MARS-GM: Multi headed attention Recommender Systems using Graphical Modelling

Ayushi Agarwal
Parth Shettiwar
Satvik Mashkaria
Shardul Parab

Introduction

- Recommendation systems - useful in the today's dynamic and competitive world. Used to ensure better sales, engagement and retention of customers.
- In the current approach, we aim to model data using different spaces - currently user and item.
- Challenges:
- 1) The model has to capture the information from there 2 graphs coherently and in a weighted fashion to output the correct recommendation.
- 2) Users tend to interact in a varied and diverse manner, which needs to be differentiated well in the latent space by the network model.

Problem Statement

- *Let* $U = (u_1, u_2, ..., u_n)$ *and* $V = (v_1, v_2, ..., v_m)$ *be the sets of users and items respectively, where n is the number of users, and m is the number of items. We assume that n×m is the user item rating matrix, which is also called the user-item graph.* r_{ij} *is the rating score, to represent the rating from user* u_i *to item* v_j*. The user user social graph T is modelled such that* T_{ij} *= 1 if user* u_j *is related to user* u_i*. be the set of known ratings. . Given the user-item graph R and social graph T, we aim to predict the missing rating value in R.*

Literature Review

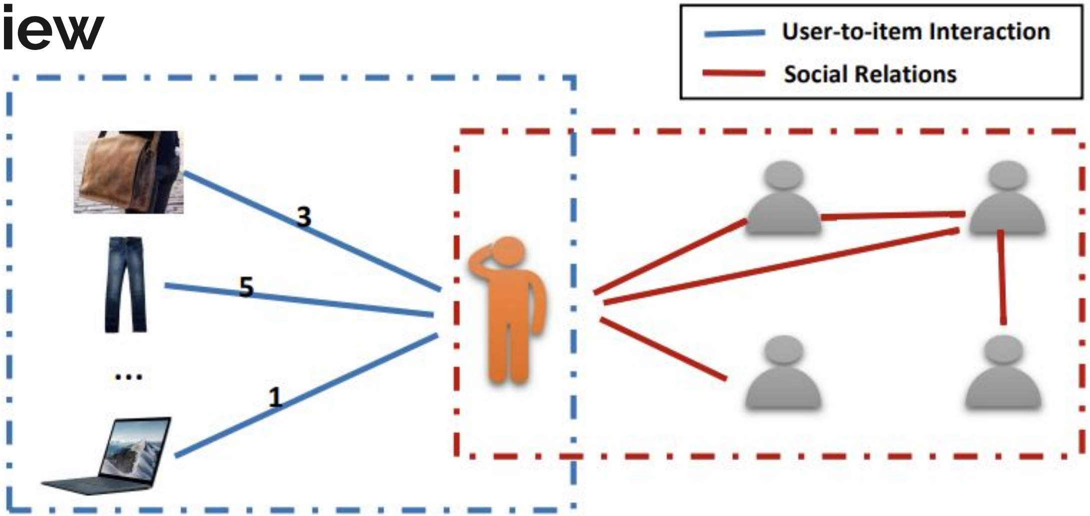


Figure 1: Graph Data in Social Recommendation. It contains two graphs including the user-item graph (left part) and the user-user social graph (right part). Note that the number on the edges of the user-item graph denotes the opinions (or rating score) of users on the items via the interactions.

Literature Review

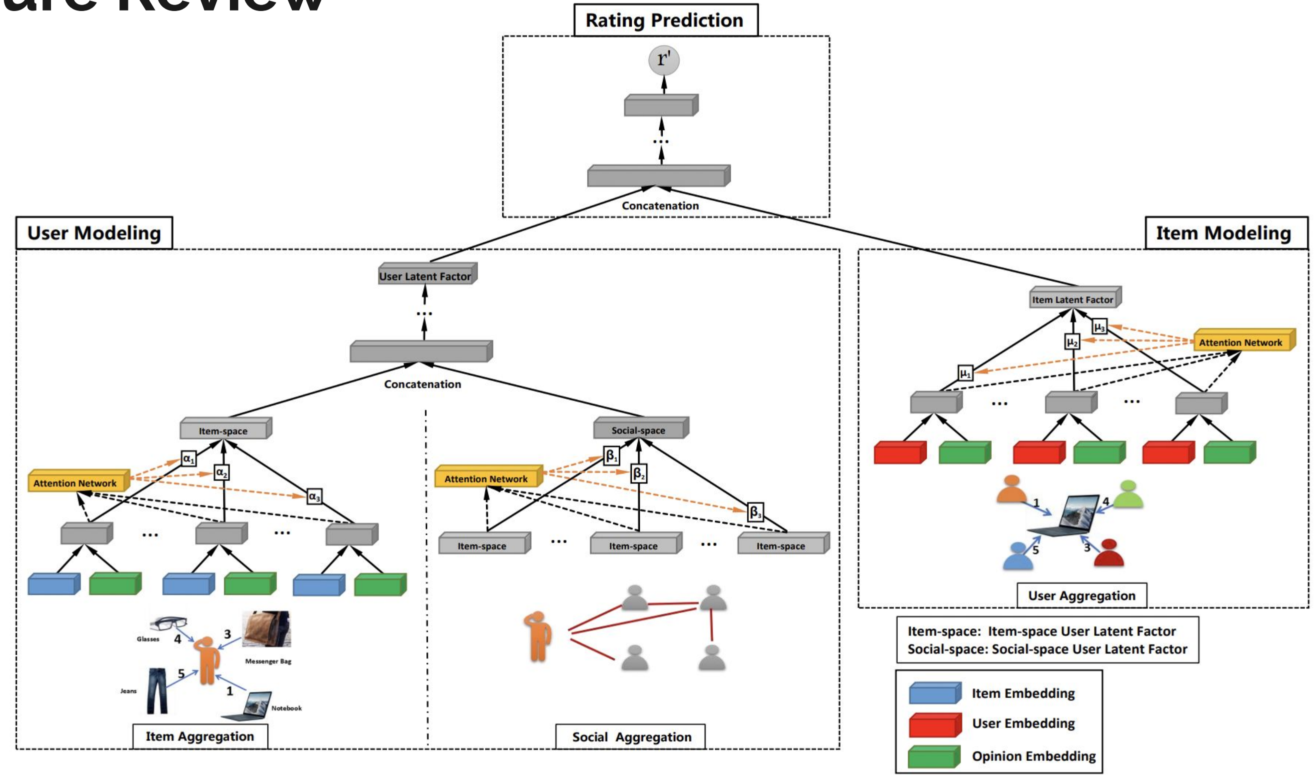


Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

Problems tackled

- Implementing(existing buggy code base) and training the end to end baseline model
- Adding different attention technique to the model
- Expanding the problem to new spaces
- Using different loss functions.
- Testing adversarial techniques
- Explore performance on new dataset - FilmTrust

Dataset details

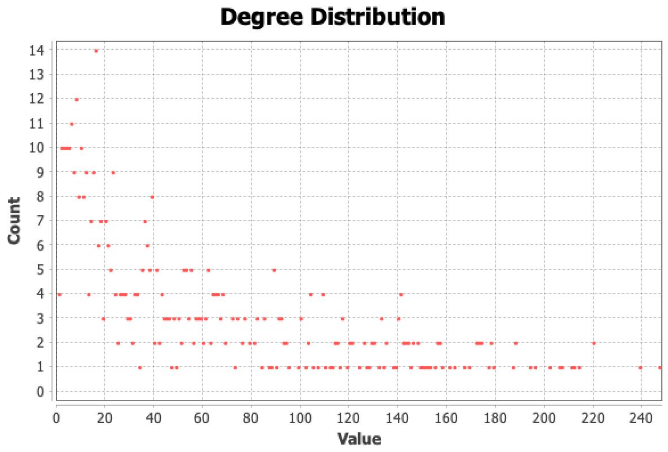
1. **Epinions**
    a. online social network of a general consumer review site Epinions.com
    b. 75,879 Nodes and 50,8837 Edges. node → user & edge → trust
2. **Ciao**
    a. five columns and they are userid, productid, categoryid, rating, helpfulness, respectively.
    b. trustnetwork.mat includes the trust relations between users. Two columns of userids.
    c. 7357 users and 106797 items
3. **FlimTrust**
    a. ratings.txt: 35497 item ratings w format: userId, movieId, movieRating
    b. trust.txt: 1853 directed trust ratings w format: trusterId, trusteeId

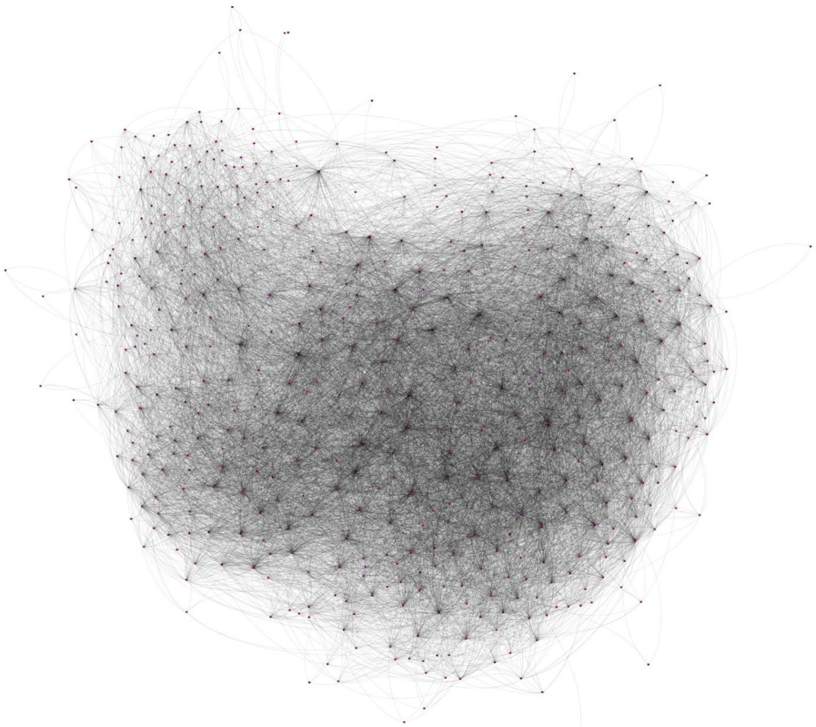
Ciao Dataset Visualization - USER USER MODELLING

Visualization for a subgraph over first 500 users
User-user interaction is highly connected

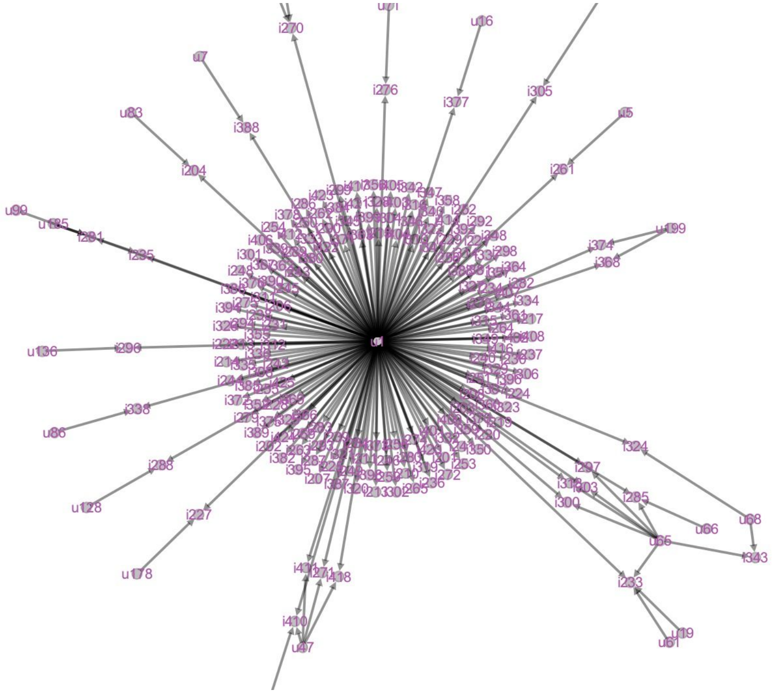
Results:

Average Degree: 28.820





Ciao Dataset Visualization - USER ITEM MODELLING

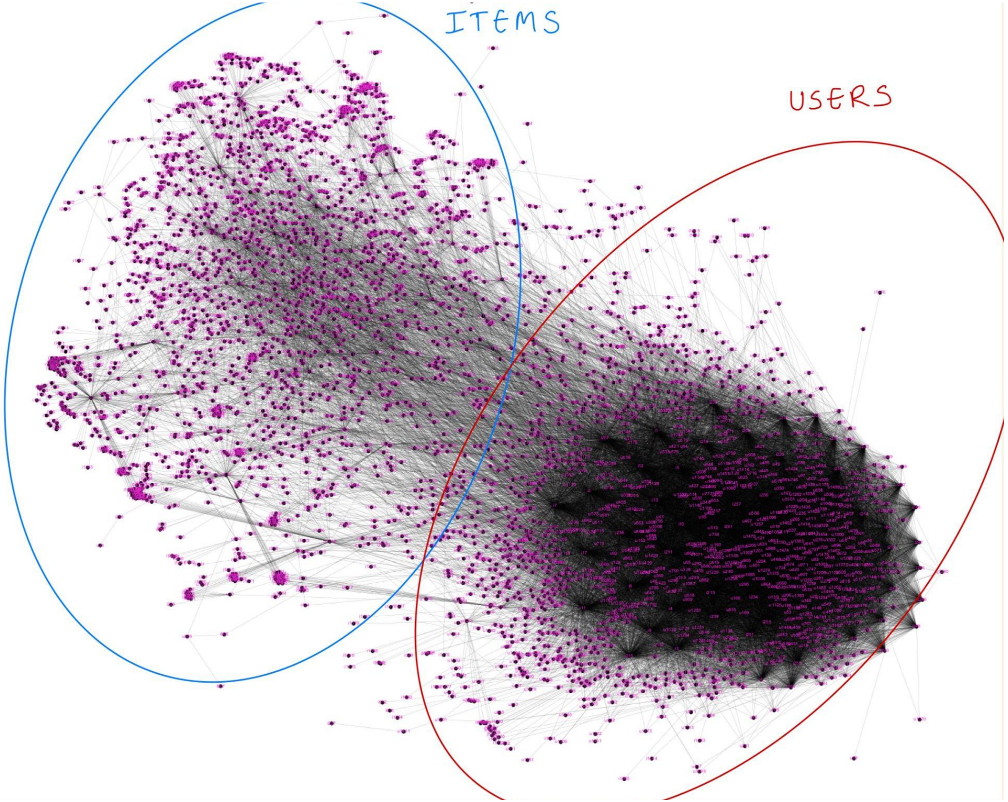


FilmTrust - User User Interaction

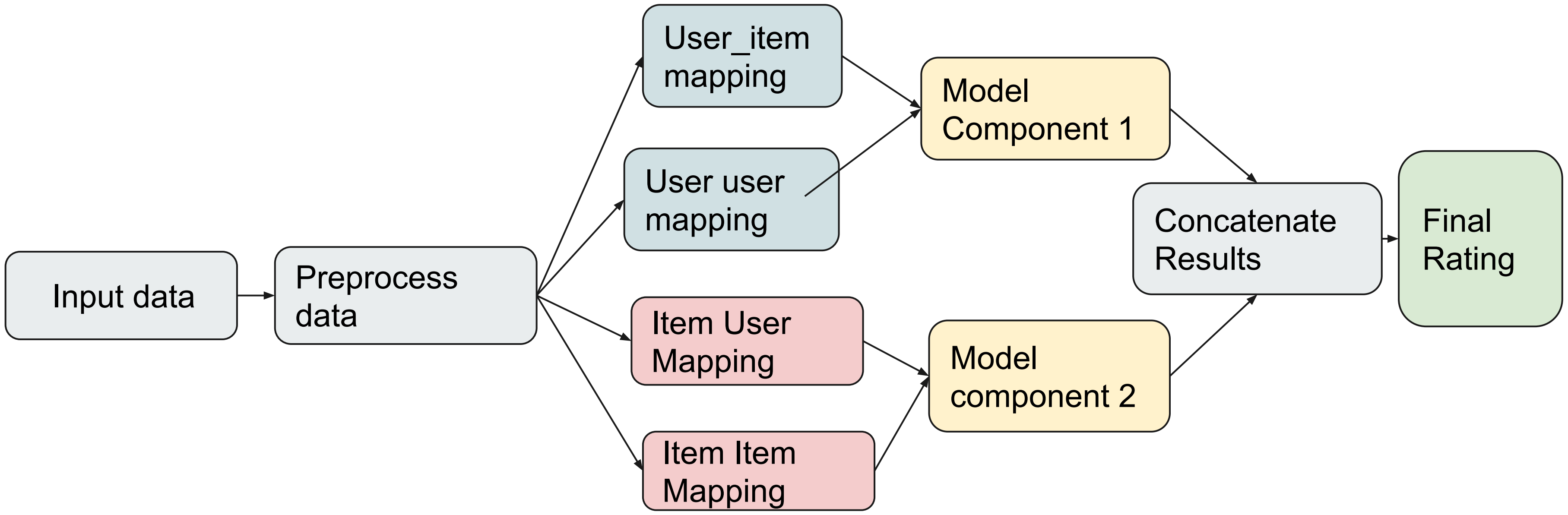
As we see that the above dataset is huge, we decided to use FilmTrust

User user modelling - largest connected component among 51 connected components. Total 705 nodes

FilmTrust - User Item Interaction (bipartite graph)



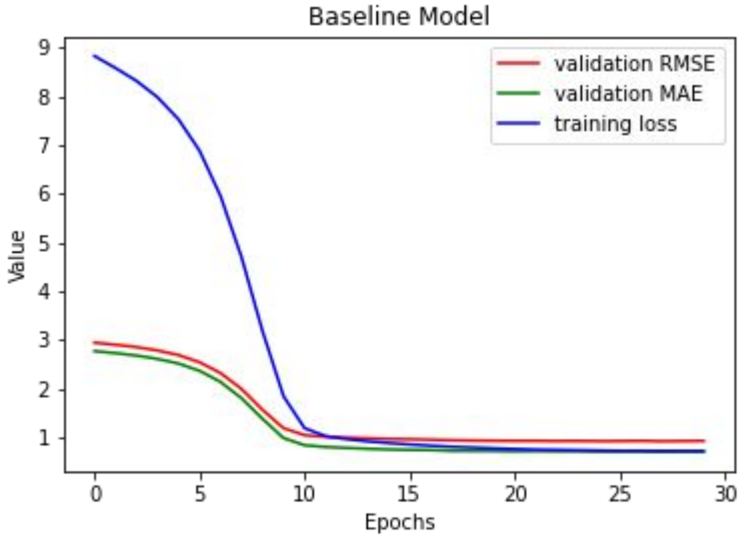
Model Pipeline



Metrics on Baseline Model and comparison to other datasets

Training	Metrics	Algorithms								
		PMF	SoRec	SoReg	SocialMF	TrustMF	NeuMF	DeepSoR	GCMC+SN	**GraphRec**
Ciao (60%)	MAE	0.952	0.8489	0.8987	0.8353	0.7681	0.8251	0.7813	0.7697	0.7540
	RMSE	1.1967	1.0738	1.0947	1.0592	1.0543	1.0824	1.0437	1.0221	1.0093
Ciao (80%)	MAE	0.9021	0.8410	0.8611	0.8270	0.7690	0.8062	0.7739	0.7526	0.7387
	RMSE	1.1238	1.0652	1.0848	1.0501	1.0479	1.0617	1.0316	0.9931	0.9794
Epinions (60%)	MAE	1.0211	0.9086	0.9412	0.8965	0.8550	0.9097	0.8520	0.8602	0.8441
	RMSE	1.2739	1.1563	1.1936	1.1410	1.1505	1.1645	1.1135	1.1004	1.0878
Epinions (80%)	MAE	0.9952	0.8961	0.9119	0.8837	0.8410	0.9072	0.8383	0.8590	0.8168
	RMSE	1.2128	1.1437	1.1703	1.1328	1.1395	1.1476	1.0972	1.0711	1.0631

Baseline Results on FilmTrust dataset

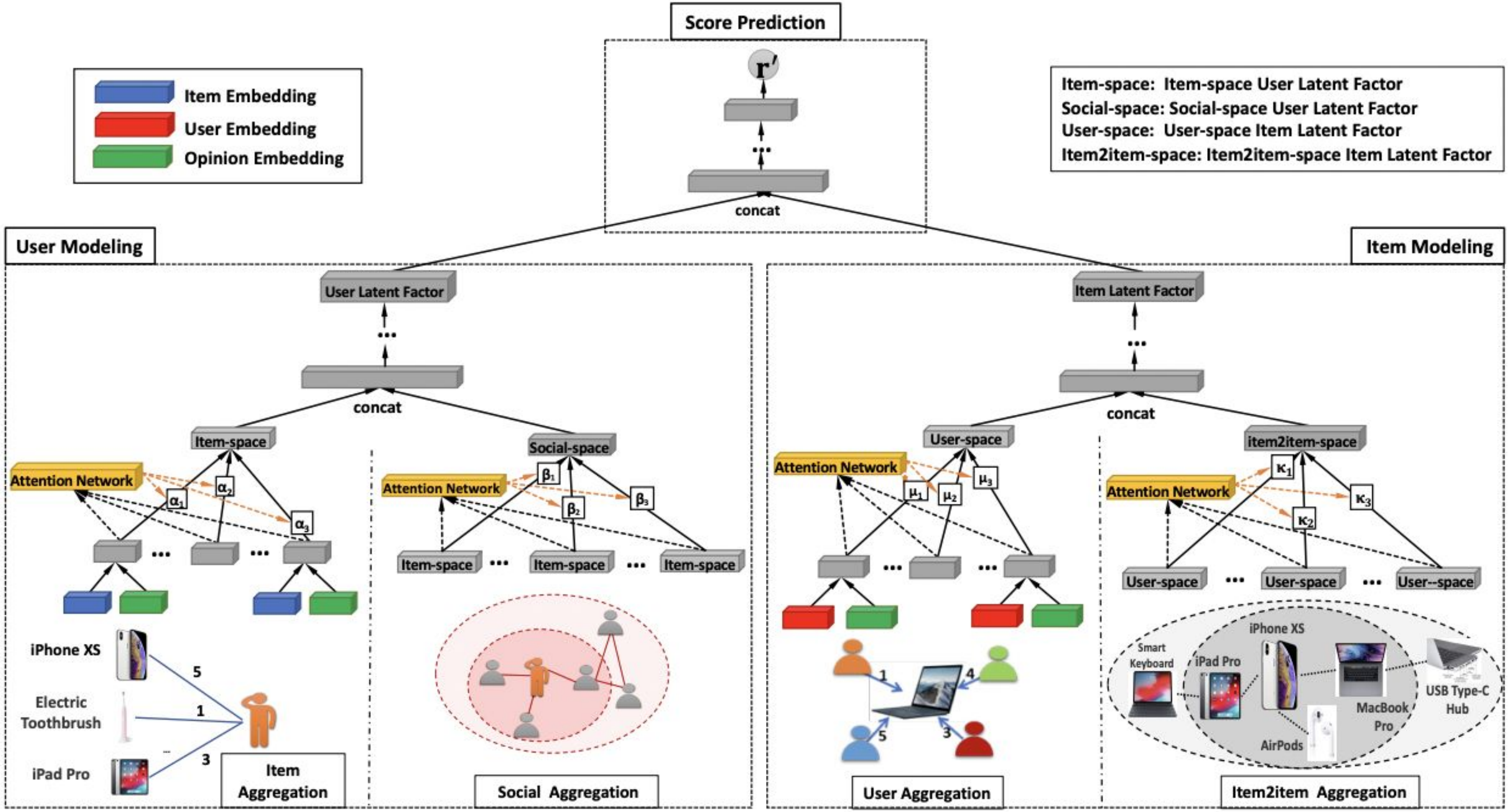


Test Results:

MAE: 0.7153

RMSE: 0.9217

Expanding the problem to more spaces



Item-item Modelling (Expanding to more spaces)

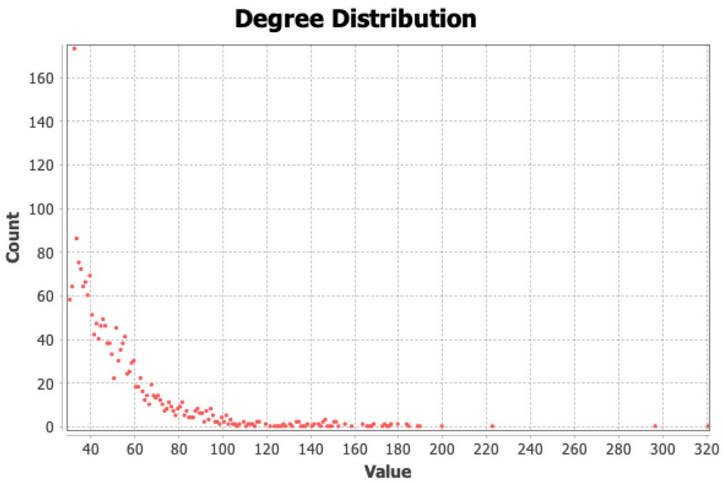
- Models discussed so far doesn't explicitly incorporate the similarity between items i.e. If a user likes scarf, with high chances it will like muffler as well.
- To capture similarity between two items i_1 and i_2, we form vectors $\mathbf{u_1}$ and $\mathbf{u_2}$ such that j^{th} element of $\mathbf{u_1}$ will be the rating given by j^{th} user to item i_1. If the user hasn't given a rating for i_1, the entry will be 0.
- Then, the similarity between items i_1 and i_2 will be the Pearson correlation coefficient between $\mathbf{u_1}$ and $\mathbf{u_2}$.
- Top 30 items were selected to be related to a particular item.

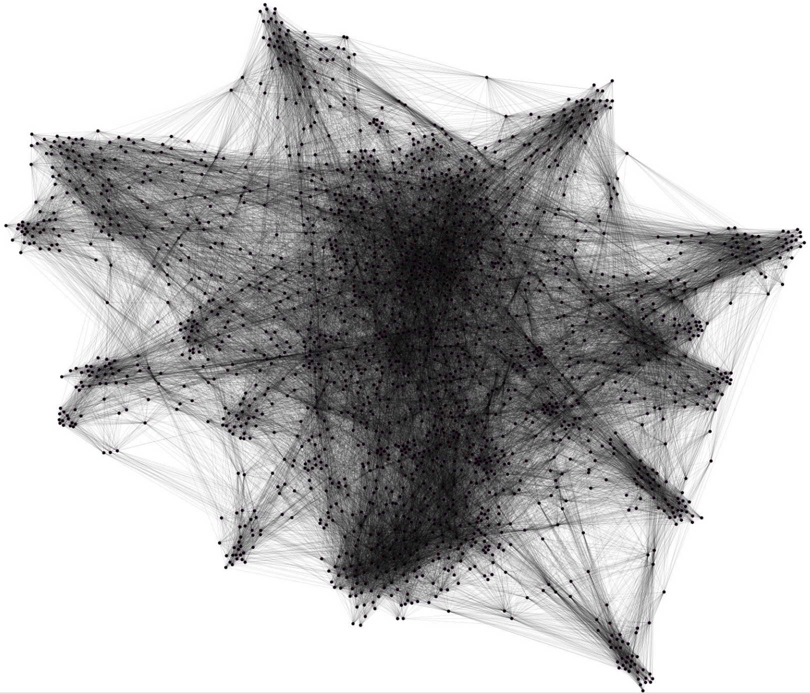
$$Sim(i_1, i_2) = \frac{\sum_j (u_1(j) - \bar{u_1})(u_2(j) - \bar{u_2})}{\sqrt{\sum_j (u_1(j) - \bar{u_1})^2 \sum_j (u_2(j) - \bar{u_2})^2}} \quad (1)$$

Item-item modeling visualizations for FilmTrust

2072 nodes and 54776 edges. Moderately connected but structured graph.

verage Degree: 52.873





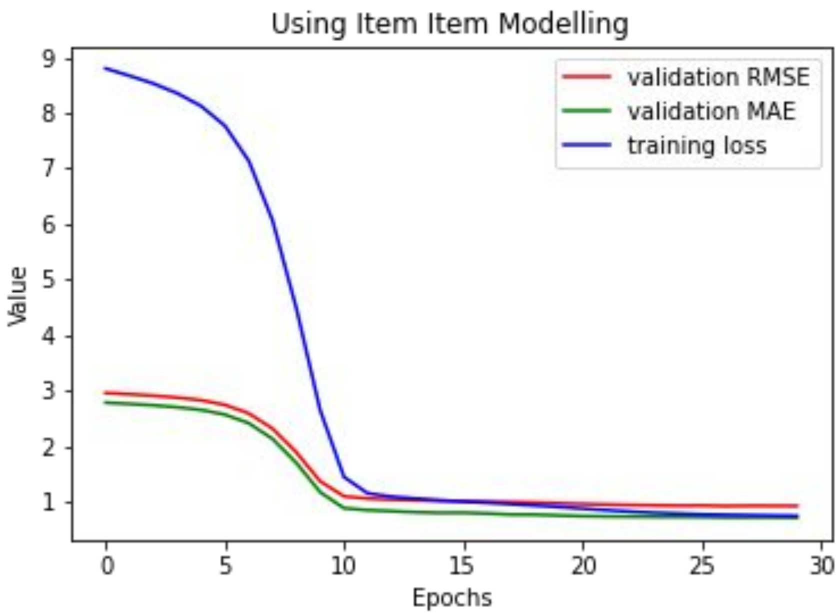
Item-Item Modelling

Likewise after performing Item Item modelling, we now have four spaces to utilize:

a) User - User Space
b) User - Item Space
c) Item - Item Space
d) Item - User Space

With more information being available for access, we hope that this can give much better performance than the original baseline.

Item-Item modelling Results



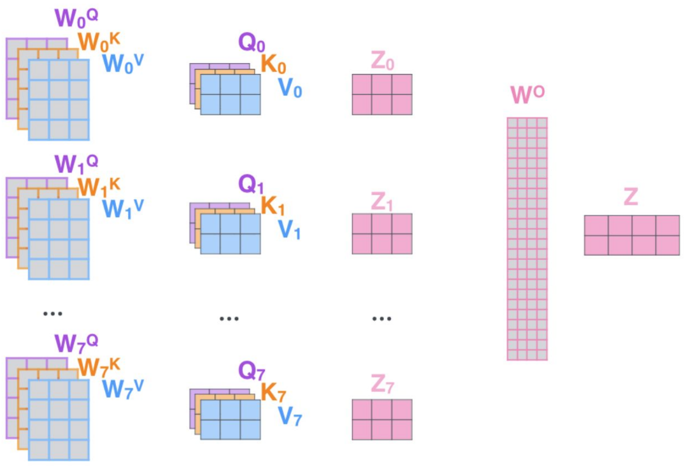
Test Results:

MAE: 0.7042

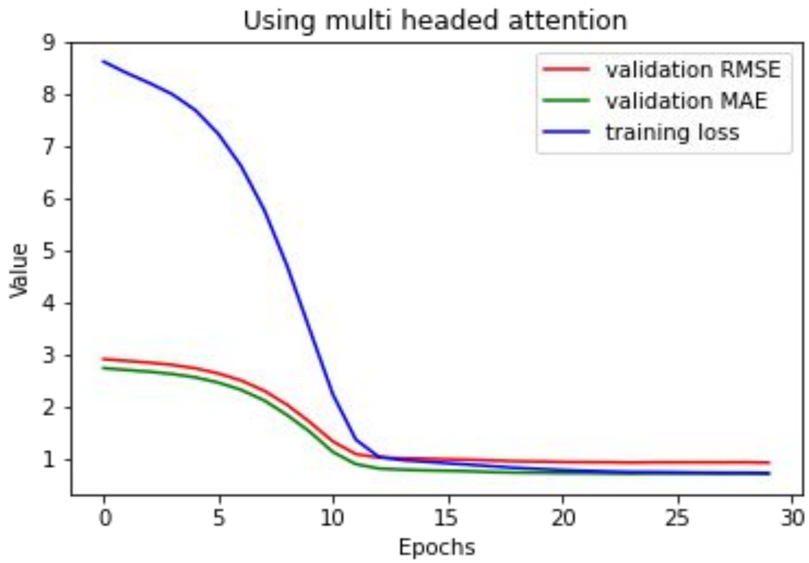
RMSE: 0.9063

Multi-headed attention

- We leveraged the state of art multihead attention method used in Transformers model
- Runs through a self attention mechanism several times in parallel (Computes Query, Key, Value matrices)
- The multi-headed attention model was used in all 4 sub-networks - user-user, user-item, item-user and item-item
- The intuition was to heterogeneously affect the various interactions in sub-networks depending on the graph structure
- Two methods to combine heads
    - Take mean of all the heads
    - Concatenate and pass through a FCN



Results of Multi Headed Attention

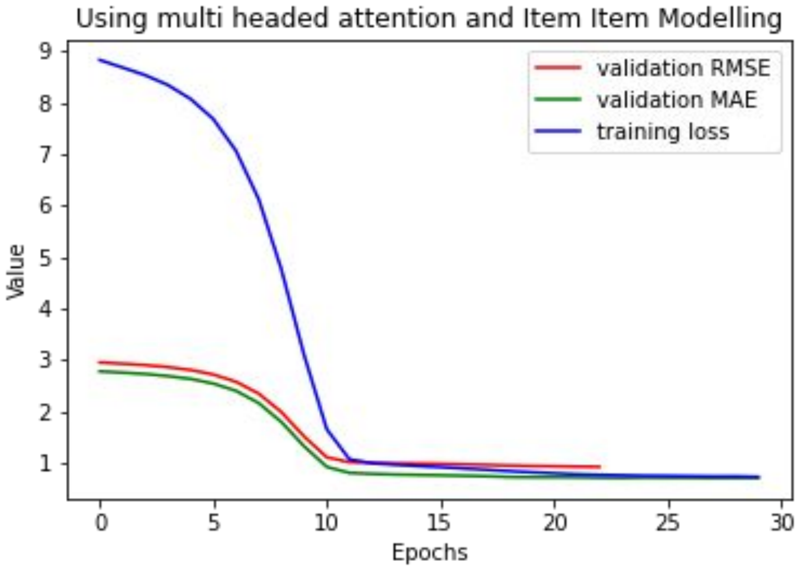


Test Results:

MAE: 0.7116

RMSE: 0.9185

Results of Multi Headed Attention + item-item modelling



Test Results:

MAE: 0.7023

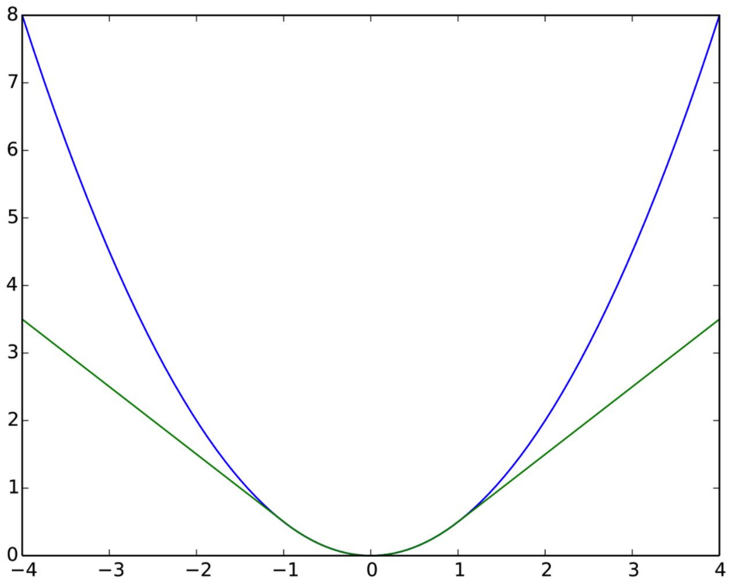
RMSE: 0.9045

Huber Loss

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

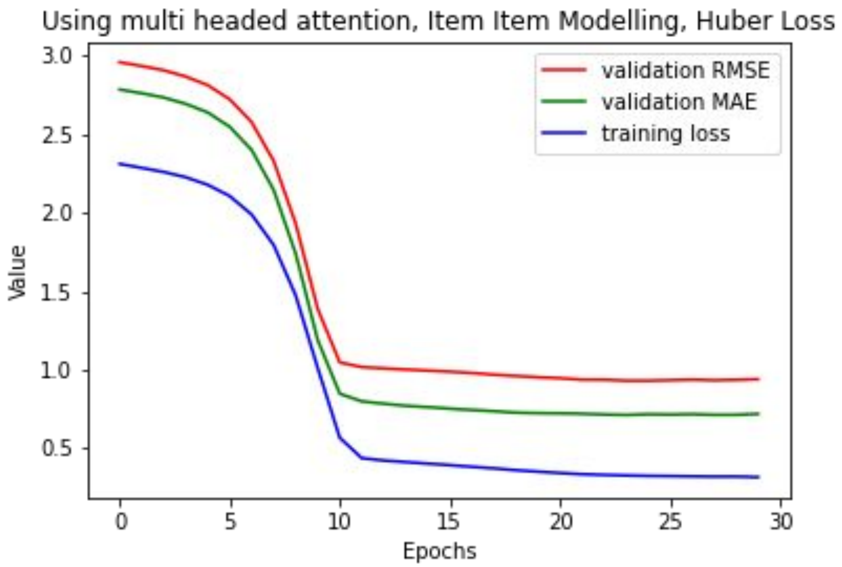
How is Huber Loss useful?

- Penalizes outlier points less severely.
- Helps the loss stabilize and not get impacted too much by outlier points
- Since we have quite a lot of outliers in our data, we saw performance improvement by using this loss



Blue line - usual MSE loss
Green line - Huber loss

Results of Huber Loss + Multi Headed Attention + Item to Item modelling



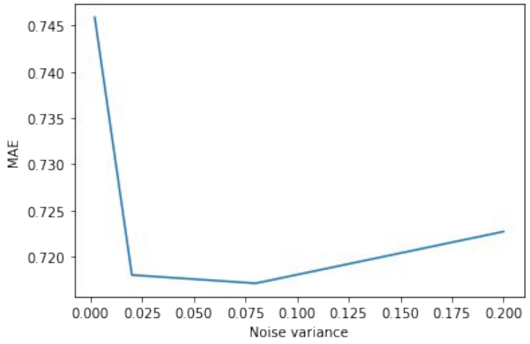
Test Results:

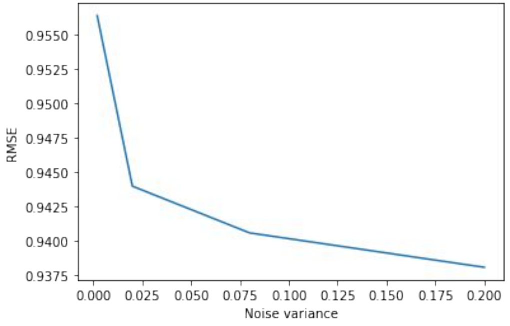
MAE: 0.7003

RMSE: 0.9001

Using Adversarial Techniques

- As part of the adversarial techniques, we tried to model the ratings given by user added with some noise
- Specifically, we added a gaussian random noise with zero mean and standard deviations ranging from 0.002 to 0.2
- The intuition behind this was to simulate the real world data samples where users usually don't put the ratings exactly and might put some other rating close to what they intended
- We analysed the MAE and RMSE scores (lower is better)
- Observed that higher noise might help as it acts as a regulariser too and avoids overfitting of the model.





Ablation Studies on FilmTrust Dataset

Model	MAE	RMSE
Baseline	0.7153	0.9217
Item-Item Modelling	0.7042	0.9063
Multi-headed attention	0.7116	0.9185
Multi-headed attention + item item modelling	0.7023	0.9045
Huber Loss + Multi-headed attention + item item modelling	**0.7003**	**0.9001**

Future Scope

1. Try different Adversarial Techniques.
2. Extend this technique to new datasets.
3. Extending it to even more spaces. (Example: Users, Items, Music, Movies).
4. Dynamic Recommendation Systems.

References

1. Fan, Wenqi, et al. "Graph neural networks for social recommendation." *The world wide web conference*. 2019.
2. Fan, Wenqi, et al. "A graph neural network framework for social recommendations." IEEE Transactions on Knowledge and Data Engineering (2020)
3. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017)

Questions?