Community-Aware Diversification of Recommendations

Mesut Kaya
Insight Centre for Data Analytics
University College Cork, Ireland
mesut.kaya@insight-centre.org

Derek Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
derek.bridge@insight-centre.org

ABSTRACT

Intent-aware methods for recommendation diversification seek to ensure that the recommended items cover so-called *aspects*, which are assumed to define the user's tastes and interests. Most typically, aspects are item features such as movie or music genres. In recent work, we presented a novel intent-aware diversification method, called Subprofile-Aware Diversification (SPAD). In SPAD, aspects are subprofiles of the active user's profile, detected using an item-item similarity method. In this paper, we propose Community-Aware Diversification (CAD), in which aspects are again subprofiles but are detected indirectly through users who are similar to the active user. We evaluate CAD's precision and diversity on four different datasets, and compare it with SPAD and an intent-aware diversification method called xQuAD. We show that on two of the datasets SPAD outperforms CAD, but for the other two CAD outperforms SPAD. For all datasets, both CAD and SPAD achieve higher precision than xQuAD. When it comes to diversity, xQuAD sometimes results in more diverse recommendations but it is more prone to paying for this diversity with decreases in precision. Arguably, SPAD and CAD strike a better balance between the two.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Diversity; intent-aware; subprofiles.

ACM Reference Format:
Mesut Kaya and Derek Bridge. 2019. Community-Aware Diversification of Recommendations. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297280.3297439

1 INTRODUCTION

A recommender system aims to help its users to discover resources that align with their tastes and interests. Early recommender systems research focused on accurately predicting the user's ratings for unseen items or generating, from the unseen items, a top-n list of relevant recommendations. However, it soon became clear that being merely accurate or relevant is not enough [14]. The pursuit of accuracy or relevance alone may result in recommending items that are too similar to each other or too similar to the user's profile. In many domains, recommendations must not only be accurate or relevant, but they should also be novel to the user or serendipitous, and a set of recommendations must be diverse [3, 9]. Here we focus on the diversity of the recommendation set.

Diversity is one response to uncertainty. A recommender cannot be certain of a user's short-term or longer-term interests. Short-term interests are, almost by definition, ephemeral and therefore difficult to predict. The user's profile can reveal her longer-term interests. But there can still be uncertainty both because some user profiles are small and others, while they may not be so small, will contain preferences over different kinds of items. In the face of uncertainty, ensuring that the recommendations are not too similar to each other can make it more likely that one or more of the recommendations will satisfy the user. However, a recommender system must strike a balance between accuracy or relevance on the one hand and diversity on the other hand. Indeed, early work on diversification implied a trade-off between the two: increasing one typically resulted in a decrease in the other [1].

Early work assumes that the diversity of a set of items is an aggregate of the all-pairs dissimilarity of the items within the set, where dissimilarity is measured by distance functions over item meta-data (such as movie genres), item ratings or latent factors. A common approach to producing a diverse set of recommendations is to incrementally construct the set by adding the item with the highest marginal contribution [2]. The marginal contribution is typically a linear combination of the relevance of the item to the user and its dissimilarity to the items in the result set so far. By this approach, the balance between relevance and diversity is controlled by a parameter of the linear combination. In principle, this allows the degree of diversity to be personalized: different users could have different values for this parameter (although this has been rare in practice). Relevance is also personalized (since it is based on the individual user's profile). But the way that diversity is measured is not personalized (since it is based simply on item dissimilarity). This might be problematic. Consider, for instance, a music recommender. For a user who only likes jazz and blues, a set of recommendations containing only jazz and blues music, although low in diversity in terms of item dissimilarity, may possibly be as diverse as the user wishes. Another user who likes jazz, blues and some classical might find a set of recommendations that contains only jazz and blues to be lacking in diversity, even though the diversity measure is the same in both cases. Note that this is not about how important diversity is relative to relevance (which is controlled by the parameter that we mentioned earlier); this is about what it means for a set to be diverse, and how this can vary from one user to another.

More recent work takes a so-called *intent-aware* approach to diversity. Intent-aware diversification seeks to ensure that the set

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297439

of recommended items covers the different tastes and interests revealed by the user's profile; indeed, it seeks to ensure that they are covered to approximately the same degree that they appear in the user's profile. In our examples from earlier, the first user will receive jazz and blues recommendations, roughly in the proportion that they appear in her profile; the second user will receive jazz, blues and classical music recommendations similarly. The most common way to characterize a user's tastes and interests is as a probability distribution over so-called *aspects* of the items. Aspects are either explicit item features (such as movie genres or news categories) or implicit item features such as the latent factors found, e.g., by a matrix factorization recommender. Since the aspect probabilities may differ across users, diversity is personalized to a certain extent.

There is a potential weakness in using item features (such as genres, keywords or tags) for the aspects in intent-aware diversification. In some domains, features may not be readily available. Even when available, they may fail to capture the subtleties of a person's preferences; they may be too coarse; they may be inconsistently applied to the items; and they may be noisy. In recent work, we proposed a new intent-aware diversification framework called Subprofile-Aware Diversification (SPAD). It does not use item features at all. Instead, aspects are subprofiles of a user's profile, and we define subprofiles in terms of item-item similarities on the items' ratings vectors, rather than on item feature [12]. The main contribution of SPAD is that, across multiple datasets, it can improve both accuracy and diversity. This is notable because early approaches to diversification sacrifice accuracy for diversity; even other intent-aware approaches sometimes make this sacrifice.

In this paper, we propose a new method for detecting subprofiles that can be used as the aspects in intent-aware diversification. We call this new approach Community-Aware Diversification (CAD). It uses user-user similarities on user ratings vectors, but it still results in subprofiles, each of which is a subset of the *items* in the user's profile. Thus it explores the idea that a user's community (similar users) correlates with her tastes or interests (subprofiles).

We report the results of offline experiments on four real-world datasets, each belonging to different domains. We analyze the datasets and both the SPAD and CAD subprofiles. The results show that on two datasets CAD performs better than SPAD; on the other two SPAD performs better. In all cases, SPAD and CAD have higher accuracy than an existing intent-aware diversification method (xQuAD), while always increasing the diversity.

The rest of this paper is structured as follows. In the next section, we briefly review related work. Section 3 contains the details of our proposed intent-aware diversification framework, CAD. Details of the datasets, recommenders and methodology in our experiments can be found in Section 4. In Section 5, we analyze the CAD and SPAD subprofiles for each dataset and we give accuracy and diversity results for all four datasets.

2 RELATED WORK

There is a small amount of work in which diversity is explicitly a part of the objective function that the recommender optimizes when generating recommendations, e.g. [8, 16, 18]. But the dominant approach to diversification is greedy re-ranking, e.g. [24].

2.1 Greedy re-ranking

The greedy re-ranking approach starts with a recommendation set RS, generated by a baseline recommender (such as one that uses matrix factorization). Each item in RS has a score, $s(u, i)$, computed by the baseline recommender. $s(u, i)$ measures how relevant item i is to user u. In greedy re-ranking, a ranked list of recommendations RL is built by iteratively selecting items from RS based on an objective function that uses a linear combination whose parameter, λ, balances each item's score with the diversity that would be achieved by adding the items to the already-selected members of RL:

$$f_{obj}(i, RL) = (1 - \lambda)s(u, i) + \lambda \,\mathrm{div}(i, RL) \tag{1}$$

In the past, the diversity of a set of recommendations $\{i\} \cup RL$ was given by the average all-pairs dissimilarity between the items in the set (known as the intra-list distance, ILD [13, 17]), or their sum, or the minimum dissimilarity between i and the members of RL [2, 24]. However, as we argued in the previous section, this measures diversity in the same way for all users.

More recently, under the name of Intent-Aware Diversification, items have been chosen in such a way as to cover the user's tastes and interests, as revealed by her profile. These tastes and interests are modeled by a probability distribution over a set of so-called *aspects*. Most commonly, aspects are item features such as genres or categories, keywords, or tags [22, 23]. For example, in a movie recommender, if a user's profile contains lots of Sci-Fi, a little Romance and a medium amount of Action, then RL can be constructed from RS in a way that tries to preserve the same distribution in RL as there is in the profile: several Sci-Fi movies, a few Action movies and maybe one Romance, if there is room.

2.2 Subprofile-Aware Diversity

Recently, we proposed a new intent-aware diversification method that does not rely on item features [10–12]. Subprofile-Aware Diversification (SPAD) is a form of intent-aware diversification which uses *subprofiles* of the user's profile as its aspects. The subprofiles are not defined in terms of item features but in terms of rating similarity between items. Hence, SPAD may be useful in domains where item features are not readily available or where they suffer from problems in their quality. Not only that, but SPAD is proving to be highly competitive against other approaches, usually achieving highest accuracy out of several intent-aware and non-intent-aware diversification techniques, and always resulting in an increase in accuracy and diversity over the baseline. In other words, SPAD is less prone to trading-off accuracy for diversity.

Since the approach in this paper, CAD, is also based on subprofiles, we will explain SPAD in more detail and also set up some notation that we can use in the rest of the paper.

Consider set of items I and, for user u, a subset of those items $I_u \subseteq I$. We will refer to I_u as the user's liked-item-set: simply, it contains items she has liked. In a recommender that uses unary implicit feedback, for example, then I_u are the items that the user has interacted with (liked, clicked on, purchased, etc.), i.e. ones for which a unary rating, r_{ui}, has been recorded: $I_u = \{i \in I | r_{ui} \neq \perp\}$. In the case of a recommender that uses explicit, numeric ratings (e.g. on 1–5 star or 1–10 star scale), then I_u must be defined in terms

of items the user liked, which will usually involve thresholding the ratings, e.g. on a 1–5 scale, $I_u = \{i | r_{ui} \geq 4\}$.

A user's subprofiles, $\mathcal{S}_u$, are subsets of I_u, each one intended to capture one of the user's tastes or interests. Consider one of these subprofiles, $S \in \mathcal{S}_u$. Then, the set RS is greedily re-ranked using the objective function given as Equation 1 with $div(i, RL) = \text{div}_{\text{SPAD}}(i, RL)$, where:

$$\text{div}_{\text{SPAD}}(i, RL) = \sum_{S \in \mathcal{S}_u} [p(S|u)p(i|u, S) \prod_{j \in RL} (1 - p(j|u, S))] \quad (2)$$

Let $\mathcal{S}_u$ denote all of u's subprofiles. Then, $p(S|u)$ is estimated as:

$$p(S|u) = \frac{|S|}{\sum_{S' \in \mathcal{S}_u} |S'|} \quad (3)$$

Finally, $p(i|u, S)$, the probability of choosing i from a set of recommendations RS given subprofile S of user u, is estimated as:

$$p(i|u, S) = \frac{\mathbb{1}(i, S)s(u, i)}{\sum_{j \in RS} \mathbb{1}(j, S)s(u, j)} \quad (4)$$

where:

$$\mathbb{1}(i, S) = \begin{cases} 1 & \text{if } i \in \bigcup_{j \in S} \text{KNN}(j) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In Equation 5, KNN(j) is the set of j's k-nearest-neighbours in I. In other words, i must be a neighbour of a member of S. Neighbours are computed using cosine similarity on item rating vectors.

What this does not yet explain is how we compute the subprofiles, $\mathcal{S}_u$. This is the part of our work that has undergone most refinement. Early versions of how this is done are in [11, 12]. In this paper, we use a simpler approach, which works better [10]. We create a candidate subprofile for each $i \in I_u$. The candidate subprofile for $i \in I_u$ contains i itself and also $j \in I_u$ if j's nearest-neighbours contain i, i.e. the candidate subprofile for i, S_u^i, is given by:

$$S_u^i = \{j \in I_u | i \in \text{KNN}(j), i \neq j\} \cup \{i\} \quad (6)$$

Again, KNN(j) are the top-k most similar items to j.

The candidate subprofiles are pruned to a final set of subprofiles by excluding those that are wholly contained in any of the others.

It follows that different subprofiles can be of different lengths; and the number of subprofiles differs from user to user; but there can be no more than $|I_u|$ subprofiles because we constructed one candidate subprofile for every $i \in I_u$.

Compared with other intent-aware diversification frameworks, which use item features as aspects, SPAD's set of possible aspects is much larger. There are $2^{|I|}$ different aspects, or $2^{|I|} - 1$ if we exclude the empty subprofile. A particular user can have non-zero probabilities for at most $2^{|I_u|} - 1$ aspects (i.e. subprofiles of her liked-item-set).

SPAD uses item-item similarity to detect the user's subprofiles. Correctly defining user tastes and interests is a complex task and user interactions involve a great deal of uncertainty. In this paper, we ask whether a user's community (her nearest-neighbours) might reveal her tastes and interests. We propose a new intent-aware diversification framework that is also based on subprofiles, but detected instead by using user-user similarity. In the next section, we give details of this new approach.

3 COMMUNITY-AWARE DIVERSITY

In this section, we explain our new approach to diversification in recommender systems, which we call Community-Aware Diversification (CAD). It is a greedy re-ranking approach; it is intent-aware; like SPAD, it uses subprofiles of the user's liked-items-set I_u as aspects; but it identifies subprofiles within I_u by using her nearest-neighbours, i.e. other *users* similar to her, not by using the item's neighbours.

We define a candidate subprofile for each item $i \in I_u$. Hence, the number of candidate subprofiles is $|I_u|$. To simplify, consider user u and one of her k-nearest-neighbours, v. If item i (which we know to be a member of u's liked-item-set) is also a member of v's liked-item-set (I_v), then we take the intersection of u's and v's liked-item-sets, $I_u \cap I_v$. Inevitably, this intersection will contain item i, but it may contain other item's too.

We compute $I_u \cap I_v$ for all of u's neighbours who like i (KNN(u), $i \in I_v$). So now we have up to k sets, one for each $v \in \text{KNN}(u)$ provided $i \in I_v$. We aggregate these sets to give a candidate subprofile. More formally, we have that, for user u, the candidate subprofile that corresponds to item i, S_u^i, is given by:

$$S_u^i = \bigstar_{v \in \text{KNN}(u), i \in I_v} I_u \cap I_v \quad (7)$$

But this leaves open how to do the aggregation of the intersections, $\bigstar$.

One possibility it to take their intersection (an intersection of intersection): an item j is in i's candidate subprofile if it is liked by *all* of the users in KNN(u) who liked i:

$$S_u^i = \bigcap_{v \in \text{KNN}(u), i \in I_v} I_u \cap I_v \quad (8)$$

Instead of using intersection, another possibility is to take the union of the intersections, as follows:

$$S_u^i = \bigcup_{v \in \text{KNN}(u), i \in I_v} I_u \cap I_v \quad (9)$$

in which case an item j is in i's candidate subprofile if it is liked by *any* of the users in KNN(u) who liked i.

With both approaches there are problems. Equation 8 can result in a lot of singleton subprofiles: i might be the only item that the neighbours have in common. There is nothing wrong with singleton profiles if they contain idiosyncratic items. But, in general, our goal is to try to group items into subprofiles that capture tastes and interests, which Equation 8 may fail to do very often.

On the other hand, taking the union, as in Equation 9, results in large subprofiles, perhaps even some for which $S_u^i = I_u$. It may place into the same subprofile items that represent different tastes and interests.

In CAD, we use a unified approach, the q-relaxed set intersection, $q \in [0, 1]$

$$S_u^i = \bigcap_{v \in \text{KNN}(u), i \in I_v}^{\{q\}} I_u \cap I_v \quad (10)$$

where an item j will be in i's candidate subprofile if it is liked by a proportion of at least q of the users in KNN(u) who liked i. When $q = 1.0$ Equation 10 is the same as Equation 8, and when $q = 1/|\{v | v \in \text{KNN}(u), i \in I_v\}|$, it gives the same results as Equation 9.

For CAD, q is a hyperparameter, whose value will be set using a validation set (see Section 4.3). As we will show, it tends to pick quite high values for q, between 0.7 and 1.0, so it tends to be closer to Equation 8 than Equation 9.

What we have at this stage are candidate subprofiles, one per $i \in I_u$. As in SPAD, the final step is to prune the candidate subprofiles: we eliminate any that are subsets of the others. The remaining subprofiles are treated as aspects in Equation 2.

Before moving on to the experiments, we will mention two other approaches that did not work and that we discarded. In one of the rejected approaches, instead of using proportion q in Equation 10, we tried an integer threshold θ, requiring items to be members of least θ of the set intersections. We also tested a variation of CAD in which aspects were not sets of items (subprofiles of I_u); rather, we used neighbours more directly than is done in Equation 7 — aspects were sets of users, nearest-neighbours who liked i, $S_u^i = \{v \in \text{KNN}(u) \mid i \in I_v\}$. Neither of these two other approaches worked well enough for us to show their results in this paper.

4 EXPERIMENTS

We have evaluated CAD in offline experiments, which we describe in this section.

4.1 Datasets

The datasets we use are the MovieLens 1M dataset,[1] the LastFM dataset,[2] the LibraryThing dataset [5], and a Facebook dataset [6]. We modify the MovieLens 1M, LastFM and LibraryThing datasets in the same way as in [9]. For example, the listening event frequencies in the LastFM dataset are converted into ratings on the scale 1–5 and the dataset is augmented with additional meta-data (namely, user-generated tags). For LibraryThing, we retrieved a maximum of the 10 most popular tags for every book and kept the tags that appeared in the profiles of at least 10 books. The Facebook dataset contains implicit ratings ('likes') unlike the rest of the datasets we use in our experiments and it is multi-domain since it consists of user preferences for Facebook pages pertaining to movies, music and books. For the Facebook dataset, we keep users who have at least 10 likes and pages that are liked by at least 10 users; for meta-data, we use categories crawled from DBpedia [6].

Table 1 summarizes the characteristics of the datasets.

Table 1: Datasets

MovieLens	6040 users	3706 movies	~1M ratings
	18 genres in total; avg. 1.65 per movie		
	avg. 165.6 (σ = 192.74) movies per user		
	avg. 95.25 (σ = 105.0) liked movies per user		
LastFM	992 users	7280 artists	~500k ratings
	71833 tags in total; avg. 8 per artist		
	avg. 515.94 (σ = 475.14) artists per user		
	avg. 195.35 (σ = 194.95) liked artists per user		
LibraryThing	7279 users	37232 books	~750K ratings
	4800 tags in total; avg. 9.08 per book		
	avg. 102.95 (σ = 132.68) books per user		
	avg. 66.43 (σ = 82.56) liked books per user		
Facebook	104178 users	15374 pages	~4.75M ratings
	33,660 categories in total; avg. 10.62 per page		
	avg. 45.94 (σ = 49.43) pages per user		
	avg. 45.94 (σ = 49.43) liked pages per user		

4.2 Recommender systems

We compare CAD with SPAD and one other intent-aware diversification technique, xQuAD [20], using the implementation which is available in the RankSys library.[3] We have also compared CAD and SPAD with a number of other intent-aware diversification methods (RxQuAD [21], SxQuAD & SRxQuAD [19], and c-pLSA [23]) and a more classical (non-intent-aware) method, MMR [2]. In this paper, we only show the results for xQuAD since, across all datasets, it was the most competitive of these other techniques.

All of these approaches to diversification use greedy re-ranking, therefore they need a baseline recommender, whose recommendation sets are re-ranked. In this paper, we report results for two baseline recommenders: a fast alternative least-squares matrix factorization recommender (MF) [15] and probabilistic Latent Semantic Analysis (pLSA) [7]. Again we use their RankSys implementations. Two baselines paired with four approaches to re-ranking (CAD, SPAD, xQuAD and none at all) gives eight systems to compare on each dataset.

Both baseline algorithms have hyperparameters. In our cross-validation methodology (next section), as in [21], we select hyperparameter values that maximize precision for top-10 recommendations. For pLSA and MF, we choose the number of latent factors (d) from $V = \{10, 30, 50, \ldots, 330, 350\}$ and MF's confidence level (α) from $\{1, 2, \ldots, 10\}$. This resulted in the following values:

- pLSA: $d = 50$ for MovieLens; $d = 30$ for LastFM; $d = 270$ for LibraryThing; $d = 30$ for Facebook.
- MF: $d = 30, \alpha = 1.0$ for MovieLens; $d = 30, \alpha = 1.0$ for LastFM; $d = 330, \alpha = 1.0$ for LibraryThing; $d = 50, \alpha = 10.0$ for Facebook.

CAD and SPAD also have hyperparameters. SPAD has a hyperparameter, k (Equation 6), the number of neighbouring items (referred to below as k_{IB}), whose value we select from V. CAD also has hyperparameter k (Equation 7), this time the number of neighbouring users (referred to below as k_{UB}). For the MovieLens, LastFM and LibraryThing datasets, we also select its value from V. However, for the Facebook dataset, we found we needed a greater range of candidate values and so we tested with values up to 1500. CAD also has hyperparameter, q (Equation 10), used in the relaxed set intersection, whose values we select from $[0.1, 0.2, \ldots, 1.0]$. Finally, Equation 5, which is part of both CAD and SPAD, also has hyperparameter k (referred to below as k_{ind}) and its value is also selected from set V.

The values selected for the MovieLens dataset are:

- pLSA: $k_{IB} = 10$, $k_{ind} = 50$ for SPAD; $k_{UB} = 150$, $k_{ind} = 10$, $q = 0.7$ for CAD.
- MF: $k_{IB} = 10$, $k_{ind} = 30$ for SPAD; $k_{UB} = 170$, $k_{ind} = 10$, $q = 0.7$ for CAD.

The values selected for the LastFM dataset are:

[1] http://grouplens.org/datasets/movielens/
[2] http://www.dtic.upf.edu/ocelma/MusicRecommendationDataset/lastfm-1K.html
[3] https://github.com/RankSys

- pLSA: $k_{IB} = 30$, $k_{ind} = 10$ for SPAD; $k_{UB} = 130$, $k_{ind} = 10$, $q = 0.9$ for CAD.
- MF: $k_{IB} = 50$, $k_{ind} = 10$ for SPAD; $k_{UB} = 130$, $k_{ind} = 10$, $q = 1.0$ for CAD.

The values selected for the LibraryThing dataset are:

- pLSA: $k_{IB} = 10$, $k_{ind} = 10$ for SPAD; $k_{UB} = 150$, $k_{ind} = 10$, $q = 1.0$ for CAD.
- MF: $k_{IB} = 30$, $k_{ind} = 10$ for SPAD; $k_{UB} = 170$, $k_{ind} = 10$, $q = 0.8$ for CAD.

The values selected for the Facebook dataset are:

- pLSA: $k_{IB} = 10$, $k_{ind} = 10$ for SPAD; $k_{UB} = 1500$, $k_{ind} = 10$, $q = 0.8$ for CAD.
- MF: $k_{IB} = 10$, $k_{ind} = 10$ for SPAD; $k_{UB} = 1500$, $k_{ind} = 10$, $q = 0.8$ for CAD.

4.3 Methodology

In our experiments, we randomly partition the ratings in each dataset into training, validation and test sets such that 60% of each user's ratings are in the training set, 20% of them are in the validation set and 20% are in the test set. Results are averaged over five runs with different random splits. In experiments, we measure accuracy using precision and diversity. We measure diversity using α-nDCG [4], which is a redundancy-aware version of nDCG, except in Section 5.4, where we also use ILD.

We emphasize that all of the hyperparameter values are found using the validation set. We select hyperparameter values for each baseline recommender that optimize precision on the validation sets [21]. Then, for each user, we generate a recommendation set RS, where $|RS| = 100$ using the baseline recommender with its best hyperparameter values. We re-rank RS to produce ranked list RL using each of the re-ranking methods with each of their combinations of hyperparameter values. Then, from each RL, we select the top-N recommendations, $N = 10$. Finally, for each re-ranking method, we select hyperparameter values that give the best α-nDCG on the validation set.

Now we train the baselines using their selected hyperparameter values on the union of the training and validation sets and, for each user, generate a recommendation set RS, where $|RS| = 100$. Then, we re-rank each RS to produce ranked lists RL using each of the re-ranking methods with their selected hyperparameter values. Then, from each RL, we select the top-N recommendations, $N = 10$ and measure precision and α-nDCG on the test set.

5 RESULTS

We divide this section into four: first we analyze the subprofiles that CAD and SPAD find in each dataset; next we give results that compare CAD with SPAD and xQuAD; then, we show the results for different values of λ; lastly, we show the trade-off between precision and diversity.

5.1 Analysis of the subprofiles

In this section we compare the subprofiles detected by CAD and SPAD on each dataset. Table 2 shows the average number of subprofiles per user and the average length of the subprofiles. We also show the average similarity of the subprofiles with each other,

Table 2: Subprofile statistics

Movie Lens	SPAD	avg. 51.09 (σ = 61.5) subprofiles per user avg. len of subprofiles is 7.78 (σ = 6.25) avg. sim of subprofiles is 0.0379
	CAD	avg. 60.22 (σ = 76.39) subprofiles per user avg. len of subprofiles is 34.01 (σ = 25.28) avg. sim of subprofiles is 0.4007
Last FM	SPAD	avg. 134.37 (σ = 161.41) subprofiles per user avg. len of subprofiles is 30.08 (σ = 28.11) avg. sim of subprofiles is 0.1045
	CAD	avg. 52.04 (σ = 29.95) subprofiles per user avg. len of subprofiles is 16.62 (σ = 35.81) avg. sim of subprofiles is 0.0523
Library Thing	SPAD	avg. 32.49 (σ = 46.71) subprofiles per user avg. len of subprofiles is 8.8 (σ = 10.72) avg. sim of subprofiles is 0.044
	CAD	avg. 21.55 (σ = 12.41) subprofiles per user avg. len of subprofiles is 6.1 (σ = 9.85) avg. sim of subprofiles is 0.059
Face book	SPAD	avg. 24.29 (σ = 25.83) subprofiles per user avg. len of subprofiles is 5.04 (σ = 5.66) avg. sim of subprofiles is 0.0208
	CAD	avg. 32.59 (σ = 31.23) subprofiles per user avg. len of subprofiles is 4.85 (σ = 6.63) avg. sim of subprofiles is 0.101

which, for user u, is an all-pairs average:

$$\text{spsim}(u) = \frac{\sum_{S \in \mathcal{S}_u} \sum_{S' \in \mathcal{S}_u, S \neq S'} \text{jsim}(S, S')}{|\mathcal{S}_u|(|\mathcal{S}_u| - 1)} \tag{11}$$

where here $\mathcal{S}_u$ is the *final* set of subprofiles for u. Subprofiles are just sets of items (movies, books, etc) and so the similarity between two subprofiles, $\text{jsim}(S, S')$ that we want here is simply how much they overlap, for which Jaccard similarity ($\text{jsim}(S, S') = \frac{|S \cap S'|}{|S \cup S'|}$) seems appropriate.

Consider, the MovieLens dataset first. On average, SPAD extracts ~51 subprofiles per user and they consist of ~8 movies. CAD extracts more subprofiles (~60) and those subprofiles has a lot more movies (~34) compared with SPAD's subprofiles; CAD's subprofiles are also much more similar to each other.

Next, consider the LastFM dataset. SPAD extracts an average of ~134 subprofiles, having ~30 artists, per user. CAD extracts fewer subprofiles (~52) with a little over half as many artists (~17). SPAD's subprofiles are more similar to each other than CAD's.

Then, consider the LibraryThing dataset. There are on average ~32 subprofiles per user, which have ~9 books, using SPAD. Using CAD gives fewer subprofiles (~22), having fewer books (~6). SPAD subprofiles are not very similar to each other, and neither are CAD subprofiles.

Finally, consider the Facebook dataset. Using CAD results in more subprofiles (~33) than SPAD (~24), and they are more similar to each other. The average length of the subprofiles are almost the same (~5).

Table 3: Results using MF as the baseline. The best result for each metric is highlighted in bold for each block. The value of λ that optimizes α-nDCG for each baseline and re-ranking strategy is given. All of the results are statistically significant with respect to the MF baseline (Wilcoxon signed rank with $p < 0.05$). For CAD and SPAD, if improvements over xQuAD are statistically significant, they are highlighted with ▲; and if their improvements over each other are statistically significant, they are marked with △

		Metrics		% change over baseline	
	λ	Precision	α-nDCG	Precision	α-nDCG
MovieLens					
MF		0.2916	0.3197		
xQuAD	0.5	0.2739	**0.3668**	−6.08%	14.72%
SPAD	0.4	**0.3005**▲△	0.3351	3.03%	4.81%
CAD	0.3	0.2982▲	0.3354	2.27%	4.89%
LastFM					
MF		0.4654	0.4244		
xQuAD	0.3	0.4701	**0.4354**	1.01%	2.61%
SPAD	0.2	**0.4742**▲	0.4296△	1.9 %	1.24%
CAD	0.3	0.472	0.4275	1.42%	0.75%
LibraryThing					
MF		0.1733	0.2412		
xQuAD	0.5	0.1866	**0.264**	7.7 %	9.44%
SPAD	0.4	0.1896▲	0.2588	9.4 %	7.28%
CAD	0.4	**0.1899**▲	0.2603△	9.59%	7.92%
Facebook					
MF		0.1341	0.1613		
xQuAD	0.4	0.13	**0.1791**	−3.0 %	10.99%
SPAD	0.4	0.1428▲	0.1721	6.53%	6.7 %
CAD	0.4	**0.1436**▲△	0.1758△	7.74%	8.82%

Table 4: Results using pLSA as the baseline. The best result for each metric is highlighted in bold for each block. The value of λ that optimizes α-nDCG for each baseline and re-ranking strategy is given. All of the results are statistically significant with respect to the pLSA baseline (Wilcoxon signed rank with $p < 0.05$). For CAD and SPAD, if improvements over xQuAD are statistically significant, they are highlighted with ▲; and if their improvements over each other are statistically significant, they are marked with △

		Metrics		% change over baseline	
	λ	Precision	α-nDCG	Precision	α-nDCG
MovieLens					
pLSA		0.2639	0.2842		
xQuAD	0.7	0.2456	**0.3428**	−6.93%	20.61%
SPAD	1.0	**0.2803**▲△	0.3171	6.2 %	11.57%
CAD	0.7	0.2731▲	0.316	3.48%	11.19%
LastFM					
pLSA		0.3804	0.3426		
xQuAD	0.5	0.41	0.3847	7.78%	12.28%
SPAD	0.6	**0.4299**▲△	**0.3878**▲△	13.0 %	13.19%
CAD	0.5	0.4124	0.3738	8.39%	9.09%
LibraryThing					
pLSA		0.0965	0.1376		
xQuAD	0.8	0.1233	0.1816	27.76%	31.94%
SPAD	0.6	0.1407▲	0.1937▲	45.8 %	40.75%
CAD	0.7	**0.1423**▲	**0.1983**▲△	47.36%	44.1 %
Facebook					
pLSA		0.1028	0.1217		
xQuAD	0.6	0.0996	0.148	−3.19%	21.65%
SPAD	0.7	0.1201▲	0.1445	16.78%	18.76%
CAD	0.8	**0.1209**▲△	**0.1501**▲△	17.6 %	23.36%

5.2 Results for different algorithms

In this section, we compare CAD with SPAD, xQuAD and the baseline algorithms (no re-ranking).

The results for the experiments that use MF as the baseline algorithm are in Table 3. In each block of the table, results for the MF baseline are presented first, and then results for each re-ranking method are given. For each method, we report the results using the value of λ that gives highest α-nDCG on the validation set.

Consider precision first. For all four datasets, CAD and SPAD have higher precision than xQuAD and the MF baseline. Their advantage over the MF baseline is statistically significant in all cases, and their advantage over xQuAD is also statistically significant in all but the case of LastFM. For the MovieLens and LastFM datasets, SPAD has highest precision; for the LibraryThing and Facebook datasets, CAD has the highest precision. However, these differences are only statistically significant in the case of MovieLens and Facebook. These results show that, for these four datasets, using subprofiles as aspects (CAD and SPAD) in intent-aware re-ranking of MF recommendations gives high precision.

Next, consider the diversity metric, α-nDCG. xQuAD is the best re-ranking method for all datasets. But note that CAD and SPAD are at a disadvantage. α-nDCG is a metric that measures diversity in terms of explicit item features. CAD and SPAD make no use of item features at all, whereas xQuAD uses them as aspects. Indeed, α-nDCG is a metric that has some similarities to what is used for re-ranking in xQuAD. Even so, CAD and SPAD have statistically significantly higher diversity than the MF baseline for all datasets. Besides, if we look at the percentage changes, for the MovieLens and Facebook datasets, xQuAD achieves highest diversity at the expense of a decrease in precision: it trades-off accuracy for diversity. CAD and SPAD, on the other hand, always increase both accuracy and diversity. Compared with each other, CAD has higher diversity than SPAD for all but the LastFM dataset where SPAD is statistically higher than CAD.

The results for the experiments that use pLSA as the baseline algorithm are in Table 4. Comparing Tables 4 and 3, we see that pLSA has lower precision and α-nDCG than MF on all four datasets, and in no case does re-ranking pLSA give a higher results than the *corresponding* result for MF. Otherwise, the story is very similar. Here, SPAD has highest precision for the MovieLens and LastFM datasets, and CAD the second highest. For the LibraryThing and Facebook datasets, CAD has the highest precision, and SPAD the second highest. Again, despite making no use of explicit features,

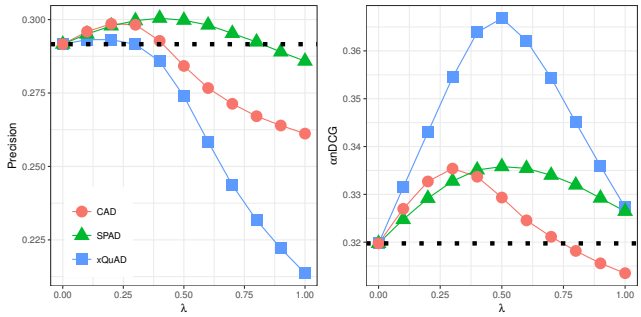

Figure 1: MovieLens dataset, precision and α-nDCG values for varying λ. Values for MF are shown by dotted lines.

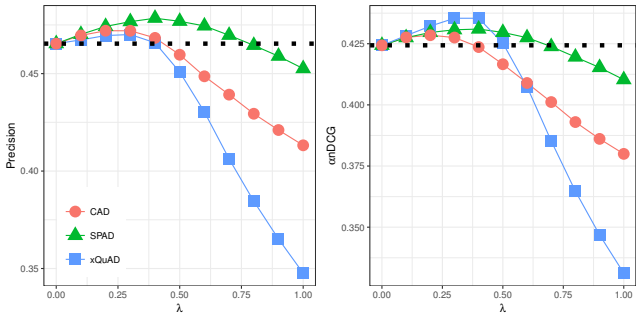

Figure 2: LastFM dataset, precision and α-nDCG values for varying λ. Values for MF are shown by dotted lines.

CAD and SPAD always increase diversity and, in fact, now for the LastFM dataset SPAD gives the highest α-nDCG and for the LibraryThing and Facebook datasets CAD gives the highest α-nDCG. Only in the MovieLens dataset does xQuAD achieve the highest diversity, again at the expense of a decrease in accuracy.

5.3 Results for different values of λ

Here, we look at the effect of parameter λ, which controls the balance between relevance and diversity in Equation 1. The results we have shown so far use whichever values for λ give highest α-nDCG. Instead, here we plot precision and α-nDCG on the test set for different values of λ; see Figures 1, 2, 3 and 4.

We see that the results we discussed in Section 5.2 are fairly robust over different values for λ. For example, for the MovieLens dataset (Figure 1), SPAD's precision is comparable with, or higher than, CAD's and xQuAD's, only becoming worse than the baseline for very high values of λ. For α-nDCG, SPAD is competitive with CAD for low values of λ and then outperforms CAD. xQuAD has high α-nDCG for many values of λ but soon suffers from decreases in precision. Results for LastFM (Figure 2) are similar but less marked. For LibraryThing (Figure 3), SPAD and CAD switch places and so it is CAD that does best across different values of λ, and for higher values of λ xQuAD is competitive with CAD since, for these values, it has higher α-nDCG and competitive precision. For Facebook (Figure 4), xQuAD is not competitive for any values of λ: even where its α-nDCG is a little higher, this comes at the cost of large drops in precision. Across different values of λ, SPAD and CAD perform quite closely on this dataset, although we know from Section 5.2 that, at $\lambda = 0.4$, CAD outperforms SPAD on both metrics.

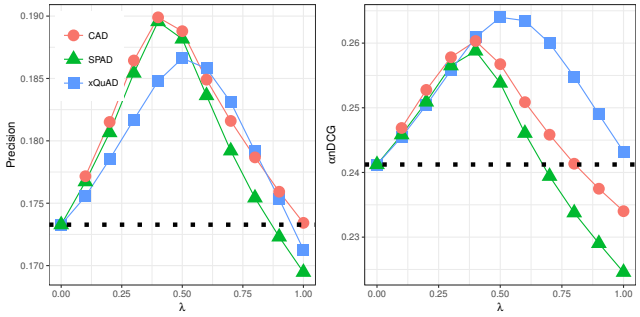

Figure 3: LibraryThing dataset, precision and α-nDCG values for varying λ. Values for MF are shown by dotted lines.

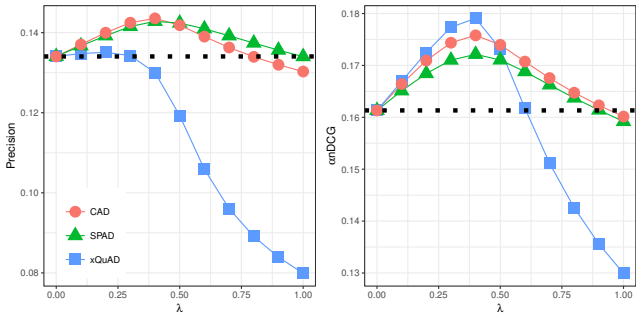

Figure 4: Facebook dataset, precision and α-nDCG values for varying λ. Values for MF are shown by dotted lines.

5.4 Results that reveal trade-offs

In this section, we show the trade-off between relevance and diversity for the re-ranking methods. Figure 5 plots their precision against diversity, where diversity is measured by α-nDCG. The dotted lines show the precision and diversity of the MF basseline, dividing each subfigure into four quadrants. The 'sweet spot' is the top-right quadrant, where both precision and diversity are higher than the baseline. We can see that, for all datasets, CAD and SPAD more often occupy this 'sweet spot' because they are less likely to trade off precision for diversity.

Figure 6 shows the same, this time measuring diversity using ILD, which we mentioned earlier. It is worth showing these results because ILD is in some sense a 'purer' measure of diversity than α-nDCG, which mixes ranking and diversity. The results are similar to those in Figure 5, with CAD and SPAD more likely to increase both precision and diversity. The one exception is the LastFM dataset, where CAD and SPAD have lower values for ILD. CAD and SPAD are again at a disadvantage because, unlike xQuAD, they make no use of item features but ILD, like α-nDCG, measures diversity with respect to these features.

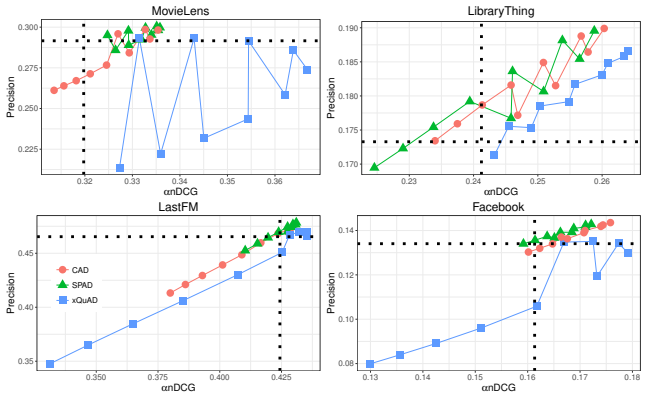

Figure 5: For all datasets, precision versus α-nDCG for different λ. Values for MF are shown by dotted lines.

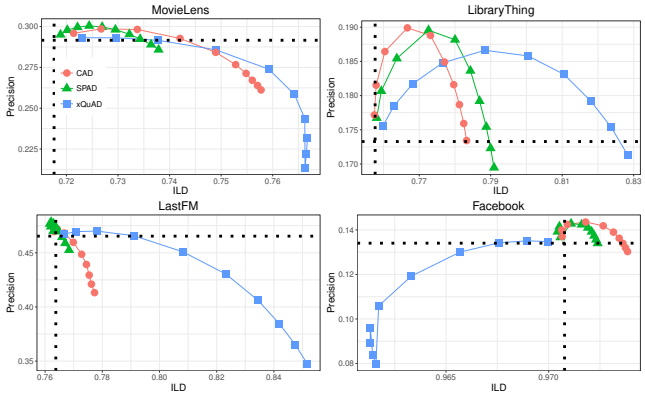

Figure 6: For all datasets, precision versus ILD for different λ. Values for MF are shown by dotted lines.

6 CONCLUSIONS

In this paper, we have presented a new approach to recommendation set diversity that we call Community-Aware Diversification (CAD). It is an intent-aware approach and uses subprofiles of the items that the user likes as its aspects, as does Subprofile-Aware Diversification (SPAD). CAD detects subprofiles using a user-user similarity approach, unlike SPAD, which uses an item-item similarity approach. We compare performance of CAD to SPAD and to xQuAD [20] on four datasets. We find that, CAD and SPAD produce recommendations that are always the most accurate. We also show that CAD and SPAD are less prone to trading-offs accuracy for diversity. In some cases using CAD subprofiles performs better than using SPAD subprofiles; in other cases, the reverse is the case. We conclude that it is meaningful and useful to define subprofiles indirectly through a user's community (her nearest neighbours).

ACKNOWLEDGMENTS

This paper emanates from research supported by a grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 which is co-funded under the European Regional Development Fund.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2009. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In *Procs. of the 19th Workshop on Information Technologies and Systems*. 79–84.
[2] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Procs. of the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval*. ACM, 335–336.
[3] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook (2nd edition)*, F. Ricci and others (Eds.). Springer, 881–918.
[4] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Procs. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 659–666.
[5] Maarten Clements, Arjen P de Vries, and Marcel JT Reinders. 2008. Optimizing Single Term Queries using a Personalized Markov Random Walk over the Social Graph. In *Workshop on Exploiting Semantic Annotations in Information Retrieval*.
[6] Ignacio Fernández-Tobías, Paolo Tomeo, Iván Cantador, Tommaso Di Noia, and Eugenio Di Sciascio. 2016. Accuracy and iversity in Cross-domain Recommendations for Cold-start Users with Positive-only Feedback. In *Procs. of the 10th ACM Conference on Recommender Systems*. ACM, 119–122.
[7] Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems* 22, 1 (2004), 89–115.
[8] Neil J Hurley. 2013. Personalised ranking with diversity. In *Procs. of the 7th ACM Conference on Recommender Systems*. ACM, 379–382.
[9] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2016), 2:1–2:42.
[10] Mesut Kaya. forthcoming. *Subprofile Aware Diversificaton of Recommendations*. Ph.D. Dissertation. University College Cork, Ireland.
[11] Mesut Kaya and Derek Bridge. 2017. Intent-Aware Diversification using Item-Based SubProfiles. In *Procs. of the Poster Track of the 11th ACM Conference on Recommender Systems*, Domonkos Tikk and Pearl Pu (Eds.). CEUR Workshop Proceedings, vol-1905.
[12] Mesut Kaya and Derek Bridge. 2018. Accurate and Diverse Recommendations Using Item-Based SubProfiles. In *Procs. of the Thirty-First International Florida Artificial Intelligence Research Society Conference*. AAAI, 462–467.
[13] John Paul Kelly and Derek Bridge. 2006. Enhancing the diversity of conversational collaborative recommendations: a comparison. *Artificial Intelligence Review* 25, 1-2 (2006), 79–95.
[14] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 1097–1101.
[15] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast ALS-based Matrix Factorization for Explicit and Implicit Feedback Datasets. In *Procs. of the Fourth ACM Conference on Recommender Systems*. ACM, 71–78.
[16] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Procs. of the 10th ACM Conference on Recommender Systems*. ACM, 15–22.
[17] Barry Smyth and Paul McClave. 2001. Similarity vs. diversity. In *Procs. of the International Conference on Case-Based Reasoning*. Springer, 347–361.
[18] Ruilong Su, Li'Ang Yin, Kailong Chen, and Yong Yu. 2013. Set-oriented personalized ranking for diversified top-n recommendation. In *Procs. of the 7th ACM Conference on Recommender Systems*. ACM, 415–418.
[19] Saúl Vargas and Pablo Castells. 2013. Exploiting the diversity of user preferences for recommendation. In *Procs. of the 10th Conference on Open Research Areas in Information Retrieval*. 129–136.
[20] Saul Vargas, Pablo Castells, and David Vallet. 2011. Intent-oriented diversity in recommender systems. In *Procs. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1211–1212.
[21] Saúl Vargas, Pablo Castells, and David Vallet. 2012. Explicit relevance models in intent-oriented information retrieval diversification. In *Procs. of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 75–84.
[22] Saúl Vargas Sandoval. 2015. *Novelty and Diversity Evaluation and Enhancement in Recommender Systems*. Ph.D. Dissertation. Universidad Autónoma de Madrid, Spain.
[23] Jacek Wasilewski and Neil Hurley. 2016. Intent-Aware Diversification Using a Constrained PLSA. In *Procs. of the 10th ACM Conference on Recommender Systems*. ACM, 39–42.
[24] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Procs. of the 14th International Conference on World Wide Web*. ACM, 22–32.