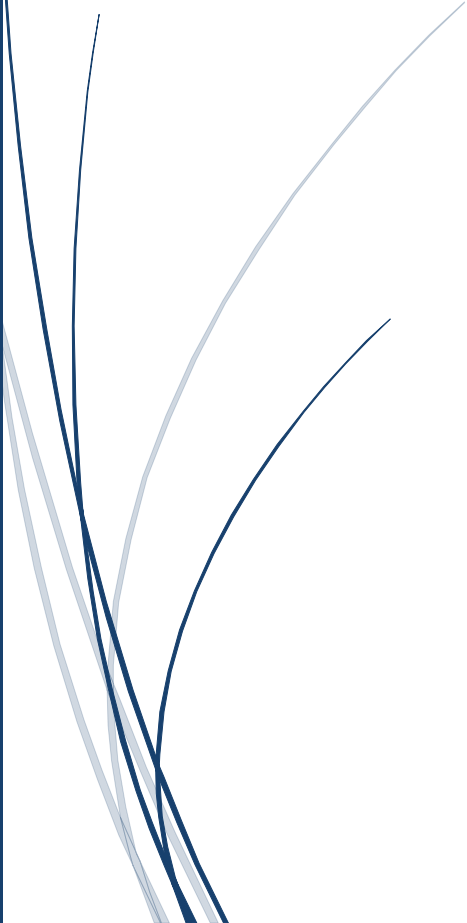




20th June 2021

Movie Night Group Recommendation



Shreyan Biswas, Francesca Drummer, Caroline
Freyer, & Stefan Petrescu

General information

Introductory description of the topic (200 words maximum)

With the rise of the internet, large amounts of information are presented to users, most of which is of low relevance and may become a burden for users. To avoid information overload, we use recommender systems to filter the information presented to users. One branch of recommender systems focuses on movies. Movie recommenders are an established type of recommender system with initiatives such as MovieLens dating back to 1997 (Ojeda-Zapata, 1997). The surge in online streaming companies such as Netflix has only fuelled the interest in such systems. However, despite their abundance, recommender systems have traditionally only concentrated on single users. Yet, the increase in social activities has shown that group recommendation is an equally important problem to address (Dara et al., 2019).

In this report, we hope to contribute to the active research field of group recommendation and help address the unmet public interest in such systems. To increase the quality of the recommended content, the report focuses on group dynamics. More specifically, the activity levels of users and group interdependencies. We hypothesize that the core of a desirable recommendation system captures the implicit signals and dependencies between group members. The system presented primitively addresses this by assigning different weights to users and movies based on their activity level and popularity. The proposal addresses this further.

Motivation (300 words maximum)

Famous recommender systems such as those owned by Netflix and Amazon, primarily focused on serving content to individual users, while group recommendation remains quite distant (Dara et al., 2019). Berry et al. (2010) show that there is a public interest in group recommendation systems and that the need for these systems is prevalent due to the rise of social activities. Moreover, the current abundance of streaming companies amplifies this need as watching movies at home with friends or family has become the norm.

A group recommendation system recommends items to a group of users collectively, given their preferences (Dara et al., 2019). This group consists of a wide range of audiences of different ages and backgrounds and can consist of friends from the farthest distances or a family living under the same roof. Figure 1 shows a general framework of a group recommender. Dara et al. (2019) note that current group recommenders either merge the individual preferences into a single preference and then perform prediction, or perform single user prediction that are later aggregated into a single group recommendation. We focus on the latter. Moreover, current group recommenders focus on homogenous groups where all members have similar interests and activity levels. However, it is known that this assumption is unrealistic as people usually have different opinions and expectations. Thus, to highlight its importance we include *Group models* in the framework given in Figure 1.

More specifically, we focus on modelling group dynamics, which refers to the actions, processes and changes that occur within a group such as the relation between group members, their actions or feelings (Recio-Garcia et al., 2009). Modelling these aspects is quite challenging due to their complexity and dynamic properties. However, we hope that the ubiquity of platforms with such functionalities provides us with a unique opportunity to develop a social based recommendation system that can identify appropriate recommendations for a given group at any given time.

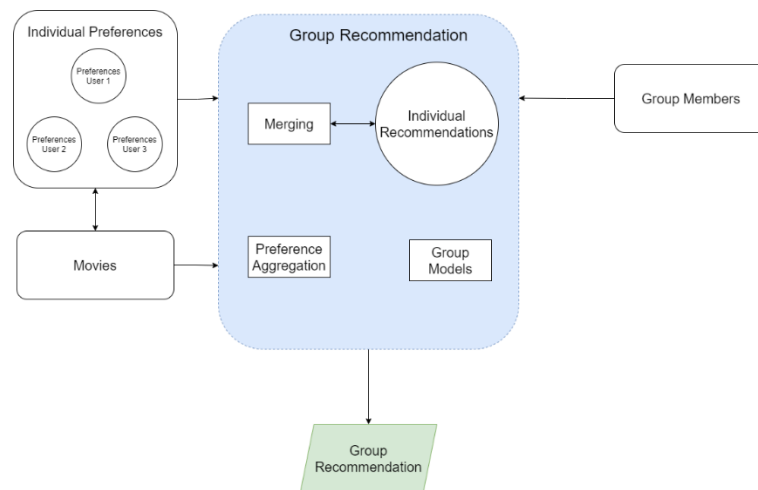


Figure 1: General framework of group recommendation systems based on the work of Dara et al (2019).

Link to CS4065 (300 words maximum)

This topic mainly relates to the recommendation lecture, especially the contextual awareness aspect in recommendation systems. Extracting group dynamics such as the relations between group members, their thoughts or feelings allows us to gain a better understanding of the decision process for choosing a movie. For example, some members of the group can have more influence on the decision. This must be captured by the system to maximise the collective satisfaction of the group. Thus, including contextual information with the user preferences is pivotal for obtaining quality group recommendations.

The system presented in this paper models the user's preferences by extracting features from a questionnaire that explicitly asks the focus group users to rate movies. We use an ordered rating to query the users, possibly resulting in different interpretations of the rating scale by the users. The system part of the report addresses this further. Moreover, as we cannot ask users to rate millions of movies, the data obtained is usually sparse, a common issue in explicit data acquisition. Therefore, we use collaborative filtering to predict the missing user ratings by using the rating from other users. Matrix factorisation (MF) is one of the most influential methods of collaborative filtering which decomposes a rating matrix into latent factors that are used to predict user ratings for unseen movies. In the latent spaces, users and their preferred items are brought together via user and item vectors. Using a dot product, the proximity between movies and users can be quantified and used for prediction. However, MF is known to be an NP-hard problem and is thus usually modelled as an optimisation problem that only approximates the latent factors from the observed ratings. Additionally, MF has considerable limitations such as quality bias towards users and items with many ratings leaving performance for long-tail items and less active users poor. Methods to address this limitation are discussed in the system part of this report.

Bibliography

- Banjac, J., Altinigne, C., Kypraiou, S., & Sioulas, P. (2020). *Movinder: A Movie Recommendation System for Groups*. Lausanne: EPFL.
- Berry, S., Fazzio, S., Zhou, Y., Scott, B., & Francisco-Revilla, L. (2010). Netflix recommendations for groups. *Proceedings Of The American Society For Information Science And Technology*, 47(1), 1-3. <https://doi.org/10.1002/meet.14504701402>
- Chen, H., & Chen, P. (2019). Differentiating Regularization Weights -- A Simple Mechanism to Alleviate Cold Start in Recommender Systems. *ACM Transactions On Knowledge Discovery From Data*, 13(1), 1-22. <https://doi.org/10.1145/3285954>
- Dara, S., Chowdary, C., & Kumar, C. (2019). A survey on group recommender systems. *Journal Of Intelligent Information Systems*, 54(2), 271-295. <https://doi.org/10.1007/s10844-018-0542-3>

- Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). Evaluating Group Recommender Systems. *Springerbriefs In Electrical And Computer Engineering*, 59-71. https://doi.org/10.1007/978-3-319-75067-5_3
- Ojeda-Zapata, Julio (1997). *New Site Personalizes Movie Reviews*. St. Paul Pioneer Press. p. 3E.
- Recio-Garcia, J., Jimenez-Diaz, G., Sanchez-Ruiz, A., & Diaz-Agudo, B. (2009). Personality aware recommendations to groups. *Proceedings Of The Third ACM Conference On Recommender Systems - Recsys '09*. <https://doi.org/10.1145/1639714.1639779>

Movie Night Group Recommendation

Link to video and repository

Youtube hidden link: <https://www.youtube.com/watch?v=twg5SDrTw3U>

We apologise for the length of the video, however, please note that it was originally almost 8 minutes long. We tried our best to shorten it, but it is still 6:26 minutes. We hope this is still acceptable as we believe that shortening it more would hinder its quality.

GitHub repo: <https://github.com/GroupMovieRec>

System documentation

1. Problem description (0.5 page)

While there has been much research conducted on the recommendation for single users there are few methods exploring recommendation to a group of users, despite the clear public interest (Berry et al, 2010). For group recommendation, it is essential to maximize the collective satisfaction of a group of users. How to integrate the different user preferences and find a movie that satisfies possibly conflicting preferences is still an active field of research. Most group recommendation algorithms process the preferences of each user and potentially those of other similar users, to predict whether a user might like a movie. These individual predictions are then aggregated into a single group recommendation. Given the framework in Figure 1, the focus of the system will be on preference aggregation, with a prototype incorporation of a group model where we estimate activity levels of users based on the ratings they provide. These activity levels attempt to capture imbalance in user preferences i.e. more active users are more likely to be disappointed if they are not satisfied than less active users.

The challenges that need to be overcome are the sparseness of user preference data, quality bias towards users and items with many ratings, and incorporating group dynamics in the system to improve the quality of predictions. The first challenge is addressed using the most influential method of collaborative filtering, matrix factorisation (MF). The method chosen to reduce the quality bias is based on the differentiating regularisation weights described by Chen & Chen (2019). The advantage of this method is that it does not require additional information about the users and can directly be obtained from the user ratings. Lastly, as group dynamics is very complex to model, we implement only a basic version that estimates the activity levels of users in the focus group based on the ratings they provided. We use these activity levels to assign different importance levels to users in the minimization. To validate our approach we use the publicly available datasets from MovieLens.

2. Related work / products (0.75 page)

This system is inspired by Movinder, a multi-user recommender system created by Banjac et al (2020). In this system, the focus group is given a questionnaire where they are asked to rate fifteen movies on a scale from 1 to 5. Although we are confident that this data accurately represents the user's preferences, only fifteen of the available movies are rated as asking users to rate more movies renders the system impractical. This resulting in very sparse data. Thus, collaborative filtering is used to include the preferences of other users (different from those in the focus group) to aid in the recommendation. Movinder uses the most influential method of collaborative filtering, called *non-negative matrix factorisation* (NMF), a more specific ML algorithm that is tailored for non-negative values such as ratings. To implement this, an older MovieLens dataset is used in addition to the ratings provided from the focus group. NMF predicts each user's rating of each item by building a rating matrix of all known user-item ratings and decomposing it into two low-rank matrices. These low-rank matrices represent the latent factors describing the users and the items, respectively, which are used to predict unknown ratings. However, as NMF does not completely solve the sparsity issue and is an NP-hard problem, it is usually modelled as an optimisation problem that approximates the latent factors from the observed ratings using strategies based on stochastic gradient descent. Banjac et al. (2020) then aggregate the per user preference predictions to produce recommendations that collectively satisfy the group. The aggregation strategy used is *aggregated voting* which determines the items that increase the overall individual satisfaction score of all the users in the group.

Despite its fame, Chen & Chen (2019) argue that NMF performs poorly for long-tail items and less active users because the system has limited clues on these items and users. In literature, this is called the cold-start problem. Moreover, when more training data is collected, we have more information for certain items and users, but not necessarily others. This result is contradictory to most other machine learning tasks where we obtain more "clues" about the relationship between the target variables and the features when more training data is collected. Thus, the authors argue that assigning the same regularisation weight to all users and items in the minimisation object function as done in NMF is too naive. To correct this imbalance, Chen & Chen (2019) propose a simple technique of assigning lower constraints to latent factors associated with popular items and active and higher constraints to latent factors associated with long-tail items and inactive users. The lower constraint on popular items and active users allows us to leverage the additional information these users and items have, as it is less restricted by the minimization. This allows us to find better latent factors and improve prediction accuracy. Conversely, as there is minimal information for long-tail users and items, the latent variances are normally noisy. With stronger constraints, they are less influenced by few extremes and thus, the

system makes more conservative predictions on these long-tail items and users resulting in better predictions as well. Therefore, the authors argue that this simple technique can partially alleviate the cold-start problem and improve the overall prediction performance.

3. Main functionalities, challenges and contributions (1.5 page)

As mentioned previously, our system is inspired by Movinder system. Although only our interface was built upon Movinder’s, in this report we will only focus on five notable improvements. Firstly, during the recommendation lecture, it was mentioned that ordered rating scales could result in different interpretations of the rating scale. Thus, we introduced a new rating scale, [Didn’t watch, worst movie, dislike, neutral, good, best movie] and internally mapped it to a zero to five scale. Note that “Didn’t watch” is mapped to the value of zero as it is regarded as an unknown rating that still needs to be estimated. This change in rating scale is motivated by the infamous qualitative-quantitative debate. Trochim (2020) argues that phrases are less ambiguous than numbers as all numerical information involves numerous judgments about what the number means. Moreover, interpreting phrases requires less effort from the user, simulating them to participate more actively. Thus, this rating scale allow us to better capture the user preferences with the qualitative scale.

Secondly, we ran the algorithms on a new MovieLens dataset from 2018 as opposed to the 1998 dataset used in Movinder. Although of similar size, the dataset consisted of newer movies. We hoped that the newer dataset would allow us to better capture today’s user’s preferences. This switch was important as if a user selects “Didn’t watch” the system regards the rating as one that still needs to be estimated. Thus, with unknown movies, we would have even less information about these users. However, an initially expected simple switch resulted in a great deal of work as new movie posters had to be found and the datasets had different formats. Moreover, when testing the interface, we were disappointed to discover that we still did not recognise many of the movies. To solve this issue, we computed a trending factor for each movie based on the top-rated IMDb movies. We used this trending factor to extract more well-known movies, thus, increasing the chances that the focus group users know the movie.

The third improvement is based on one of the findings of Dara et al. (2019) which states the importance of explaining the group recommendations so that the users can build confidence and trust in them. Moreover, they state that recommender systems that give multiple options to the users are preferred. Thus, the system presents the focus group with the top ten movies ranked from one to ten and shows the average score of the individual users for each item.

The last two improvements are based on differentiating regularisation weights proposed by Chen & Chen (2019). NMF models ratings in a large matrix and attempts to decompose this matrix into the product of two low-ranked matrices interpreted as the latent factors of the users and the latent factors of the items. These factors are determined based on two objectives; to best approximate the known rating scores and to maintain a low Frobenius norm for the low-ranked matrices to prevent overfitting. As there is a trade-off between these two objectives, the relative importance is usually determined using a hyperparameter referred to as a regularization weight. NMF assigns a regularization for the user latent factors and a separate one for the item latent factors. However, the training data used for the rating prediction task usually follow a long-tail distribution, with the majority of items and users contributing to the long-tail of the distribution. Thus, the authors argue that assigning the same regularization weight to all users and items is too naïve and suggest assigning lower constraints to latent factors associated with items or users that reveal more information and higher constraints to long-tail items and less active users. In both cases, the rating predictions are better as explained in Section 2. We verify the imbalance claim by investigating the MovieLens dataset used in the system. Table 1 shows a set of summary statistics for this dataset. We see that the bottom 10% of items only have 1 rating, however, the top 10% has 27-329. This is indicative of a long-tail distribution.

	Min	Max	Mean	Standard Deviation	Range Bottom 10%	Range top 10%
User	20	2698	165.31	269.26	20-24	403-2698
Item	1	329	10.37	22.40	1	27-329

Table 1: Summary statistics for MovieLens dataset.

In addition, we consider the loss function of the optimization in Equation (1) to better quantify this approach. The loss function is composed of a training error (in blue) which is the mean squared error between the known ratings and predicted ratings and the Frobenius norms of the two low-ranked matrices (in red) which is used to estimate the inverse of the generalization power to unseen data. The formula is given below (See appendix B for notation):

$$\frac{1}{2} \sum_{\forall (u,i) \in \mathcal{K}} (r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_p}{2} \sum_{\forall u \in \mathcal{K}_U} \|\mathbf{p}_u\|_2^2 + \frac{\lambda_q}{2} \sum_{\forall i \in \mathcal{K}_I} \|\mathbf{q}_i\|_2^2 + \frac{\lambda_U}{2} \sum_{\forall u \in \mathcal{K}_U} \|b_U(u)\|_2^2 + \frac{\lambda_I}{2} \sum_{\forall i \in \mathcal{K}_I} \|b_I(i)\|_2^2 \quad (1)$$

In the above equation, a universal weight λ_p is assigned to all the user latent factors, similarly, λ_q is assigned to all the item latent factors, λ_U to all user bias terms, and λ_I to all item bias terms. To implement the differentiating regularisation weights, the authors define new weights that are inversely correlated with the number of ratings received by the item or the number of items the user rated. More specifically,

$$\lambda'_p(u) = \frac{\lambda_p}{f(R_U(u))}, \quad \lambda'_q(i) = \frac{\lambda_q}{f(R_I(i))}, \quad \lambda'_U(u) = \frac{\lambda_U}{f(R_U(u))}, \quad \lambda'_I(i) = \frac{\lambda_I}{f(R_I(i))} \quad (2)$$

where $R_U(u)$ is the set of items rated by the user u and $R_I(i)$ is the set of all users who rated item i . We highlight the dependence of the weight on the specific item or user by making the weight a function of u or i . Note that, if $f(x) = 1 \forall x$, the object functions are identical. The authors propose three different possible functions to specify the weights; a linear, square root, and logarithmic function. Based on their results, the logarithmic function is chosen as it performed best for NMF. More specifically, the chosen function is $f(x) = \log(|x| + \exp(1))$ where $\exp(1)$ was added to ensure the function only returns positive values (necessary by definition of regularisation weights). Given this function, we see that if, for example, an item is rated more often the weight will be smaller making the term weaker in the minimization. This allows the system to leverage the additional information that this item provides and learn latent factors that better summaries the properties of the item. Chen & Chen (2019) verify this with experiments on multiple datasets and the improvement can be noted in our results as well. In contrast, this approach assigns higher constraints to long-tail items and less active users, making the latent variables less susceptible to few extremes. This improved the predictions on the less active users and long-tail items, which is seen in the results of Chen & Chen (2019) and ours. Thus, we argue that this technique partially alleviates the cold start problem and reduces the quality bias towards more active and popular items.

Lastly, we saw that Movinder treated the users in the focus group and the users in the MovieLens dataset equally. We believe that more importance must be given to the users in the focus group as these are the users we are recommending to. Intuitively, one would consider decreasing the weight for our users so that the system regularises less and better fits the preferences of the users in the focus group. However, we noticed that the users of the focus group make up the bottom 0.7% of all users as each user in the MovieLens dataset rated at least 20 movies. To better comprehend this discrepancy, we note that user rated 165 movies on average in the MovieLens dataset. Thus, we have the least information for the users in the focus group, as they only rate 15 movies. This makes the latent factors found noisy and very susceptible to extreme ratings. We considered increasing the number of movies rated, however, rating fifteen movies already requires substantial effort from the user and any higher number would render the system unusable. To address their extreme long-tail status, we add an activity term to these users, increasing their weight. This resulted in more conservative predictions allowing the system to generalise better for these users. Moreover, the reasoning behind this corresponds to the reasoning behind Chen & Chen's (2019) technique.

Additionally, we prototype the incorporation of group dynamics in the group recommendation system by defining different activity levels for the users and chose this additional weight based on which activity level the user is categorised in. We assigned an activity level for each user in the focus group based on two assumptions: (1) A high variance in ratings implies high user activity, and (2) the importance ranking for the mean of the ratings is Low, High, Neutral. As the movies are randomly chosen from the dataset there should be high variance in the user ratings. Thus, higher variance implies that a user is more active. For the second assumption, we note that a user with a low mean is harder to please, and hence will be more difficult to satisfy. A higher mean should be easier to satisfy, thus they are given lower importance. This assumption is based on Dara et al. (2019) claim stating that users prefer not getting what they want than to explicitly get what they do not want. Moreover, a neutral mean also implies less activity as the movies are selected randomly.

Given these assumptions, we define six different activity level groups based on the mean and variance of the ratings that the user-provided. These activity levels are defined below:

Activity Level:	Low mean High variance	High mean High variance	Neutral mean High variance	Low mean Low variance	High mean Low variance	Neutral mean Low variance
Weight:	1/6	1/5	1/4	1/3	1/2	1

The thresholds used to determine which activity level is assigned to the user are given below:

High mean: > 3.5	High variance: > 1
Neutral mean: $2.5 < \dots < 3.5$	Low variance: ≤ 1
Low mean: < 2.5	

We recommend analysing these thresholds further. In the system evaluation, these thresholds were used to create the different groups tested. However, it would be better to learn these thresholds based on data.

After assigning each user their activity level we define their weights as follows:

$$\lambda''_p(u) = \frac{\lambda_p}{f(R_u(u))} + \lambda_p \cdot \text{activity_level_weight}(u) \quad (3)$$

where $\text{activity_level_weight}(u) \in \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}\}$. With this addition, we assign a larger weight to the users of the focus group highlighting their greater importance in the minimisation compared to the users in the MovieLens dataset. Zooming in further, we assign a lower weight on the more active users (1/6, 1/5, 1/4) and a larger weight on the more inactive users (1/3, 1/2, 1) as we can learn more from the active users given that they reveal more information about themselves. This again corresponds to Chen & Chen's (2019) technique.

4. Techniques and technologies used (0.5 page)

The system is implemented in Python using Flask, one of the most common Python APIs for building web applications. For the interface, HTML and CSS were used. No additional memory or CPUs were used for the project. All algorithms can be run within two minutes on a standard laptop. To make the application easily accessible to users, it is hosted online using Heroku, a well-known cloud platform as a service.

The main technique used in this system is the regularisation differentiating functions non-negative matrix factorisation (RDFNMF) defined by Chen & Chen (2019). In this section, we detail the prediction process for the method. As described above, NMF models ratings in a large matrix \mathbf{R} and attempts to approximately decompose this matrix into a user latent factor matrix \mathbf{P} and an item latent factor matrix \mathbf{Q} , such that $\mathbf{R} \approx \mathbf{P} \cdot \mathbf{Q}^T$. By including non-uniform weights for the users and the items, \mathbf{P} and \mathbf{Q} can better represent the ratings. Then the rating prediction of user u on item i is

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{q}_i^T \cdot \mathbf{p}_u, \quad (4)$$

where μ is the global mean i.e. the average rating for all known ratings, b_u is the user bias i.e. the tendency for user u to over or underrate an item compared to other users, b_i is the item bias and \mathbf{p}_u and \mathbf{q}_i are the latent factors for user u and item i . If \mathbf{p}_u and \mathbf{q}_i are similar, the inner product will be large and thus the rating will be high. Otherwise, the predicted rating will be low. Chen & Chen (2019) argue that this prediction method allows for the best results as it uses both biases and the latent factors in addition to the global mean.

5. Evaluation methodology and results (2 page)

The evaluation of group recommenders is related to the evaluation techniques used for a single recommendation (Felfernig et al., 2018). Due to time constraints, online metrics were not considered as they usually involve user studies. Moreover, these are usually very subjective metrics. Instead, we focus on offline metrics which measure the prediction quality based on datasets split into test and training sets. The most common metric used for rating quality analysis is the root mean square error (RMSE) defined as follows:

$$\sqrt{\frac{1}{|\mathcal{K}_{test}|} \sum_{(u,i) \in \mathcal{K}_{test}} (r_{ui} - \hat{r}_{ui})^2}, \quad (5)$$

where \mathcal{K}_{test} is the set of ratings in the test set, r_{ui} is the ground truth and \hat{r}_{ui} is the system prediction. This is a negative-oriented score and thus a lower RMSE implies more accurate predictions.

To validate our approach, we use the publicly available MovieLens dataset which contains 100,000 ratings of 9,000 movies by 600 users. In addition, we generate ratings for five different types of focus groups. Each focus group consists of four users (different to those in the MovieLens dataset) and each user rates 15 movies randomly selected from the MovieLens dataset. The differences in these focus groups are the activity levels of the users. We consider five different groups where one contains user with average activity levels, one with all users active, one with all users inactive, one where half the users are active and the other two inactive, and lastly, one where three users are active and one is not. The two goals we focus on in this evaluation is the comparison between NMF, RDFNMF, and RDFNMF with activity levels and how the system performs for the different focus groups with users having different activity levels.

First, we compare the results of NMF, RDFNMF, and RDFNMF with activity levels. For consistency, we used the NMF and RDFNMF code provided by Chen & Chen (2019). For all methods, the regularisation term chosen for all latent variables and bias variables was 0.02 as chosen by Chen & Chen for the MovieLens dataset. The number of epochs was set to 50, the learning rate to 0.005, the number of latent factors to 15, and the learning rate shrinkage rate to 0.9 as chosen by Chen & Chen (2019) as well. We did not perform hyperparameter tuning as it requires extensive experiments like cross-validation which we did not have time for in this project and just used the values determined by Chen & Chen (2019). However, during testing, it was found that hyperparameter tuning can improve the results. More specifically, we suggest tuning the regularisation terms used for the latent factors (see parts highlighted red in loss function) for better results. This was a surprising discovery as we thought that the effect of the regularisation term’s value would diminish given the division by the regularisation differentiating function. Furthermore, for the below experiments, all regularisation terms were set to 0.02 whereas the algorithm may benefit from different regularisation values per term.

After our initial run for RDFNMF, we saw a less significant improvement in the results than initially anticipated. After analysing the code we realised that in the provided code for RDFNMF the latent variable was still constant for all users and items and thus, the improvement was merely due to hyperparameter tuning. Therefore, we had to significantly refactor the RDFNMF code to match the pseudocode provided in Algorithm 1 of Chen & Chen (2019). In hopes to verify this error, an email was also sent to the authors, however, no response was received at the time of submission. Nonetheless, the refactoring produced results consistent with those given in Chen & Chen (2019). Lastly, RDFNMF with activity levels was implemented based on our refactored RDFNMF code. All code can be found in the GitHub link above.

For the evaluation of the methods, we use the dataset from the focus group with average activity levels for all users. From both the MovieLens dataset and this we randomly sample 20% of the ratings for the test set and use the remaining 80% for training. We do this 50 times and record the average RSME scores in Table 2.

NMF	RDFNMF	RDFNMF with activity levels
1.6284 ± 5.0925	1.4029 ± 3.2622	1.0348 ± 0.2421

Table 2: Average RMSE scores for the three different methods with standard deviation.

In Table 2, we see a clear improvement from the baseline methods. The improvement between NMF and RDFNMF is consistent with the results of Chen & Chen (2019), however, both means are slightly higher. This discrepancy can be explained by the fact that a random algorithm is by the fact that we only took an average of 50 splits where there are millions of different possible splits.

As previously mentioned, a key improvement of the system was to place greater importance on the users of the focus group compared to the users in the MovieLens dataset by increasing their weight with an assigned activity level. Additionally, Chen & Chen (2019) claimed that this weighting technique improved ratings for less active users. As the users of the focus group make up the bottom 0.7% of the long-tail users, RDFNMF considers them as the least active users. We compute the average RMSEs for these users separately to verify this quality improvement claim. The average RMSE scores for 50 splits are given in Table 3.

NMF	RDFNMF	RDFNMF with activity levels
2.0047 ± 0.4241	1.5121 ± 0.3501	1.5021 ± 0.3340

Table 3: Average RMSE scores for the three different methods for focus group users.

Table 3 shows a clear improvement between NMF and RDFNMF as reported in the paper. Adding activity levels only has a small improvement. However, note that for this comparison the focus group contains users with average activity levels. The mean values are lower for groups with more variety in user activity levels. In addition, these users make up the bottom 0.7% of users whereas Chen & Chen (2019) considered the bottom 10%. When considering the bottom 10% the differences are more notable as shown in Table 4. Despite the improvement, the predictions are still less accurate than an average user due to their long-tail status.

NMF	RDFNMF	RDFNMF with activity levels
1.9301 ± 1.5742	1.4508 ± 0.2710	1.3020 ± 0.2196

Table 4: Average RMSE scores for the three different methods for bottom 10% of users.

Lastly, we extracted the 10% of movies that received the fewest numbers of ratings and also computed their mean RSME for the three methods. These are recorded in Table 5.

NMF	RDFNMF	RDFNMF with activity levels
3.0042 ± 4.8588	2.0937 ± 3.6741	2.0155 ± 0.1435

Table 5: Average RMSE scores for the three different methods for bottom 10% of movies.

In Table 5, we see that the values are significantly higher. This because that all the bottom 10% of movies only receive one rating (See Table 1) and thus the latent factors extracted are very noisy and can predict less accurately. Nonetheless, these results explain the higher standard deviation for the overall means in Table 2. Moreover, we see that our system does improve predictions for long-tail movies as well. Thus, we conclude that this improvement can better serve less active users and items less than mainstream items, and addresses users in the focus group better.

Secondly, we focus on how the system handles the five different focus groups. We again sample 20% of ratings for the test set and use the remaining 80% for training. We take the average of 50 random splits and use the same hyperparameters as noted above. To better evaluate this goal, we present both the overall means and the means of the focus group users in Table 6.

	All average activity	All active	All inactive	Two active, two inactive	Three active, one inactive
Overall average RMSE	0.9937 ± 0.30487	1.4471 ± 0.5229	1.2657 ± 0.2556	1.1818 ± 0.3018	1.02944 ± 0.2911
Focus group average RMSE	1.5186 ± 0.3915	1.5214 ± 0.4857	1.6498 ± 0.1982	1.4263 ± 0.3110	1.4539 ± 0.2633

Table 6: Average RMSE scores for RDFNMF with activity levels for five different focus groups.

In Table 6, we see that the scores for the average activity are consistent with those seen in Table 2 and 3 as the same datasets were used. The difference highlights the stochasticity of RDFNMF and the effect of the different dataset splits. When all users are active, the mean for focus group users does not change too much. We expected it to decrease more, given that the users provide more information and thus can be better modelled by the latent factors. However, with all users active there may have also been more conflicts in the predictions as only a basic aggregation strategy was used. The high standard deviation further supports this hypothesis. Regardless, the users of the focus group are still long-tail users and thus the effects might not be so obvious. The higher overall mean can be explained by the fact that more information is used from the users in the focus group for the latent factors, making predictions for other users less stable. When all users are inactive, stronger constraints are placed on the users of the focus group. Thus, we expect the predictions to be more conservative. This is verified by a higher mean and a lower standard deviation for the focus group users. The overall mean is in-between the overall means for the two aforementioned cases. This is explained by the fact that more importance is given to the training set users as inactive users have higher weight and do not offer sufficient information. This results in predictions with a high bias. The focus groups with mixed activity levels perform best; in particular when two members are active and the other two are inactive. This demonstrates the effect of including the activity levels. The higher mean for the user of the focus group when more users are active can be explained by the higher chance of conflicts with user preferences. Thus, the best performance for the 50-50 mixed group can be explained by a balance between enough information provided to the system and conflicting preferences. Lastly, we still see the effects of the long-tail distribution as the means for the users of the focus group are significantly higher than the overall mean.

Finally, we address the model and time complexity of the models. Dara et al. (2019) note that many past works evaluate their proposed models for quality, but not by model complexity, despite the apparent need to incorporate these algorithms in real-time scenarios. In all three models, the parameters to be learnt are and (See appendix B for notation). Thus, the number of parameters to be learnt is $mk + nk + m + n = (m + n)(k + 1)$. Chen & Chen (2019) show that the run-time for RDFNMF is $O(|\mathcal{K}|k)$ for each epoch, which is the same for NMF. RDFNMF with activity levels has the same asymptotic run-time, however, the coefficients hidden in the big-O notation are larger. Therefore, it may take slightly longer in practice. However, prediction can be computed in $O(k)$ time for all methods (See section 4). Thus, the additions do not severely affect the runtime of the current state-of-the-art.

6. References (does not count towards page limit)

- Banjac, J., Altinigne, C., Kypraiou, S., & Sioulas, P. (2020). *Movinder: A Movie Recommendation System for Groups*. Lausanne: EPFL.
- Berry, S., Fazio, S., Zhou, Y., Scott, B., & Francisco-Revilla, L. (2010). Netflix recommendations for groups. *Proceedings Of The American Society For Information Science And Technology*, 47(1), 1-3. <https://doi.org/10.1002/meet.14504701402>
- Chen, H., & Chen, P. (2019). Differentiating Regularization Weights -- A Simple Mechanism to Alleviate Cold Start in Recommender Systems. *ACM Transactions On Knowledge Discovery From Data*, 13(1), 1-22. <https://doi.org/10.1145/3285954>
- Dara, S., Chowdary, C., & Kumar, C. (2019). A survey on group recommender systems. *Journal Of Intelligent Information Systems*, 54(2), 271-295. <https://doi.org/10.1007/s10844-018-0542-3>
- Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). Evaluating Group Recommender Systems. *Springerbriefs In Electrical And Computer Engineering*, 59-71. https://doi.org/10.1007/978-3-319-75067-5_3
- Ojeda-Zapata, Julio (1997). *New Site Personalizes Movie Reviews*. St. Paul Pioneer Press. p. 3E.
- Recio-Garcia, J., Jimenez-Diaz, G., Sanchez-Ruiz, A., & Diaz-Agudo, B. (2009). Personality aware recommendations to groups. *Proceedings Of The Third ACM Conference On Recommender Systems - Recsys '09*. <https://doi.org/10.1145/1639714.1639779>
- Trochim, W. (2020). The Qualitative-Quantitative Debate [Blog]. Retrieved 18 June 2021, from <https://conjointly.com/kb/qualitative-debate/>.

A. Process (1 page)

At the very beginning, the entire group performed a literature search and proposed topics. After a brainstorming session, we decided on the topic of group recommendation as a group. Each of us then dove into literature for this topic to grasp the topic better. This enabled us to construct a plan for the coming weeks for the system and the proposal. After this second, more focused literature search, we decided to split into two subgroups for the system and the proposal due to time constraints. Stefan and Caroline were responsible for the system.

The system was implemented based on the Movinder interface and the code provided by Chen & Chen (2019) for RDFNMF. From the beginning, Stefan familiarised himself with the Movinder code to implement the improvements for the interface and integrate the new dataset. Meanwhile, Caroline read Chen & Chen's (2019) RDFNMF paper to understand the approach and familiarised herself with the provided code. Later she also extended Chen & Chen's (2019) approach to include activity levels. Lastly, Stefan was in charge of merging the code and cleaning it up, and also recorded the video. On the other hand, Caroline wrote up the report sections. In the next paragraphs, the actions taken by both group members are further detailed.

Stefan started investigating the Movinder code and adapted the interface based on the additions mentioned in the report. One of the first changes was the rating scale. Then Stefan moved his attention towards generating the dataset files from the inputs given by the focus groups via the interface. After that, Stefan spent many hours integrating the newer MovieLens dataset. Not only was the format different, but the movies in the datasets were also different. Thus, for the interface to work correctly, posters had to be provided for each movie. Therefore, Stefan had to scrape the IMDb dataset for the posters of the movies. Lastly, Stefan added a trending factor to the movies so that the movies prompted to the users were more likely to be known by them. Stefan also took care of merging Caroline's code with the interface and back end and worked on making the interface more visually appealing. He also used the Heroku application to host our system online. Moreover, Stefan scripted and recorded the video for the system.

Caroline started investigating the regularisation differentiating weights introduced by Chen & Chen (2019). After reading the paper, she dove into the code. After the first couple runs she realised that the implementation provided by Chen & Chen (2019) had an error as the RDFNMF weight was the same for all users and items, which was inconsistent with the pseudocode provided in the paper. Therefore, she refactored the code to match the pseudocode. To verify this error, an email was sent to the authors, however, no response was received at the time of submission. In addition, Caroline extended the RDFNMF with the activity levels concept defined above. This was both theoretically described in the report and implemented in Python. To validate the approaches, Caroline also performed the testing. The results were detailed in this report. Lastly, Caroline wrote up the report sections for the system.

Both Caroline and Stefan met regularly and helped each other throughout the quarter. In general, the whole group was very motivated and we got along well. We met often together and in subgroups and always kept each other updated. Due to time constraints, we thought it was best to split up into two subgroups, one for the system and one for the proposal. However, looking back it would have been nicer to work together more as a group. Nonetheless, we were able to work all together for the brainstorming phase, the intake, and the midterm presentation.

B. Notation

Please note that notation was chosen to be consistent with that of Chen & Chen (2019).

Symbol	Definition
n_{train}	Number of users in MovieLens dataset.
n_{group}	Number of users in focus group.
$n = n_{train} + n_{group}$	Total number of users.
m	Number of items.
k	Number of latent factors.
$\mathbf{R} = [r_{ui}]$	Rating matrix with n rows and m columns for $u = 1, \dots, n$ and $i = 1, \dots, m$.
μ	Global mean.
$\mathbf{P} = [\mathbf{p}_u]^T$	Low-ranked n by k matrix representing the user latent factors.
$\mathbf{Q} = [\mathbf{q}_i]^T$	Low-ranked m by k matrix representing the item latent factors.
$b_U(u)$	Bias of user u , $u = 1, \dots, n$.
$b_I(i)$	Bias of item i , $i = 1, \dots, m$.
\mathcal{K}	The set of all (user, item) pairs from known ratings.
\mathcal{K}_U	The set of all known users. Note, $ \mathcal{K}_U = n$.
\mathcal{K}_I	The set of all known items. Note, $ \mathcal{K}_I = m$.
λ_x	Regularization term for given parameter x .
$R_U(u)$	Set of all items rated by user u .
$R_I(i)$	Set of all users who rated item i .

Incorporating Group Interdependence into Group Movie Recommendation

Proposal summary (300 words)

Recommending movies for a group is a challenging problem to solve. In a standard recommendation system, one only needs to worry about an individual user but in a group recommender system, one needs to consider individual preferences as well as the complex intertwined relationship that exists between different members. The process of modelling these intertwined relationships, albeit alien in computer science, is empirically defined in fields like psychology and social science. However, translating these concepts from other disciplines is a difficult task. Thus, many works in the field of group recommendation assume that the group structure is homogenous, that is, individual members are all equal. This prevents group recommendation systems from being robust. To incorporate heterogeneity, we attempt to incorporate a concept called group interdependence from the field of psychology. Group interdependence is simply a measure of influence each member has on their peers within a group. We hypothesise that incorporating this concept within a model allows us to effectively recommend movies to a wide variety of groups. We utilise the Twitter platform to record the complex interactions that foster within a group and then utilise this information to generate a concrete mathematical model that embeds the concept of group interdependence.

Extended synopsis of the project proposal

1. Problem description (0.5 page)

Currently, many group recommendation systems focus on recommendations for homogenous groups (Dara et al., 2019). Homogeneity assumes that members have similar interests which might be the case for friends but cannot be the default assumption for most cases. For example, in the case of families, the individual movie preferences between mother, father, and children probably differ (Forsyth, 2019). Therefore, most groups should be described as heterogeneous according to their composition.

However, by assuming heterogeneous groups the challenge of integrating inconsistent influence imparted by different users within a group arises (Cantador and Castells, 2012). Evaluation of traditional group recommendation systems suited for homogenous groups show considerably worse performance on heterogeneous group structures. Also, homogeneity is a rare phenomenon. On the other hand, recommender systems that include heterogeneous groups must establish a way to aggregate or combine the individual ratings based on the pattern of heterogeneity. Most aggregation strategies applied in this domain are static and focuses on obtaining a recommendation that satisfies every individual member. However, group satisfaction is not always equal to the aggregation of group members satisfaction, because people can have different opinions and expectations (Recio-Garcia et al., 2009).

Therefore, instead of assuming equal contribution of each group member, the aggregation strategy should incorporate various aspects of group dynamics. Group dynamics describes a broad concept and to accommodate each dimension into a particular group recommendation framework can constitute a multitude of research work. However, to solve the problem of group recommendation we need not consider all aspects of group dynamics as there exists a concept called the group interdependence that encircles many of these concepts relevant for recommending movies to a group. The term group interdependence defines how other people can influence one's outcomes, actions, thoughts, feelings, and experiences to some degree (Forsyth, 2019). Group interdependence enables us to understand how group member's outcomes, thoughts, actions, or feelings are indirectly or directly influenced by other members of the group (Recio-Garcia et al., 2009). Thus, when modelling group decisions, it is necessary to consider the influence patterns between group members.

The problem of this research proposal can be summarized as: *Based on both user-movie rating matrix and extracted group interdependences from Twitter data, recommend a list of movies to the group.*

2. Previous work (0.75 page)

Previous group recommendation systems are commonly divided into three categories; the first category includes all approaches using a predefined aggregation strategy for a combination of individual ratings ((Masthoff, 2002), (O'Connor et al., 2001), (Gorla et al., 2013)). As previously mentioned, this strategy does not consider group dynamics and is therefore not suitable for heterogeneous groups. The second category considers group dynamics by collecting additional information except for item ratings from users. For example, the recommendation system by Quijano-Sanchez et al. (2013) considers the personality of group members and their trust among each other. To analyse a user's personality, they use the Thomas-Kilmann Conflict Mode Instrument (TKI) test. This strategy requires that every user fills in the TKI upfront. The disadvantage of this approach is its reliance on the additional effort of the users to finish the TKI test before being able to use the recommendation system. The challenges behind these approaches that require additional information input from the user are unclarity about what additional information is necessary to obtain and if the user can provide them. Lastly, the third category describes approaches where domain experts guide the combination process considering user ratings and their group dynamic. A popular example is the INTRIGUE tourist recommendation system from Ardissono et al. (2003) which recommends sightseeing destinations and itineraries to heterogeneous travel groups.

Although the approaches described in the second and third category are suitable to create group recommendation systems for heterogeneous groups, they require substantial additional information. Obtaining this additional information comes with an increase in time and effort for the users as well as an increase in the time and space complexity of the method itself. Another potential limitation is that it becomes unclear to the user how the recommendation is suggested. Thus, the algorithm might suffer from understandability and expansibility.

A method proposed by Chen et al. (2008) predicts the interaction between members of a group with a genetic algorithm. They focus on predicting the interaction without increasing the effort and time required from the user by only defining the GA according to users and groups specified ratings. Although the method shows significant improvement it assumes the group to be persistent and relies on known preferences of the group which are not always available.

Another example of social-based recommendation systems developed by Soletto et al. (2009) analyses a user's social network and connections to recommend items that other trusted users liked. The group recommendation system by Soletto et al. (2009) uses different consensus strategies to recommend TV programs to a group of users depending on their composition. The authors use TV content meta-data to define different user profiles based on four different attributes. Each user is modelled by four vectors representing their preferences for the defined attributes. Depending on the correlation between each pair of users a consensus strategy is selected which differs for homogeneous and heterogeneous groups. Although this approach considers heterogeneous groups it does not differentiate between the different group dynamics in heterogeneous groups, such as hierarchical, sequential, or symmetric (Forsyth, 2019).

3. Scientific contribution (0.75 page)

The aim of the research proposal is the exploration of modelling heterogeneous groups for movie group recommendation given their group dynamics, focusing specifically on group interdependence. Modelling heterogeneous groups is a relevant but not as active research field. This is summarized in many reviews about group recommendation, such as by Dara et al. (2019). Further research about group recommendation systems showed that if heterogeneous group modelling occurred this seldomly incorporates group interdependence and rather focuses on modelling similarities in group members' preferences, as done by Chen et al. (2008).

Again, let us understand why group interdependence is important for group recommendation; imagine you want to watch a movie together with your friend. You will likely not only consider your favourites but also your friend's favourite movies. However, this may not be the case for every group of friends i.e., all friends may not be this considerate towards the other. In the end, you want to watch a movie but how you determine what to watch often depends upon the type of interdependence exhibited between group members. The definition of group interdependence states that individual members outcomes, actions, thoughts, and experiences are partially determined by others in the group. If we can capture this group interdependence, then we can establish a dynamic model that can recommend movies to a wide variety of heterogeneous group effectively. These are henceforth described as the main contribution of our paper.

Formally – We explore a new framework to captures interdependency between members in a group using social network data by incorporating the group members' interdependence factor to recommend movies to heterogeneous groups.

The task to model interdependence is crucial to our project's success but it is also the most difficult aspect because it requires translating a conceptually defined framework from the field of psychology to a concrete methodology in computer science. An additional difficulty is the evaluation procedure. To the best of our knowledge, there are no datasets or benchmarks to validate the effectiveness of our group recommender system. We either must synthesize the data set or rely upon subjective evaluation (Felfernig et al., 2018).

To evaluate the efficacy of our group recommendation subjectively we need to test the system on groups of people belonging to different interdependence categories. This may require us to deploy our work on a production-ready movie streaming platform. Although these are achievable, they may require a certain budget. Additionally, it might be challenging to find groups corresponding to the different interdependence categories as the group members themselves might not be explicitly aware of their group interaction themselves. For example, if one group members preference is dominated by others then the person does not necessarily need to be aware of this (Forsynth, 2019).

Despite the challenges explained above, the group recommendation system can be considered as state-of-the-art as it allows recommendation for various types of heterogeneous groups. Compared to previous work our model's ability to adapt to different types of groups makes it stand out. Thereby, it is novel as it does not require groups to be static and is believed to improve over time after multiple use cases as it will learn more about the group's interdependence. Furthermore, we believe that if this model shows to be successful it can also be employed in other domains such as music recommendation as it does not rely on content-based information but rather on the group's interaction.

4. Methodology (1 page)

We hypothesize that interdependence between the members of the group positively co-relates with the amount of influence they have. In this context, influence refers to the impact group members have on each other's decision-making process. Figure 2 specifies the different interdependence structure that exists within a group. The arrows in the image represent influence. In Figure 2.c we also see the concept of reciprocity. Reciprocity means the ability of a member to reciprocate the influence imparted on him/her by other members.

To build this influence network (from here onwards we refer to interdependence network as influence network) we take the help of social network data. In social networking platform like Twitter often influence is negatively correlated with interaction. Consider the case of a very popular influencer (A) on social media. Apart from having a skewed following/follower count, they tend to have limited interaction with contents created by non-influencers. Alternatively, content consumers are more likely to interact with the content shared by influencers. Now imagine the influencer goes to watch a movie in an ephemeral group that consists of his/her followers (B, C, D). The non-influencers (followers) are more likely to cede ground to please the influencer creating a hierarchical interdependence (Figure 2.b). At times, the non-influencers can also influence influencers to some degree (Figure 2.c), but the degree of such influence will be comparatively lower. Similarly, interaction amongst a group of friends will be approximately equal forming a symmetric interdependence (Figure 2.a). Now to establish a proper weightage to individual ratings we need to capture the influence a member has on other members of the group. To properly translate this hypothesis into a concrete method we divide our

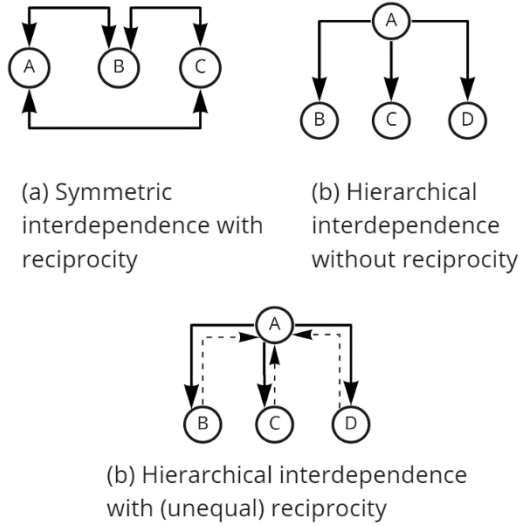


Figure 2: Example of interdependence among group members (*Forsyth, 2019*).

authors define four time series related metrics namely - 1) non-zero days (count the non-zero days the users interacted), 2) mean and variance over non-zero count, 3) daily exponentially decayed interaction count, 4) days since last interaction that aims to capture the frequency, intensity, and recency of each interaction. For our work, we want to focus primarily on the visible interactions and try to develop different algorithm variants utilising one or more interaction metrics in conjunction in order to find the most effective one. After establishing the network, we use the weights of the directed edges between each pair of nodes (users) in the next stage of our methodology. So, if there are three nodes in the graph then we will have six weights representing directed pairwise interaction (two per user). These weights can be represented by w_{ij} indicating the amount of interaction done by user i on the contents shared by user j .

Stage 2: Converting interaction to Influence: In the next stage of our methodology, we determine the influence of user i on user j . To do this we simply divide w_{ji} with w_{ij} . If the ratio is greater than 1 it means i has more influence on j since $w_{ji} > w_{ij}$ (recall that the relationship between interaction and interdependence is reciprocal). If the value is 1 then it would mean they both impart symmetrical influence. Given this, we define interdependence factor y_j to represent the influence score imparted by user j on all other users (hence the summation),

$$y_j = \frac{1}{n} \sum_{i=1}^n \frac{w_{ji}}{w_{ij}}$$

Since these individual weights (w) are unbounded, we will also normalize the interdependence factor to remain within the 0-1 range.

Stage 3: Factoring in interdependence factor: Once we have established the influence score, we can utilise it in our original system design framework. Recall that in our system design we add specific weights based on user activity. In this stage, we simply replace the weight in equation (1) with the interdependence factor calculated in the previous step. One thing to note that when we incorporate the interdependence factor, we will take its reciprocal as we want more importance to be put on the users who have more influence. This concept is similar to assigning lower weights to highly active users to give them more importance.

5. Project objectives, organization and planning (1.5 page)

Our methodology allows us to divide the main task into multiple subtasks which can be performed in parallel. Each of these subtasks can be further divided into more fine-grained subtasks. The study will be performed in a group of four over the period of one academic year. In week 1 the project will start off with a kick-off meeting amongst all the stakeholders where tasks for the first phase will be divided democratically based on member preferences. At the end of each week, the members are supposed to present their findings and brainstorm ideas about implementation.

Table 1 shows the timeline of our project and four different focuses (Implementation, pilot, evaluation and paper) that will be worked on. The general objectives of our project are: 1) link to Twitter, 2) setting up an online server, 3) improving the system design, 4) implementation of the interaction network described in the methodology, 5) evaluation of the system and 6) writing up the research paper. How and which of these objectives will be addressed throughout the year are described more in detail in the following paragraphs.

	2021				2022							
	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June	July	Aug
Implementation	1	1	1	2	2	2	2	3	3	3	3	
Pilot												
Evaluation												
Paper												

Table 1: Timetable

Pilots

We envision running two pilot studies; the first pilot runs from December to January followed by a second one from April to May. The goal of the pilot studies is to receive feedback from users about which recommender system is most accurate and how to improve the system design.

During the first pilot study, we will investigate which features (as described in the methodology i.e., mean and variance of non-zero days) obtained from the Twitter data can be used to most accurately model the interaction network. To do so we will publish multiple system versions, each integrating a different set of Twitter features to recommend movies. After each evaluation session, a questionnaire will be sent to each user regarding their satisfaction with the recommended movies and the system design functionalities. The results of the questionnaires will be used for guidance during the second implementation phase. For the second pilot study, an improved system design is used. Besides that, the second pilot will be using the Twitter features that have proven to be most useful during the first pilot study.

Each of the two pilot studies will be structure in the same manner. Every group participating in the pilot will be testing the movie recommendation system three times. We aim to have a different variety of groups participating, e.g. groups of friends, a collection of random people, or a group of colleagues. During the first pilot, every user will connect their Twitter profile to the system. Note that for simplification purposes we will only select individuals that have a Twitter account. Before each movie recommendation, the user is asked to give fifteen ratings for different movies. After each of the three sessions takes place, a questionnaire is sent to the different users to evaluate their satisfaction with the recommendations.

Implementation

The implementation phase is divided into three subphases; The first subphase runs from September to November and focuses on finishing a prototype for the first pilot study. During the second subphase, we will focus on improving the system design and interface as well as enhancing the generation of interaction networks using the features that proved as most helpful from the first pilot study. After the second pilot study, the third implementation phase focuses on finetuning and error correction to finish up a working movie group recommendation system.

First implementation phase: The prototype will be based upon the already existing system from the system presented above. Therefore, the biggest part of the first implementation phase is to retrieve data from Twitter, establish the interaction networks of users, and develop a live server that can be used for the pilot study. We already performed a proof of study to obtain some user features from Twitter using natural language

processing. This proof of study will be further extended during this study. The team will split up over the three tasks in a formation of 1-2-1. The first subgroup focuses on obtaining data from Twitter and integrating it into the system. The second subgroup implements the interaction network described in the methodology. Besides the implementation of the three stages described in the methodology, this objective also requires performing further literature research to better understand the network. Lastly, the third subgroup focuses on establishing the live server and possible integrating the new features.

Second implementation phase: In the second implementation phase the team will split up into a formation 2-2. The first subgroup will focus on improving the system design based on the feedback of the users from the first pilot phase. Additionally, we are planning to implement further functionalities such as the implementation of a machine learning system that adapts the movies prompted for rating based on previous information. The focus of the second subgroup is on improving the creation of the interaction networks by focusing on the factors that have shown the most success according to the first pilot study.

Third implementation phase: Similarly, as in the second implementation phase, the team will again split up into a 2-2 formation. The first group will focus on finetuning the system design and the second on the network integration generation. If there is extra time, the team will look into further feature improvements such as providing a short motivation per movie for the group.

Evaluation

For our final evaluation, we envision three movie sessions per group. The groups will be formed of a varying degree of members (two to six members per group). Additionally, groups should represent different group interdependencies, i.e. group of friends or group of colleagues. Before each movie session, the participants will be asked how likely they are to see the movie on their own and afterwards the participants are asked how much they liked the movie. The latter will indicate the satisfaction of the group member with the movie recommendation while the difference between the first and the second presents a measure of effectiveness.

6. Risks and mitigation plan (0.5 page)

We expect to encounter two main risks during our project development: 1) Difficulties to empirically evaluate and objectify the results and 2) complications to gather Twitter data from users due to privacy or trust issues.

The first addresses the issue that our group recommendation system is connected to the empirical concept of group interdependence from the field of psychology, translating it into a concrete theoretical function will be challenging and may bring unsatisfactory results, but we assume this to be true for any form of exploratory research. We also rely upon social media data, that might not reflect the true personality of the user. For example, if a user uses Twitter for only professional reasons the information about the personality obtained from Twitter might not reflect how the user interacts among friends. To counteract this issue, we aim to perform multiple pilot studies. During the first pilot study, as explained in the previous section, the group recommendation system will rely on multiple features. After finishing the first pilot study, the intermediate results will be evaluated such that the most promising features are extracted. We hope to improve the evaluation and reflection of a user by considering using pilot studies to adjust our system to better validate the hypothesis. When subjectively evaluating our model, we need to recruit/invite different types of groups. We foresee another risk stemming from under/over-representation of specific group structure that can lead to our evaluation being biased. Hence, we need to pay very close attention to this process to ensure we have a balanced representation.

Secondly, since the methodology relies upon having access to the users' social network, data privacy issues can be a concern. On similar grounds, the evaluation hinges on the fact that we can find groups with different interdependencies. As group members may not be aware of these interdependencies or may not be willing to have these interdependencies exploited, we see it as a risk as well. But our design does not rely directly upon personal data, it only focuses upon the interactions between the users. Due to this we can simply omit personal information in our recommendation system. Furthermore, we only require data during the training phase, as this is not an online learning model, thus, we can delete the data as soon as the recommendation is made.

References (does not count towards page limit)

- Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 17(8–9), 687–714. <https://doi.org/10.1080/713827254>
- Cantador, Iván & Castells, Pablo. (2012). Group Recommender Systems: New Perspectives in the Social Web. Intelligent Systems Reference Library. 32. 10.1007/978-3-642-25694-3_7.
- Chen, Yen-Liang & Cheng, Li Chen & Chuang, Ching-Nan. (2008). A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl.*. 34. 2082-2090. 10.1016/j.eswa.2007.02.008.
- Dara, Harsha & Chowdary, Ravindranath & Kumar, Chintoo. (2020). A survey on group recommender systems. *Journal of Intelligent Information Systems*. 10.1007/s10844-018-0542-3.
- Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). Evaluating Group Recommender Systems. *Springerbriefs In Electrical And Computer Engineering*, 59-71. https://doi.org/10.1007/978-3-319-75067-5_3
- Forsyth, Donelson R. (2019) *Group Dynamics*. Seventh edition. Boston: Cengage
- Gorla, J., Lathia, N., Robertson, S., Wang, J. (2013). Probabilistic group recommendation via information matching. In Proceedings of the 22Nd international conference on world wide Web, WWW '13 (pp. 495–504). New York: ACM.
- Kamath, K., Sharma, A., Wang, D., & Yin, Z. (2014). RealGraph: User Interaction Prediction at Twitter.
- Masthoff, Judith (2002). Modeling a Group of Television Viewers. In Proceedings of the workshop future TV in intelligent tutoring systems conference (pp. 34–42).
- O'Connor, M., Cosley, D., Konstan, J., & Riedl, J. (2001). PolyLens: a recommender system for groups of users. In Proceedings of the European conference on computer-supported cooperative work (pp. 199–218). Germany.
- Quijano-Sanchez, Lara & Recio-García, Juan & Diaz-Agudo, Belen & Jimenez-Diaz, Guillermo. (2013). Social Factors in Group Recommender Systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 4. 10.1145/2414425.2414433
- Recio-García, Juan & Jiménez-Díaz, Guillermo & Sánchez-Ruiz, Antonio & Díaz-Agudo, Belén. (2009). Personality aware recommendations to groups. RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems. 325-328. 10.1145/1639714.1639779.
- Sotelo, Rafael & Blanco-Fernández, Yolanda & López-Nores, Martín & Gil, Alberto & Pazos-Arias, Jose. (2009). TV program recommendation for groups based on multidimensional TV-anytime classifications. *Consumer Electronics, IEEE Transactions on*. 55. 248 - 256. 10.1109/TCE.2009.4814442.

A. Process (1 page)

Before writing this proposal, we undertook an extensive search for scientific literature in this field. Every member of the team took part in this literature survey. Once we have explored a satisfactory amount of literature. The idea behind every member's participation in literature survey was to make both our systems design and research proposal consistent with each other. That is we wanted to implement a basic system and then either extend or tackle the problems in our current system design by means of the research proposal. After the literature survey stage two members started focusing on actual system design (Stefan and Caroline) while two others (Shreyan and Francesca) began working on the research proposal. In our initial iteration we were thinking about incorporating social media data to offset the cold start problem. We also thought about utilising multi arm bandit framework to implement an online learning paradigm that iteratively improves the recommendation made by the system. However, during our midterm feedback session, we realised we were working on multiple directions and did not have a concrete motivation. After that we decided to streamline our proposal's direction a bit more, so we thought about performing an exploratory study that captured influence of individual preferences on the outcome of group learning. We shared our motivation and contribution to the teaching team and based on their feedback we realised we were still falling short. We then decided to change our approach, we went back to the drawing board, brainstormed together multiple times, and decided that we want to explore more about group dynamics. To understand group dynamics, we moved away from literatures in computer science and focused on literature from psychology that deals with group dynamics. Once we found the concept of group interdependence, we again brainstormed and tried to translate it to a concrete methodology that can be implemented. We wanted our method to be concrete and hence we also tried to derive the exact formulations of how we wanted to represent the interdependence factor. After much trial and error, we arrived at a formulation that we all agreed upon. While deriving these formulations we also explored a few more literatures, this time in the field of computer science to validate our argument to some extent. At each state of writing this proposal we wanted to validate our hypothesis with proper citation from the prior work.

Performing literature survey and brainstorming ideas were all done jointly. We also met during the lab sessions in person to further discuss about our direction. At every checkpoint we also consulted other two members who worked on the systems design to have a different perspective.

In terms the actual writeup Shreyan oversaw the scientific contribution, methodology and risk and mitigation plan while Francesca focused on writing problem description, previous works and Project objectives, organization, and planning. But in the end it was a collaborative effort where we both helped each other and completed the writeup for the proposal.

Reflections

Caroline:

One of the main things I learned scientifically was how regularisation worked. I feel like these topics are generally quickly addressed in machine learning courses, yet are extremely powerful techniques that can substantially improve results. Investigating RDFNMF helped me better grasp the concept and learn how regularisation can be used to improve results for very different objectives and items. In a more technical sense, I found it challenging to understand a newly published technique and implement it. Not only was it difficult to understand the approach of Chen & Chen (2019), it was difficult to explain it accurately in the report and to implement the technique, especially given the fact that their code was incomplete. However, I am proud of the work we achieved in the end, in particular, if we consider the time frame of the project which was essentially 6 weeks (intake was at the end of week three and final submission at the end of week nine). Moreover, I think it was difficult to complete such a large assignment in the given time frame, especially with other courses and commitments. I am usually a person who likes to have detailed planning so that I can ensure everything is completed on time. Due to other commitments, this was not possible and I had to learn how to manage my time without explicitly planning things. As this is a vital skill to have, especially in the future when I will have more commitments, I am happy to have experienced it in this group setting.

Stefan:

In my opinion, I think that I've learned about how movie recommender systems actually function (all the way from abstract mathematical concepts to actual interface design), as we've implemented & delivered a fully working group recommendation system. From a scientific perspective, as our project's design was mainly based on literature research, it was very interesting to read & understand how to go about maximizing the amount & validity of the information that one collects by querying users. Furthermore, as our recommendation algorithm (weighted matrix factorization) was (in my opinion) a little bit math-heavy, the process of trying to understand how to apply these abstract concepts in our actual application was kind of challenging & required a lot of effort. However, I think that due to this reason I got to understand what these concepts are and how they can be applied successfully in practice. Technically, I think that I've managed to get some real experience by deploying the system and combining everything together (interface, web-app, algorithm & cloud hosting). As this was a group project, communication was crucial, and I can say that I had the most amazing group members - always active, always down to work & always down to help. Overall, I thoroughly enjoyed working on this project and I can say that this course was a very interesting and useful experience.

Shreyan:

I have always wanted to learn how recommender systems work but was severely limited by the high-level abstraction that is shrouded by state-of-the-art recommender systems. By means of this project I was able to learn in depth, how different techniques in this field work, starting from SVD to deep learning-based models. Another aspect that I learnt is how intertwined the domain of recommender system is with other domains such as information retrieval, natural language processing, reinforcement learning and even non computer science domains such as psychology sociology etc. Through this project I was able to explore how these different interdisciplinary topics influence the field of recommender system and was also able to a proof-of-concept use case by incorporating social web data, natural language processing to make a recommender system. In the final stage of the project, I was also able to explore various cross domain scientific literatures, especially the book on group dynamics by Donelson R. Forsyth was exciting to read. Previously I had little to no idea about how difficult it is to come up with a concrete research proposal. I had lot of ideas to start with but as I started to think about them in detail and write down how I wanted to implement them I came across a lot of roadblocks. Almost of those seemingly "amazing ideas" had to be scrapped because either they were not concrete enough or would require a lot of effort to build them. Eventually when I started concretely put down a proposal, I came across Condorcet paradox. After two days of thinking and writing down equations I finally realised that my work was causing a paradox and hence I was unable to come up with a result. This was unfortunate but at the same time enthralling and motivated me to further explore more about this topic. I was also able to explore concepts like self-attention from the field of deep learning, regularization concepts in machine learning. Overall, it was very hectic, but I am glad to come out with a plethora of knowledge. Before concluding I would like to

thank my entire team, I was lucky to have them as team members, their passion and motivation throughout the project made it all possible.

Francesca:

This course was the first time that I learned about multimedia computing, its challenges, complexities and interconnections to other fields such as machine learning. To be honest, I was quite overwhelmed by all the possibilities initially when performing research about which direction we could concentrate on for our project. Because most courses I took focused on machine learning methods I mainly focusing on how ML techniques could be reused for recommendation systems. However, soon I realised that this exceeded the scope of the project. In general, due to the overwhelming amount of new information one of the main things I learned was about how important it is to narrow down and concretize your ideas. This step is not only important for a good and realistic project outcome but also helps group communication. Overall, I am really satisfied with the topic we picked because it provided me with a good introduction to the basic techniques of recommendation systems such as matrix factorization and the common challenges, e.g. sparsity of the data. Furthermore, I am happy to have picked the research proposal to focus on as it allowed me to research further challenges in recommendation systems and showed me the interconnection of multimedia computing to other domains such as social and behavioural studies. Besides that, this was the first real research proposal that I needed to write, including a planning, which I am sure will be helpful for starting my thesis next year. I really enjoyed working on this project with my group so, thanks for the great course organization and interesting lectures!