Diversity in recommender systems

Toni Mikkola, Andy Valjakka, Heng Gui, Wilson Poon
Introduction

● two steps of recommender systems:
  ○ estimate a rating for item
  ○ recommend according to the rating
● standard methods can lead to over-specialization
  ○ collaborative filtering (CF)
Collaborative filtering

- using ratings by other people *collaboratively* to filter recommendations
- **user-based**
  - based on similar users, e.g. horror fans
- **item-based**
  - based on similar items, e.g. horror genre
Because you like

You might also like

→ Trivial; need for diversification
Diversification

- diversification happens by “searching from further away”
- balancing diversity and relevance
- diversity from user behaviour and ratings
  - no need for semantic information
- Yu et al. 2009
  - notion of explanations
    - set of items from which the recommendation comes
- Boim et al. 2011
  - Priority cover trees (PCT) and priority medoids
    - structures that denote the distance
Yu et al. 2009

What is diversity of recommender system?

The definition of diversification and how to estimate it.

before that, we will talk about:

  Relevance

  Explanation
Relevance

The goal of recommendation strategies is to estimate the user’s rating of unrated items.

$$\text{relevance}(u, i) = \sum_{u' \in U} \text{UserSim}(u, u') \times \text{rating}(u', i)$$

$$\text{Item Based Strategies}$$

$$\text{relevance}(u, i) = \sum_{i' \in I} \text{ItemSim}(i, i') \times \text{rating}(u, i')$$
**Explanation**

Explanation is a set of elements that contribute to recommending the item to the user. An explanation for a recommended item depends on the underlying recommendation strategy used.

For content-base:

\[
\text{Expl}(u, i) = \{i' \in I | \text{ItemSim}(i, i') > 0 \land i' \in \text{Items}(u)\}
\]

or

\[
\text{Expl}(u, i) = \{i' \in I | \text{ItemSim}(i, i') \times \text{rating}(u, i') > 0 \land i' \in \text{Items}(u)\}
\]

For CF strategy:

\[
\text{Expl}(u, i) = \{u' \in U | \text{UserSim}(u, u') > 0 \land i \in \text{Items}(u')\}
\]

or

\[
\text{Expl}(u, i) = \{u' \in U | \text{UserSim}(u, u') \times \text{rating}(u', i) > 0 \land i \in \text{Items}(u')\}
\]
Diversity Distance

\[ DD_u^J(i, i') = 1 - \frac{|\text{Expl}(u, i) \cap \text{Expl}(u, i')|}{|\text{Expl}(u, i) \cup \text{Expl}(u, i')|}. \]

The distance between items depends on the ratio between the number of users who recommend both items and the total number of users who recommend these items.

For a set of items \( S \subseteq \text{RecItems}(u) \), we define:

\[ DD_u(S) = \text{avg}\{ DD_u(i, i') \mid i, i' \in S \} \]
Diversification Algorithms

Intuitively, the optimal scenario for recommendation is finding a set of items with highest aggregated relevance and highest diversity. But it is impossible. In most of the situation, item pairs which have higher diversity will have lower relevance.

So we need to find a balance between relevance and diversity.

We will introduce some methods or ideas about the diversity of recommendation system.
Algorithm Swap

Set of Item Candidate
Sorted by Relevance

- \( i_1 \)
- \( i_2 \)
- \( i_3 \)
- \( i(m) \)
- \( i(k) \)
- \( i(n+1) \)
- \( i(n) \)

K items will be recommended to user

Switch the item which contribute least diversity with the next highest scoring item among the remain items.

In this way, the diversity of the set increase but the Relevance decrease.

Loop this switching until the diversity or Relevance reach an optimal threshold.
Algorithm Greedy

- Items sorted by relevance score.
- Classify the items by distance to the DivList.

**DivList**
- Items are highly distant from each other (highly diversity)
  - Upper-bound
  - KeepList
  - DiscardList
  - Lower-bound

**SimList**
- Items that have short distance to some of the item in DivList

- If DivList have enough items to recommend, then end the loop.
- If DivList do not have enough items, then refine the threshold (Upper-bound and Lower-bound) and repeat.
Explicit vs Implicit Network

Explicit network
e.g. friends network in delicious where users come friends which by explicit declaration. covering only 10% of users in del.icio.us. 90% of user will not be recommended any results.

Implicit network
based on users implicit actions e.g. in del.icio.us by creating link between two users how have shared common URL. cover 40% of users. Still average user share URL with only small number of users -> User X User sim matrix is sparse.
ALGORITHM: Item-based Similarity Computation

For generating implicit shared-item user networks. Efficient because it organizes items to buckets based on how many users have tagged them and eliminates items which are associated only with 1 user.

Only does comparison between two users if the comparison is likely to create a similarity link.

Iterates items and associated users from buckets and start from the top most bucket.

Network could computed offline and stored to SQL-database for later online use to generate recommendations.
Evaluation: Quality of Diversification Algorithms

Sampling approach:
1. active users organized by network size and divided to 5 buckets
2. for evaluation select 3 buckets – low, medium and high

<table>
<thead>
<tr>
<th></th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>del.icio.us friendship # users</td>
<td>121</td>
<td>110</td>
<td>118</td>
</tr>
<tr>
<td>avg. network size</td>
<td>1.0</td>
<td>1.5</td>
<td>12</td>
</tr>
<tr>
<td>del.icio.us shared-url # users</td>
<td>334</td>
<td>319</td>
<td>330</td>
</tr>
<tr>
<td>avg. network size</td>
<td>1.0</td>
<td>7.2</td>
<td>155</td>
</tr>
<tr>
<td>Y! Movies shared-movie # users</td>
<td>540</td>
<td>545</td>
<td>494</td>
</tr>
<tr>
<td>avg. network size</td>
<td>3.9</td>
<td>38.4</td>
<td>192.2</td>
</tr>
</tbody>
</table>

Table 4: Statistics of sample groups.
Evaluation: Performance

Figure 1: Performance comparison between naïve and item-based similarity computation algorithms: the naïve algorithm cannot finish on the full data sets for either del.icio.us or Y! Movies, and we randomly sample 1% and 0.1% of the data, respectively.

Table 2: Summary of Data Sets

<table>
<thead>
<tr>
<th></th>
<th>del.icio.us</th>
<th>Y! Movies</th>
</tr>
</thead>
<tbody>
<tr>
<td># distinct active users</td>
<td>413K</td>
<td>3.3M</td>
</tr>
<tr>
<td># distinct items</td>
<td>3.7M</td>
<td>52K</td>
</tr>
<tr>
<td># total actions</td>
<td>6.5M</td>
<td>21M</td>
</tr>
<tr>
<td>avg. # items per user</td>
<td>15.7</td>
<td>6.4</td>
</tr>
<tr>
<td>avg. # users per item</td>
<td>1.8</td>
<td>403</td>
</tr>
</tbody>
</table>

Table 3: Summary of explicit (Friends) and implicit user networks (SI for shared interests) under various similarity thresholds (th) over the study period.

<table>
<thead>
<tr>
<th></th>
<th>SI/th=0.9</th>
<th>SI/th=0.5</th>
<th>SI/th=0.1</th>
<th>Friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>del.icio.us</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># links user cow</td>
<td>88K</td>
<td>262K</td>
<td>6.2M</td>
<td>131K</td>
</tr>
<tr>
<td># users cow</td>
<td>3.1%</td>
<td>6.9%</td>
<td>39.5%</td>
<td>7.5%</td>
</tr>
<tr>
<td>Y! Movies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># links user cow</td>
<td>5.3M</td>
<td>8.3M</td>
<td>17.9M</td>
<td>N/A</td>
</tr>
<tr>
<td># users cow</td>
<td>4.5%</td>
<td>5.9%</td>
<td>8.3%</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Figure 3: Average costs for recommendation diversification for users in high group.

Figure 2: Average costs for generating recommendations and explanations.
Boim et al. 2011

- previously there was the need for manual adjustments
  - using weight to calculate similarity/diversity
  - thresholds to limit similarity/diversity
- instead, use existing ratings to formulate item clusters and structure them
  - priority medoids
  - priority cover trees
Priority medoids

- **medoid** is a representative element of a cluster
- Conventionally, the distance from the representative member to other members of the cluster is minimal
- **Priority medoids** require the representative to have the highest rating
- Effective but computationally ineffective (NP hard)
Priority cover tree (PCT)

- **Cover Tree** - data structure specialized on nearest neighbour search
  - Each node is associated with an item, but chosen arbitrarily
  - Does not take item ratings into consideration
- **PCT**: Has following attributes
  - **Ordered Insertion** - Algorithm first sorts items with respect to ratings, then inserts in descending order
  - **Tight Insertion** - During item insertion into tree, nodes with smallest distance measure are preferred as parents
  - Each node has a rating higher or equal to that of any of its children
  - Combines both ratings and distance measured as best as possible
PCT Representative Selection

- Choose best subset of items from PCT to be presented to user.
- Following methods were tested by experimentation:
  - Max-Rating (PCT-R)
    - Elements having maximal ratings
  - Max-Diversity (PCT-D)
    - Elements farthest from previously chosen elements (maximal distance between nodes)
  - Max-Coverage (PCT-C)
    - Elements having maximal number of descendants
    - Has more items (from descendants) to recommend
Boim et al. 2011 - Netflix Experiment

- Average quality of item sets generated from random set of 1000 users
- optR, optD, CT are baseline (control) tests
- Rating measure: Sum of element ratings
- Distance-based: Pairwise distance of items
- Sequel diversity: # of distinct series in item set. From 0 to 1
  - Using info from IMDB -> $\text{sequelDiversity}(I_k) = \frac{|\{\text{sequel}(i) | i \in I_k\}|}{k}$
- PCT-R is best overall performing algorithm

<table>
<thead>
<tr>
<th></th>
<th>Rating</th>
<th>Distance-based diversity</th>
<th>Sequel diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCT-D</td>
<td>-</td>
<td>+</td>
<td>=</td>
</tr>
<tr>
<td>PCT-C</td>
<td>-</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>Swap</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>CT</td>
<td>-</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>optR</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>optD</td>
<td>-</td>
<td>++</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Summary of the results (relative PCT-R)
Overall Conclusion

- Proposals to recommender systems with respect to balancing ratings and diversity and without the need for semantic information.
- Problems to discuss in next presentation
  - Item ratings based on subjective user feedback.
  - Even though semantic information is not needed for these algorithms, how do we know whether or not movies are sequels without semantics!
  - Diversity is hard to evaluate with offline evaluation methods
  - COLD START (when user or item is new) problem even bigger problem

Thank you! Any questions?