Exploiting the Diversity of User Preferences for Recommendation

Saúl Vargas and Pablo Castells
{saul.vargas, pablo.castells}@uam.es
Item Recommendation

User

User profile

Amazon.com

You may also like...

X  Y  Z

Recommendation
Collaborative Filtering

- Collaborative filtering techniques match users with similar preferences, or items with similar choice patterns from users, in order to make recommendations.
Diversity in Recommendation (I)

- Somebody could receive the following recommendations from a music on-line retailer:

  - Lack of diversity: pop albums from female singers.
  - Some of them are redundant.

- This is not a good recommendation.
Some time ago, I received the following set of music recommendations:

- Wrecking Ball by B. Springsteen
- Not your Kind of People by Garbage
- Like a Prayer by Madonna
- Choice of Weapon by The Cult
- Sweet Heart Sweet Light by Spiritualized
- The Light the Dead See by Soulsavers
- Little Broken Hearts by Norah Jones

Some observations:
- Different authors and genres.
- Not similar between them.

These are much better recommendations!
Relation to Search Result
Diversification (I)

$q = "java"$
Some concepts need to be translated:
- Query → User and Profile
- Document → Item
- Subtopic → Category of items

We considered two recommendation domains with different categorizations (units of diversity):
- Movie recommendations: genres
- Music artists recommendation: user-generated tags
Re-Ranking for Diversification

- comedy
- drama
- action

Recommender

top 5 not diverse

Re-ranking

Ziegler et al. 2005
Zhang et al. 2008
Vargas et al. 2011

top 5 diverse
Previous work has adapted search result diversification techniques by considering \textit{explicitly} the diversity of the items in an initial top-N recommendation.

Using the same principle, we can adapt the xQuAD re-ranking algorithm (Santos et al.).
Explicit Diversification (II)

\[
S \leftarrow S \cup \{ \text{argmax}_{i \in R \setminus S} \lambda p(i|u) + (1 - \lambda) p(i, \bar{S}|u) \}
\]

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

\[
p(c|u) = \frac{\sum_{j \in u} r(u, j) \left[ c \in j \right]}{\sum_{c'} \sum_{j \in u} r(u, j) \left[ c' \in j \right]}
\]

\[
p(i|c, u) = \frac{s(u, i) \left[ c \in i \right]}{\sum_{j \in R} s(u, j) \left[ c \in j \right]}
\]
Explicit Diversification (III)

- The aspect-specific item probability $p(i|c,u)$ could be further refined and integrated in the recommendation process.

- **The diversity is not a property of the initial recommendation list, but of the user profile.**

- We adapt the idea of query reformulation of the xQuAD framework.
Query reformulations

• We adapt the idea of query reformulation of the xQuAD framework:

\[ q = \text{“java”} \]
\[ q_1 = \text{“java island”} \]
\[ q_2 = \text{“java programming”} \]
\[ q_3 = \text{“java coffee”} \]

\[ p(d|c, q) = p(d|q_c) \]
Sub-Profiles (I)

\[ p(i|c, u) = p(i|u_c) \]
User Pools for CF (I)

- As mentioned, collaborative filtering approaches use other users' profiles to generate recommendations.
- Now we have the original complete profiles and different sub-profiles, what can we do with them?
- We consider different user pools for recommendation.
User Pools for CF (II)

Sub-users and Users
User Pools for CF (III)

Sub-users only
User Pools for CF (IV)

Category
Sub-users
Experiments (I)

• Datasets:
  – MovieLens1M: 6040 users, 3706 movies with genres.
  – Last.fm 1K (by Ò. Celma): ~1000 users, ~150.000 artists with user-provided tags.

• Recommendation algorithms:
  – Baselines: pLSA, and MF.
  – Re-ranking strategies: xQuAD-adapted explicit and sub-profile diversifications (with all three considered user pool selections).
Experiments (II)

• Evaluation methodology:
  - MovieLens1M: 5-fold cross-validation.
  - Last.fm: 60-40% temporal split.
  - TestItems: the recommender is asked to rank the items in the user's test set and the rest of the items in the other users' test (assumed to be not relevant).

• Metrics:
  - Accuracy: nDCG@20
  - Accuracy & Diversity: $\alpha$-nDCG@20, ERR-IA@20
  - Pure diversity: S-recall@20
Scalability of Diversification Algorithms

• The proposed approach has a high computational cost for Last.fm experiments with user tags:
  – MovieLens1M: 17.58 sub-profiles per user.
  – Last.fm: ~12,007 sub-profiles per user

• We propose to consider only the top-20 sub-profiles of each user.
Results (I)

- Explicit diversification degrades accuracy.
- Sub-profile diversifications show improvements in all metrics.
- CategorySubusers is slightly better than the others.

**pLSA in MovieLens1M**

- Explicit diversification degrades accuracy.
- Sub-profile diversifications show improvements in all metrics.
- CategorySubusers is slightly better than the others.
Results (II)

pLSA in Last.fm

- Sub-profile diversifications differ.
- SubusersOnly degrades S-recall, SubusersAndUsers does not improve it.
- CategorySubusers is clearly better than the others.
Conclusions

• We exploited the diversity within user-profiles to enhance the diversity of search results.

• The proposed approach is very competitive compared to explicit diversification approaches.

• We proposed a simple yet effective solution for when the number of sub-profiles is large.
Thanks for your attention! Questions?