Exploiting the Diversity of User Preferences for Recommendation

Saúl Vargas and Pablo Castells
Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid, Spain
{saul.vargas,pablo.castells}@uam.es

ABSTRACT

Diversity as a quality dimension for Recommender Systems has been receiving increasing attention in the last few years. This has been paralleled by an intense strand of research on diversity in search tasks, and in fact converging views on diversity theories and techniques from Information Retrieval and Recommender Systems have been put forward in recent work. In this paper we research diversity not only as a target property for a recommender system, but as an element in the input data, within and between user behaviors, that a recommender system can leverage to enhance the quality of its output in terms of the balance between accuracy and diversity. We propose an adaptation of search result diversification methods to recommender systems based on query reformulation: we identify the diversity within user profiles and generate partial recommendations based on homogeneous subsets of user preferences (sub-profiles), which we combine later to produce a final recommendation. We report experiments on movie and music recommendation datasets showing that our approach improves indeed the quality of state-of-the-art recommenders, and is competitive against diversification methods that use explicitly item categories as the units for diversification. Our approach shows further advantages in cases where the high cardinality of the explicit category spaces can pose a problem in terms of computational cost.

1. INTRODUCTION

Diversity as a quality dimension for Recommender Systems (RS) has been receiving increasing attention in the last few years. Diversity in recommendations avoids redundancy and enhances the array of choice for the user, which is a good strategy to cope with the inherent uncertainty involved in guessing the user's preferences. This perspective contrasts with the traditional view in Recommender Systems, where the focus has been in the accumulation of relevance in the delivered recommendations.

OAIR '13, May 22-24, Lisbon, Portugal.

Diversity in recommendation has a close precedent in Information Retrieval (IR), where diversification of search results is envisioned as a means to deal with the ambiguity in queries, and to reduce the redundancy of retrieved documents. The most recent approaches to the problem to date usually consider an explicit representation of the interpretations or aspects behind ambiguous or underspecified queries. A particularly effective approach for the extraction of query aspects use query reformulations returned by a search engine as proxies for query aspects [13]. Drawing from this perspective, we propose the adaptation of the notion of query reformulation to recommender systems through the extraction of user sub-profiles. Considering subsets of user interests is a natural idea, since people's preferences have different sides (sports, politics, work, leisure, music or movie genres, etc.), as well as we have different facets in our lives, and different attitudes in different contexts. The definition of user sub-profiles seeks to make specific recommendation to a user according to every single facet or interest. The basis of the approach we propose in this paper is the intuition that better (more accurate and better diversified) recommendations can be produced by taking into account this polyfacetic nature of user interests. The idea is that, for instance, user preferences in classical music can be more useful than rock music favorites to recommend a classical music piece.

We thus research diversity not only as a target property for a recommender system, but as an element in the input data, within and between user behaviors, that a recommender system can leverage to enhance the quality of its output in terms of the balance between accuracy and diversity. Our approach identifies the diversity within user profiles and generates partial recommendations based on homogeneous subsets of user preferences (sub-profiles), which we combine later to produce a final recommendation. We report experiments on movie and music recommendation datasets showing that our approach improves indeed the quality of state-of-the-art recommenders, and is competitive against diversification methods that just use explicitly item categories as the units to diversify for. Our approach shows further advantages in cases where the high cardinality of the explicit category spaces can pose a problem in terms of computational cost.

The rest of this paper is structured as follows. In Section 2 we overview the related work. Section 3 introduces some notation that will be used along the following sections. Section 4 presents an initial diversification approach adapted from previous state-of-the-art proposals for search result and item recommendation diversification that will serve us as a baseline to compare with our proposal. In Section 5 we present our sub-profile diversification alternative. Section 6 studies some scalability problems of the diversification approaches proposed in Section 4 and Section 5 and proposes a solution for the approach based on sub-profiles. Section 7 describes the experimental evaluation carried out to assess the validity of our proposal and discusses the obtained results. Finally, Section 8 offers a summary and final comments of this work.

2. RELATED WORK

Recommender Systems [2] can be seen as a particular type of Information Retrieval system where the information need is not expressed by means of an explicit query, but it is implicit in the user records of interaction with items in the system. We refer by *user profile* to the set of items the user has interacted with, either by explicitly assigning ratings to them (indicating the degree to which the user likes them, such as when rating movies on a 1-5 star scale), or by just selecting them for consumption with a certain frequency (e.g. listening to music), where the frequency is taken as a hint of positive preference.

In this work we focus on recommendation algorithms based on collaborative filtering. Collaborative filtering techniques match users with similar preferences, or items with similar choice patterns from users, in order to make recommendations. These techniques do not require any knowledge of the content of the items, and can therefore be applied to generate recommendations for virtually any item domain. Additionally, collaborative filtering approaches also have the advantage that a user may benefit from other users' experiences, thereby being exposed to potentially novel recommendations beyond her own experience [2]. A variety of collaborative filtering algorithms have been applied to recommendation, including nearest neighbors strategies [2], matrix factorization techniques [11, 15], probabilistic latent semantic analysis [10] and the adaptation of text retrieval models for recommendation [5, 19]. The latter are particularly interesting for us, since they have brought forward the view of recommendation as an Information Retrieval task, and have opened a way for adapting other text retrieval techniques and problems to recommender systems.

The Recommender Systems field has traditionally focused on retrieving items that maximize their individual probability of being relevant to the user. However, it has become increasingly apparent that this strategy alone is sub-optimal in general [21], as recommendations focusing only on accuracy tend to suffer from a lack of diversity: the recommended items are typically too similar to each other, in such a way that the users draw limited benefit from them.

Strong analogies can be established between diversity in Recommender Systems and the problem as stated and addressed in Information Retrieval [16, 17]. Diversification of search results is generally posited as a means to handle the ambiguity in user queries. In this work we investigate the application of a similar principle to recommendation. While the proposed Recommender Systems approaches to date generally disregard the diversity contained in user preferences, we consider the principle that a good recommendation should be not only diverse in itself, but should also realize the diversity within the user interests and behavior to provide recommendations that embody a better fitted, more comprehensive view of user preferences.

Common approaches to enhance diversity of item recommendations are greedy re-ranking [23], quadratic optimization [21] and user profile partitioning [22]. Our research is complementary to prior fundamental diversification approaches [21, 23] in that we explicitly investigate the notion of sub-profiles within users, based on how people's preferences typically work in practice, and their polyfacetic nature, a vision which may open further possibilities beyond the ones we focus on here, related to recommendation diversity and effectiveness. Later work by Zhang and Hurley [22] does explore a notion of user profile partitioning which can be related to our research, with significant differences nonetheless:

1. In our approach, user profiles are partitioned using a previously available categorization of the item domain, whereas Zhang and Hurley [22] require elaborate clustering algorithms to define their partitions based on the similarity of between user ratings.
2. In [22], once partition-specific recommendations have been generated, a selection of the most novel ones is combined into a final recommendation by uniformly allocating items from each partition-specific recommendations. We propose a combination of sub-profile recommendations by means of a non-trivial adaptation of aspect-based search result diversification algorithms [13], which performs a rank-aware allocation of items by taking into account the relative importance of each sub-profile while maximizing the number of user tastes represented while avoiding redundancy in a top-N recommendation.

Other proposals in recommendation diversity include that of [18], where diversity is seen as a minimization of risk based on Portfolio Theory, temporal diversity [12], aggregated diversity [1] and the adaptation of intent-oriented methods from Information Retrieval to Recommender Systems [16].

There has also been significant recent work on search result diversification [3, 6, 8, 13, 18, 20]. In this work we build on the proposal of Santos et al. [13], which considers query reformulations as proxies for interpretations or aspects behind ambiguous or underspecified queries. As we shall see, we adapt the concept of query reformulation and its application in search result diversification to the Recommender Systems context.

3. NOTATION

Before presenting our researched approach, we briefly introduce the notation to be used henceforth. The general recommendation task considers a community of users $u \in \mathcal{U}$ –such as potential customers of an e-commerce platform– and a recommendation domain consisting of items $i \in \mathcal{I}$ – e.g. movies, music, hotels, clothes, etc. On top of this, we assume a categorization C of these items is available, where the relation between items and categories $c \in C$ is weighted by a real value $w(c, i)$ –if the relation between categories and items is binary, then weights take the values 1 and 0 accordingly.

Personalized recommendations for a given user u are produced based on her profile $\mathbf{u}$, which consists of a subset of the domain $\mathcal{I}$ of items the user has evidenced interest for in the past, and numeric scores $r(u, i)$ that reflect the intensity of the interest the user has evidenced. The evidence may consist of the frequency of interaction, purchase actions, a direct assignment of score values by the user, etc. The output of the recommender is a list R of recommended items whose ordering is determined by an output score $\hat{r}(u,i)$ produced by the recommender system, where the score value may reflect a specific semantic meaning –generally as a predicted rating–, or just a number for ranking purposes.

4. EXPLICIT ASPECT-BASED DIVERSIFICATION

Most diversification strategies for recommender systems (and Information Retrieval systems for that matter) use a greedy algorithmic scheme where the recommendation returned by a baseline recommender system is re-ranked for a balance between preference-matching (seeking item relevance and recommendation accuracy) and diversity. This re-ranking is often performed in the light of some kind of categorization of the items in the recommendation domain (such as genres, tags, etc), as the basis to assess the diversity of recommended items. The greedy scheme (see Algorithm 1) iteratively selects an item from the original list R and places it at the end of a re-ranked list S until there are no items left and the diversified ranking is complete. The core of the greedy scheme is an objective function f_{obj} which embodies the accuracy-diversity balance, so that the algorithm picks at each step the document that maximizes this function as the next one to place in the diversified reranked list. The diversification methods essentially distinguish themselves from each other by this objective function (and naturally, the principles from which the function results).

Algorithm 1 A greedy selection of the items in recommendation list R to produce a re-ranked list S.

$S = \emptyset$
while $|R| > 0$ **do**
  $i^* = \arg\max_{i \in R \setminus S} \ f_{obj}(i;S)$
  $R = R \setminus \{i^*\}$
  $S = S \cup \{i^*\}$
end while
return S

In previous work, we studied the adaptation of search result diversification methods to recommender systems [16]. Using search result diversification approaches based on an explicit representation of query aspects, we introduced a notion of user profile aspects as a direct equivalent of query interpretations. In our present approach we follow up on this idea and take it a step further by explicitly extracting user sub-profiles based on the categorization of items, as we will see. As the base search result diversification scheme to build upon, we shall use here the xQuAD algorithm [13], which has been shown to be among the most competitive methods (if not the best) in recent evaluation campaigns [14].

Equating a user profile u (which is the representation of the information need in the recommendation task) to a search query, and items to documents, the greedy objective function in xQuAD is a mixture of an initial ranking score function, denoted as $p(i|u)$ (relevance component) after a document likelihood retrieval model, and a diversity score. After Santos et al. [13], the latter is stated as $p(i,\bar{S}|u)$ (diversity component), denoting the probability that an item i is of interest for the user but no other item in the re-ranked list S under construction is relevant. In other words, this term represents the marginal utility of the next item i, after the items in S have been already ranked.

Santos et al. [13] develop the diversity component by marginalizing the probabilities over an explicit set of user need aspects (which in our adaptation are item categories), resulting in the final expression of the xQuAD objective function:

$$f_{obj}(i;S) := \lambda\, p(i|u) + (1-\lambda)\ p(i,\bar{S}|u) \quad (1)$$

$$p(i,\bar{S}|u) = \sum_c p(c|u)\, p(i|c,u) \prod_{j \in S} (1 - p(j|c,u)) \quad (2)$$

where $p(c|u)$ represents the importance of category c for the user (derived from the information of the user profile) and $p(i|c,u)$ the likelihood of item i being chosen by user u when she is interested in category c.

For the adaptation of xQuAD to recommendation, the distributions involved in the objective function can be estimated as follows [16]:

$$p(i|u) = \frac{\hat{r}(u,i)}{\sum_{j \in R} \hat{r}(u,j)} \qquad i \in R \quad (3)$$

$$p(c|u) = \frac{\sum_{i \in \mathbf{u}} r(u,i) w(c,i)}{\sum_{c'} \sum_{i \in \mathbf{u}} r(u,i) w(c',i)} \quad (4)$$

$$p(i|c,u) = \frac{\hat{r}(u,i) w(c,i)}{\sum_{j \in R} \hat{r}(u,j) w(c,j)} \qquad i \in R \quad (5)$$

The estimation for the probability $p(i|c,u)$ was adapted in [16] based on the work by Agrawal et al. [3] for search result diversification, where (again, rephrasing documents and queries as items and users), the likelihood of a document d satisfying the user intent –represented by a document category c– given the query q is determined by the scoring function for the document and the query, weighted by the likelihood that the document belongs to a particular category.

We showed in past work that this direct estimation of the aspect-specific item relevance $p(i|c,u)$ allows to effectively diversify recommendations [16]. We find nonetheless that the estimation of the recommendations for a specific user taste or interest can be further refined. Specifically, as a new principle, we consider now the integration of this estimation in the very recommendation process. For this purpose, we adapt the idea of [13] of using query reformulations as proxies for the multiple interpretations or aspects of an ambiguous or underspecified query. The search results of these reformulated queries can then be combined to produce a diversified search result list. We take a similar approach by producing recommendations specific to each of the tastes or interests represented in the user profile. Each item category c defines a subset of the user profile that we denote as sub-profile. In the next section we describe how these sub-profiles are generated and used to produce diverse recommendations.

5. DIVERSIFICATION USING SUB-PROFILES

In contrast to the approach described in the previous section, where the diversification of the recommendation is exclusively applied after an initial list is generated by a baseline recommender, we embrace the diversity of user preferences in the recommendation process itself. For this purpose, we introduce the notion of user sub-profiles. We establish an analogy between query reformulations and user sub-profiles. Consider an ambiguous query like "java" and some reformulations like "java island", "java coffee" or "java programming", which specify or disambiguate to some extent the original query. Such reformulations can be obtained from commercial search engines, and can serve as a proxy of query intents [13]. Similarly, users can also have different tastes or interests in the context of the recommendation domain. This moves us to consider adapting the use of retrieved results from reformulated queries, as proposed by Santos et al.[13], to recommendation diversification using sub-profile recommendations.

Given a generic user $u \in U$, whose information need is represented by a user profile $\mathbf{u}$, we define the concept of sub-profile as a subset of the original, complete profile that clearly represents a single taste or interest of the user. Each sub-profile can be understood as the profile of an abstract sub-user, representing the general information need of the original user particularized to an exclusive interest.

In our approach, the use of sub-profiles involves three steps:

1. Extraction of sub-profiles: we propose two different alternatives for two types of input data, namely movie recommendation with genre information and music recommendations with social tags. This step is explained in Section 5.1.

2. Generation of sub-profile recommendations: as explained in Section 2, collaborative filtering methods generate recommendations matching users with similar preferences, or items with similar patterns of choice by users. Consequently , the extracted sub-profiles from the previous step can be used to help issuing recommendations to other sub-profiles. Several alternatives are considered in Section 5.2, depending on the grouping of sub-profiles and the use or not of the complete original profiles for the collaborative filtering recommendation.

3. Combination of sub-profile recommendations: the items recommended to the sub-profiles of each user need to be combined to generate a single recommendation to the original user. In Section 5.3 we adapt the greedy approach of Section 4 for combining sub-profile recommendations.

5.1 Extraction of Sub-profiles

Given a generic user u, we extract her sub-profiles using a categorization of the items $i \in \mathbf{u}$ in the complete profile. Given a category $c \in C$ we define the associated user sub-profile $\mathbf{u}_c$ as a subset of the profile $\mathbf{u}$ where only items having some relation with category c are included, that is, $w(c, i) > 0$. For instance, in the example of Figure 1, where there are three users and two item categories, six different sub-users are generated. The preference value for items in a user sub-profile are computed as the original preference value $r(u, i)$, weighted by $w(c, i)$. Formally:

$$\mathbf{u}_c := \{i \in \mathbf{u} : w(c, i) > 0\} \tag{6}$$

$$r(u_c, i) := w(c, i) r(u, i) \tag{7}$$

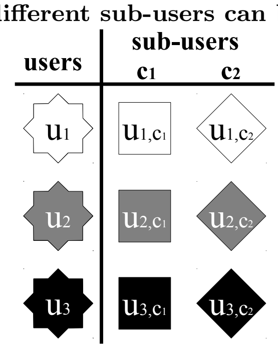
Depending on the type of data we are using, the specific range and interpretation of $r(u, i)$ and $w(c, i)$ may vary. If user preferences are expressed as explicit ratings, then $r(u, i)$ is a rating value provided by user u for item i. This is the case for instance of the MovieLens datasets[1] in the movie recommendation domain, where interactions between users and movies are recorded as rating values from 1 to 5 (the range 1-2 usually expressing dislike, 3 indifference, and 4-5 positive preference).

If the evidence of user interest available to the system consists of transactional records of user interaction (purchases or access), then $r(u, i)$ would be a frequency value representing implicit evidence of user interest for the item. For instance, in the music recommendation domain, $r(u, i)$ is the number of times user u has played item i (a music track, an album or an artist).

Similarly the category weight $w(c, i)$ can be understood in slightly different ways. It can be defined as a binary value expressing whether or not item i belongs to (or has) the category c. This can be appropriate, for instance, in MovieLens data, where movies are classified into e.g. 19 different genres in the 100K version. Even though movies often are assigned more than one genre (e.g. "The Piano" is classified as Drama and Romance), the cardinality is below 2 on average, and as editorial classifications they have very low noise.

This is quite different in crowd-based categorization, where items get a high number of social tags by the crowd. For instance in the music domain, Last.fm[2] records a rather large number of tags associated with artists. In such cases, we define $w(c, i)$ as the probability that item i has category c. In the absence of any specific information about how much an item i should truly reflect a tag c, we assume a uniform distribution among the tags the item has, that is, $w(c, i) = 1/\,|\{c \in C \mid i \text{ tagged with } c\}|$. Thus, by equation 7, $r(u_c, i)$ amounts in this case to the original rating divided by the number of tags of the item i.

Figure 1: Given three users and two item categories, a total of six different sub-users can be defined.



5.2 Generation of Sub-Profile Recommendations

Once the sub-profiles have been extracted, the next step consists of generating recommendations for each of them, as if they represented complete users. As previously mentioned, collaborative filtering algorithms generate recom-

[1] http://movielens.umn.edu

[2] http://www.last.fm/

mendations for a user by combining preferences of similar users. The availability of profiles and sub-profiles – which represent different "sides" of user interests, by item categories– in our approach opens several alternatives for their use in a collaborative filtering approach:

- Do we group sub-profiles generated from the same item category to generate recommendations, or should we rather process them all together?

- Do we use the profiles of the original users to help generate recommendations to the sub-users, or should we not mix profiles and sub-profiles?

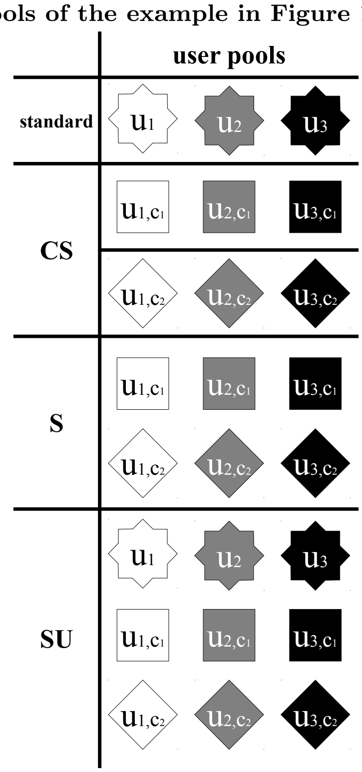
One may foresee that the different alternatives for sets of profiles and sub-profiles that collaborate to the generation of a specific recommendation can provide different outputs. We shall refer to the set of profiles that is taken as input to produce a recommendation for a specific user or sub-user as a *user pool*. To consider the basic variants envisioned in the above questions, we name the following methods (see Figure 2) for creating users pools:

- Method **CS**: a different user pool for each item category c is created only including those sub-profiles derived from c.

- Method **S**: all sub-profiles are included in a single user pool without the original profiles.

- Method **SU**: all sub-profiles and the original profiles are included in a single user pool.

Figure 2: Diagram of the three different methods for user pools of the example in Figure 1.



5.3 Combination of Sub-profile Recommendations

Once the recommendations to the sub-profiles have been generated, the final step consists of combining them to create a diversity-aware recommendation list. This is done by adapting the greedy scheme defined in Section 4 to our diversification based on sub-profiles. Specifically, we replace the probability $p(i|c, u)$ by $p(i|u_c)$, which represents the likelihood of the item i being selected by u_c, that is, the abstract user defined by the sub-profile $\mathbf{u}_c$ representing a drive (a sub-taste of the original user) for items related to the category c. This probability $p(i|u_c)$ is proportional to the score $\hat{r}(u_c, i)$ of the recommendation R_c for the sub-profile $\mathbf{u}_c$:

$$p(i|u_c) = \frac{\hat{r}(u_c, i)}{\sum_{j \in R} \hat{r}(u_c, j)} \quad i \in R_c \tag{8}$$

6. SCALABILITY OF DIVERSIFICATION ALGORITHMS

The different alternatives for forming user pools have potential implications on the computational cost of the recommendation and diversification steps. Along with the user pooling approach, the effect is determined by the cardinality of the relation between items and categories.

From the point of view of the greedy re-ranking approaches, the computational complexity of the objective function is proportional to the number of item categories –more accurately, to the number of categories such that $p(c|u) > 0$ for each user, which we refer to as *sub-user cardinality*. In the datasets we use in our experiments (see Section 7 below), while MovieLens 1M has an average of 17.58 sub-users per user, Last.fm has a much higher average cardinality of 12,007 sub-users per user, which heavily impacts the cost of the re-ranking algorithm.

Moreover, for the recommendation step, the sub-user cardinality of Last.fm has a combinatorial effect whereby for nearly 1,000 original users the resulting total number of sub-profiles is around 12 million, for which a recommendation should be computed. In such situations where the sub-user cardinality may lead to a impractical computational load, we propose a simple and efficient solution: defining sub-profiles for the top N tags of each user (those with a higher $p(c|u)$) – where N should take a much smaller value than the sub-user cardinality. An experiment with this approach is carried out in the next section to test whether this proposal is valid.

7. EXPERIMENTS

In order to test whether our proposal of using sub-profile diversifications is competitive, we use two well-known datasets: the 1M version of the MovieLens collection and an extract from Last.fm provided by Ò. Celma [7]. The MovieLens 1M dataset includes one million ratings (on a 1-5 scale) by 6,040 users for 3,900 movies. The movies are categorized in 19 different genres. The Last.fm dataset includes the full listening history of 992 users till May 2009, comprising 19,150,868 scrobblings (i.e. instances of a user playing a music track) for 176,948 different artists. Using the Last.fm API we have extracted the most popular tags for each artist, resulting in 122,624 different tags for 96,612 artists.

We perform a 5-fold cross validation for training and test in the MovieLens 1M dataset, where only the 500 users with most diverse tastes are given recommendations. By user

Figure 3: MovieLens 1M

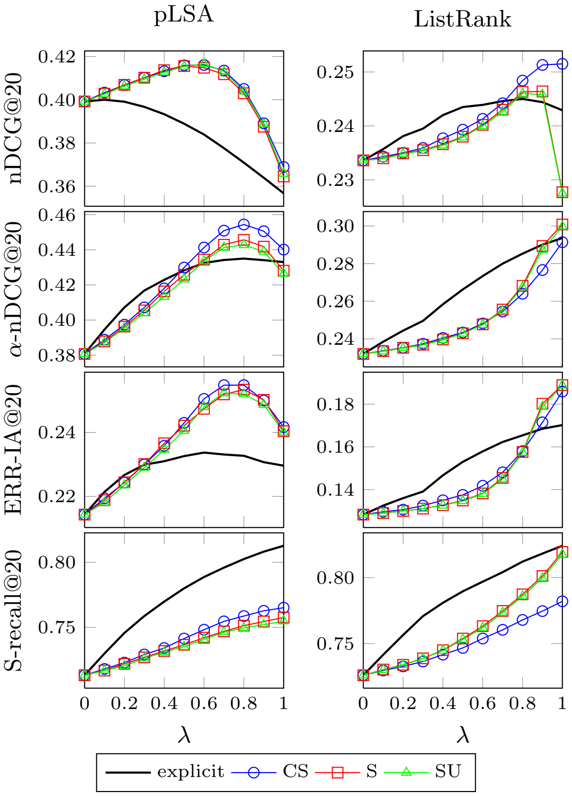


Table 1: Results in MovieLens 1M for xQuAD with $\lambda = 0.5$, $\lambda = 1.0$ and best λ value for each metric using the explicit diversification with genres (expl) and the methods S, SU and CS with sub-profiles. The superscripts a and b indicate statistically significant differences (Wilcoxon signed rank with $p < 0.05$) with respect to the baseline and expl, respectively.

		nDCG	α-nDCG	ERR-IA	S-recall
pLSA		0.3992	0.3807	0.2143	0.7132
$\lambda = 0.5$	**expl**	0.3889^{a}	0.4283^{a}	0.2326^{a}	$\mathbf{0.7800}^{a}$
	CS	$\mathbf{0.4159}^{ab}$	$\mathbf{0.4299}^{a}$	$\mathbf{0.2430}^{ab}$	0.7414^{ab}
	S	0.4156^{ab}	0.4258^{a}	0.2421^{ab}	0.7366^{ab}
	SU	0.4154^{ab}	0.4232^{a}	0.2406^{ab}	0.7351^{ab}
$\lambda = 1.0$	**expl**	0.3567^{a}	0.4330^{a}	0.2296^{a}	$\mathbf{0.8127}^{a}$
	CS	$\mathbf{0.3689}^{ab}$	$\mathbf{0.4401}^{a}$	$\mathbf{0.2417}^{ab}$	0.7651^{ab}
	S	0.3645^{a}	0.4281^{a}	0.2404^{a}	0.7576^{ab}
	SU	0.3658^{a}	0.4264^{a}	0.2400^{a}	0.7548^{ab}
best λ	**expl**	0.4000	0.4350^{a}	0.2337^{a}	$\mathbf{0.8127}^{a}$
	CS	0.4161^{ab}	$\mathbf{0.4544}^{ab}$	$\mathbf{0.2548}^{ab}$	0.7651^{ab}
	S	0.4156^{ab}	0.4457^{a}	0.2534^{ab}	0.7576^{ab}
	SU	$\mathbf{0.4166}^{ab}$	0.4428^{a}	0.2523^{ab}	0.7548^{ab}
ListRank		0.2336	0.2320	0.1283	0.7260
$\lambda = 0.5$	**expl**	$\mathbf{0.2435}^{a}$	$\mathbf{0.2663}^{a}$	$\mathbf{0.1530}^{a}$	$\mathbf{0.7896}^{a}$
	CS	0.2393^{ab}	0.2434^{ab}	0.1375^{ab}	0.7467^{ab}
	S	0.2380^{ab}	0.2430^{ab}	0.1349^{ab}	0.7538^{ab}
	SU	0.2379^{ab}	0.2429^{ab}	0.1347^{ab}	0.7531^{ab}
$\lambda = 1.0$	**expl**	0.2429^{a}	0.2936^{a}	0.1702^{a}	$\mathbf{0.8241}^{a}$
	CS	$\mathbf{0.2515}^{ab}$	0.2914^{a}	0.1859^{ab}	0.7820^{ab}
	S	$0.2277^{\ b}$	$\mathbf{0.3008}^{a}$	$\mathbf{0.1889}^{ab}$	0.8194^{a}
	SU	$0.2275^{\ b}$	0.2997^{a}	0.1888^{ab}	0.8175^{a}
best λ	**expl**	0.2450^{a}	0.2936^{a}	0.1702^{a}	$\mathbf{0.8241}^{a}$
	CS	$\mathbf{0.2515}^{ab}$	0.2914^{a}	0.1859^{ab}	0.7820^{ab}
	S	0.2464^{a}	$\mathbf{0.3008}^{a}$	$\mathbf{0.1889}^{ab}$	0.8194^{a}
	SU	0.2460^{a}	0.2997^{a}	0.1888^{ab}	0.8175^{a}

taste diversity we mean the entropy of the conditional distribution $p(c|u)$ of categories in the training of the user profile. In the Last.fm dataset, we take a single temporal 60-40% split of the scrobblings for training and test.

We use two recommender system baselines: a probabilistic Latent Semantic Analysis (pLSA) recommender [10] and a List-wise Matrix Factorization Recommender (ListRank) [15]. For each baseline, their explicit aspect-based diversification and the three described variants (CS, S, SU) of the sub-profile diversification are generated. In MovieLens 1M the extraction of sub-profiles and the defined aspects of explicit diversification (*expl*) use genres as categories. In the case of Last.fm, given the high cardinality of the categories space, sub-profiles are extracted using the top 20 tags of each user, while the explicit diversification is applied using all tags (*all*) and the top 20 tags of each user (*top*).

We have evaluated the previous recommendations using the *TestItems* methodology described in [4]. In this methodology, given a user, the recommendation algorithm is requested to rank the set of all items for which there is some rating by some user in the test split. The items with a test rating by the target user are taken as relevant for computing the metrics, and the rest as non-relevant. We evaluate all recommendations with nDCG@20, its diversity-aware version α-nDCG@20 [9], the intent-aware expected reciprocal rank ERR-IA@20 [3], and subtopic recall S-recall@20 [20]. The diversity-aware metrics use genres and tags for MovieLens 1M and Last.fm, respectively, as the equivalent of query subtopics for the diversity metrics.

The results for the experiment on the MovieLens 1M are shown in Table 1 and Figure 3, where we select two fixed values of the λ parameter of xQuAD (0.5 as a balance of accuracy-diversity and 1.0 for full diversification) and the best λ value for each combination of diversification and metric. Let us observe first the results for the pLSA recommender. In terms of accuracy (nDCG), *expl* implies a loss or a non-significant gain over the baseline, while sub-profiles variants improve over the baseline significantly for $\lambda = 0.5$ and their best λ, but also decay for $\lambda = 1.0$. For α-nDCG and ERR-IA all the diversifications improve over the baseline, showing that they achieve a much better combination of accuracy and diversity than the original recommendation. Specifically, sub-profile methods show the best results when the best λ is used. In terms of pure subtopic recall, *expl* has always the best results, although sub-profile alternatives also achieve better results than the baseline. As to the ListRank recommender, there are some differences with respect to pLSA. For example, on ListRank, *expl* overperforms the baseline in all cases, and is better than the sub-profile methods with $\lambda = 0.5$. Once we move to $\lambda = 1.0$ and best λ, sub-profile methods become again the best alternatives in terms of nDCG, α-nDCG and ERR-IA, but not for S-recall, where *expl* consistently achieves better results despite the improvements of sub-profile methods over the baseline. In this dataset the three proposed methods for sub-profiles do not show significant differences between them, although in general CS achieves the best results.

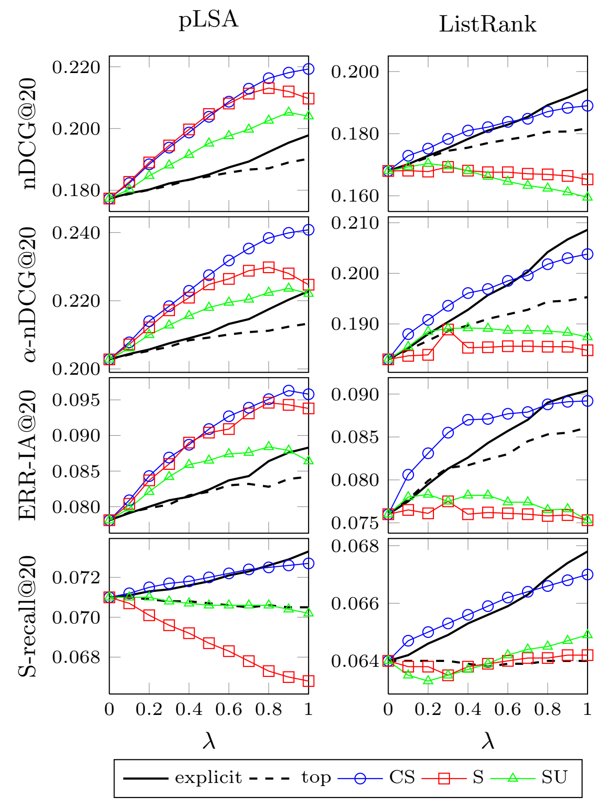
Table 2: Results in Last.fm for xQuAD with $\lambda = 0.5$, $\lambda = 1.0$ and the best λ value for each metric using the explicit diversification with all tags (all) and the top 20 tags (top), and the methods S, SU and CS with sub-profiles also using the top 20 tags. The superscripts a, b and c indicate statistically significant differences (Wilcoxon signed rank with $p < 0.05$) with respect to the baseline, all and top, respectively.

		nDCG	α-nDCG	ERR-IA	S-recall
pLSA		0.1773	0.2028	0.0781	0.0710
$\lambda = 0.5$	**all**	0.1851$^{a\ c}$	0.2106$^{a\ c}$	0.0824$^{a\ c}$	0.0718$^{a\ c}$
	top	0.1845^{a}	0.2092^{a}	0.0821^{a}	0.0707^{ab}
	CS	**0.2038**^{abc}	**0.2275**^{abc}	**0.0909**^{abc}	**0.0720**$^{a\ c}$
	S	0.2047^{abc}	0.2248^{abc}	0.0904^{abc}	0.0687^{abc}
	SU	0.1952^{abc}	0.2180^{abc}	0.0865^{abc}	0.0706^{ab}
$\lambda = 1.0$	**all**	0.1978$^{a\ c}$	0.2228$^{a\ c}$	0.0883$^{a\ c}$	**0.0733**$^{a\ c}$
	top	0.1902^{ab}	0.2133^{ab}	0.0842^{ab}	0.0705^{ab}
	CS	**0.2193**^{abc}	**0.2408**^{abc}	**0.0958**^{abc}	0.0727^{abc}
	S	0.2097^{abc}	0.2247$^{a\ c}$	0.0938^{abc}	0.0668^{abc}
	SU	0.2040^{abc}	0.2221$^{a\ c}$	0.0864^{a}	0.0702^{abc}
best λ	**all**	0.1978$^{a\ c}$	0.2228$^{a\ c}$	0.0883$^{a\ c}$	**0.0733**$^{a\ c}$
	top	0.1902^{ab}	0.2133^{ab}	0.0842^{ab}	0.0710^{ab}
	CS	**0.2193**^{abc}	**0.2408**^{abc}	**0.0963**^{abc}	0.0727^{abc}
	S	0.2131^{abc}	0.2298^{abc}	0.0946^{abc}	0.0710^{abc}
	SU	0.2052^{abc}	0.2236$^{a\ c}$	0.0884$^{a\ c}$	0.0710^{abc}
ListRank		0.1680	0.1830	0.0760	0.0640
$\lambda = 0.5$	**all**	0.1810$^{a\ c}$	0.1957$^{a\ c}$	0.0843$^{a\ c}$	0.0656$^{a\ c}$
	top	0.1770^{ab}	0.1910^{ab}	0.0824^{ab}	0.0638^{ab}
	CS	**0.1821**$^{a\ c}$	**0.1969**$^{a\ c}$	**0.0871**$^{a\ c}$	**0.0659**^{abc}
	S	0.1676^{abc}	0.1854^{abc}	0.0762^{abc}	0.0639^{abc}
	SU	0.1662$^{\ bc}$	0.1891^{a}	0.0782^{a}	0.0639^{abc}
$\lambda = 1.0$	**all**	**0.1943**$^{a\ c}$	**0.2086**$^{a\ c}$	**0.0904**$^{a\ c}$	**0.0678**$^{a\ c}$
	top	0.1817^{ab}	0.1953^{ab}	0.0861^{ab}	0.0640^{ab}
	CS	0.1890^{abc}	0.2038^{abc}	0.0892$^{a\ c}$	0.0670^{abc}
	S	0.1653^{abc}	0.1848^{abc}	0.0753^{abc}	0.0642^{abc}
	SU	0.1595$^{\ bc}$	0.1874^{ab}	0.0753$^{\ bc}$	0.0649^{abc}
best λ	**all**	**0.1943**$^{a\ c}$	**0.2086**$^{a\ c}$	**0.0904**$^{a\ c}$	**0.0678**$^{a\ c}$
	top	0.1817^{ab}	0.1953^{ab}	0.0861^{ab}	0.0640^{ab}
	CS	0.1890^{abc}	0.2038^{abc}	0.0892$^{a\ c}$	0.0670^{abc}
	S	0.1693^{abc}	0.1889^{abc}	0.0775^{abc}	0.0642^{abc}
	SU	0.1704^{abc}	0.1892^{ab}	0.0783^{abc}	0.0649^{abc}

The results for the Last.fm dataset are reported in Table 2 and Figure 4. With the pLSA recommender, sub-profiles always work better than explicit aspects (*all* and *top*) in the diversification algorithm. In terms of α-nDCG, CS obtains, by far, the best results for the three alternatives of λ, where S and SU are quite similar to the optimal of the explicit aspect approach, and they are better than *top*. For ERR-IA, CS and S are very similar and show the best results, but SU is weaker, below *all* in $\lambda = 1.0$ though still better than *top*. Regarding S-recall, CS achieves improvements comparable to those of *all* with respect to the baseline, while the other methods with sub-profiles and *top* obtain similar or much worse results, as is the case of S. With the ListRank recommender, there is a clear observation: methods S and SU do not work, since they do not offer results much different from those of the baseline for all the considered metrics. The other sub-profile method, CS, obtains comparable results to *all* and is always better than *top*, specially in S-recall, where *top* does not improve and CS is similar to *all*.

Overall, results show that sub-profile methods can achieve better results than baseline recommendations, although S and SU do not seem to be very robust and may fail in some situations (as seen for ListRank in Last.fm). Compared to the diversification based on explicit aspects, our methods achieve better results or, at least, competitive ones while ensuring the scalability of the procedure. Specifically, our approach in Last.fm of using only the top 20 tags for each user to extract sub-profiles achieves similar results to the much costlier process of considering all tags in a explicit diversification and is much better than the explicit diversification using only the same top 20 tags.

Figure 4: Last.fm



8. CONCLUSIONS

We have proposed a way of exploiting the diversity of user preferences to enhance recommendations in terms of accuracy and diversity. Our experiments show that this approach is competitive against diversifications based on explicit aspects, particularly so with aspect spaces of high cardinality which raise scalability problems.

The consideration of the different sides of user tastes and behaviors arises as a natural idea which on the one hand aims to take into account people are not one-block entities, and on the other seeks to draw improvements in personalization techniques by reflecting this fact in the models. This is a quite general principle which can be pursued towards multiple potential applications beyond the one addressed here.

We plan to expand this work by considering more complex category spaces from where to extract sub-profiles. Concretely, we think that the combination of different features of items (such as genre, language, location or decade of movies) or a hierarchical structure of categories (such as that of the Amazon web store, where the products are classified in a taxonomy of categories) have potential to be adapted to our approach.

9. ACKNOWLEDGMENTS

This work was supported by the national Spanish projects TIN2011-28538-C02-01 and S2009TIC-1542.

10. REFERENCES

[1] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. on Knowl. and Data Eng.*, 24(5):896–911, May 2012.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, 2009.

[4] A. Bellogín, P. Castells, and I. Cantador. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 333–336, 2011.

[5] A. Bellogín, J. Wang, and P. Castells. Text retrieval methods for item ranking in collaborative filtering. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 301–306, 2011.

[6] 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*, SIGIR '98, pages 335–336, 1998.

[7] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 179–186, 2008.

[8] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 429–436, 2006.

[9] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 659–666, 2008.

[10] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, Jan. 2004.

[11] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[12] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 210–217, 2010.

[13] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 881–890, 2010.

[14] R. L. Santos, C. Macdonald, and I. Ounis. On the role of novelty for search result diversification. *Inf. Retr.*, 15(5):478–502, Oct. 2012.

[15] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 269–272, 2010.

[16] S. Vargas, P. Castells, and D. Vallet. Intent-oriented diversity in recommender systems. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1211–1212, 2011.

[17] S. Vargas, P. Castells, and D. Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 75–84, 2012.

[18] J. Wang. Mean-variance analysis: A new document ranking theory in information retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 4–16, 2009.

[19] J. Wang, S. Robertson, A. P. Vries, and M. J. Reinders. Probabilistic relevance ranking for collaborative filtering. *Inf. Retr.*, 11(6):477–497, Dec. 2008.

[20] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 10–17, 2003.

[21] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 123–130, 2008.

[22] M. Zhang and N. Hurley. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 508–515, 2009.

[23] 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*, WWW '05, pages 22–32, 2005.