Experience-based Personalized Diversification of Recommendations

K.L. Jayaratne
University of Colombo School of Computing
University of Colombo
Colombo 07, Sri Lanka
klj@ucsc.cmb.ac.lk

***Abstract*— Accuracy of the recommendations has long been regarded as the primary quality aspect of Recommender Systems (RS), but there's an increasing cognizance that there are other factors such as diversity that users also value. Despite the increased interest of researchers to improve diversification of recommendations, we find that personalization of diversification has been overlooked. As the preference for diversity changes from person-to-person, we propose a personalized diversification technique which is capable of controlling the trade-off between accuracy and diversity, where personalization is achieved by diversifying the recommendation list with more novel items if the user has shown diverse preferences in the past, and diversifying the recommendation list with more relevant items if the user has shown homogeneous preferences in the past. Moreover, we also introduce a novel recommendation technique which uses the past preferences of a user and the ratings of experienced item category experts in recommendation generation process. As post-filtering approaches generate the final diversified recommendation list by selecting items from a list generated from some RS, we use the recommendation technique we propose in order to generate an initial recommendation list with both novel and relevant items to improve the personalized diversification process. Our experiments and evaluation provides evidence to illustrate the properties of proposed techniques and indicate the proposed approach has comparable results to state-of-art techniques. Moreover, unlike other techniques, our approach can promote both novel and relevant items and also make the diversification process personalized.**

Keywords—Recommender Systems; diversity; personalization; novelty; experts;

I. Introduction

Information accessible to people has increased with the advancement of technology. But people face hard times when choosing something relevant from this vast amount of available information. Recommender Systems (RS) are software tools and techniques that assist people in choosing something useful [1] from such vast amount of information by providing worthwhile and relevant suggestions and saving the time in decision making.

Recommendations to a user are provided based on the past preferences of users. Hence, an item recommended to a user is quite similar to the items the user had shown preferences for in the past and this increases the accuracy of the RS. But solely focusing on improving the accuracy of recommendations is not sufficient to achieve user satisfaction [2]. There are other factors such as novelty and diversity users expect [2]. For example, in Amazon a customer who has purchased many proses of Hermann Hesse may obtain a recommendation list where all top five books would be of that author only [3]. When pure accuracy is considered, this list is optimal as all the recommendations are relevant to the user as indicated by his past preferences. But if that user had other interested authors as well, the recommendation list would disappoint the user as the recommendations are homogenous and lacks diversity. One reason for this hindrance is that the recommendation list is not considered as an individual entity rather it's considered as an aggregation of individual items [2]. Hence, the quality of RS is assessed based on the relevance of each individual items in the recommendation list and it cannot capture that the list might be monotonous which will lead to less user satisfaction.

This is where *Diversity* comes into play. Diversity refers to how different the items in a list or a set are, with respect to each other [4] based on the properties of the items. For example, a list of books of different authors is a diverse list compared to a list of books of a single author. Providing a diverse list of recommendation instead of homogenous alternatives has various advantages [5]. Those can be explained as follows. The main utility of a RS is to help a user to find new unknown and relevant items. Thus if the list of recommendations contains highly similar items, the usefulness of the recommendations might reduce. But with diversity the users will be able to explore unknown and a wide variety of items in the items space. Also, since it's not feasible to perfectly capture the complete spectrum of user's preferences, as it's difficult to get user's preferences to all the items in the item space of the RS, it's better to generate the recommendation list with diverse items as then the user would find at least one item useful. But if the list is highly monotonous and the user dislikes one item, it would make the whole list useless as the items are similar with each other. A diversified list would also be helpful to cover most of the interests of the user. It should be noted that there's a trade-off between accuracy and diversity. Although both are desired features, it is impossible to achieve both at the same time. Hence techniques to balance these competing objectives are needed for the advancement of RS.

DOI: 10.5176/2251-3043_5.3.377

Although various approaches have been proposed in the literature to diversify the recommendations, most of the diversification strategies are not personalized. Hence diversification is carried out in the same manner for all users. Although diversity is a desired feature, the desire for diversity may vary from person-to-person. Some may be willing to explore novel items whereas some may prefer to stick to only a certain set of item categories [5].Thus, we propose a technique to personalize the diversification process in order to diversify the recommendation list of a user with more novel items that are different from his past preferences, if the user has shown high diversity in his past preferences and to diversify the list with more items that are similar to his past preferences if the user has shown less diversity and have only few categories in his preferences.

In order to further support the diversification process we also propose an experience-based recommendation technique. If a RS uses only the user's past preferences or the preferences of similar users, it's not easy to add novel items to the recommendation list that are different from user's past preferences and it will negatively affect diversification and personalized diversification. In some item domains there are experienced experts that have much higher experience with certain items categories. Hence our technique uses user's ratings along with experts' ratings to allow the recommendations to include novel items which are rated by experts along with items that are similar to the items the user had preferred in the past. Since these novel items are rated by experts, there's a lesser risk of that item being an unsatisfactory item to the user. We report experiments to show that the proposed techniques perform well with respect to diversity and relevance.

The remainder of the paper is structured as follows. In section I and II we provide the background and related work. In section III we further explain the proposed approach. Next in section IV we provide the experiments we carried out and finally in section V we provide final summary and future directions of this work.

II. Background

RSs generate recommendations to a user based on the explicit preferences such as ratings and likes given to items and implicit preferences such as purchasing an item, which are the preferences indirectly indicated. The preferences are maintained in a user profile and are used as input to generate recommendations to users. Content-based RSs and user-based collaborative filtering (*UserCF*) are some of the prominent traditional recommendation techniques. In content-based RSs, an item is recommended to a user, if that item is similar to the items the user has preferred in the past. For example, an Action movie is recommended to a user if the movies that user has preferred in the past indicates he prefers Action movies. *UserCF* is based on the nearest neighbor concept where a user *u* becomes a neighbor of another user *v*, if *u* and *v* have similar past preferences. Thus, in *UserCF* an item is recommended to a user based on the ratings the neighbors have given to that item. Predicting the rating a user would give to an unseen item and producing a ranked list of items to be recommended based on the predicted user preference are some of the tasks a RS carries out [6].

In this section we provided some basic concepts in RSs that are related to our research. In the next section we discuss the work related to diversification.

III. Related Work

Traditionally, Recommender Systems are focused on improving the accuracy of individual recommendations. Hence the RSs try to improve the accuracy of rating prediction and to recommend items that are highly relevant to the user based on the user's past preferences. One of the main reasons for researchers to focus on the accuracy of recommendations is that RSs are generally evaluated specifically with accuracy focused metrics [7]. Traditional Content-based RSs suffer from overspecialization which tends to recommend items that are too similar to the items user has previously rated. Due to this, the recommended items are not diverse from the items the user had previously rated, although the recommendations have a higher level of accuracy. Although *UserCF* can recommend more novel items, compared to content-based approach, it also has diversity issues. Since *UserCF* recommends items with highest ratings from similar neighbors, the recommendations can be biased towards popular items. Also if the neighbors' preferences are highly similar to the user's preferences, the items recommended from neighbors will also be similar to the user's past preferences and will not be much novel. Also if the neighbors are also similar with each other, the recommendation list generated from the items preferred by neighbors will lack diversity as the items can be similar with each other.

Since RSs are aimed at improving the accuracy of recommendations, a recommendation list is generated by adding the items with highest predicted rating and ranking them in the list based on this predicted rating, focusing on the accuracy of individual items [8]. But since the recommendation list is considered as an aggregation of individual items rather than a single entity, it isn't considered whether the items in the recommendation list are similar with each other. Hence it is not possible to achieve diversity in the list. For example, consider a list with five books of a user's favorite author. When these five items are considered individually, each of them has a higher predicted rating, thus, high accuracy. But when the complete list is considered, it doesn't provide much new information to the user as all the items are similar to each other and lacks diversity.

It is apparent from the above facts that the main reasons for lack of diversity in recommendations is that the traditional evaluation metrics are solely focused on accuracy and that the recommendation lists are not considered as an entity in their own right, rather they are considered as an aggregation of individual items which doesn't indicate the real worth of the item with respect to the list.

But the work in [2] report how accuracy metrics have hurt RSs and it has also identified that there are properties different from accuracy such as novelty and diversity which are concerned with user satisfaction in RSs [7]. Due to the heavy focus on accuracy, a recommendation list presented to a user often includes homogenous items that are predicted to be highly relevant to the user. But when the recommended items are too similar with each other, the benefit users obtain becomes limited. *Individual diversity* refers to recommending a diverse set of items to a single user instead of providing a list of homogenous items. If the list consisted of homogenous items, for example books from the user's favorite author, it might give low benefits to the user. One reason for this consequence is that, if all books are from the user's favorite author, the user may have already heard of at least some of those books [3]. Hence the user doesn't receive much new information from the RS. Also since it's not possible to capture all the preferences of the user from the large item space, some uncertainty lies in the user profile [9]. Thus, recommending homogenous items would be risky as the complete list would be useless if the user dislikes one item in the list and a list of diverse items would be more appropriate. Recommending novel items that are different from a user's past preferences would help the user to discover new products and would also facilitate the RS to improve the user profile from user's reaction to that item [5]. The importance of diversity in the recommendations has been emphasized by several researchers in the RS literature [4] [10] and has also been validated with user studies [11]. The user study done in [11] provides valuable evidence on the necessity for diversity in recommendations where a majority of users have highlighted diversity as a desired feature in recommendations. With respect to all the mentioned facts from literature, it can be concluded that diversifying recommendations is an essential requirement in the field of RSs.

It should also be noted that there's a trade-off between diversity and relevance and it's not possible to maximize the diversity of the list and maximize the relevance of items in the list at the same time. Several approaches are proposed in the literature to control the trade-off between diversity and relevance in recommendations and achieve a proper diversification strategy. Some of these approaches modify the RS itself and other approaches suggest post-filtering strategies. Post-filtering strategies generate a larger list of recommendations from some RS and select k items from the list according to some maximization criteria and generate the final recommendation list. Post-filtering strategies are greedy and are proposed because directly trying to maximize diversity is considered a NP-complete problem [12]. Most of these post filtering approaches [12][3][10] are inspired from Maximal Marginal Relevance(MMR) criteria [13], given in (1), used in Information Retrieval. In (1), $sim_1(D_i,Q)$ represent the similarity or relevance of document to the target query and $sim_2(D_i,D_j)$ is the similarity between two documents and λ controls the trade-off between relevance and diversity.

$$MMR = arg \max_{D_i \in R \setminus S} [\lambda sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} sim_2(D_i, D_j)] \quad (1)$$

We can summarize some of the post-filtering diversification approaches as follows. The work in [3] presents a topic diversification approach to diversify recommendation lists focusing on covering the wide spectrum of user interests of topics. It carries out a post-filtering approach using lists generated from item-based collaborative filtering. B. Smyth and P. McClave [10], in their algorithm, "Diversity 3 - Bounded Greedy Selection", also perform a post-filtering approach in a case-based RS. A recent work in [14] introduces a post- filtering binomial framework for genre diversity of recommendation lists. G. Adomavicius and Y. Kwon [15] also suggest a post-filtering approach based on neighbors' rating variance in collaborative filtering. A. Pathak and B. K. Patra [16] propose a clustering based post-filtering approach for diversification. However it doesn't use any objective function. G. Adomavicius and Y. Kwon [17] also state that it is possible to take item re-ranking approaches (post-filtering) in order to diversify recommendation lists as these techniques are efficient and also flexible to be used with any existing commercial RS.

However, in these earlier approaches diversification is not personalized, as the same diversification process is carried out for each user. But the desire for diversity, changes from person-to-person. Some people may prefer diversity and some may not. Even who prefers diversity may have different levels of diversity needs. Some user surveys [18] [19] indicate that the need for diversity changes according to the personality. Work in [20] has considered personalization and have diversified a user's recommendation list based on the category the user had shown more diversity towards in the past. As identified by A. Barraza-Urbina, B. Heitmann, C. Hayes, and A. Carrillo-Ramos [5], exploration and exploitation are two aspects of diversity. S. Vargas and P. Castells [21] have shown an intent-aware diversification strategy. But to the best of our knowledge earlier diversification approaches have not considered personalized diversification which considers, if the user has shown diverse preferences in the past, he may prefer a diverse list with more novel items whereas a person who has shown less diversity in the past where most of his preferences are focused on few item categories, may prefer a diverse list with more items similar to his past preferences. Although work in [20] analyses the past behavior, that approach only identifies which category based the list should be diversified and doesn't consider whether the list should be diversified based on novelty or relevant items. XPLODIV framework [5] is proposed to control exploration and exploitation but it has not considered users past behavior to tune these parameters. Thus we identify a limitation in the diversification literature related to personalization.

In post-filtering diversification approaches, the achievable diversity of the final recommendation list is dependent on the candidate item list generated from some existing RS. But due to heavy focus on accuracy, traditional RSs vaguely promote diversity as items are only recommended if the items are similar to a user's past preferences. Hence, it is beneficial to identify means to promote novel and diverse items from the RS used in post-filtering approaches. When a certain item domain is considered, one can find experienced experts in that domain who has high experience on some item categories whom a user can get good recommendations from. For example, in movie domain a user who has high experience in Romance movies can give good recommendations for Romance movies. Thus, it is safe to recommend an item which is not similar to user's preferences if the experts have rated it positively. This will be helpful to achieve diversity in the final recommendation list. Therefore, it'd be beneficial to consider these expert's opinions along with the user's opinion in order to generate recommendations as it will help to carry out the diversification process in a personalized manner, by promoting relevant and novel items in the recommendation generation. Thus we believe it's important to identify means to find experts based on experience in an item domain and to see how to make use of their ratings in the recommendation phase along with user's preferences in order to further improve personalized diversification.

In this section we explained why diversity is a desired property in RSs and why it is lacking. We reviewed related work of diversification and identified that personalization of diversification is not properly addressed. In the next section we present the approach we propose to carry out the diversification process in a personalized manner.

IV. Methodology

In this section we introduce our post-filtering diversification technique consisting of two main modules, namely, *ExpertRec* and *PersonalizedDiv*. Overall architecture of our proposed approach is presented in Fig.1. Post-filtering approaches selects a subset of items according to a given criteria from a recommendation list generated from some RS. We use *ExpertRec* to generate a recommendation list to serve as the input to *PersonalizedDiv* in order to generate the final recommendation list. Since our solution is a post-filtering approach, the Initial Recommendation List can be generated from any available recommendation technique such as *UserCF* as well. Although we explain our approach using examples from movie domain it should be noted that this approach is applicable to any item domain which has a form of item categorization. For example, movies can be categorized based on genre.

A. Notations

We will first present some notations we'll be using in our design. Rating a user u has given to item i, is given by $r_{u,i}$. User profile of user u is given by U_u and this set includes all the items u has rated. The set of categories of item j is given by I_j ,for example the set of genres of a movie j. Probability of how much user u prefers category c is given by $C_{u,c}$ and C_u indicates the category preference profile of u which contains how much u prefers each item category ($C_{u,c} \in C_u$).

B. Expert-based Recommender Engine

Expert-based Recommender Engine (*ExpertRec)* is a recommendation technique which predicts ratings for items and generates a recommendation list based on both past preferences of the user and the ratings of experts. Experts are defined as the users who have higher experience in a certain item category. For example, in movie domain, a user who has rated about hundred Romance movies has higher experiences and knowledge in Romance category than a user who has rated about hundred movies of a wide variety of genres including Romance genre. The reason is, since the all the movies the first user has rated are Romance movies, the user has more experiences in Romance movies and that experiences and knowledge lead him to be an expert in Romance movies. But the second user has only few experiences with Romance movies. *ExpertRec* also considers how relevant an item is to

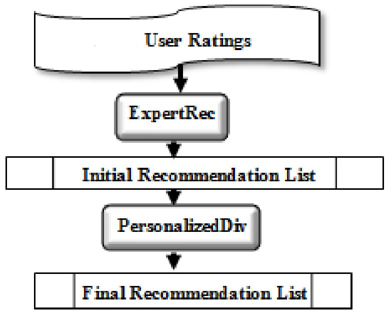


Fig. 1. *Overall Architecture*

the user based on how similar the categories of an item are to the categories of items of user's past preferences. To predict ratings based on expert ratings and user's past preferences and thereby to generate the final recommendation list, *ExpertRec* goes through several steps which are presented next.

1) User preference diversity measurement

First phase of *ExpertRec* is to find the diversity of preferences of each user. To do so, we first create the user's category preference profile, C_u, based on the items user has rated in the past and use it to measure diversity. For example, in movie domain, C_u indicates how much the user u prefers romance, crime and etc., genres. How much user u prefers a category c, is calculated according to (2). For each category in the item domain we calculate the probability user u prefers that category and finally create C_u. In (2), when an item has only a single category, the proportion it contributes towards that category is 1, and if it has more than one category, the contribution for each of those categories would be 1/Number of categories of the item.

$$C_{u,c} = \frac{\sum_{i \in U_u \ \& \ c \in I_i} \frac{1}{|I_i|}}{|U_u|} \tag{2}$$

Next we adapt Shannon Entropy as in (3) to calculate how diverse the user's preferences are. If a user has rated items of a wide variety of categories, (3) will give a higher value and if the user has rated items of only a single or few categories, (3) will give a lower value. We used $|C|$ as the log base in order to keep the diversity value of user u, given by D_u normalized in the range of 0 – 1.

$$D_u = -\sum_{i=1}^{|C|} C_{u,c} \log_{|C|} C_{u,c} \tag{3}$$

We generate the user's category preference profile and calculate the user's diversity value for each user and use them in the next phase.

2) Expert family creation

Expert family creation phase selects experts and create expert families for each item category. In *ExpertRec*, users who have shown low diversity in their preferences are identified as experts in the category they have rated most. The reason is, since the identified expert has shown low diversity of preferences, almost all of his ratings would be focusing on a single or few categories. Hence, he has high experiences in that category making him an expert in that category than someone with diverse preferences and low experiences on that category. Thus, for example, this phase identifies Romance-genre experts, Thriller-genre experts and etc., based on the previously calculated C_u and D_u.

To select the experts we first select the users who have a lower D_u than a given threshold T. From these users we select experts for each of the item category, by identifying the category with highest preference probability (i.e., *argmax* $C_{u,c}$) in that user's C_u. If the user has same highest preference probability for more than one category, we identify that user as an expert in all those categories. Thus, at the end of this phase, for each of the item categories, we have an expert family which includes experts of that category.

3) Relevance Prediction

Relevance prediction is the final phase of *ExpertRec* which predicts the relevance of an item for a user that has not seen the item before. To predict the rating for an item, we combine a score we obtain by considering the user's past preferences and a score we obtain from the experts. We consider this expert score as an expert advice on how good the item is.

We obtain the final relevance score for an item *j* from (4), where $UScore_j$ is a score based on user's past preferences and $EScore_j$ is a score based on ratings given by Experts. Division by 2 indicates the same contribution from both of the scores. But if needed, one can change the proportion of each contribution.

$$RScore_j = \frac{UScore_j + EScore_j}{2} \tag{4}$$

a) User-based Score

User-based Score (*UScore*) is gained from a content-based similarity measurement. The main concept behind *UScore* is to calculate how much a user would like an item j, based on how much user likes the categories of item j. A user shows his preference to an item by giving a rating. User's profile consists of all the items user has rated. By considering the ratings the user has given to items that are similar to item j in the past, based on the item categories, we can predict how much user would like item j. *UScore* is calculated according to (5).

$$UScore_j = \frac{\sum_{i \in U_u \ \& \ I_i \cap I_j \neq \emptyset}^{\forall i} \sum_{c \in \{I_i \cap I_j\}}^{\forall c} \frac{r_{u,i}}{|I_i|}}{\sum_{i \in U_u \ \& \ I_i \cap I_j \neq \emptyset}^{\forall i} 1} \tag{5}$$

To calculate $UScore_j$, we consider all the items in the user profile U_u. First for each of the items *i* in the user profile we check if that item's category set I_i and the set of categories of the item *j* we are considering, I_j have any common elements, i.e., we check if there are any common categories between the two sets. If so, for each such common category of the item *i*, we calculate the proportion of rating item *i* gives for that category *c* by dividing the rating user *u* has given to item *i* by the total number of categories of the item, $|I_i|$. A single item can have many categories. That's the reason for dividing the rating user has given to item *i* by $|I_i|$.

Ultimately *UScore* will give a rating on how relevant an item is to the user based on the user's past preferences.

b) Expert-based Score

Expert-based Score (*EScore*) is a score obtained based on experts' ratings for an item. To calculate *EScore* for an item *j* we first identify the Expert families of categories of item *j*. For example, if item *j* has Romance and Thriller genres, we first select the Romance expert family and Thriller expert family from previously generated set of expert families. Thereafter we select the Experts from those families who have rated item *j*. Once the set of expert members who have rated item *j*, given by EM_j, is identified, all their ratings are aggregated to obtain the *EScore* according to (6).

$$EScore = \frac{\sum_{u \in EM_j}^{\forall u} r_{u,j}}{|EM_j|} \tag{6}$$

ExpertRec predicts the ratings for items which a user has not rated, based on *RScore* given in (4), which uses (5) and (6). Once the ratings are predicted, the items are ordered according to the predicted ratings and the Top-N items are recommended to the user. *ExpertRec* considers both user's past preferences and the ratings given by experts when generating recommendations.

ExpertRec is capable of generating a recommendation list with not only items that are similar to a user's past preferences but also novel items which are recommended by experts. Thus, we believe *ExpertRec* has the capacity to generate a diversified recommendation list unlike the traditional RSs. Traditional Content-based RSs are prone to overspecialization, hence it's hard for content-based RSs to recommend novel items which are different from a user's past preferences. Since *UserCF* is based on nearest neighbors, the items that are recommended based on neighbors' ratings will also be similar to the user's past preferences if the neighbors have very similar preferences to the user. This is also not very helpful for diversification. But *ExpertRec* not only uses user's past ratings, but also experts' ratings. Hence even if the user has not rated any item similar to the item to be recommended, it will still be recommended to the user if the experts have rated it well. Hence *ExpertRec* unlike other techniques, reinforce diversity by allowing novel items to be added to a user's candidate item list. It is also safe to recommend such novel items as that item has not been recommended by just an ordinary user, but by a group of experts who have higher experience in that item category. Since they have higher experiences in that item category, the ratings they have given can be considered as an expert opinion of how good or bad the item is. Hence recommending a novel item that is positively rated by relevant experts would not only be helpful to improve diversification but would also be helpful for users to explore worthwhile novel items. It should also be noted that *ExpertRec* considers relevance as well since it also considers user's past ratings. Hence an item will also be added to the candidate item list if it's similar to the items user has rated in the past.

By generating a recommendation list with both relevant and novel items, *ExpertRec* supports *PersonalizedDiv* to carry out the diversification in a personalized manner. After generating the initial recommendation list from *ExpertRec*, we serve it as input to *PersonalizedDiv* and generate the final recommendation list by selecting a subset of items from the initial list.

C. Personalized Diversification Module

Personalized diversification module (*PersonalizedDiv*) carries out the diversification process in a personalized manner while allowing controlling of trade-off between diversity and relevance. *PersonalizedDiv* is formulated as a greedy optimization problem that generates the final recommendation list by greedily selecting items that maximize the objective function (7), from the provided set of candidate items which were generated from *ExpertRec*. It should be noted that one can also use *PersonalizedDiv*, with a recommendation list generated from any other technique as well since our solution approach is a post-filtering approach. The adopted greedy strategy is given in Algorithm 1.

In each iteration of the Algorithm 1, the item *i**, from the candidate item set *S*, which maximizes (7) is added to the final recommendation list *R*, to generate an ordered Top-k recommendation list.

Equation (7) adopts Maximal Marginal Relevance (MMR) [13] to allow controlling the trade-off between diversity and relevance. We use λ ($0 \leq \lambda \leq 1$) in (7) to control the trade-off between diversity and relevance where λ=1 indicates maximum relevance and λ=0 indicates maximum diversity. Relevance of an item *i* to a user is indicated by $RScore_i$. If the candidate item list is generated from *ExpertRec*, $RScore_i$ is the rating calculated by (4). *MaxPossibleRating* is the maximum rating a user can give to an item. In a 1-5 star scale, *MaxPossibleRating* would be 5. In order to normalize the relevance score, $RScore_i$ is divided by *MaxPossibleRating*. In (7), $PDiv(i,R,C_u,D_u)$ is a score based on personalized diversification and on how diverse the item is with respect to the previously selected items in the list R and it is calculated by (8).

$$PersonalizedDIV(i,R,C_u,D_u) = \lambda \frac{RScore_i}{MaxPossibleRating} + (1-\lambda) PDiv(i,R,C_u,D_u) \quad (7)$$

The main concept behind *PersonalizedDiv*, is to provide a personalized diverse list. We earlier measured the diversity of user's preferences D_u. Based on that, we identify whether the user prefers various item categories or prefers only a few categories. Our main modeling hypothesis for the *PersonalizedDiv* is that, users who have a highly diverse profile are explorative and tend to prefer diverse and novel items. Thus, it is the duty of the RS to diversify the user's recommendation list with more novel items that are diverse with respect to the user's past preferences. Whereas a user with a less diverse profile tend to prefer items similar to what the user has rated before as that user has only a smaller spectrum of interests. Thus, it is the duty of the RS to diversify that user's recommendation list with more items similar to what the user has consumed before. The goal of *PersonalizedDiv* is to obtain this personalized diversification behavior.

$R = \emptyset$
while $|R| < k, S \;\&\; S \neq \emptyset$ **do**
  $i^* = argmax_{i \in S \setminus R}\; PersonalizedDIV(i,R,C_u,D_u)$
  $S = S \setminus \{i^*\}$
  $R = R \cup \{i^*\}$
end while
return R

Input : k - size of final list, S - candidate item set, C_u - User's category preference profile, D_u - User's diversity value
Output : R - final recommendation list

Algorithm 1: Personalized Greedy Re-ranking

Equation (8), which was used in the objective function (7), gives a score based on personalization and diversity of an item with respect to the rest of the items in a list. In (8), *div(i,R)*is calculated by (10). This is multiplied by $W_{personalized}$, calculated by (9), in order to get the diversity score *PDiv*.

$$PDiv(i,R,C_u,D_u) = W_{personalized} * div(i,R) \quad (8)$$

Our unique diversification concept lies in the personalized-weight $W_{personalized}$. Equation (9) presents the personalized-weight $W_{personalized}$ we introduce. By using $W_{personalized}$, we control how much novel items to offer and how much similar items to offer depending on user's past behavior. This is further explained as follows; suppose we have to decide whether to select item *i* in the candidate item list to be added to the final recommendation list. We first look at how diverse the user profile is to decide if the user prefers more novel items or more related items. Then if the user prefers more novel items, which is indicated by a highly diverse user profile, and item *i* has categories the user has not seen before, $W_{personalized}$ will give a higher weight to that item and if item *i* has categories the user has seen before, $W_{personalized}$ will give it a lower weight. If the user prefers more related items indicated by a low diverse user profile and item *i* has categories user mostly prefers, $W_{personalized}$ will give it a higher weight and if item has novel categories user has never seen before, $W_{personalized}$ will give it a lower weight. Accordingly, $W_{personalized}$ gives weight values in a personalized manner.

$$W_{personalized} = \frac{\sum_{c \in I_i} |D_u - C_{u,c}|}{|I_i|} ; 0 \leq W_{personalized} \leq 1 \quad (9)$$

In (8), we multiply the weight $W_{personalized}$ with *div(i,R)* given in (10) , which will measure how diverse the item *i* is with respect to the other items in the list. As diversity can be regarded as the opposite of similarity, (10) represent the diversity score as the inverse of similarity score (*sim(i,R)*).

$$div(i,R) = 1 - sim(i,R) \quad (10)$$

If by adding the item *i* to the list R, the list will be more diverse, *div(i,R)* will give a higher score and vice versa. We measure the similarity of item *i* to the list *R* as the aggregated pairwise similarity between item *i* and each of the item *j* in *R*, as in (11). In (11), $sim_c(i,j)$ is the similarity between item i and j.

$$sim(i,R) = \frac{\sum_{j \in R} sim_c(i,j)}{|R|} \tag{11}$$

We use Jaccard similarity to measure $sim_c(i,j)$ based on the item categories as in (12), which gives a ratio between the number of common categories of both items and the total number of categories in both items.

$$sim_c(i,j) = \frac{|I_i \cap I_j|}{|I_i \cup I_j|} \tag{12}$$

The main procedure of *PersonalizedDiv* is to select items from the candidate item list generated by *ExpertRec* in order to create a personalized diverse recommendation list. Personalization is obtained based on the diversity of the user's past preference and giving novel or related items based on that behavior. The final recommendation list is diversified with more novel items if the user has shown very diverse preferences in the past and it will be diversified with more items similar to what user has rated in the past if the user has shown less diversity in the past. While maintaining this personalization behavior, diversity of the final list is maintained by comparing the similarity of the item to the rest of the items already added to the final recommendation list. Thus, *PersonalizedDiv* is capable of generating a personalized diverse recommendation list.

In this section we explained our proposed approach which consisted of two main phases, *ExpertRec* and *PersonalizedDiv*. The uniqueness of our approach lies in both of these phases. On the one hand, *ExpertRec* predicts rating for items the user has not consumed before and creates a sorted list of recommendations. Ratings are predicted based on both user's past preferences and expert ratings. Since we not only consider user's ratings but also experts' ratings for an item, *ExpertRec* helps to generate a candidate item list which promotes diversity by including novel items based on experts' ratings. On the other hand, *PersonalizedDiv* generates the final diversified recommendation list in a personalized manner by diversifying the list of a user who prefers diversity with more novel items and diversifying the list with more related items that are similar to the user's past preferred items if the user has a low diverse past preferences. In the next section we present the experimental validation for the proposed approach.

V. EXPERIMENTS AND RESULTS

In order to evaluate the capabilities of our proposed approach we carried out experiments with the MovieLens 100k[1] dataset which contains ratings users have given to movies. We evaluated the results on diversity, relevance and rating prediction accuracy. For our evaluation we carried our two phases of experiments. In the first phase we carried out experiments for *ExpertRec* and in the second phase we carried out experiments for *PersonalizedDiv*.

In both phases we used the traditional *UserCF* algorithm available in Apache Mahout[2] configured with Pearson correlation similarity as the user similarity measure and a neighborhood of size 50. We selected 100 as the size of the initial recommendation list of candidate items and we selected the size of the final recommendation list as 10 for our experiments.

A. *Experiments for ExpertRec*

We carried out experiments in order to evaluate whether our *ExpertRec* approach is competitive with respect to rating prediction accuracy and for promoting diversity in recommendation lists. Before carrying out the quantitative experiments we had to decide the threshold value T to be used in *ExpertRec*.

As explained in the methodology, *ExpertRec* selects a user u as an expert if u's diversity value given by D_u is smaller than a given threshold T. Hence we first analyzed the dataset to decide what value to be chosen for T. Since D_u is calculated according to (3), the value of T should be in the range of 0 – 1. By selecting each of the values in set {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} as T, we calculated the total number of experts for each genre according Expert Family Creation phase previously described in section IV.B.2). When T <0.4 there were no or very few Experts in the dataset. Hence value of T had to be in the range of 0.5 - 1.0. Thus, we chose T =0.8 as it's the mid-point of the possible value range. All the experiments involving *ExpertRec* were conducted with the configuration T =0.8.

Our experiments for *ExpertRec* are focused on rating prediction accuracy and diversity. We used *UserCF* as the baseline to compare *ExpertRec* with and generated candidate item list from both techniques. We selected *UserCF* because among the available traditional RSs, such as content-based and item-based collaborative filtering, *UserCF* is more capable of generating potentially novel recommendations beyond the user's past preferences [21]. For measuring rating prediction accuracy we used RMSE metric and used the hold- out method by keeping 70% of ratings of each user to build the training set and the remaining 30% for the test set and carried out validation for 5 rounds while randomly creating training and testing sets in each round. Final RMSE score was obtained by averaging the values obtained in each five rounds. To measure diversity we used pairwise ILD@k[10] metric with Jaccard Similarity as the similarity measure. We generated candidate item lists for each user using the selected recommendation technique, i.e., either *ExpertRec* or *UserCF*. Then we measured the diversity of the generated list of each user using ILD and the final ILD score was calculated by averaging the ILD values of each user's lists. The obtained results are given in Table I.

TABLE I. RESULTS OF BENCHMARK EXPERIMENT

	ILD	RMSE
UserCF	0.2336	1.1128
ExpertRec	0.2694	1.4737

[1] http://grouplens.org/datasets/movielens/

[2] http://mahout.apache.org/

According to the results, *UserCF*'s error in rating prediction, measured by RMSE, is 1.1128. This indicates that the ratings predicted by *UserCF* are varied by 1.1128 stars from the real rating, in 1-5 star scale. The ratings predicted by *ExpertRec* are varied by 1.4737 stars. Thus, rating prediction error is greater in *ExpertRec* by 0.3 stars compared to *UserCF*. The outcomes also show that ILD score of *ExpertRec* is greater than *UserCF* by 0.0358. This indicates that *ExpertRec* performs better than *UserCF* with respect to diversity.

By the analysis of results, we observed that *ExpertRec* doesn't perform better than *UserCF* in rating prediction task. There could be many explanations for this behavior. *ExpertRec* uses a combination of content-based and collaborative filtering like approach. As explained in section IV.B.3) in the methodology, *UScore* is calculated by a content-based approach and the only content we considered was genre of movies in movie recommendation. As *UScore* contributes significantly to the final predicted rating, the above hindrance could be decreased if more features were used for the content-based approach. This could also be due to some of the limitations inherent in accuracy measuring metrics.

When it comes to diversity, as expected, *ExpertRec* has given better results. The reason for this is clear and can be explained as follows. *UserCF* technique recommends movies which the neighbors of the user have rated. Due to that, the quality of the recommendations is highly affected by the selection of neighbors. If the similarity function, which identifies the neighbors with a high resemblance to the user, is excellent, then the selected neighbors would have very similar preferences to the user. Based on the transitive property, it is a clear-cut declaration that, since each neighbor is similar to the user, each neighbor would be similar to each other. As these neighbors have homogeneous preferences, it is obvious that the set of recommendations generated from their preferences would naturally be homogeneous. This explains why *UserCF* had a lower ILD score. Although *ExpertRec* technique includes a collaborative-filtering like approach, it gives a higher ILD. The rationale behind this is that, *ExpertRec* is not based on similar neighbor concept. When recommending a movie to the user, *ExpertRec* considers the ratings given by Experts of the respective genre, not the ratings of user's neighbors. These Experts may or may not bear any resemblance to the preferences of the user. We are considering their ratings although they may not have similar preferences to the user, because we believe that since they are experts in a certain genre, the movies they have highly rated are actually good and useful movies and it's safe to recommend such a movie to a user. We should point out that, we consider if the user prefers movies of that genre as well by *UScore*. But a movie which has no resemblance to user's past preferences has the chance of being recommended in *ExpertRec* if it's highly rated by Experts. Thus, *ExpertRec* has inherent qualities to promote diversity and it is validated from the obtained results.

With the carried out experiments we could observe, that performance of *ExpertRec* with respect to accuracy was inferior to *UserCF*, but with respect to diversity, its performance is superior. Even though the accuracy is inferior to *UserCF*, we have achieved a positive level of accuracy. Thus, we can validate that *ExpertRec* is capable of predicting ratings and generating recommendation lists with diverse items due to its capability to include novel items along with related items to the recommendation list.

B. *Experiments for PersonalizedDiv*

We conducted qualitative and quantitative experiments to validate that *PersonalizedDiv* technique can carry out diversification process in a personalized manner and achieve diversity and relevance of the final recommendation list while controlling the trade-off between diversity and relevance.

1) *Qualitative Experiments*

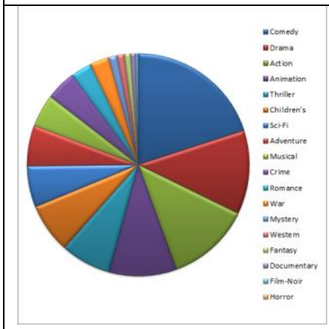
In this paper, we emphasized what we interpret by personalized diversification. In simple terms, personalized diversification is obtained by diversifying a user's recommendation list based on the user's past behavior. If we elaborate this concept more; a user who had shown diverse tastes in the past, i.e., a user who prefers to explore novel things, should receive a recommendation list diversified with more novel items, whereas a user who had homogeneous tastes in the past should receive a recommendation list diversified with more relevant items. In this qualitative experiment we hope to investigate if our *PersonalizedDiv* technique is capable of showing the aforementioned behavior.

As the initial step of the experiment, we analyzed the dataset to identify users with different user profile diversity values, D_u, in order to select users for this qualitative experiment. We defined three diversity levels and randomly selected a user for each level. The three levels are; High-Level : $0.7 \leq D_u \leq 1.0$, Medium-Level : $0.4 \leq D_u < 0.7$ and Low-Level : $0.0 \leq D_u < 0.4$. We randomly selected three users for these three levels who had provided almost the same number of ratings. For each of the three users, we generated the candidate item list using the proposed *ExpertRec* technique. Then we generated the final recommendation list using *PersonalizedDiv*. Since we are experimenting on the diversification behavior, we set $\lambda=0$ in (7) to achieve maximum Diversity. We also generated user's category preference profile, C_u, as explained in section IV.B.1) for each of the three users. As the final step of the experiment, we analyzed the generated final recommendation list and user's profile to see if the diversification process has been carried out in a personalized manner.

As depicted in Table II, user profile indicates the user has diverse tastes and the user has not been able to consume many movies from Mystery, Western, Fantasy, Documentary and Film-Noir genres. Horror is the most novel genre for this user as the profile doesn't consist of any horror movie. The Recommendation list generated for this user consists of three Horror movies which are novel for this user. It also consists of movies of Film-Noir, Fantasy, Western and Mystery genres, which are the genres the user had only few experiences with. It should be noted that the number of items from user's mostly consumed genre, Comedy, is lower than the number of items from the novel genre, Horror, in the recommendation list.

Table III depicts the results for Medium-Level diversity user. According to the user profile in Table III, it is visible that it consists of various genres, but it's not a highly diverse profile as there's a notable contrast between the number of consumed movies from each genre whereas in Table II we noted, that user

TABLE II. RECOMMENDATIONS FOR A HIGH-LEVEL DIVERSITY USER

User Profile- *High-Level*	**Recommendation List**
	• **567** Horror • **770** Crime | Film-Noir | Mystery | Thriller • **217** Horror | Romance • **1004** Drama | Western • **182** Crime | Drama • **860** Horror | Thriller • **1472** Comedy | Sci-Fi • **951** Adventure | Children's | Fantasy • **835** Comedy | Musical | Romance • **483** Drama | Romance | War

had consumed almost the same number of movies from each genre leading to a diverse profile. This user's preferences are also not much biased to a single genre as the user has preferred movies of various genres. Thus we can observe that the user has a medium level diversity preference profile. If we analyze the Recommendation list of this user, we can observe that it consists of various genres including the user's most favorite genres and novel genres. Among the novel genres, the list has covered Sci-Fi, Musical and Children's genres. List has also covered user's mostly preferred genres, which are Drama, Comedy and Romance. Thus, we can observe that our recommendation technique has been able to diversify the Medium-level diversity user's recommendation list with items of novel genres as well as with items of user's favorite genres.

If we analyze the results of low-level diversity user, as depicted in Table IV, Action and Thriller are the chiefly

TABLE III. RECOMMENDATIONS FOR A MEDIUM-LEVEL DIVERSITY USER

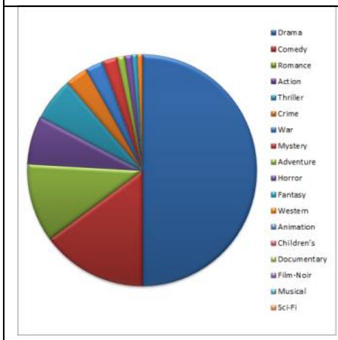
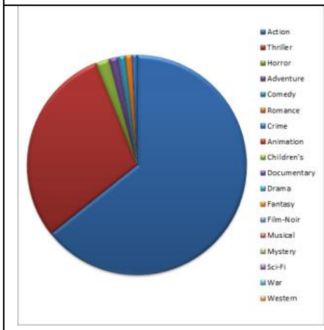
User Profile- *Medium-Level*	**Recommendation List**
	• **558** Drama | Fantasy| Romance | Thriller • **1472** Comedy | Sci-Fi • **132** Adventure | Children's | Drama | Musical • **835** Comedy | Musical | Romance • **1500** Comedy • **855** Action | Drama | Mystery | Romance| Thriller • **1298** Comedy | Musical • **172** Action | Adventure | Drama | Romance| Sci-Fi | War • **135** Drama | Mystery | Sci-Fi | Thriller • **965**Comedy | Musical

TABLE IV. RECOMMENDATIONS FOR A LOW-LEVEL DIVERSITY USER

User Profile- *Low-Level*	**Recommendation List**
	• **977** Action • **273** Action | Crime | Thriller • **925** Thriller • **1250** Action • **1239** Action | Adventure | Romance • **773** Thriller • **174** Action | Adventure • **33** Action | Romance | Thriller • **1139** Action | Crime | Thriller • **771** Action | Sci-Fi | Thriller

preferred genres of this user and the user has homogeneous preferences as most of his preferences are centered on a few number of genres. If we scrutinize the recommendation list, there are many movies of Action and Thriller genre which are the most preferred genres of the user. The list also consists of Adventure, Romance and Crime movies which the user had shown preferences for in the past. These results indicate that this recommendation list has been diversified with more items of user's most preferred genres.

If we further analyze the results, analysis of results of High-Level diversity user proclaimed that our approach generated a diversified recommendation list with more novel items to a user with a highly diverse preference profile. However, this is impossible if novel items were not present in the candidate item list. For this experiment, candidate item list was generated using our *ExpertRec* technique. Recommendation generation process of *ExpertRec* considers user's past preference for genres as well as ratings given by experts. Novel items are anticipated due to the consideration of ratings given by experts. The user profile of the High-Level diversity user indicated that Horror is a novel genre to that user as the user profile didn't include Horror genre. Thus the only prospect of a Horror movie being added to the candidate item list is, due to the ratings given by Horror experts to the recommended Horror movies. This indicates that the ratings of the experts have an influence on the addition of novel items to the final recommendation list. The analysis also indicated that Low-Level diversity user received a recommendation list diversified with more items similar to his past preferences. The carried out qualitative experiment indicates that *PersonalizedDiv* technique is capable of diversifying the recommendation list in a personalized manner.

2) Quantitative Experiments

We carried out experiments to measure the diversity and relevance that can be achieved with different configurations of *PersonalizedDiv*. Diversity was measured as before using pairwise ILD metric. To measure relevance we used nDCG@k metric where the Ideal DCG was obtained from the top k results of the candidate item list.

To generate the candidate item list for a user we used both *ExpertRec* and *UserCF*. Different configurations of our *PersonalizedDiv* approach is compared with the following baseline and state-of-art techniques. (i) Random Diversity: Randomly selecting k items from the candidate items list. Represents maximum achievable diversity. (ii) No Diversity: Selecting the top k items from the candidate items list. Represents maximum achievable relevance. (iii) MMR[13](λ=0.5). Represents same level of diversity and relevance. We used the generated candidate item lists as input to each of these approaches and generated the final recommendation lists to carry out the evaluation. Obtained results are presented in Table V.

MaxDiv results of Table V is graphically presented in Fig. 2. According to the MaxDiv results, Random Diversity has obtained a higher diversity score than *PersonalizedDiv* for both *ExpertRec* and *UserCF*. This is the anticipated behavior with respect to maximum diversity. Since randomization is the key to attain best level of diversity, it is nearly impossible for *PersonalizedDiv* to obtain a higher or same level of diversity while maintaining personalization aspect. If we compare the diversity scores of *ExpertRec* and *UserCF*, we can observe that there's a remarkable contrast between the scores and *ExpertRec* shows a higher diversity score. The reason for this could be the quality of candidate item list. From the results of *ExpertRec* experiment presented in Table I, we saw that the candidate item list generated from *ExpertRec* has obtained a higher diversity score than the list generated by *UserCF*.

When considering the MaxRel results in Table V, both No Diversity and *PersonalizedDiv* has achieved maximum relevance score of 1.0 using candidate item lists generated from both *ExpertRec* and *UserCF*. No Diversity technique returns the top-k items from the candidate item list and according to (7), *PersonalizedDiv* also returns the top-k when configured with λ=1. Since nDCG metric compares the final recommendation list with the top-k item list from candidate recommendation list, it is expected that we achieve the maximum possible score of 1.0 for both No Diversity and *PersonalizedDiv* for maximum relevance configuration.

To discuss about Equal Diversity & Relevance configuration, i.e., achieving both diversity and relevance in the final recommendation list, it would give more insight if average of diversity and relevance results are considered. This averaging is possible as both ILD and nDCG metrics give results in the same scale of 0-1. Results are given in Table VI.

TABLE V. DIVERSITY AND RELEVANCE RESULTS FOR PERSONALIZEDDIV

		ExpertRec		*UserCF*	
		ILD	nDCG	ILD	nDCG
MaxDiv	RandomDiversity	0.2654	-	0.2333	-
	PersonalizedDiv λ=0	0.1965	-	0.0574	-
MaxRel	NoDiversity	-	1.000	-	1.000
	PersonalizedDiv λ=1	-	1.000	-	1.000
Equal Div & Rel	MMR α=0.5	0.2965	0.9481	0.0985	0.9620
	PersonalizedDiv λ=0.5	0.1742	0.9424	0.0978	0.9666

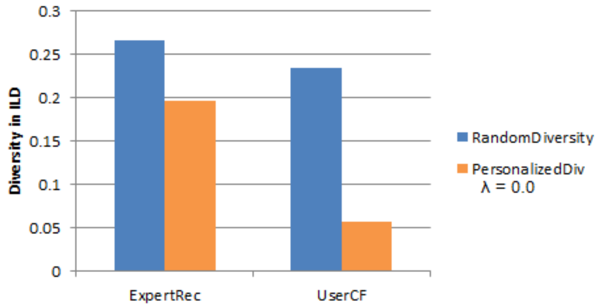


Fig. 2. *Max Diversity Results*

The results in Table VI are graphically presented in Fig.3. According to the results, *ExpertRec* has supported both MMR and *PersonalizedDiv* to achieve higher results in equal diversity and relevance configuration, compared to *UserCF*. Results also depict that, when the user is in need of both diversity and relevance, we should consider generating thecandidate list from *ExpertRec* and use MMR post-filtering technique. When the user is in need of diversity and relevance along with personalization, we should consider generating the candidate list from *UserCF* along with the *PersonalizedDiv* technique. Even with the additional personalization step, we have achieved better results than the non-personalized MMR, by using candidate list of *UserCF*.

From these quantitative experiments, we can conclude that the candidate list generated by *ExpertRec*, is the overall winner for maximum diversity and equal div & rel configurations for all post-filtering approaches and for maximum relevance both *ExpertRec* and *UserCF* are winners. Although *PersonalizedDiv* hasn't outperformed Random Diversity in maximum diversity configuration, it shares the winner position with NoDiversity for maximum relevance configuration. When configured to maximum diversity, *PersonalizedDiv* performs better with *ExpertRec* than with *UserCF* which shows that *ExpertRec* approach has positively affected *PersonalizedDiv* to achieve higher diversity even with the additional personalization step. *PersonalizedDiv* has also achieved better results than the non-personalized state-of-art MMR technique, when configured to generate final recommendation list with equal level of diversity and relevance, by using the candidate list of *UserCF*.

Overall, our *ExpertRec* approach is capable of promoting novel items to improve diversity than the traditional *UserCF* although it's not superior in rating prediction. *PersonalizedDiv* together with *ExpertRec* approach is capable of personalizing the diversification of recommendations while allowing controlling the trade-off between diversity and relevance. Also, some configurations of *PersonalizedDiv* perform better than state-of-art diversification approaches as well.

TABLE VI. DIVERSITY AND RELEVANCE COMBINED RESULTS FOR PERSONALIZEDDIV

		ExpertRec	*UserCF*
		avg(ILD,nDCG)	avg(ILD,nDCG)
Equal Div & Rel	MMR α=0.5	0.6223	0.5302
	PersonalizedDiv λ=0.5	0.5583	0.5322

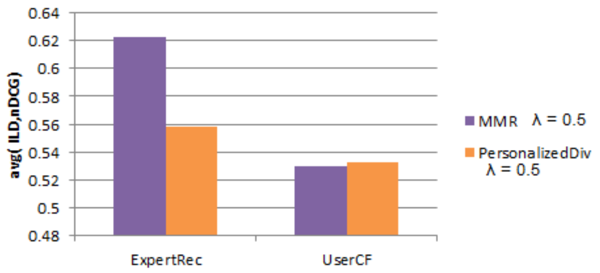


Fig. 3. *Equal Diversity and Relevance Results*

VI. Conclusion and Future Work

In this work, we proposed a technique to utilize ratings given by experts and user's past preferences in the recommendation process to promote diversity along with relevance in the recommendation generation. We also devised a diversification technique named *PersonalizedDiv* that is not only capable of controlling the trade-off between diversity and relevance of the recommendations, but is also capable of diversifying the recommendation list in a personalized manner. Experiments showed that our approach carries out the diversification in a personalized manner and provides comparable results to state-of-art approaches.

As future work we plan to find the feasibility of utilizing *ExpertRec* for non-personalized diversification where diversified recommendation lists are generated in a non-personalized manner. We also plan to find means other than based on experience to identify experts for novel item recommendation. We wish to investigate if *ExpertRec* and *PersonalizedDiv* can be combined by some means to directly generate a personalized diversified recommendation list instead of using a post-filtering approach. Since we believe a user study would give a more interesting perception of our proposed approach through a user-centric evaluation, we also wish to conduct a user study. Finally we also wish to investigate if using more categories in *ExpertRec* would have any impact on the recommendations it generate.

References

[1] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood based recommendation methods," in Recommender systems handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, vol. 1, pp. 107–144.

[2] J. Sean, M. McNee, and J. Konstan, "Accurate is not always good: How accuracy metrics have hurt recommender systems," extended abstracts on Human factors in computing systems (CHI06) p, pp. 1097–1101.

[3] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in Proceedings of the 14th international conference on World Wide Web. ACM, 2005, pp. 22–32.

[4] P. Castells, S. Vargas, and J. Wang, "Novelty and diversity metrics for recommender systems: choice, discovery and relevance," 2011.

[5] A. Barraza-Urbina, B. Heitmann, C. Hayes, and A. Carrillo-Ramos, "Xplodiv: An exploitation-exploration aware diversification approach for recommender systems," in The Twenty-Eighth International Flairs Conference, 2015.

[6] A.Gunawardana and G. Shani. "A survey of accuracy evaluation metrics of recommendation tasks." The Journal of Machine Learning Research vol.10, 2009, pp. 2935-2962.

[7] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 5–53, 2004.

[8] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, "Introduction to recommender systems handbook," in Recommender systems handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, vol. 1, pp. 1–38.

[9] A. Barraza, "Diversification technique for recommender systems that controls the tradeoff between exploitation of the user profile and exploration of novel products," Master's thesis, Facultad de IngenierÃ-a, Pontificia Universidad Javeriana, Colombia, 2014.

[10] B. Smyth and P. McClave, "Similarity vs. diversity," in Case-Based Reasoning Research and Development. Springer, 2001, pp. 347–361.

[11] L. Candillier, M. Chevalier, D. Dudognon, and J. Mothe, "Diversity in recommender systems: bridging the gap between users and systems," Centrics, to appear, 2011.

[12] S. V. Sandoval, "Novelty and diversity evaluation and enhancement in recommender systems," Ph.D. dissertation, Departamento de IngenierÃ-a InformÃatica, Escuela PolitÃlcnica Superior ,Universidad AutÃsnoma de Madrid, Spain, 2015.

[13] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998, pp. 335–336.

[14] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells, "Coverage, redundancy and size-awareness in genre diversity for recommender systems," in Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014, pp. 209–216.

[15] G. Adomavicius and Y. Kwon, "Overcoming accuracy-diversity tradeoff in recommender systems: A variance-based approach," in Proceedings of the 18th Workshop on Information Technology and Systems, WITS. Citeseer, 2008.

[16] A. Pathak and B. K. Patra, "A knowledge reuse framework for improving novelty and diversity in recommendations," in Proceedings of the Second ACM IKDD Conference on Data Sciences. ACM, 2015, pp. 11–19.

[17] G. Adomavicius and Y. Kwon, "Improving recommendation diversity using ranking based techniques," IEEE Transactions on Knowledge and Data Engineering.

[18] L. Chen, W. Wu, and L. He, "How personality influences users' needs for recommendation diversity?" in CHI'13 Extended Abstracts on Human Factors in Computing Systems. ACM, 2013, pp. 829–834.

[19] N. Tintarev, M. Dennis, and J. Masthoff, "Adapting recommendation diversity to openness to experience: A study of human behaviour," in User Modeling, Adaptation, and Personalization. Springer, 2013, pp. 190–202.

[20] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio, "An analysis of users' propensity toward diversity in recommendations," in Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014, pp. 285–288.

[21] S. Vargas and P. Castells, "Exploiting the diversity of user preferences for recommendation," in Proceedings of the 10th Conference on Open Research Areas in Information Retrieval. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire, 2013, pp.129-136.