommender Systems

Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation (MoSAN)

This model goes one step further by modeling apart from the expertise, the influence of users on each other.

$$a(c, u) = \mathbf{w}^T \Phi(\mathbf{W}_c \mathbf{c} + \mathbf{W}_u \mathbf{u} + \mathbf{b}) + d$$

$$o(c, u, G) = \text{softmax}(a(c, u)) = \frac{\exp a(c, u)}{\sum_{\substack{u' \in G \\ u' \neq c}} \exp a(c, u')}$$



Group Recommender Systems

MoSAN Attention Module Adapted

$$a(c, u, i) = \mathbf{w}^T \Phi(\mathbf{W}_c \mathbf{c} + \mathbf{W}_u \mathbf{u} + \mathbf{W}_i \mathbf{i} + \mathbf{b}) + d$$

$$s_g(i, G) = \frac{1}{|G|} \sum_{\substack{c,u \in G \\ c \neq u}} o(c, u, i, G) \cdot s_u(u, i)$$

$$\alpha(u, i, G) = \frac{1}{|G|} \sum_{\substack{c \in G \\ c \neq u}} o(c, u, i, G)$$

Implementation

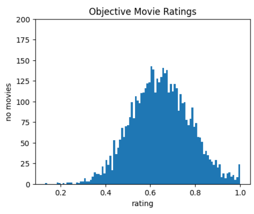
Synthetic Movie Dataset

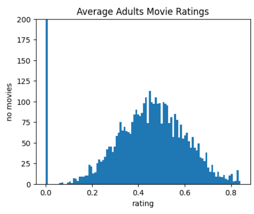
The generation process considers movie categories, different preferences of users for these categories, expertise, two different user types namely adults and children, three kinds of relationships between users, and three group types with different decision-making processes. Users are assigned to groups based on their relationships.

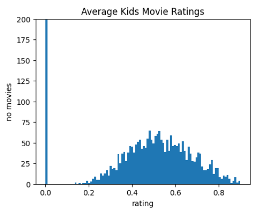
1. Close Friends (Friends, Families, Couples): this group type uses the maximum satisfaction decision approach
2. Acquaintances: they use a weighted average satisfaction approach with weights adjusted based on the expertise of users on the movie category and on the user type (children have more priority)
3. Strangers: by trying to be polite they make sure no one is too disappointed. They, therefore, use the least misery rating approach which assigns a movie the lowest of its ratings among the group members.

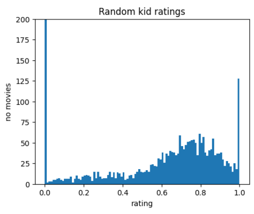
Implementation

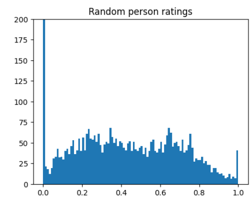
Synthetic Movie Dataset

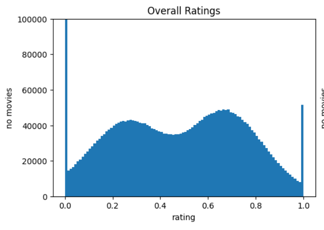


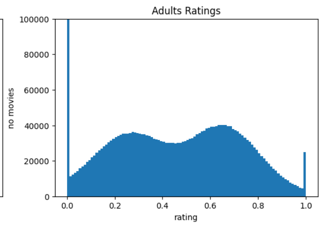


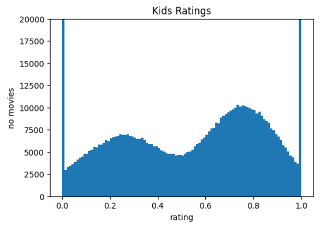






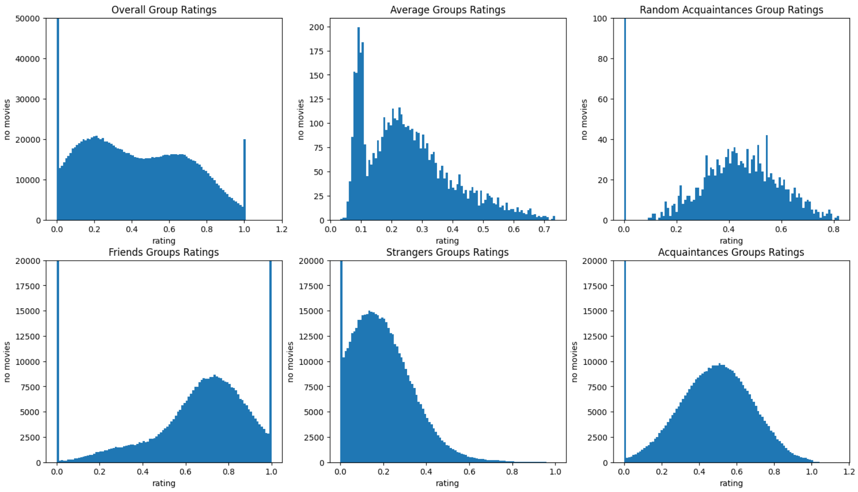






Implementation

Synthetic Movie Dataset



Implementation
Challenges

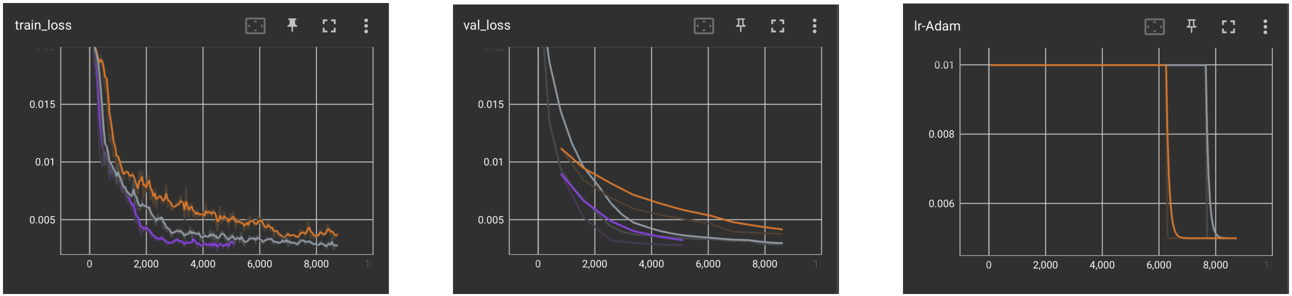
The bottleneck of the two models is the fact that batches can not be processes as whole but for-loops need to be used. In the MoSAN network, even two for-loops are required since the context embeddings of all group member paired with all other group member embeddings are needed. This results in a very slow batch processing time making the training practically impossible.

To alleviate this problem, the implementation moves part of the burden outside the batch processing function into a pre-processing step.

RESULTS

	MSE AGREE	MSE MoSAN	L1 AGREE	L1 MoSAN
16	0.00381	0.00650	0.061	0.080
32	0.00287	0.00587	0.053	0.076
64	0.00259	0.00656	0.050	0.080

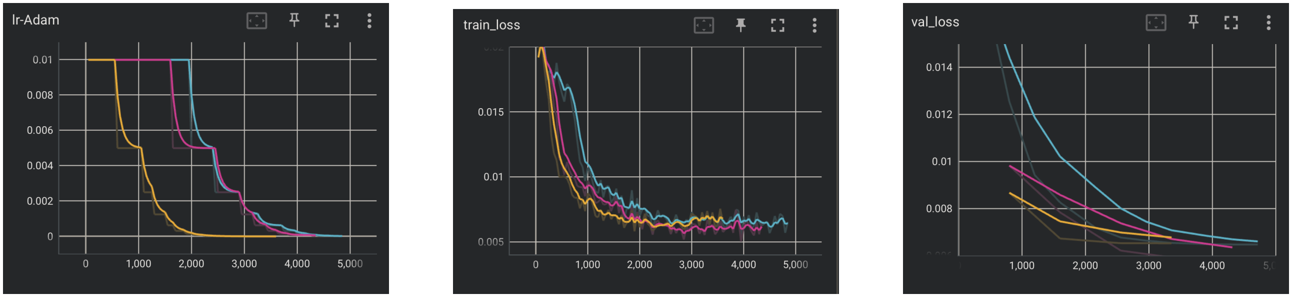
RESULTS



(a) learning rates **(b)** training losses **(c)** validation losses

Figure. Training of the AGREE-like model with various embedding dimensions.

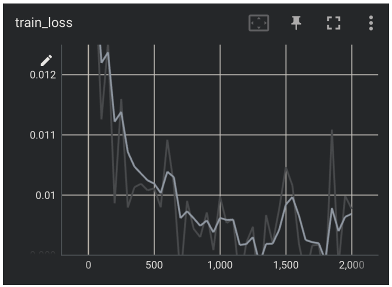
RESULTS



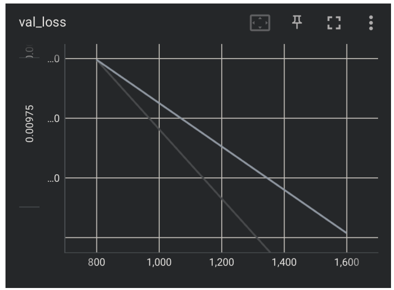
(a) learning rates **(b)** training losses **(c)** validation losses

Figure. Training of the MoSAN-like model with various embedding dimensions.

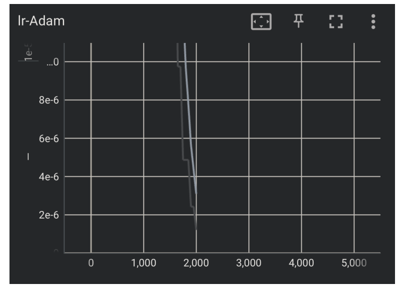
RESULTS



(a) learning rates



(b) training losses



(c) validation losses

Figure. Training of the AGREE-like model with encoder and the 32-dimensional embedding space.