Learning from Sets of Items in Recommender Systems

Mohit Sharma
University of Minnesota, USA
Email: mohit@cs.umn.edu

F.Maxwell Harper
University of Minnesota, USA
Email: max@umn.edu

George Karypis
University of Minnesota, USA
Email: karypis@umn.edu

***Abstract*—Most of the existing recommender systems use the ratings provided by users on individual items. An alternate source of preference information is to use the ratings that users provide on sets of items. The advantages of using preferences on sets are two-fold. First, a rating provided on a set conveys some preference information about each of the set's items, which allows us to acquire a user's preferences for more items that the number of ratings that the user provided. Second, due to privacy concerns, users may not be willing to reveal their preferences on individual items explicitly but may be willing to provide a single rating on a set of items, since it provides some level of information hiding. This paper investigates two questions related to using set-level ratings in recommender system. First, how users' item-level ratings relate to their set-level ratings. Second, how collaborative filtering-based models for item-level rating prediction can take advantage of such set-level ratings. We have collected set-level ratings from active users of Movielens on sets of movies that they have rated in the past. Our analysis of these ratings shows that though the majority of the users provide the average of the ratings on a set's constituent items as the rating on the set, there exists a significant number of users that tend to consistently either under- or over-rate the sets. We have developed collaborative filtering-based methods to explicitly model these user behaviors that can be used to recommend items to users. Experiments on real data and on synthetic data that resembles the under- or over-rating behavior in the real data, demonstrate that these models can recover the overall characteristics of the underlying data and predict the user's ratings on individual items.**

***Keywords*–Recommender systems; Collaborative filtering; Sets or lists of items; User-behavior modeling.**

I. Introduction

Recommender systems help consumers by providing suggestions that are expected to satisfy their tastes. They are successfully deployed in several domains such as e-commerce, multimedia content providers and mobile app stores. Collaborative filtering [1], [2], which takes advantage of users' past preferences to suggest relevant items, is one of the key methods used by recommender systems.

Most collaborative filtering approaches rely on past preferences provided by users on individual items. An alternate source of preferences is the user's preferences on sets of items. Example of such set-level ratings includes ratings on song playlists, music albums, reading lists, and watchlists. A rating provided by the user on a set of items conveys some information about the user's preference on each of the set's items and, as a result, it is a mechanism by which some information about user's preferences can be acquired for many items. At the same time, due to privacy concerns, users that are not willing to explicitly reveal their true preferences on individual items may provide a single rating to a set of items, since it provides some level of information hiding.

This paper investigates two questions related to using set-level preferences in recommender systems. First, how users' item-level ratings relate to the ratings that they provide on a set of items. Second, how collaborative filtering-based methods can take advantage of such set-level ratings towards making item-level rating predictions.

To answer the first question, we collected ratings on sets of movies from users of Movielens, a popular online movie recommender system [3]. Our analysis of these ratings leads to two key findings. First, for the majority of the users, the rating provided on a set can be accurately approximated by the average rating that they provided on the set's constituent items. Second, there is a considerable user population that tends to consistently either over- or under-rate the set, especially for sets that contain items on which the user's item-level ratings are diverse. Using these insights, we developed different models that can predict a user's rating on a set of items as well as on individual items. These methods solve these problems in a coupled fashion by estimating models to predict the item-level ratings and by estimating models that combine these individual ratings to derive set-level ratings.

The key contributions of the work are the following: (i) collection and analysis of a dataset that contains users' ratings both on individual items and on various sets containing these items; (ii) introduction of *Variance Offset Average Rating Model* (VOARM) to model a user's consistency to over- or under-rate the set as a function of his/her ratings on the set's constituent items; and (iii) development of collaborative filtering-based methods that take advantage of different rating models in order to estimate users' preferences on sets of items as well as on individual items.

The rest of the paper is organized as follows. Section II describes the relevant prior work. Section III describes the dataset creation process along with the analysis of the set ratings in relation to the users' ratings on their constituent items. Section IV presents the methods that we developed to estimate the item-level models from the set ratings. Section V provides information about the evaluation methodology. Section VI presents the results of the experimental evaluation. Finally, Section VII provides some concluding remarks.

II. Related Work

There has been little published work on using set-level ratings to improve the accuracy of item-level recommendations. The one exception is a recent study in which relative preference information on different groups of items was collected during a new user signup process and these preferences were then used to assign a user to a set of pre-built recommendation profiles [4]. This approach significantly reduced the time required to learn the user's preferences in order to generate recommendations for the new user. The principal difference from this approach is that in our work we try to model the user behavior that determines his/her estimated rating on a set and then use that to develop fully personalized recommendation methods that are not limited to new users.

In addition, there has been some work that has focused on recommending lists of items or bundles of items, e.g., recommendation of music playlists [5]–[7], travel packages [8]–[10], reading lists [11] and recommendation of lists under user specified budget constraints [12], [13]. However, these are not directly related to the problems explored in this paper because our focus is on learning the user's ratings on items from ratings on lists of items.

III. MOVIELENS SET RATINGS DATASET

In this section, we will present details and analysis of the ratings elicited from Movielens users on sets of movies. Additionally, we will describe the modeling of users' rating patterns on sets of movies.

A. Data collection

Movielens is a recommender system that utilizes collaborative filtering algorithms to recommend movies to their users based on their preferences. We developed a set rating widget to obtain ratings on a set of movies from the Movielens users. The set rating widget could be rated from 0.5 to 5 with a precision of 0.5. For the purpose of data collection, we selected users who were active since January 2015 and have rated at least 25 movies. The selected users were encouraged to participate by contacting them via email. The sets of movies that we asked a user to rate were created by selecting five movies at random without replacement from the movies that they have already rated. Furthermore, we limited the number of sets a user can rate in a session to 50, though users can potentially rate more sets in different sessions. The set rating widget went live on February 2016 and, for the purpose of this study, we used the set ratings that were provided until April 2016.

B. Data processing

From the initially collected data, we removed users who have rated sets within a time interval of less than one second to avoid users who might be providing the ratings at random. After this pre-processing, we were left with ratings from 854 users over 29,516 sets containing 12,549 movies. Figure 1(a) shows the distribution of the number of sets rated by the users, which shows that roughly half of the users have rated at least 45 sets in a session.

C. Analysis of the set ratings

In order to analyze how consistent a user's rating on a set is with the ratings provided by the user on the movies in the set, we computed the difference of the average of the user's ratings on the items in the set and the rating assigned by a user to the set. We will refer to this difference as *mean rating difference* (MRD). Figure 1(b) shows the distribution of the MRD values in our datasets. The majority of the sets have an MRD within a margin of 0.5 indicating that the users have rated them close to the average of their ratings on set's items. The remaining of the sets have been rated either significantly lower or higher from the average rating. We refer to these sets as the under- and the over-rated sets, respectively. Moreover, an interesting observation from the results in Figure 1(b), is that the number of under-rated sets is more than that of the over-rated sets.

In order to understand what can lead to a set being over- or under-rated, we investigated if the *diversity* of the ratings of the individual movies in a set could lead a user to under- or over-rate the set. We measured the diversity of a set as the standard deviation of the ratings that a user has provided to the individual items of the set. As shown in Figure 1(c), the sets that contain more diverse ratings (i.e., higher standard deviations) tend to get under- or over-rated more often when compared to less diverse sets. This trend was found to be statistically significant (p-value of 0.01 using t-test).

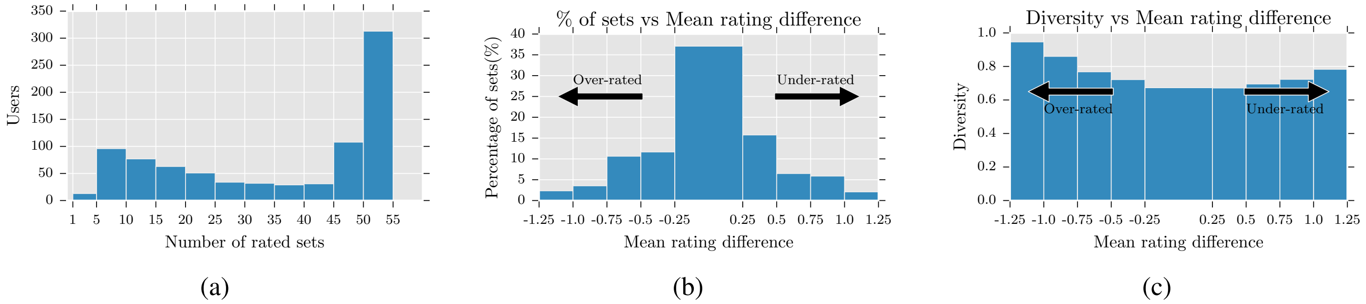
Additionally, we studied if there are users that tend to consistently over- or under-rate sets. To this end, we selected users who have rated at least 50 sets and computed the fraction of their under- and over-rated sets. We also computed the fraction of under- and over-rated sets across a random population of the same size. We generated this random population by randomly permuting the under- and over-rated sets across the users. Figures 2(a) and 2(b) show the fraction of under- and over-rated sets for both the true and random population of users, respectively. In the true population, some users tend to under- or over-rate sets significantly more than that of the random population. Using the Kolmogorov-Smirnov 2 sample test, we found this behavior of true population to be statistically different (p-value $<$ 1e-16) from that of random population.

D. Modeling users' under- and over-rating patterns

The above analysis reveals that our dataset contains users that when they are asked to assign a single rating to a set of items, some of them consistently assign a rating that is lower than the average of the ratings that they provided to the set's constituent items (they under-rate), whereas others assign a rating that is higher (they over-rate). Thus, some users are very demanding (or picky) and tend to focus on the worst items in the set, whereas other users are less demanding and tend to focus on the best items in the set.

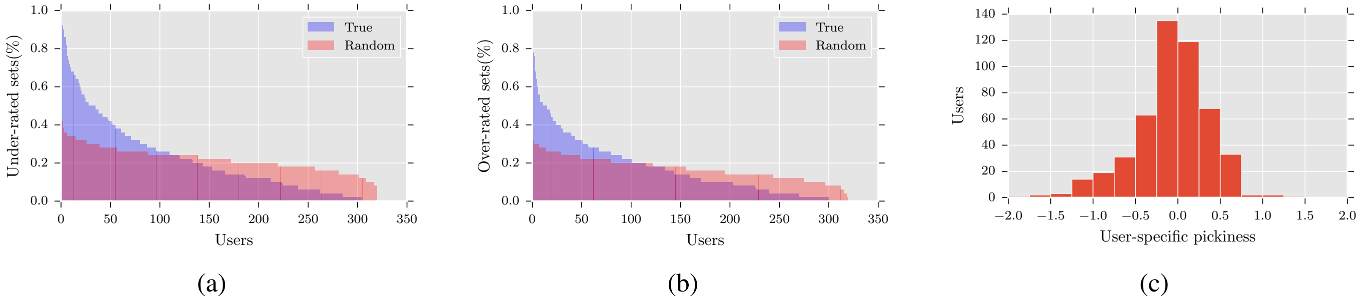
In order to capture this user-specific *pickiness*, we investigated a model that postulates that a user rates a set by considering both the average rating of the items in the set and also the diversity of the set's items. In this model, the set's rating is determined as the sum of the average rating of the set's items and a quantity that depends on the sets diversity (e.g., the standard deviation of the set's ratings) and the user's level of pickiness. If a user is very picky, that quantity will be negative and large, resulting to the set being (severely) under-rated. On the other hand, if a user is not picky at all, that quantity will be positive and large, resulting to the set being (severely) over-rated. We will refer to this model as *Variance Offset Average Rating Model* (VOARM).

In order to determine how well this model can explain the ratings that the users in our dataset provided, we performed the following analysis. We selected the users that rated at least 20 diverse sets (their standard deviation was ≥ 0.5) and for each of these users (493 in total), we computed a user's level



(a) (b) (c)

Figure 1. (a) The distribution of number of sets rated by the users. (b) Histogram of percentage of sets against Mean rating difference. (c) Histogram of diversity against mean rating difference.



(a) (b) (c)

Figure 2. (a) Fraction of under-rated sets across users in the true and random population. (b) Fraction of over-rated sets across users in the true and random population. (c) The number of users and their computed level of pickiness.

of pickiness (β_u) as

$$\beta_u = \frac{1}{n_s} \sum_{s=1}^{n_s} \frac{r_{us} - \mu_s}{\sigma_s}, \tag{1}$$

where n_s is the number of sets rated by user u, r_{us} denotes the rating provided by user u on set $\mathcal{S}$, μ_s is the mean rating of the items in set $\mathcal{S}$ and σ_s is the standard deviation of the ratings of the items in set $\mathcal{S}$. Figure 2(c) shows the histogram of the users' level of pickiness. As can be seen from the figure, certain users tend to under- or over-rate sets with high standard deviation, and interestingly more users tend to under-rate sets than over-rate them.

We computed how well the VOARM compares against the *Average Rating Model (ARM)*, where a user rates a set as the average of the ratings that he/she gives to the set's items. The RMSE of VOARM (0.521) was found to be lower than that of the ARM (0.597), thereby suggesting that modeling users' level of pickiness could lead to better estimates.

IV. Methods

In this section, we describe various methods that use the set ratings alone or in combination with individual item ratings towards solving two problems: (i) predict a rating for a set of items, and (ii) predict a rating for individual items. Our methods solve these problems in a coupled fashion by estimating models for predicting the ratings that users will provide to the individual items and by estimating models that use these item-level ratings to derive set-level ratings.

A. *Modeling users' ratings on sets*

In order to estimate the preferences on individual items from the preferences on the sets, we need to make some assumptions on how a user derives a set-level rating from the ratings of the set's constituent items. Informed by our analysis of the data described in Section III, we investigated two approaches of modeling that.

Average Rating Model (ARM): This approach assumes that the rating that a user provides to a set reflects his/her average rating on all the items in the set. Specifically, if the rating of user u on set $\mathcal{S}$ is denoted by r_u^s and the size of set $\mathcal{S}$ is represented by $|\mathcal{S}|$, then the estimated rating of user u on set $\mathcal{S}$ is given by

$$\hat{r}_u^s = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} r_{u,i}. \tag{2}$$

As the analysis in Section III showed, such a model correlates well with the actual ratings that the users provided on the majority of the sets, especially when the ratings of the constituent items are not very different.

Variance Offset Average Rating Model (VOARM): This approach is based on the VOARM method described in Section III-D. If β_u denotes the pickiness level of user u, then the estimated rating on a set is given by

$$\hat{r}_u^s = \mu_s + \beta_u \sigma_s, \tag{3}$$

where μ_s and σ_s are the mean and the standard deviation of the ratings of items in the set $\mathcal{S}$, respectively. Both μ_s and σ_s

are given by

$$\mu_s = \frac{1}{|S|}\sum_{i\in\mathcal{S}} r_{u,i}, \quad \sigma_s = \sqrt{\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}(r_{u,i}-\mu_s)^2}. \tag{4}$$

B. Modeling user's ratings on items

In order to model a users' ratings on the items, similar to matrix factorization method [2], we assume that the underlying user-item rating matrix is low-rank, i.e., there is a low-dimensional latent space in which both the users and the items can be compared to each other. The rating of user u on item i can be computed as an inner product of the user and the item latent factors in that latent space. Thus, the estimated rating of user u on item i, i.e., $\hat{r}_{u,i}$, is given by

$$\hat{r}_{u,i} = \boldsymbol{p_u}^T\boldsymbol{q_i}, \tag{5}$$

where $\boldsymbol{p_u} \in \mathbb{R}^f$ is the latent representation of user u, $\boldsymbol{q_i} \in \mathbb{R}^f$ is the latent representation of item i and f is the dimensionality of the underlying latent space.

C. Combining set and item models

Our goal is to estimate the item-level ratings by learning the user and item latent factors of Equation 5; however, the ratings that we have available from the users are at the set-level. In order to use the available set-level ratings, we need to combine Equation 5 with Equations 2 and 3. To solve the problem, we assume that the actual item-level ratings used in Equations 2 and 3 correspond to the estimated ratings given by Equation 5. Hence, the estimated set-level ratings in Equations 2 and 3 are finally expressed in terms of the corresponding user and item latent factors.

D. Model learning

The parameters of the models that estimate item- and set-level ratings are the user and item latent vectors (p_u and q_i) and in the case of the VOARM method the user's pickiness level (β_u). These parameters are estimated using the user-supplied set-level ratings by minimizing a square error loss function given by

$$\mathcal{L}_{rmse}(\Theta) \equiv \sum_{u\in U}\sum_{s\in\mathcal{R}_u^s}(\hat{r}_u^s(\Theta) - r_u^s)^2, \tag{6}$$

where U represents all the users, $\mathcal{R}_u^s$ contains all the sets rated by user u, r_u^s is the original rating of user u on set $\mathcal{S}$ and $\hat{r}_u^s$ is the estimated rating of user u on set $\mathcal{S}$.

To control model complexity, we add regularization of the model parameters thereby leading to an optimization process of the following form

$$\underset{\Theta}{\text{minimize}}\, \mathcal{L}_{rmse}(\Theta) + \lambda(||\Theta||^2), \tag{7}$$

where λ is the regularization parameter. The L2-regularization is added to reduce the model complexity thereby improving its generalizability. This optimization problem can be solved by Stochastic Gradient Descent (SGD) algorithm. Also, in the VOARM method we add a fixed constant, i.e., ϵ in [0, 1], to computed σ for robustness.

If we also have ratings for the individual items, then we can incorporate these ratings into model estimation by treating each item as a set of size one.

V. Experimental Evaluation

In this section, we will describe the datasets and the evaluation methodology used to assess the proposed methods.

A. Dataset

We evaluated the proposed methods on two datasets: (i) the dataset analyzed in Section III, which will be referred to as MLRSET, and (ii) a set of synthetically generated datasets that allow us to assess how well the optimization algorithms can estimate accurate models and how their accuracy depends on various data characteristics.

The synthetic datasets were derived from the Movielens 1M dataset [14], which contains 1 million ratings from approximately 6000 users on 4000 movies. We created synthetic low-rank matrices of rank 5, 10 and 20 as follows. We started by generating two matrices $A \in \mathbb{R}^{n\times k}$ and $B \in \mathbb{R}^{m\times k}$, where n is number of users, m is number of items and $k \in [5, 10, 20]$, whose values are uniformly distributed at random in $[0, 1]$. We then computed the singular value decomposition of these matrices to obtain $A = U_A\Sigma_A V_A^T$ and $B = U_B\Sigma_B V_B^T$. We then let $P = \alpha U_A$ and $Q = \alpha U_B$ and $R = PQ^T$. Thus, the final rank k matrix R is obtained as the product of two randomly generated rank k matrices whose columns are orthogonal. Note that the parameter α was determined empirically in order to produce ratings in the range of $[-10, 10]$. We randomly selected 1000 users without replacement from the dataset and for each user we created sets containing five movies. The movies in a user's set were selected at random without replacement from the movies rated by that user. For each user, we created at least 20 and at most 50 such sets of movies. We generated VOARM-based rating for a user on a set by choosing the user's level of pickiness (the β_u parameter) at random from the range of [-2.0, 2.0]. A random $\mathcal{N}(0, 0.1)$ Gaussian noise was added to all item- and set-level ratings. For each rank, we generated 15 different synthetic datasets by varying the user-item latent factors and the users' pickiness levels.

B. Evaluation methodology

To evaluate the performance of the proposed methods we divided the available set-level ratings for each user into training, validation and test splits by randomly selecting five set-level ratings for each of the validation and test splits. The validation split was used for model selection. In order to assess the performance of the methods for item recommendations, we used a test set that contained for each user the items that were not present in the user's sets (i.e., these were absent from the training, test, and validation splits) but were present in the original user-item rating matrix used to generate the sets.

VI. Results and Discussion

The experimental evaluation of the proposed methods is done in two phases. First, we evaluated the performance of the methods using the synthetically generated datasets in order to assess how well the underlying optimization algorithms can recover the underlying data generation models and achieve good prediction performance at either the set- or item-level. Second, we evaluated the performance of the methods on the real dataset that we obtained from a subset of Movielens users (described in Section III).

TABLE I. THE AVERAGE RMSE OBTAINED BY THE PROPOSED METHODS ON SYNTHETIC DATASETS WITH RATINGS IN THE RANGE [-10, 10].

	Rank 5		Rank 10		Rank 20	
Method	Set	Item	Set	Item	Set	Item
ARM	1.206	2.949	1.498	3.545	1.619	3.880
VOARM	1.211	2.372	1.480	2.686	1.597	2.830

Underlined entries indicate the best performing scheme for each experiment.

TABLE II. THE AVERAGE RMSE OF THE PROPOSED METHODS ON SYNTHETIC DATASETS THAT CONTAIN DIVERSE SET OF ITEMS (RANK 5).

	$\sigma \geq 1$		$\sigma \geq 2$		$\sigma \geq 3$	
Method	Set	Item	Set	Item	Set	Item
ARM	1.183	3.057	1.098	3.487	1.140	4.326
VOARM	1.129	2.339	1.068	2.269	1.075	2.507

Underlined entries indicate the best performing scheme for each experiment. Each dataset was generated by keeping only the sets in which the standard deviation of the constituent item ratings (σ) is greater than or equal to the specified value.

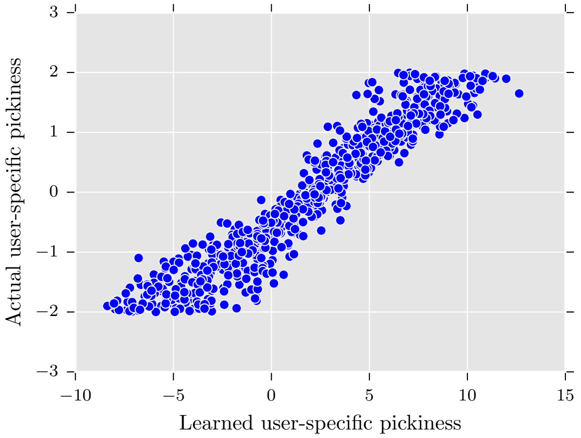


Figure 3. A scatter plot of the estimated and actual parameters that model a user's level of pickiness in VOARM (Rank 5).

TABLE III. AVERAGE RMSE PERFORMANCE OF VOARM WHEN USING ADDITIONAL ITEM-LEVEL RATINGS FROM THE SAME USERS OR A DIFFERENT SET OF USERS (RANK 5).

	set only	+items	+users
Set	1.211	1.190	0.447
Item	2.372	2.169	0.757
MF	—	2.373	—

The entries marked with "—" correspond to combinations that are not applicable.

A. *Performance on the synthetic datasets*

1) Accuracy of set- and item-level predictions: Table I shows the performance achieved by the various methods on the synthetic datasets. In these experiments, ARM acts as a baseline method and its performance relative to VOARM provides insights on the latter's ability to recover the known properties of the underlying data (that this scheme was specifically designed for). These results show that VOARM is able to achieve lower RMSE at the item-level predictions than the corresponding RMSE values obtained by ARM. However, for the set-level predictions, ARM's performance is better than VOARM's for rank 5, but for the greater ranks, i.e., 10 and 20, VOARM performs better than ARM.

In order to study how the performance of the various methods is affected by the diversity of the sets, we followed the approach described in Section V-A to generate a new set of datasets (with rank 5) in which we only kept the sets in which the standard deviation of the set's item ratings is greater than or equal to 1, 2, and 3. The RMSE results that were obtained by the different methods are shown in Table II. These results show that the performance advantage of VOARM over ARM increases with the rating diversity of the items in the sets. This is true for both the set- and item-level predictions.

The results shown in Tables I and II indicate that VOARM is able to recover the known underlying characteristics of the dataset and consequently lead to better prediction performance. To further illustrate this, Figure 3 plots the actual vs estimated weights that model a user's level of pickiness in VOARM (i.e., β_u parameters), which shows that VOARM is able to recover the overall characteristics of the underlying data.

2) Effect of adding item-level ratings: In most real-world scenarios, in addition to set-level ratings, we will also have available ratings on individual items as well, e.g., users may provide ratings on music albums and as well as on tracks in the albums. Also, there may exist some users that are not concerned about keeping their item-level ratings private. To assess how well VOARM can take advantage of such item-level ratings we performed two sets of experiments. In the first experiment, we added in the synthetic datasets a set of item-level ratings for the same set of users for which we have set-level ratings. The number of item-level ratings was kept to 35% of their set-level ratings and the items that were added were disjoint from those that were part of the sets that they rated. Additionally, we used the matrix factorization (MF) method to estimate the user and item latent factors without any set-level ratings by utilizing only the added item-level ratings. In the second experiment, we selected 500 additional users (beyond those that exist in the synthetically generated datasets) and added a random subset of 60 ratings per user from the items that belong to the existing users' sets.

The performance that was achieved by VOARM on these datasets along with the performance in the original set-only dataset is shown in Table III. The "set only" columns show the results of the models that were estimated using only set-level ratings. The "+items" columns show the results of the models that were estimated using the sets of "set only" and also some additional ratings on a different set of items from the same users that provided the set-level ratings. The "+users" columns show the results of the models that were estimated using the sets of "set only" and item-level ratings of a different set of users. We also show the item-level RMSE of the MF models estimated using only the additional item-level ratings from the same users that provided set-level ratings. These results show that by adding these additional item-level ratings

TABLE IV. THE RMSE PERFORMANCE OF THE PROPOSED METHODS ON MLRSET DATASET.

	ARM			VOARM		
	set only	+items	+users	set only	+items	+users
Set	0.633	0.633	0.605	0.632	0.632	0.618
Item	1.082	0.972	0.866	1.005	0.966	0.894
MF	—	1.077	—	—	1.077	—

The meaning of these columns is same as that of Table III.

VOARM's performance improves considerably. Also, VOARM outperforms MF for the task of item-level rating prediction when additional item-level ratings are available for the users. Furthermore, it is promising that when item-level ratings is available for another set of users, the prediction performance for those users for which only set-level ratings is available also improves considerably. Hence, using both item- and set-level ratings can lead to better item recommendations for the users.

B. *Performance on the Movielens-based real dataset*

Our final experiment used the two different methods (ARM and VOARM) to estimate both set- and item-level rating prediction models using the real set-level rating dataset that we obtained from Movielens users. In addition, we assessed how well the proposed methods can take advantage of additional item-level ratings. In the first experiment, we added 20% of the users' set-level ratings as additional item-level ratings and the items that were added were disjoint from those that were part of the sets that they rated. In the second experiment, we added ratings from 500 additional users (beyond those that have participated in the survey), and these users have provided on an average 20,000 ratings for the items that belong to the existing users' sets. The results of these experiments are shown in Table IV.

In the case when we have only set-level ratings, for prediction of item-level ratings, VOARM achieves lower RMSE than ARM. In terms of the accuracy of the set-level predictions, similar to the trends that we observed in the earlier experiments, VOARM does somewhat better than ARM.

For the experiments that include both set- and item-level ratings from the same set of users, we see that performance of both methods improves for item-level predictions. Moreover, VOARM outperforms not only ARM but also MF for item-level predictions. Finally, for the experiments that include set-level ratings of a set of users and item-level ratings from a disjoint set of users we see a significant improvement in performance for both the set- and item-level predictions.

Similar to our results on synthetic datasets, it is promising that the item-level ratings from additional users have significantly improved the performance for the users who have provided only the set-level ratings. The overall consistency of the results between the synthetically generated and the real dataset suggests that VOARM is able to capture the tendency that some users have to consistently under- or over-rate diverse sets of items.

VII. CONCLUSION

In this work, we studied how users' ratings on sets of items relate to their ratings on the sets' individual items. We collected ratings from active users of Movielens on sets of movies and based on our analysis we developed collaborative filtering-based models that try to explicitly model the users' behavior in providing the ratings on sets of items. Through extensive experiments on synthetic and real data, we showed that the proposed methods can model the users' behavior as seen in the real data and predict the users' ratings on individual items.

For future work, we plan to study how the performance of the proposed approaches varies with the different number of items in sets. Furthermore, it will be interesting to investigate if, similar to the diversity of ratings in sets, there exist other properties at item-level or set-level that can affect a user's ratings on sets of items.

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.

[2] Y. Koren, R. Bell, C. Volinsky et al., "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, 2009, pp. 30–37.

[3] "Movielens recommender system," 2017, URL: http://www.movielens.org/ [accessed: 2017-02-26].

[4] S. Chang, F. M. Harper, and L. Terveen, "Using groups of items for preference elicitation in recommender systems," in Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work &#38; Social Computing, ser. CSCW '15. New York, NY, USA: ACM, 2015, pp. 1258–1269. [Online]. Available: http://doi.acm.org/10.1145/2675133.2675210

[5] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims, "Playlist prediction via metric embedding," in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 714–722. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339643

[6] N. Aizenberg, Y. Koren, and O. Somekh, "Build your own music recommender by modeling internet radio streams," in Proceedings of the 21st international conference on World Wide Web. ACM, 2012, pp. 1–10.

[7] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, "Learning to embed songs and tags for playlist prediction." in ISMIR, 2012, pp. 349–354.

[8] R. Interdonato, S. Romeo, A. Tagarelli, and G. Karypis, "A versatile graph-based approach to package recommendation," in 2013 IEEE 25th International Conference On Tools with Artificial Intelligence. IEEE, 2013, pp. 857–864.

[9] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu, "A cocktail approach for travel package recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 2, Feb 2014, pp. 278–293.

[10] M. Xie, L. V. Lakshmanan, and P. T. Wood, "Comprec-trip: A composite recommendation system for travel planning," in Data Engineering (ICDE), 2011 IEEE 27th International Conference on. IEEE, 2011, pp. 1352–1355.

[11] Y. Liu, M. Xie, and L. V. Lakshmanan, "Recommending user generated item lists," in Proceedings of the 8th ACM Conference on Recommender Systems, ser. RecSys '14. New York, NY, USA: ACM, 2014, pp. 185–192. [Online]. Available: http://doi.acm.org/10.1145/2645710.2645750

[12] M. Xie, L. V. Lakshmanan, and P. T. Wood, "Breaking out of the box of recommendations: from items to packages," in Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010, pp. 151–158.

[13] I. Benouaret and D. Lenne, "A package recommendation framework for trip planning activities," in Proceedings of the 10th ACM Conference on Recommender Systems, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 203–206. [Online]. Available: http://doi.acm.org/10.1145/2959100.2959183

[14] "Movielens 1M dataset," 2017, URL: https://grouplens.org/datasets/movielens/1m [accessed: 2017-02-26].