Precedence Mining in Group Recommender Systems

Venkateswara Rao Kagita, Vineet Padmanabhan, and Arun K. Pujari

School of Computer and Information Sciences,
University of Hyderabad, India
venkateswar.rao.kagita@gmail.com, {vineet,akp}cs@uohyd.ernet.in

Abstract. We extend the `Precedence mining` model for *personal recommendation* as outlined in Parameswaran et.al., [6] in three different ways. Firstly, we show how precedence mining model can be used for recommending items of interest to a `group` of users (*group recommendation*) and compare and contrast our model with traditional group recommendation models like *collaborative* and *Hybrid*. Secondly, we extend the precedence mining model to incorporate `ratings` for items and experimental results show that the *goodness* of recommendation is improved. The third extension is related to the issue of *new items* being ignored which is a fundamental problem plaguing collaborative and precedence mining algorithms. When recommendations are based on other users interests (like in Collaborative recommender systems) the possibility of not recommending a new item which has not been consumed by many of the users is high though the new item may be of interest to the target user. We outline two models, `Vector precedence-mining` and `Hybrid precedence-mining` that addresses this issue.

1 Introduction

Collaborative filtering [2] is one of the commonly used methods for developing personal as well as group recommender systems [1,8,5,3,4]. Two main techniques used in collaborative filtering are called *Memory-based* and *Model-based*. Recently, Parameswaran et.al., [6] proposed a precedence mining approach for personal recommender system. The nicety and novelty of this approach is the use of pairwise precedence relations between items. Most of the earlier approaches of Collaborating Filtering do not take into consideration the relationship between objects. Though precedence mining, as demonstrated by Parameswaran et. al., [6], can be an important approach for recommender systems, current research is mostly limited to personal recommendation.

In this work we propose a group recommender system based on precedence mining model. Based on the precedence relations between pairs of items, a precedence probability of consuming an item by a user is calculated. The item with highest precedence probability for a user is recommended in a personal recommender system. We extend this concept of precedence probability to a group by introducing a virtual user. A virtual user, in principle, represents all users of the

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 701–707, 2013.

group. We derive the precedence probability of the virtual user of a group from precedence probabilities of individual members of the group. The group recommender system is extended further by incorporating users' ratings of items and by considering recommendation of unknown (new) item. The group recommender methods, proposed here, are experimented with real-life data on a movie database and is demonstrated that the user's satisfaction is substantially higher than earlier techniques.

2 Recommender Systems

The problem of recommender system is defined as follows. Let $O = \{o_1, o_2, \ldots, o_n\}$ be the set of items and $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users. $profile(u_j)$ is the sequence of items consumed by user u_j. For given O, U, k and user $u \in U$, the Personal Recommender Problem (PRP) is to recommend k items to be consumed by user u. The recommended k items are presumed to be absent in $profile(u)$. The Group Recommender Problem (GRP) is to recommend k items to a group $G \subseteq U$ for given O, U and k. Any recommender system (personal or group) aims at selecting items for recommendation such that these items are expectedly preferred to other items by the user for whom it is recommended. Identifying most probable items from the set of profiles for a group of users (GRP) is harder than the same for individual users (PRP).

In general, collaborative filtering techniques find users having similar profiles as $profile(u)$ and then restricts its search to items consumed by this subset of users and not consumed by u. Thus certain patterns of consumption of items exhibited by the whole set of users, U, is not captured as the search is restricted. Precedence mining gets over this shortcomings and attempts to capture pairwise precedence relation occurring frequently among all users. Precedence mining approach as proposed by Parameswaran et.al. [6] is limited to recommending items for individual users. We, in this paper, propose a novel technique of use of precedence mining for group recommendation.

2.1 Precedence Probabilities

Given an item o_i, $support_i$ is the number of users who consumed item o_i. We define p_{ij} as the number of users having item o_i preceding o_j in their profiles. The precedence probability for items o_i and o_j, denoted as $PP(o_i|o_j)$ represents the probability of o_i preceding o_j. We define $PP(o_i|o_j) = \frac{p_{ij}}{support_i}$. There may arise cases where p_{ij} is zero and hence $PP(o_i|o_j)$ becomes zero. In order to avoid this situation we perturb PP by adding 1 to both numerator and denominator. Thus when p_{ij} is zero, we have $PP(o_i|o_j) = \frac{p_{ij}+1}{support_i+1}$. Let CI_j be the set of items consumed by user $u_j \in U$. Then we define

$$Score(o_i, u_j) = const \times \frac{support_i}{n} \times \prod_{o_l \in CI_j} PP(o_l|o_i) \tag{1}$$

For a group $G \subseteq U$, a trivial way of computing the score is to aggregate scores of individual members [9,7]. We propose a novel method of virtual user strategy. A virtual user $v(G)$ represents the whole group G. $profile(v(G))$ is generated from $profile(u_j), u_j \in G$. For group G, the set of consumed items CI_G is $\bigcup_{u_j \in G} CI_j$. We define weight for an item consumed by an individual user as

$$weight(o_i, u_j) = \begin{cases} 1 & \text{if } o_i \in CI_j \\ score(o_i, u_j) & otherwise \end{cases}$$

We define weight of an item for group G as

$$weight(o_i, G) = \sum_{u_j \in G} weight(o_i, u_j)/|G|$$

An item $o_i \in profile(v(G))$ if $weight(o_i, G) \geq \tau$ where τ is a predefined threshold. Algorithm 1 gives pseudo-code of constructing $profile(v(G))$.

Algorithm 1. Virtual User Strategy in Probabilistic Model

Input: O, U, $profile(u, \forall u \in U)$, G, τ
Output: Recommended Item(s)
$profile(v(G)) = \emptyset$;
for *each item* $o_i \in CI_G$ **do**
  Calculate $weight(o_i, G)$;
  if $weight(o_i, G) > \tau$ **then** $profile(v(G)) = profile(v(G)) \cup o_i$;
  delete o_i from CI_G
for *each item* $o_i \in O \setminus CI_G$ **do** Calculate $score(o_i, v(G))$;
Recommend top k items that maximizes $score(o_i, v(G))$.

2.2 Precedence Mining Model with Ratings

In many real life situations a user not only consumes an item, but also gives a satisfaction rating for the item. A group recommendation system becomes more meaningful if such ratings are also incorporated in the user profile. We assume here that $profile(u_j)$ is a sequence of ordered pairs (o_i, r_i) indicating the item o_i consumed by user with a rating of r_i. In this situation we propose a new method of computing the precedence probability for a pair of items. There is a lower bound t such that cases with $r_i \geq t$ are only considered. We define $p_{ij}(r, t)$ as number of user profiles in which item o_i precedes item o_j with $r_i = r$ and $r_j \geq t$. Similarly $support_i$ is number of user profiles that contain item o_i with rating $r_i \geq t$. $PP(o_i : r|o_j) = \frac{p_{ij}(r,t)}{support_i}$ and

$$Score(o_i : r, u_j) = const \times \frac{support_i}{n} \times \prod_{o_l \in CI_j} PP(o_l : r|o_i) \qquad (2)$$

In Parameswaran et.al., [6] incorporating ratings into the precedence mining model was left out as part of future work.

3 Vector Precedence Mining

Majority of the group recommender systems try to recommend an item o_i belonging to the set of known items O. There are situations when a new item is introduced in the market and it becomes necessary to consider recommending this new item. One of the drawbacks of the existing precedence mining models is their inability to incorporate new items. To handle such situations, $profile(u_j)$ is extended as a sequence of vectors representing multiple features of items. Thus instead of counting the number of occurrences of an item, the number of occurrences of feature-set is counted. Therefore an unknown item which is similar in features to any of the known items could be treated as an identical item. For the case of movies, each movie is represented as a set of genres like *Action, Comedy* etc.[1] $profile(u_j)$ is a binary vector with 1 if a genre is present in the movie, otherwise zero. We take the decimal equivalent of that binary vector and represent the movie with that value.

3.1 Social Influence

Many persons, often, like an item because it is associated with persons they like. In the case of a movie recommendation scenario, people watch movies of their favorite star (hero, heroine, director, etc.,) irrespective of the content of the movie and they also watch movies of the star who is having similar characteristics or relevance to their favorite star. It is important to consider such *social influence factors* in a group recommender system. Our experimental results show that the *goodness* of recommendation is improved while taking into consideration such social factors with respect to a movie dataset. In our experiments, we augment $profile(u)$ by taking into consideration factors like actor-id, director-id etc.

3.2 A Hybrid Technique Based on Precedence Mining

In section3 we outlined a vector representation model to overcome the issue of new items not being recommended to the target user/s. We propose a hybrid technique that combines both precedence mining and content based recommendations. Our idea is as follows. First apply precedence mining technique and get top $2k$ (where k is the number of items to be recommended) items. Having done that get score for new items. Then apply content based technique and get score for these top $2k$ and new items. Add these two scores and re-rank according to the new score.

4 Experiment Results

The concepts introduced in the above sections are validated in a movie recommendation scenario. Our dataset consists of 100 users and 453 movies. User profiles with ratings were created by consulting these individuals. Our recommender

[1] In the data-set used for our experiments each movie is represented with 19 genres.

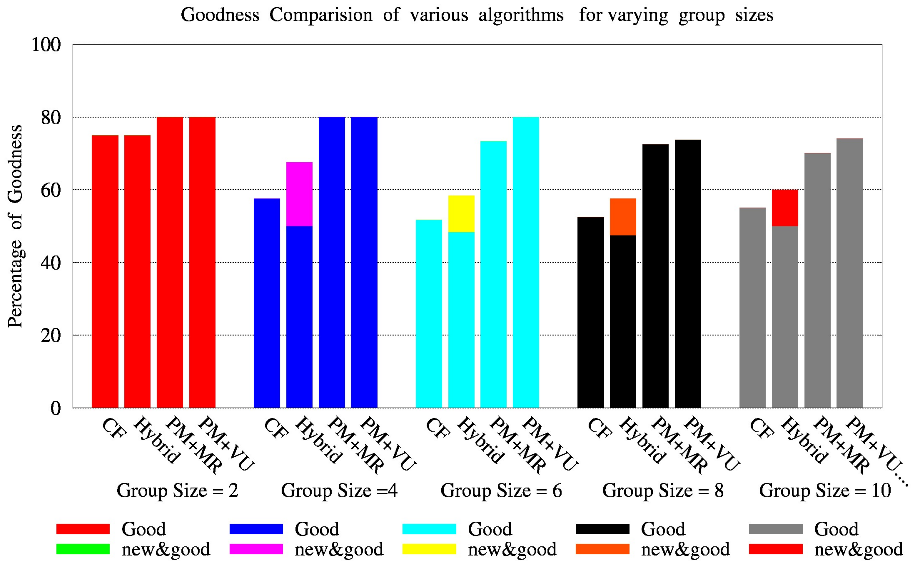

Fig. 1. Goodness comparison Collaborative and Hybrid Vs. Precedence Mining

algorithms are validated by collecting users' satisfaction on the outcomes of our algorithms as well as on the outcomes of collaborative and Hybrid algorithms. We presented the recommendations of all the algorithms (each movie only once) and asked them to give their satisfaction on that recommendation in a 1-5 scale. (Note here that the satisfaction may depend on many parameters such as theme of the movie and surprise about recommendation and many more). User/s satisfaction rating exceeding 4 are only considered to calculate the percentage of goodness as shown in equation 3 wherein $response_{ij}$ is response given by user $u_i \in G$ for movie m_j

$$percentage\ of\ goodness = \frac{\sum_{i=1}^{|G|} \sum_{j=1}^{k} response_{ij} \geq 4?1:0}{k \times |G|} \times 100 \quad (3)$$

First set of experiments are to compare the base models of precedence mining with the traditional collaborative and hybrid techniques for different group sizes. Figure 1. shows the results wherein CF represents collaborative filtering, PM+MR represents precedence mining with merging results strategy and PM+VU represents precedence mining with virtual user strategy. We observe that in Figure 1. newness is completely missing in collaborative and precedence mining models. There is an overall improvement of goodness in precedence mining models compared to collaborative and hybrid models. PM+VU has consistently better performance than PM+MR. In order to calculate *new&good* movies we consider only recently released movies from the recommendation set whereas

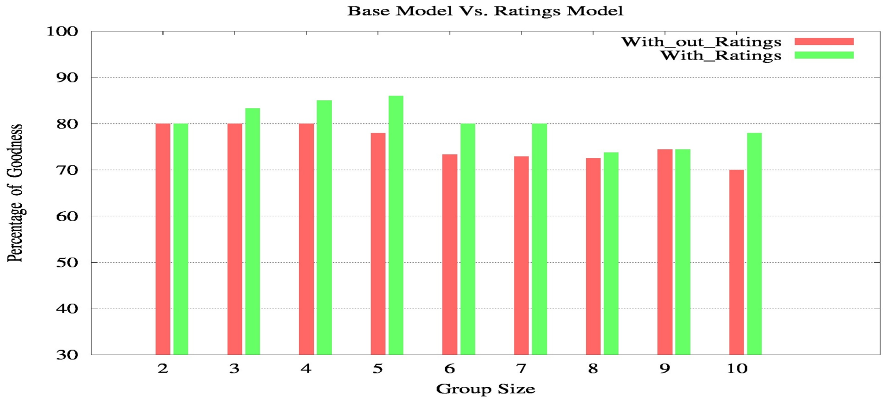

Fig. 2. Performance difference with and without incorporating ratings

to calculate *Good* we use the remaining movies in the recommendation set. The second set of experiments are related to the improvement of goodness by incorporating ratings into the precedence mining model (as described in section 2.2). This set of experiments are carried out by averaging the individual scores and not by virtual user strategy. Figure 2. shows these results comparison. If we observe

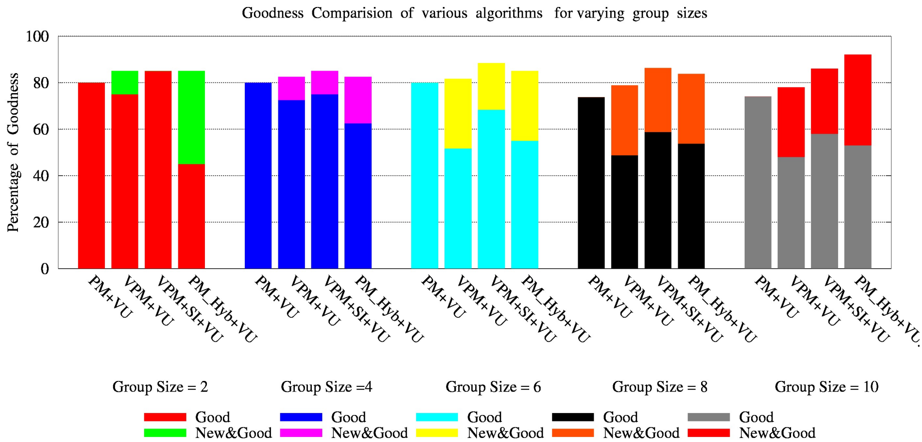

Fig. 3. Performance Improvement with vector precedence mining and social influence

the graph in Figure 2. incorporating ratings into the precedence mining model is performing better than without ratings most of the time or it is guaranteed that incorporating ratings is at-least as good as without incorporating ratings. If we observe Figure 1., Virtual User strategy is performing better than Merging Results strategy in most of the cases. So we have used virtual user strategy in vector-precedence mining model and social influence model as outlined in section 3. Results related to `newness` improvement with vector-precedence mining and `goodness` improvement by considering social influence factors such as *actor* in movie domain (described in section 3.1) is shown in Figure 3. VPM+VU in this figure represents vector precedence mining with virtual user strategy, VPM+SI+VU represents vector precedence mining and social influence with virtual user strategy and PM_Hyb+VU represents precedence mining hybrid model with virtual user strategy. The dataset related to our experimental results can be found at `http://dcis.uohyd.ernet.in/~vineetcs`.

5 Conclusions and Future Work

In this work we have developed a framework for group recommender systems (GRS) based on precedence mining. We outline two models which has the advantage of extracting more frequent patterns within the whole set of user profiles as compared to other GRS based on collaborative filtering. Our experimental results show that both *goodness* as well as *newness* of recommendation is improved using our group recommender algorithms. In the future, we would like to extend our framework to include rating prediction for virtual user items as well as to provide justification for group recommendations.

References

1. Amer-Yahia, S., Roy, S.B., Chawla, A., Das, G., Yu, C.: Group recommendation: Semantics and efficiency. PVLDB 2(1), 754–765 (2009)
2. Breese, J., Heckermanx, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI (1998)
3. Chen, Y.-L., Cheng, L.-C., Chuang, C.-N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. 34(3), 2082–2090 (2008)
4. Padmanabhan, V., Kiran, P., Sattar, A.: Group recommender systems: Some experimental results. In: 5th International Conference on Agents and Artificial Intelligence (ICAART 2013), Barcelona, Spain, pp. 370–376 (2013)
5. Padmanabhan, V., Seemala, S.K., Bhukya, W.N.: A rule based approach to group recommender systems. In: Sombattheera, C., Agarwal, A., Udgata, S.K., Lavangnananda, K. (eds.) MIWAI 2011. LNCS, vol. 7080, pp. 26–37. Springer, Heidelberg (2011)
6. Parameswaran, A.G., Koutrika, G., Bercovitz, B., Garcia-Molina, H.: Recsplorer: recommendation algorithms based on precedence mining. In: SIGMOD Conference, pp. 87–98 (2010)
7. Seko, S., Yagi, T., Motegi, M., Muto, S.Y.: Group recommendation using feature space representing behavioral tendency and power balance among members. In: RecSys, pp. 101–108 (2011)
8. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Providing justifications in recommender systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A 38(6), 1–1272 (2008)
9. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Modeling and User-Adapted Interaction 16(1), 63–82 (2006)