Accurate and Diverse Recommendations Using Item-Based SubProfiles*

Harshil Shah – 202311056
Ananya Adarsh – 202318027
Siddharth Kadam – 202318015
Sreejesh S Nair - 202318001

Background

In many approaches to recommendation diversification, a recommender scores items for relevance and then re-ranks them to balance relevance with diversity.

The paper proposes a form of intent-aware diversification, which they call **SPAD** (SubProfile-Aware Diversification), and a variant called **RSPAD** (Relevance-based SPAD). The aspects they use in SPAD and RSPAD are **subprofiles** of the user's profile. They are not defined in terms of explicit or implicit features

They compare their methods to other forms of intent-aware diversification and found that SPAD and RSPAD always improve accuracy (as measured by precision) and diversity (as measured by α-nDCG)

Introduction

It has long been recognized that it is not enough for recommendations to be accurate or relevant. Diversity is one response to uncertainty.

Early work, within both RS and IR, measures the diversity of a set of items as an aggregate of the all-pairs dissimilarity of the items within the set.

The assumption in this early work is that a set of items that are dissimilar to each other is more likely to contain one or more items that satisfy the user’s current needs or interests.

More recently within IR, there has been a body of research into what has been called intent-aware diversification.

Introduction

In intent-aware diversification, diversity is formulated in terms of coverage of aspects, where aspects are either explicit such as movie genres or implicit such as the latent factors found during matrix factorization.

The idea is to cover the different tastes or interests revealed by the user’s profile.

The most common way to characterize a user’s tastes is as a probability distribution over so-called aspects of the items. The same set of aspects is used across all users.

The paper proposes a new intent-aware diversification framework based on user subprofiles, rather than item features.

This is advantageous because item features, such as genres, do not necessarily fully represent a user’s tastes or interests and are not available in every recommendation domain

Subprofiles (Brief Overview)

A subprofile is a subset of the items in a user's profile, each such subprofile representing one of the user's distinct tastes,

Unlike the aspects used in earlier work, which are global across the set of users, subprofiles differ from user to user, making for a more personalized form of diversification.

Related Work

The dominant approach to diversification is greedy re-ranking.

Greedy re-ranking :

The greedy re-ranking approach assumes the existence of a conventional recommender algorithm (*baseline recommender*), which, for user **u**, produces a set of recommended items, **RS**, and, for each item **i** in RS, a relevance score, **s(u,i)** — the predicted relevance of recommended item i to user u.

The greedy algorithm re-ranks RS by iteratively inserting into ordered result list **RL** the item i from RS that maximizes a function, **f_obj (i, RL)**.

Greedy re-ranking (cont.)

f_obj is usually defined as a linear combination of the item's relevance score and the contribution item i makes to the diversity of RL, **div(i, RL)**, the trade-off between the two being controlled by a parameter **λ** ($0 \leq \lambda \leq 1$):

Algorithm 1 Greedy re-ranking algorithm

Input: RS, set of recommendations for user u, each with relevance score
Output: RL, ranked list containing all items in RS

$RL \leftarrow [\,]$
while $|RS| > 0$ **do**
  $i^* \leftarrow \arg\max_{i \in RS \setminus RL} f_{obj}(i, RL)$
  delete i^* from RS
  append i^* to the end of RL
return RL

Greedy re-ranking (cont.)

Most commonly, **div(i, RL)** is computed as the average (or sum) of the all-pairs intra-list distances (**ILD**).
The distance between items, can be calculated from meta-data such as movie genres or book categories or from item ratings data.

The final recommendation comprises the **top-N** members of the re-ranked list, **RL**, where **N < |RL|**.

Re-ranking using the ILD can result in a top-N that comprises items that are dissimilar to each other.

The assumption behind this form of diversification is that dissimilar items will address the different interests of the user, but there is nothing in the operation of the system to explicitly ensure this.

Item-based Top-N Recommendation

These algorithms use **item-to-item** similarities to compute the relations between the different items.

The primary motivation behind these algorithms is the fact that a customer is more likely to purchase items that are similar to the items that he/she has already purchased in the past; thus, by analyzing historical purchasing information (as represented in the user–item matrix) we can automatically identify these sets of similar items and use them to form the top-N recommendations.

At a high-level, these algorithms consist of two distinct components.

The first component builds a model that captures the **relations** between the different items, whereas the second component applies this precomputed model to derive the top-N recommendations for an active user.

Item-based Top-N Recommendation (cont.)

The model used by the item-based top-N recommendation algorithm is constructed using the algorithm given in next slide.

The input to this algorithm is the **n × m** *user–item matrix* **R** and a parameter k that specifies the number of item-to-item similarities that will be stored for each item.

The output is the model itself, which is represented by an **m × m** matrix **M** such that the j th column stores the k most similar items to item j .

In particular, if **Mi, j > 0,** then the ith item is among the k most similar items of j and the value of Mi, j indicates the *degree of similarity* between items j and i.

Item-based Top-N Recommendation (cont.)

Algorithm 4.1: $\text{BUILDMODEL}\ (R, k)$

$$
\begin{array}{l}
\textbf{for } j \to 1 \textbf{to } m \\
\textbf{do } \left\{
\begin{array}{l}
\textbf{for } i \to 1 \textbf{to } m \\
\quad \textbf{do } \left\{
\begin{array}{l}
\textbf{if } i \neq j \\
\quad \textbf{then } \mathcal{M}_{i,j} \to \text{sim}(R_{*,j}, R_{*,i}) \\
\quad \textbf{else } \mathcal{M}_{i,j} \to 0
\end{array}
\right. \\
\textbf{for } i \to 1 \textbf{to } m \\
\quad \textbf{do } \left\{
\begin{array}{l}
\textbf{if } \mathcal{M}_{i,j} \neq \textbf{among the } k \textbf{ largest values in } \mathrm{M}_{*,j} \\
\quad \textbf{then } \mathcal{M}_{i,j} \to 0
\end{array}
\right.
\end{array}
\right. \\
\textbf{return } (\mathcal{M})
\end{array}
$$

(1)

(2)

The parameterization of M on k was motivated due to performance considerations and its choice represents a performance-quality trade-off.

The resulting matrix M that contains at most k nonzero entries per column becomes the final model of the item-based algorithm

Understanding Intent Aware Diversification

Why we need Intent-Aware Diversification

- Traditional diversification techniques, such as re-ranking lists based on dissimilarity, might result in a set of items that are dissimilar to each other but may not necessarily cater to various user preferences or intents.

In IAD it aims to select items that are not only dissimilar but also explicitly relevant to different user interests or intents.

By doing so, it ensures a more personalized and relevant recommendation experience for users with diverse preferences.

- Intent-aware methods for recommendation diversification rely on a predefined set of aspects, denoted as A, which characterize the items and can be utilized to estimate user interests.

- These aspects can be explicit, such as categories (e.g., politics in a news recommender) or genres (e.g., comedy in a movie recommender).

- Alternatively, aspects can be implicit, representing latent factors discovered through techniques like matrix factorization.

Query Aspect Diversification framework

For each aspect, it estimates the probability of user interest, reflecting the likelihood that the user is interested in that particular aspect.

improve the diversity of search results or recommendations by re-ranking items based on the probability of user interest in various aspects.

It considers both the relevance of items to user interests and the diversity of aspects covered by the items.

Formulation

$$p(f|u) = \frac{|\{i \in I_u : f \in F_i\}|}{\sum_{j \in f} |\{i \in I_u : f' \in F_i\}|}$$

the aspects are explicit feature $\mathcal{F}$, i.e. $\mathcal{A} = \mathcal{F}$, hence we will write $p(f|u)$ and $p(i|u, f)$ instead of $p(a|u)$ and $p(i|u, a)$

Let $\mathcal{F}_i$ be the subset of $\mathcal{F}$ that describes item i (e.g. the genres of movie i)

let I_u denote the items that are in the user's profile.

$$p(i|u, f) = \frac{ind(i, f)s(u, i)}{\sum_{j \in RS} ind(j, f)s(u, j)}$$

$p(i|u, f)$,the probability of choosing I from a set of recommendations RS given explicit aspect f of user u

Where $ind(i, f) = 1$, if $f \in F_i$ and 0 otherwise

Formulation

- $nov_{xQuAD}(i, RL) = \sum_{a \in A}[p(a|u)p(i|u,a)] \prod_{j \in RL}(1 - p(j|u,a)]$

$p(a|u)$: The probability distribution of user u's interest in aspect a.

$p(i|u,a)$: The probability of choosing item i from the recommendation set given aspect a of user u.

RL: A list of already recommended items, used to factor in novelty.

Advantage

- Balances relevance with novelty
- Provides a diverse set of recommendations, which can increase user engagement by exposing them to a broader array of choices.

Disadvantage

- Complexity increases with the number of aspects and the size of the recommendation list due to multiple probability computations.

Relevance-Based -Query Aspect Diversification

RxQuAD focuses on maximizing relevance. It does so by considering the probability that a user finds an item relevant for a specific aspect.

It calculates the probability that a user finds an item relevant when interested in a specific aspect.

The probability that a user stops exploring the recommendation list after finding a relevant item.
This stopping probability influences the diversification strategy by balancing between exploration and exploitation.

Formulation

$$nov_{RxQuAD}(i, RL) = \sum_{a \in A} [p(a|u)p(rel|i, u, a) \prod_{j \in RL} (1 - p(rel|j, u, a)p(stop|rel))]$$

- A is the set of aspects (user interests) considered for diversification.
- $p(a|u)$ is the probability of user u being interested in aspect a.
- $p(\text{rel}|i,u,a)$ is the probability that user u finds item i relevant given the aspect a.
- $p(\text{rel}|j,u,a)$ is the probability that item j (already in the recommendation list RL) is relevant to user u for the aspect a.
- $p(\text{stop|rel})$ is the probability that a user stops exploring the recommendation list after finding a relevant item.

Other models of Intent Aware Diversification

c-pLSA (Constrained Probabilistic Latent Semantic Analysis)
This method involves a learning model that uses explicit aspects for interpretability but optimizes them through a learning algorithm to improve predictive accuracy.

Minimum Variance Criterion
A more recent approach based on portfolio theory, which aims to minimize risk (variance) in the set of recommendations. This method seeks to ensure a stable performance across various user preferences and item aspects.

SPAD & RSPAD

- It is an Intent-Aware Diversification framework based on user subprofiles,rather the the item explicit feature
- SPAD : Subprofile aware diversity
- RSPAD : Relevance Based Subprofile aware diversity
- Main Difference From Quad& RxQuad is that, quad and rxQuad uses a global set f aspects for the users, where SPAD & RSPAD uses user specific subprofile as aspects leading to personalized diversification

SubProfile Aware Diversity

- SPAD is a greedy re-ranking approach that is intent-aware and personalized based on subprofiles within the user's profile.
- What is Sub Profiles in SPAD & RSPAD ?
- It is a way to group items that a user has rated similarly ,providing a more nuanced understanding of their preferences.

SubProfile Aware Diversity

- Approach
  - Firstly, we generate a set of recommendation using IB+ recommender
  - Extract Refined Sub Profiles using this recommendation set
  - Generate Recommendation Set using MF ALS
  - By using the extracted subprofile , find out the SPAD Score for each item in Recommended Set
  - Recommend the top-N items after finding out SPAD

Candidate Subprofile Generation

- User u profile is $I_u \subseteq I$, where I be the set of all items
- $I_u = \{i|rui \geq 4\}$
- A user's subprofiles are subsets of I_u.

U_ID	M_ID	Rating
u1	m1	4
u1	m2	4
u1	m3	5
u1	m4	3
u2	m1	4
u2	m5	5
u2	m6	4
u3	m7	4
u3	m8	4
u3	m2	4

Candidate Subprofile Generation

User Profile

userId	1
movieId	
1	5.0
48	5.0
150	5.0
260	4.0
527	5.0
531	4.0
594	4.0
595	5.0
608	4.0

Generate Recommendation set using IB+

- Item-Based nearest neighbor recommender (IB+) is designed for implicit feedback (positive-only ratings)
- We will use IB + (Item Based Recommendation system) to generate a RS of N = 100,using a particular User profile.

	score
movieId	
1196	47.256524
364	46.833867
1270	45.689436
588	45.686762
1198	44.608992
...	...
3418	36.680231
596	36.654132
912	36.639494
2194	36.637970
2137	36.477518

100 rows × 1 columns

Candidate Subprofile Generation

- Finding Similarity score (Cosine Similarity) between Items in the User's Profile and RS generated by IB+

movieId	1	48	150	260	527	531	594	595	608
movieId									
32	0.330817	0.138764	0.273778	0.374442	0.315371	0.106009	0.187012	0.242043	0.376076
34	0.397539	0.210627	0.335787	0.304424	0.334375	0.212342	0.245166	0.294469	0.354851
39	0.349490	0.176087	0.285341	0.280161	0.273226	0.143741	0.205582	0.266900	0.321994
50	0.323534	0.120547	0.284646	0.351718	0.357380	0.128785	0.212211	0.248291	0.488979
110	0.322782	0.146113	0.316535	0.402857	0.361595	0.144493	0.205583	0.258209	0.373100
...	...	...	...	...	...	...	...	...	...
3396	0.308908	0.169449	0.208368	0.303287	0.233516	0.134091	0.301616	0.266761	0.249406
3418	0.287237	0.147271	0.305177	0.296203	0.315253	0.169427	0.181726	0.242639	0.376663
3421	0.314053	0.120398	0.271878	0.329903	0.282737	0.104788	0.176222	0.210652	0.308474
3448	0.294965	0.167070	0.303362	0.311944	0.307623	0.151987	0.251417	0.245868	0.307712
3471	0.281153	0.146512	0.284148	0.402358	0.308853	0.140604	0.232273	0.241177	0.343758

100 rows × 37 columns

Candidate Subprofile Generation

- For each candidate item $i \in I_u$, IB+ finds items in the Recommended set that have the candidate as one of their k-nearest-neighbours: $S^*_i = \{j \in RS | i \in KNN(j)\}$

```
1: [3114, 1265, 34, 588, 2571, 1270, 1196, 364, 1580, 356],
48: [364, 2081, 588, 2096, 2078, 2087, 2080, 2137, 1380, 1035],
150: [1704, 1682, 1784, 2268, 1393, 593, 590, 318, 296, 1610],
260: [1196, 1198, 1210, 1240, 1214, 2571, 589, 480, 1270, 2628],
527: [318, 593, 296, 1704, 2858, 1196, 1617, 1198, 858, 110],
531: [364, 34, 2137, 588, 1393, 1282, 1682, 2096, 597, 2087],
594: [2096, 2087, 2078, 596, 1282, 2080, 2081, 364, 588, 2137],
595: [364, 588, 2081, 2087, 2096, 2078, 1282, 2080, 596, 2137],
608: [296, 50, 593, 318, 1617, 2858, 1196, 457, 858, 2571],
```

Refining Candidate Subprofiles:

- It iterates through the sorted list, removing any subprofile entirely contained within another subprofile (already chosen). This avoids redundancy and ensures subprofiles capture distinct user interests.
- The system sorts the candidate subprofiles (S*i) in descending order of size (number of items).
- The remaining subprofiles after this removal process become the final subprofiles (Su) for user u.

```
[1, 3114, 1265, 34, 588, 2571, 1270, 1196, 364, 1580, 356],
[48, 364, 2081, 2096, 2078, 2087, 2080, 2137, 1380, 1035],
[150, 1704, 1682, 1784, 2268, 1393, 593, 590, 318, 296, 1610],
[260, 1196, 1198, 1210, 1240, 1214, 589, 480, 2628],
[527, 318, 2858, 1196, 1617, 858, 110],
[531, 364, 1393, 1282, 597],
[594, 2096, 596, 1282, 588],
[595, 364, 596],
[608, 296, 50, 1617, 457, 2571],
```

Key Point

- Increasing k (number of neighbors) leads to more comprehensive initial subprofile candidates, potentially capturing a wider range of similar movies.
- Subprofile refinement ensures the final subprofiles represent distinct user interests and avoids redundancy.

Generate Recommendation using other Recommenders

- We produce a set of recommendations RS using some recommender.
- This can be any recommender that
- produces scores s(u, i), for the items that
- it recommends.

- In particular, just because we detect subprofiles using IB+, we are not obliged to use IB+ to produce RS.
- We have generated RS using MF (ALS) Model.

	item	score	genre
item			
900	900	5.0	Musical|Romance
110	110	5.0	Action|Drama|War
3656	3656	5.0	Crime
2686	2686	5.0	Drama|Mystery
1387	1387	5.0	Action|Horror
176	176	5.0	Comedy
2396	2396	5.0	Comedy|Romance
1035	1035	5.0	Musical
3334	3334	5.0	Crime|Drama|Film-Noir|Thriller
1046	1046	5.0	Drama|Romance
1060	1060	5.0	Comedy|Drama
1360	1360	5.0	Drama
1358	1358	5.0	Drama|Thriller
1066	1066	5.0	Comedy|Musical|Romance

Diversification Score : SubProfile Aware Diversity

$\mathrm{SPAD}(i, RL) = \sum_{s \in S} [p(s|u)p(i|u, s) \ * \ \pi_{j \in RL} (1 - p(j|u, s))]$

Where,

- $P(S|u)$: Probability of Subprofile Given User
- $P(i|u, S)$: Probability of Choosing Item i given User u and Subprofile S

p(S|u): Probability of Subprofile Given User

- This represents how likely each subprofile reflects user u1's interests.

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

Example : P(S1|U1)

- S1 = [1, 3114, 1265, 34, 588, 2571, 1270, 1196, 364, 1580, 356]
- |S1| = 11 (number of items in Subprofile 1)
- The total size is Σ |S'| = 67 (sum of the number of items in all subprofiles).
- p(S1|u1) = |S1| / Σ |S'| = 11 / 67 = 0.164

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

Calculation for every SubProfile

- $P(S1|u)=11/67=0.16$
- $P(S2|u)=10/67=0.15$
- $P(S3|u)=11/67=0.16$
- $P(S4|u)=9/67=0.13$
- $P(S5|u)=7/67=0.10$
- $P(S6|u)=5/67=0.07$
- $P(S7|u)=5/67=0.07$
- $P(S8|u)=3/67=0.04$
- $P(S9|u)=6/67=0.08$

P(i|u, S): Probability of Choosing Item i given User u and Subprofile S

- This represents how likely user u would choose a specific movie (i) considering a particular subprofile (S).

$$p(i|u,S) = \frac{\mathrm{ind}(i,S)s(u,i)}{\sum_{j \in RS} \mathrm{ind}(j,S)s(u,j)}$$

Example : $p(i|u, S)$

- Let's consider an item $i \in RS$, generated by the MF ALS
- $i = 364$
- S2 = [48, 364, 2081, 2096, 2078, 2087, 2080, 2137, 1380, 1035]
- $p(1035|u1, S2) = ind(364, S2) * s(u1, 364) / \Sigma_{j \in RL}(ind(j, S2) * s(u1, j))$

$$p(i|u,S) = \frac{\text{ind}(i,S)s(u,i)}{\sum_{j\in RS}\text{ind}(j,S)s(u,j)}$$

Example : SubProfile Aware Diversity

Find SPAD(364,RL)

spad_divScore=0

S1_score = [p(s1|u)p(364|u, s1) * πj∈RL (1−p(j|u, s1)] = X

S2_score = [p(s2|u)p(364|u, s2) * πj∈RL (1−p(j|u, s2)]=X

S3_score = [p(s3|u)p(364|u, s3) * πj∈RL (1−p(j|u, s3)]=[0 * πj∈RL (1−p(j|u, s3)] = 0

S4_score = [p(s4|u)p(364|u, s4) * πj∈RL (1−p(j|u, s4)]=[0 * πj∈RL (1−p(j|u, s4)] = 0

S5_score = [p(s5|u)p(364|u, s5) * πj∈RL (1−p(j|u, s5)]=[0 * πj∈RL (1−p(j|u, s5)] = 0

S6_score = [p(s6|u)p(364|u, s6) * πj∈RL (1−p(j|u, s6)]=X

S7_score = [p(s7|u)p(364|u, s7) * πj∈RL (1−p(j|u, s7)]=[0 * πj∈RL (1−p(j|u, s7)] = 0

S8_score = [p(s8|u)p(364|u, s8) * πj∈RL (1−p(j|u, s8)]=X

S9_score = [p(s9|u)p(364|u, s9) * πj∈RL (1−p(j|u, s9)]=[0 * πj∈RL (1−p(j|u, s9)] = 0

spad_divScore=

S1_score+S2_score+S3_score+S4_score+S5_score+S6_score+S7_score+S8_score+S9_score

Relevance Based SubProfile Aware Diversity

$RSPAD(i, RL) = \sum_{s \in S} [p(s|u)p(rel|i, u, s) * \pi_{j \in RL}(1 - p(rel|j, u, s)p(stop|rel))]$

Where ,

- $p(rel|i, u, s)$ is probability that user u finds item i relevant given the Subprofile
- $p(stop|rel)$ is probability that a user stops exploring the recommendation list after finding a relevant item
- Means if the item i is present in the subprofile s_i then the $P(rel|i,u,s)=s(u,i)/5$

Datasets

- **Datasets Used:**
  - MovieLens 1M
  - LastFM
- **Modifications (LastFM):**
  - Listening event frequencies converted into ratings (1-5)
  - Augmented with additional meta-data (user-generated tags)
- **Characteristics (Modified Versions):**
  - Refer to Table 1 for details

- **Characteristics (Modified Versions):**

MovieLens	6040 users	3706 items	~1M ratings
	18 genres in total; avg. 1.65 per movie		
LastFM	992 users	7280 items	~500k ratings
	71833 tags in total; avg. 8 per artist		

Table 1: Datasets

Diversification Techniques Used

- Diversification Techniques Compared:
  a. SPAD
  b. RSPAD
  c. MMR (Carbonell and Goldstein 1998)
  d. xQuAD (Vargas, Castells, and Vallet 2011)
  e. RxQuAD (Vargas, Castells, and Vallet 2012)
  f. c-pLSA (Wasilewski and Hurley 2016)

Baseline Recommenders

- Three Baseline predictors have been used to see the performance of each of the diversification techniques
  - Probabilistic Latent Semantic Analysis (pLSA) with d = 50
  - Matrix Factorization (MF) wit d= 30 and α=1.0
  - A Factorization machine using Bayesian pairwise loss for ranking (FMBPR) with d = 190, lr = 0.01, regM = 0.01, regW = 0.001

  where,
  d - number of latent factors selected from {10, 30, 50, 70, 90, 110, 130, 150, 170, 190, 210}
  α - confidence level of MF from {1, 2, . . . , 10}
  lr - learning rate for FMBPR
  regW and regM - regularization parameters chosen from {0.01, 0.001}

- IB+ has its own hyperparameters: the number of neighbours (k, referred to below as kIB) and the number of recommendations to make (n), both of whose values we select from V . Another hyperparameter k (referred to below as kDC) and its value is also selected from set V .

  a. pLSA : n=50 for SPAD, and n=70 for RSPAD; kIB =10,kDC =30 for both.

  b. MF : n=50 for SPAD, n=90 for RSPAD; kIB =10 and kDC =50 for both.

  c. FMBPR : n=30, kIB =10 and kDC =10 for both SPAD and RSPAD.

Methodology

1. Data Partitioning:
   - Randomly partition ratings into training, validation, and test sets.
   - Each user's ratings are split such that 60% are in the training set, 20% in the validation set, and 20% in the test set.
   - Results are averaged over five runs with different splits to ensure robustness.

2. Hyperparameter Optimization:
   - Select hyperparameter values for each baseline recommender that optimize precision on the validation sets.
   - Hyperparameters are chosen based on their ability to maximize precision, a key evaluation metric.

3. Training Baseline Recommenders:
- Train the baseline recommenders using the selected hyperparameter values on the combined training and validation sets.
- Generate recommendation sets (RS) for each user, where the size of RS is set to 100.

4. Re-ranking Recommendation Sets:
- Re-rank each recommendation set (RS) using different re-ranking algorithms.
- Experiment with various values of λ, which control the balance between relevance and diversity in the re-ranking process.
- λ values are varied within the range [0.1, 0.2, ..., 1.0] to explore different trade-offs between relevance and diversity.

5. Top-N Recommendations:
- From each re-ranked list (RL), select the top-N recommendations.
- N is set to 10, representing the number of items to recommend to users.
- The top-N recommendations are evaluated to assess the performance of the re-ranking algorithms.

Evaluation Measures

- For accuracy we measure precision and for diversity we measure α-nDCG (Clarke et al. 2008), which is a redundancy-aware version of nDCG.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\alpha\text{-nDCG}(L) = \frac{1}{\alpha\text{-IDCG}} \sum_{i \in L} \left[\frac{1}{\log_2(\mathrm{r}(i,L)+1)} \sum_{f \in \mathcal{F}} \mathrm{rel}(i|u,f) \prod_{\substack{j \in L, \\ \mathrm{r}(j,L) < \mathrm{r}(i,L)}} (1 - \alpha\, \mathrm{rel}(j|u,f)) \right]$$

- Diversity is measured with respect to the explicit features F (the meta-data).
- We evaluate top-N for N = 10 recommendations and, we treat test set items with a rating of 4 or 5 as being relevant. In α-nDCG, we use α = 0.5.
- We use p(stop|rel) = 0.5 in RxQuAD and RSPAD too.

Result

- **Precision Evaluation**:
  - SPAD and RSPAD consistently achieve the highest precision among baselines on the MovieLens dataset.
  - Exception: When MF is the baseline, c-pLSA slightly outperforms SPAD and RSPAD in precision.

- **Diversity Evaluation**:
  - SPAD and RSPAD exhibit lower diversity due to not utilizing explicit features compared to other re-ranking methods like xQuAD and RxQuAD.
  - Despite this disadvantage, SPAD and RSPAD outperform baselines and achieve higher diversity than MMR and c-pLSA re-ranking methods.
  - xQuAD and RxQuAD demonstrate the highest diversity among re-ranking methods, surpassing SPAD and RSPAD.
  - However, xQuAD and RxQuAD often sacrifice precision for diversity, resulting in significant decreases in accuracy. Surprisingly, c-pLSA, SPAD, and RSPAD achieve both increased accuracy and diversity.

		Metrics		% change over baseline	
	λ	Precision	α-nDCG	Precision	α-nDCG
pLSA		0.2639	0.2842		
MMR	0.3	*0.2635*	0.2913	-0.17%	+2.47%
xQuAD	0.7	0.2456	**0.3428**	-6.93%	+20.61%
RxQuAD	1.0	0.2452	0.3341	-7.1%	+17.53%
c-pLSA	0.5	0.2763	0.3075	+4.69%	+8.19%
SPAD	1.0	0.2783	0.3233	+5.44%	+13.74%
RSPAD	1.0	**0.2797**	0.3213	+5.98%	+13.05%
MF		0.2916	0.3197		
MMR	0.2	0.2906	0.3243	-0.34%	+1.43%
xQuAD	0.5	0.2739	**0.3668**	-6.08%	+14.72%
RxQuAD	0.7	0.2629	0.3586	-9.85%	+12.15%
c-pLSA	0.3	**0.2978**	0.3292	+2.11%	+2.96%
SPAD	0.6	0.2947	0.337	+1.04%	+5.39%
RSPAD	0.7	0.2945	0.3368	+1.00%	+5.32%
FMBPR		0.2655	0.3025		
MMR	0.2	0.2649	0.3068	-0.22%	+1.42%
xQuAD	0.4	0.2534	**0.3376**	-4.56%	+11.61%
RxQuAD	0.5	0.2429	0.3272	-8.48%	+8.16%
c-pLSA	0.3	0.2754	0.3157	+3.75%	+4.38%
SPAD	0.4	**0.2765**	0.321	+4.16%	+6.13%
RSPAD	0.5	0.2736	0.3178	+3.07%	+5.06%

Conclusion

- **Personalized Intent-Aware Diversification**:
  - Introduce personalized diversification by utilizing subprofiles of the user's tastes.
  - Subprofiles are extracted using an extended item-based recommender (IB+) and interest estimation similar to xQuAD and RxQuAD systems.
- **Comparison of Approaches**:
  - Compare SPAD and RSPAD with other re-ranking approaches.
  - Evaluation shows that SPAD and RSPAD produce recommendations with high accuracy and diversity.
- **Accuracy and Diversity**:
  - SPAD and RSPAD recommendations are able to achieve the best balance between accuracy and diversity..
  - Noteworthy, as the diversity metric favors algorithms sensitive to explicit features, which SPAD and RSPAD do not utilize.

Thank You