DeepGroup: Group Recommendation with Implicit Feedback

Sarina Sajadi Ghaemmaghami
Ontario Tech University
Ontario, Canada
sarina.sajadighaemmaghami@ontariotechu.net

Amirali Salehi-Abari
Ontario Tech University
Ontario, Canada
abari@ontariotechu.ca

ABSTRACT

We focus on making recommendations for a new group of users whose preferences are unknown, but we are given the decisions of other groups. By formulating this problem as *group recommendation from group implicit feedback*, we focus on two of its practical instances: *group decision prediction* and *reverse social choice*. Given a set of groups and their observed decisions, *group decision prediction* intends to predict the decision of a new group of users, whereas *reverse social choice* aims to infer the preferences of those users involved in observed group decisions. These two problems are of interest to not only group recommendation, but also to personal privacy when the users intend to conceal their personal preferences but have participated in group decisions. To tackle these two problems, we propose and study DeepGroup—a deep learning approach for group recommendation with group implicit data. We empirically assess the predictive power of DeepGroup on various real-world datasets and group decision rules. Our extensive experiments not only demonstrate the efficacy of DeepGroup but also shed light on the privacy-leakage concerns of some decision making processes.

ACM Reference Format:
Sarina Sajadi Ghaemmaghami and Amirali Salehi-Abari. 2021. DeepGroup: Group Recommendation with Implicit Feedback. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482081

1 INTRODUCTION

Group decision problems are prevalent, ranging from high-stake decisions (e.g., elections, the board of directors' decisions, etc.) to casual decisions (e.g., deciding on a restaurant or movie for a group of friends). Group recommender systems play an integral role in facilitating group decision-making by recommending a set of items (or a single item) to a group of people. The applications of group recommender systems are diverse, such as tourism [17], music [6], crowdfunding [19], news/web pages [18], and TV programs [28].

Group recommendation methods (e.g., [1, 3, 5, 9, 21–23, 26]) usually encompass two integrated components: *preference assessment* and *preference aggregation*. The *preference assessment* component focuses on understanding group members' preferences. Two common approaches for preference assessment are (a) *preference elicitation* (e.g., [14, 15, 25]) via asking relevant queries for revealing user preferences, and (b) *preference learning* from historical user data represented either in the form of rankings (e.g.,[7, 13, 21]) or user-item interaction data (e.g.,[1, 3, 9, 16, 23]). The *preference aggregation* component aggregates users' inferred (or elicited) preferences into group preferences (or decisions). The aggregation methods are usually well-studied social choice (group consensus) functions [4, 8] or learned by deep learning (e.g.,[5, 11, 12, 27]). We deviate from this common approach of preference assessment and aggregation.

Problems. We focus on a group recommendation problem in situations that the group members' personal preferences cannot be assessed using typical means (e.g., preference elicitation or preference learning) due to unavailability or confidentiality of preferences. However, we assume the presence of a certain type of *implicit feedback* for some group members in the form of which other groups they belong to and those groups' decisions. The applications for this problem are prevalent, e.g., group recommendations for restaurants and vacation packages given that we observe restaurants and places that some group members have attended with their acquaintances. We focus on two special instances of this problem: *group decision prediction*, which intends to predict the decision of a new group of users, and *reverse social choice*, which aims to infer the preferences of a user involved in observed group decisions. In addition to group recommendation, these two special cases are of high importance for assessing privacy leakage. Imagine those users who intend to conceal their personal preferences on a sensitive issue (e.g., promotion, social injustice issues, etc.), but have participated in group decisions on these topics with publicly known decisions.

Contribution. To address the group recommendation problem with group implicit feedback, we propose *DeepGroup*—a deep neural network for learning group representations and decision making. We conducted extensive experiments to evaluate the effectiveness of DeepGroup for both group decision prediction and reverse social choice. Our findings confirm the superiority of DeepGroup over a set of benchmarks for both problems over various datasets. In our experiments, we also study how different group decision rules (or group decision-making processes) might affect the performance of DeepGroup. Our findings show that DeepGroup excels (compared to benchmarks) regardless of the choice of group decision rule and even performs reliably well when different unknown decision rules are used in observed groups. In the reverse social choice task, DeepGroup performance was more prominent for plurality voting. This is an interesting observation regarding privacy. Despite requiring the least personal preference data (i.e., only top choice) for decision making, plurality has the highest privacy leakage.

2 PROBLEM STATEMENT

We consider a set of n users $\mathcal{U} = \{1, \ldots, n\}$ and a set of m alternatives (items) $\mathcal{A} = \{a_1, \ldots, a_m\}$. We assume that we have observed

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482081

l groups of users $\mathcal{G} = \{G_1, \ldots, G_l\}$ with $G_i \subseteq \mathcal{U}$, and their corresponding group decisions in the form of *group-item interaction matrix* $\mathbf{Y} = [y_{ij}] \in \{0,1\}^{l \times m}$, where $y_{ij} = 1$ if $G_i \in \mathcal{G}$ has decided $a_j \in \mathcal{A}$ as their group decision.[1] We focus on the top-k recommendation problem by suggesting the k most preferred (or likely) items from $\mathcal{A}$ to a new group of users $G \subseteq \mathcal{U}$ where $G \notin \mathcal{G}$.[2] While our defined problem covers a broad range of problems, of particular interest, are two special instances of it.

Group Decision Prediction. Single-option group recommendation (i.e., when $k = 1$)—sometimes referred to as *group decision prediction*—not only applies to group recommendation but also can be used for predicting the most likely decision (or outcome) of a newly formed group G. Imagine a committee is asked to decide on a sensitive issue (e.g., promotion, social injustice issues, etc.) when various decisions are possible. The goal is to predict the final decision of the committee based on the involvement of the committee members in previous committees whose final decisions are public.

Reverse Social Choice. By letting the target group G be a singleton set of a user $u \in \mathcal{U}$, one can focus on a special instance of our problem, that we call *reverse social choice.* As opposed to social choice functions that aggregate individuals' preferences to group decisions/preferences, the reverse social choice maps group decisions to individuals' preferences. The solution to this problem not only enhances preference learning but also allows us to measure privacy leakage from publicly announced group decisions.

Regardless of our interest in these two special instances of the group recommendation problem, our solution is for the general problem: we predict the likelihood of the interaction of group G with any item in $\mathcal{A}$, and then select a rank list of k items with the highest prediction score for recommendation to the group G. Our learning task in this paper is to find the *likelihood function* $f(G, a|\boldsymbol{\theta})$ that predicts the likelihood of group G's interaction with any item $a \in \mathcal{A}$. The model parameters $\boldsymbol{\theta}$ is learned from the observed groups $\mathcal{G}$ and group-item interaction matrix $\mathbf{Y}$. We introduce DeepGroup for formulating and learning this likelihood function.

3 DEEPGROUP MODEL

The DeepGroup model takes both group $G_i \subseteq \mathcal{U}$ and item $a_j \in \mathcal{A}$ as an input (see Figure 1). The G_i is represented as the n-row sparse vector $\mathbf{g} = [g_p]$ where $g_p = 1$ if $p \in G_i$ otherwise $g_p = 0$.

The DeepGroup considers real-valued latent representations (or embedding) for all users $u \in \mathcal{U}$ and items $a \in \mathcal{A}$. The latent representations of users and items are captured by $n \times d$ matrix $\mathbf{U}$ and $m \times d'$ matrix $\mathbf{V}$ (resp.), where d and d' are the dimensions of user and item latent spaces (resp.).[3] For the input group $\mathbf{g}$, DeepGroup retrieves all its users' latent representations $\{\mathbf{U}_p | p \in \mathcal{U}$ and $g_p = 1\}$, where $\mathbf{U}_p$ denotes the latent vector of user p (i.e., the p^{th} row in the matrix $\mathbf{U}$). Similarly, DeepGroup looks up the item embedding $\mathbf{V_j}$ for input item a_j. A key idea behind DeepGroup is the aggregation

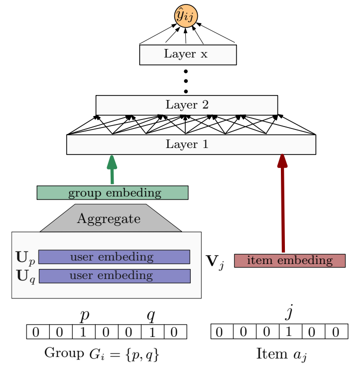


Figure 1: The architecture of the DeepGroup model.

of a group $\mathbf{g}$'s users latent representations $\{\mathbf{U}_p | p \in \mathcal{U}$ and $g_p = 1\}$ into a single fixed-length vector $\mathbf{q}$:

$$\mathbf{q} = \text{Aggregate}\left(\{\mathbf{U}_p | p \in \mathcal{U} \text{ and } g_p = 1\}\right). \quad (1)$$

The Aggregate(.) function takes any set of user latent representations and maps them into $\mathbf{q}$, which is the latent representation of the group $\mathbf{g}$. This group latent representation is expected to capture the consensus preference of group members. We discuss different aggregator functions below. The group latent representation $\mathbf{q}$ and the item embedding $\mathbf{V}_j$ are then concatenated and fed into a multilayer perceptron (MLP) neural network to predict $\hat{y}_{ij}$ (i.e., the likelihood that group g decide on item a_j).

Aggregator Functions. The aggregate function, which maps any arbitrary set of user embeddings into group representation $\mathbf{q}$, is required to satisfy at least two natural properties.

Property 1. *A function* Aggregate *acting on sets of user embeddings must be permutation invariant to the order of user embeddings in the set such that for any permutation π and any user embedding set $\{\mathbf{U}_{i_1}, \cdots, \mathbf{U}_{i_j}\}$:*

$$\text{Aggregate}\left(\{\mathbf{U}_{i_1}, \cdots, \mathbf{U}_{i_j}\}\right) = \text{Aggregate}\left(\{\mathbf{U}_{\pi(i_1)}, \cdots, \mathbf{U}_{\pi(i_j)}\}\right).$$

Property 2. *A function* Aggregate *must have a fixed-length range for any set of user embeddings: letting $\mathcal{E} = \{\mathbf{U}_i | i \in \mathcal{U}\}$ be the set of all users' embeddings, the function* Aggregate : $2^{\mathcal{E}} \to \mathbb{R}^k$ *maps any subset of $\mathcal{E}$ to a k-dimension real-valued vector.*

Given these two properties, we consider the class of *elementwise aggregators* that deploys an elementwise operator (e.g., mean, max, min, median) to reduce a group G's user embeddings $\{\mathbf{U}_p | p \in G\}$ into a group embedding $\mathbf{q}$. An important instance of this class is the Mean aggregator which computes the i^{th} element of the group embedding $\mathbf{q}$ by $q_i = mean(\{u_{pi} | p \in G\})$, where u_{pi} denotes the value of the i^{th} dimension of user p's embedding. A variant of this Mean aggregator has been widely deployed in representation learning on graphs (for example, see [10]) to aggregate features from a node's neighborhood.

Learning. One can learn DeepGroup model parameters by the *maximum likelihood estimation (MLE)* method. Given the observed groups $\mathcal{G}$ and group-item interaction matrix Y, the log-likelihood can be computed by

$$\ell(\boldsymbol{\theta}|\mathcal{G}, \mathbf{Y}) = \sum_{i=1}^{l}\sum_{j=1}^{m} y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log\left(1 - \hat{y}_{ij}\right),$$

[1] One can extend our problem to the setting in which observed decisions/outcomes are in the form of group aggregated rankings rather than a consensus option.

[2] The set of alternatives doesn't have to be the same for all group decisions. One can assume a universal set of alternatives with all possible alternatives. Regardless of which alternative has been available from the universal set for past group decisions, the alternative selected by a group signals the preferences of the group members.

[3] Assuming that the latent representations are drawn identically and independently, these representations can be initialized by sampling from any prior distributions.

where $\hat{y}_{ij} = f(G_i, a_j|\boldsymbol{\theta})$ is DeepGroup's estimated probability for the interaction of group G_i with the item a_j. The maximum likelihood estimate of the model parameters is $\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}|\mathcal{G}, \mathrm{Y})$. Equivalently, one can learn model parameters by minimizing loss $L(\boldsymbol{\theta}|\mathcal{G}, \mathrm{Y}) = -\ell(\boldsymbol{\theta}|\mathcal{G}, \mathrm{Y})$. This loss function is the same as the *binary cross-entropy loss*, which can be minimized by performing stochastic gradient descent (SGD) or any other optimization techniques.

4 EXPERIMENTS

We conduct extensive experiments to evaluate the effectiveness of DeepGroup for both problems of group decision prediction and reverse social choice. We study the efficacy of DeepGroup by comparing its testing accuracy with some other baseline algorithms.[4]

Group Datasets. Due to inaccessibility to group datasets with both group membership data $\mathcal{G}$ and group decisions Y, we create our datasets using real-world preference datasets, different group formation mechanisms, and group decision rules (or voting methods).

We consider four real-world preference ranking datasets.[5] Three datasets are from the 2002 Irish Election:[6] Dublin West with 9 candidates and 29,989 user preferences; Dublin North containing 43,942 user preferences over 12 candidates; and Meath containing 64,081 user preferences over 14 candidates. The other dataset has 5000 user rankings over 10 varieties of sushi.[7] To generate a set of groups $\mathcal{G}$ from these datasets, we deploy the *κ-participation group (KPG)* method. KPG first samples n users from a preference dataset, then κ times randomly partitions this set of users into size-constrained subsets (i.e., groups), whose sizes are bounded by $[s_{min}, s_{max}]$. The KPG outputs the collection of all unique subsets generated by these κ partitions. By varying κ, one can control the extent to which each user participated in different groups (or equivalently, the extent to which groups overlap with one another). We set $s_{min} = 2$ and $s_{max} = 10$ while varying other parameters.

To create the group-item interaction matrix Y for each generated group set $\mathcal{G}$, we aggregate user preferences of each group $G_i \in \mathcal{G}$ to a group decision $a_j \in \mathcal{A}$ by voting rules [4]. We focus on Borda and plurality—two examples of positional scoring rules [4]. For a fixed group set $\mathcal{G}$, we either use Borda for all $G_i \in \mathcal{G}$, use plurality for all $G_i \in \mathcal{G}$, or uniformly at random select between Borda and plurality for each $G_i \in \mathcal{G}$ (i.e., mixture of Borda and Plurality). All three preference aggregation strategies are unknown to the group recommendation methods studied in our experiments. The mixture strategy further challenges the group recommendation methods by stochastically diversifying the group decision rules among groups.

Benchmarks. We compare the accuracy of DeepGroup against some baseline methods. Due to the lack of any comparable algorithms for solving group decision prediction and reverse social choice, our baselines are either (i) adaptations of state-of-the-art deep learning methods to our problems or (ii) our own heuristics. *AGREE* [5] employs both user and group rating preferences for group recommendation. To make it comparable with DeepGroup, we deployed AGREE without user preferences while only inputting the groups and their top choices to this model. Except for this change, the other settings were left as default.[8] *Popularity (Pop)* predicts the most popular group decisions in the training set as the group decision of any groups in the testing set. For a group in the testing set, *Random Top Choice Plurality (RTCP)* first guesses its users' top choices, then outputs the plurality winner (ties broken randomly). To guess user top choice, if a user belongs to at least one group in the training set, RTCP randomly picks one of its groups' decisions as its top choice; otherwise, the method outputs the popular group decision in the training set. For a group in the testing dataset, *Overlap Similarity (O-Sim)* outputs the group decision of the most similar group in the training set, when the similarity is measured by the number of common members.

Setup. In all our experiments, DeepGroup has four hidden layers with 64, 32, 16, and 8 hidden units and the Relu activation function. We use dropout over hidden layers with the retaining probability of 0.8. The dimensions of user and item embeddings are set to 64 and the Mean aggregator is used. We optimize DeepGroup with Adam for 100 epochs with a learning rate of 0.001 and the batch size of 4096 (i.e., each mini-batch encompasses 4096 records of negative/positive group-user interactions along with group membership data). As our group dataset generation is stochastic, for each fixed setting (e.g., preference dataset, group decision rule, etc.), we generate 20 instances of each group dataset setting and report an average accuracy over those instances. In experiments focused on group decision prediction, for each group dataset, we randomly selected 70% of all groups and their group-item interactions as the training set and 30% for the testing set. For reverse social choice, we use each group dataset as the training set and create a testing set including the singleton groups of all users appeared in the training set.

Group decision prediction. We investigate the effectiveness of DeepGroup in group decision prediction when compared to other benchmarks. We set the group decision rule to plurality or Borda. By fixing the number of users $n = 5000$ and varying κ, we study how the performance of different methods changes with more availability of implicit data (i.e., the participation of individuals in different group decisions). Fig. 2 shows the accuracy of different methods for various group datasets for the plurality decision rule.[9] In all four datasets, DeepGroup is comparable with others for $\kappa = 1$ but outperforms the benchmarks for $\kappa \geq 3$. The performance of DeepGroup is more prominent as κ increases (e.g., about 100% improvement over the best baseline for $\kappa = 20$ and Irish datasets). These results suggest that as users participate more in various group decision-making processes, we learn more accurately their embeddings, and consequently their groups' embeddings and decisions.

The effect of group decision rules. We investigate the effect of different group decision rules on the group decision prediction task. We fixed $\kappa = 5$ and $n = 5000$. Fig. 3 shows the average accuracy for various group decision rules (i.e., Borda, Plurality, and their mixtures). For all datasets, DeepGroup outperforms others over all decision rules to various extent. It seems that DeepGroup offers the most absolute improvement over baselines for plurality and the least absolute improvement for Borda. One interesting observation is that DeepGroup still performs fairly well for the mixture of Borda

[4] The code can be found at https://github.com/sarinasajadi/DeepGroup

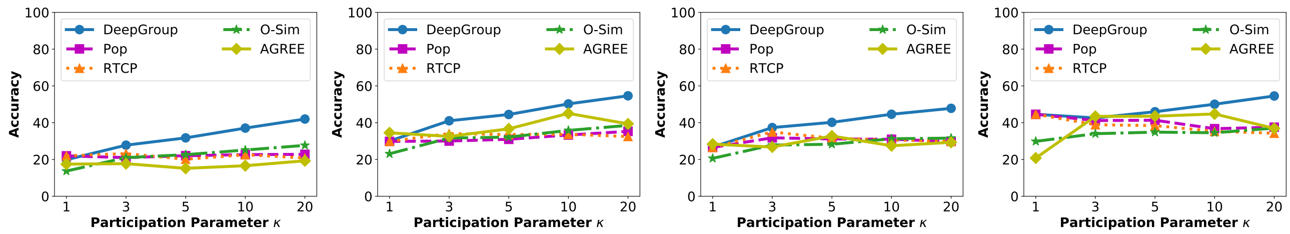
[5] As our problems are closely connected with social choice, we focus on preference rankings which are of special interest in social choice since they help circumvent the problem of interpersonal comparisons of ratings/utilities [2, 24].

[6] http://www.preflib.org/data/election/irish/

[7] http://www.preflib.org/data/election/sushi/

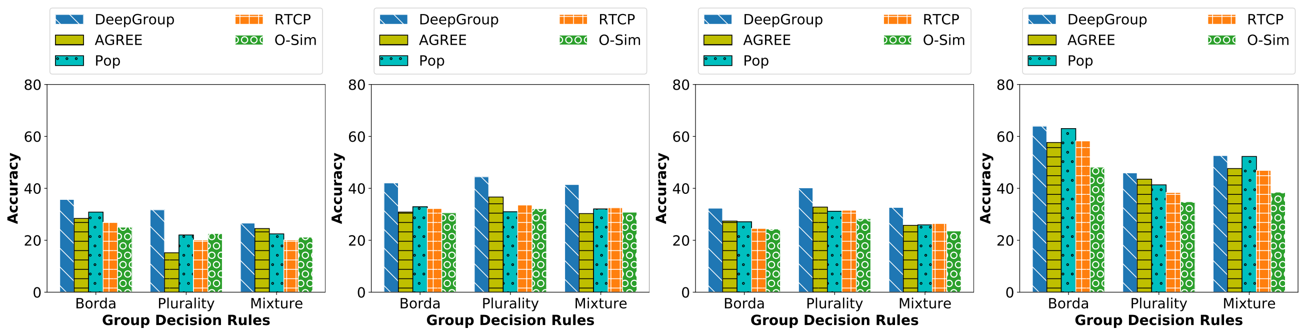
[8] https://github.com/LianHaiMiao/Attentive-Group-Recommendation

[9] The results for Borda were qualitatively similar.



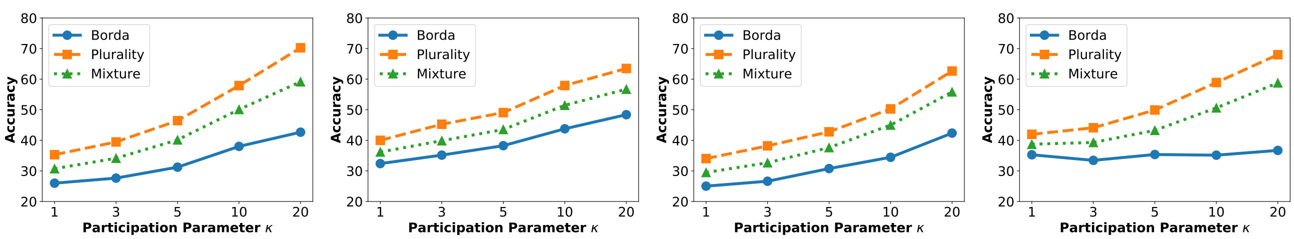
(a) Dublin North (b) Dublin West (c) Meath (d) Sushi

Figure 2: Testing accuracy (%), group datasets generated on preference datasets (a)–(d), the plurality group decision rule.



(a) Dublin North (b) Dublin West (c) Meath (d) Sushi

Figure 3: Testing accuracy (%), group datasets generated on preference datasets (a)–(d) with different group decision rule, $\kappa = 5$.



(a) Dublin North (b) Dublin West (c) Meath (d) Sushi

Figure 4: Testing accuracy (%), reverse social choice, group datasets generated by different decision rules and datasets.

and Plurality. This suggests that (a) DeepGroup does not necessarily require to be aware of group decision rules for successful prediction, and (b) DeepGroup can perform well when different groups use inconsistent decision rules.

Reverse social choice. We study the accuracy of DeepGroup for reverse social choice (i.e., predicting individual preferences of group members) when various group decision rules are applied (see Fig. 4). For all group decision rules and preference datasets, the accuracy of DeepGroup increases with participation factor κ. This implies that user personal preferences can be predicted more accurately if users participate in more group decisions. We also observe that the accuracy is always the highest in all the datasets for the plurality decision rule whereas Borda has the lowest accuracy. This observation is surprising: despite requiring the least preference data (i.e., only top choice) for decision making, plurality has the highest privacy leakage as the personal preferences can be predicted more accurately when it is deployed. In contrast, Borda has the lowest privacy leakage in this sense. Another important observation emerges from this experiment: when the decision rule is not inconsistent among the groups in a dataset (e.g., the mixture of plurality and Borda), DeepGroup is still effective in predicting the individual preferences.

5 CONCLUSION AND FUTURE WORK

We formulate the problem of group recommendation from group implicit feedback, intending to make item recommendation to a new group of users in the absence of personal user preferences. To address this problem, we introduce DeepGroup. Through extensive experiments, we show the effectiveness of DeepGroup. Our empirical results also show that different group decision rules (e.g., plurality, Borda, etc.) exhibit privacy leakage of concealed personal preferences to various extent. Surprisingly, plurality, despite requiring less information than Borda, suffers more privacy leakage than Borda. For future work, one can theoretically analyze well-known voting rules in the context of our reverse social choice problem to understand their privacy-preserving characteristics when the group decisions are publicly announced. Our DeepGroup model can be improved by incorporating ranking loss functions [20] and deploying more complex latent aggregator functions. Of practical importance is to extend DeepGroup with group and item features (e.g., descriptions or demographic information), side information (e.g., social networks of users), or context (e.g., time or location).

Acknowledgements. This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.* 2, 1 (2009), 754–765.
[2] Kenneth J. Arrow. 1951. *Social Choice and Individual Values*. Yale University Press.
[3] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group Recommendations with Rank Aggregation and Collaborative Filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems* (Barcelona, Spain) *(RecSys'10)*. 119–126.
[4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.
[5] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive Group Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor MI USA). 645–654.
[6] Andrew Crossen, Jay Budzik, and Kristian J Hammond. 2002. Flytrap: Intelligent group music recommendation. In *Proceedings of the 7th international conference on Intelligent user interfaces (IUI'02)*. 184–185.
[7] John A. Doucette, Kate Larson, and Robin Cohen. 2015. Conventional Machine Learning for Social Choice. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 858–864.
[8] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. 2018. *Group Recommender Systems: An introduction*. Springer.
[9] Mike Gartrell, Xinyu Xing, Qin Lv, Aaron Beach, Richard Han, Shivakant Mishra, and Karim Seada. 2010. Enhancing Group Recommendation by Incorporating Social Relationship Interactions. In *Proceedings of the 16th ACM International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) *(GROUP'10)*. 97–106.
[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Proceeding of the Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)* (Long Beach, CA, USA). 1024–1034.
[11] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Wei Cao. 2014. Deep Modeling of Group Preferences for Group-Based Recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1861–1867.
[12] Zhenhua Huang, Xin Xu, Honghao Zhu, and MengChu Zhou. 2020. An efficient Group Recommendation Model with Multiattention-based Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 31 (2020).
[13] David Hughes, Kevin Hwang, and Lirong Xia. 2015. Computing Optimal Bayesian Decisions for Rank Aggregation via MCMC Sampling. In *UAI*. 385–394.
[14] Meir Kalech, Sarit Kraus, Gal A. Kaminka, and Claudia V. Goldman. 2011. Practical Voting Rules with Partial Information. *Journal of Autonomous Agents and Multi-Agent Systems* 22, 1 (2011), 151–182.
[15] Tyler Lu and Craig Boutilier. 2011. Robust Approximation and Incremental Elicitation in Voting Protocols. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*. Barcelona, 287–293.
[16] Judith Masthoff. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction* 14, 1 (2004), 37–85.
[17] Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. 2006. Cats: A synchronous approach to collaborative group recommendation. In *Florida Artificial Intelligence Research Society Conference (FLAIRS) (FLAIRS'06)*. 86–91.
[18] Sebastiano Pizzutilo, Berardina De Carolis, Giovanni Cozzolongo, and Francesco Ambruoso. 2005. Group modeling in a public space: methods, techniques, experiences. In *Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications (AIC'05)*. 175–180.
[19] Vineeth Rakesh, Wang-Chien Lee, and Chandan K Reddy. 2016. Probabilistic group recommendation model for crowdfunding domains. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM'16)*. 257–266.
[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized Ranking from Implicit Feedback. In *Conference on Uncertainty in Artificial Intelligence (UAI)* (Montreal, Canada). 452–461.
[21] Amirali Salehi-Abari and Craig Boutilier. 2015. Preference-oriented Social Networks: Group Recommendation and Inference. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys-15)* (Vienna, Austria) *(RecSys'15)*.
[22] Amirali Salehi-Abari, Craig Boutilier, and Kate Larson. 2019. Empathetic decision making in social networks. *Artificial Intelligence* 275 (2019), 174–203.
[23] Shunichi Seko, Takashi Yagi, Manabu Motegi, and Shinyo Muto. 2011. Group Recommendation Using Feature Space Representing Behavioral Tendency and Power Balance Among Members. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) *(RecSys'11)*. 101–108.
[24] Amartya Kumar Sen. 1970. *Collective Choice and Social Welfare*. North-Holland.
[25] Lirong Xia and Vincent Conitzer. 2008. Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders. Chicago, 202–207.
[26] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys'17)*. 107–115.
[27] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social Influence-based Group Representation Learning for Group Recommendation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (Macua SAR, China). IEEE, 566–577.
[28] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. 2006. TV Program Recommendation for Multiple Viewers based on User Profile Merging. *User Modeling and User-Adapted Interaction* 16, 1 (2006), 63–82.