Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions

406.534 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering
Seoul National University
bibliographic details

- title: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions
- authors: G. Adomavicius and A. Tuzhilin
- journal: IEEE Transactions on Knowledge and Data Engineering
- vol: 17
- no: 6
- month: June
- year: 2005
- pp: 734-749
introduction

- abundance of practical recommender systems
- books, CDs, movies, ...
- amazon.com, MovieLens, VERSIFI, ...
problem definition

- the most common formulation
  - **estimate ratings** for the items that have not been seen by a user
  - recommend to the user the item(s) with the highest estimated rating(s)
- formal definition
  - \( C \): set of all users
  - \( S \): set of all items
  - \( u \): utility function s.t. \( u: C \times S \rightarrow R \)
  - the problem is to find
    \[
    s'_c = \arg \max_{s \in S} u(c, s), \quad \forall c \in C
    \]
- problem characteristics
  - \( u \) is not defined on the whole \( C \times S \) space, but only on some subset of it
  - \( u \) needs to be **extrapolated** to the whole space \( C \times S \) by
    - specifying **heuristics** that define the utility function
    - **estimating** the utility function that optimizes certain performance criterion
example

A Fragment of a Rating Matrix for a Movie Recommender System

<table>
<thead>
<tr>
<th></th>
<th>K-PAX</th>
<th>Life of Brian</th>
<th>Memento</th>
<th>Notorious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Bob</td>
<td>Ø</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Cindy</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Ø</td>
</tr>
<tr>
<td>David</td>
<td>3</td>
<td>Ø</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

- Ø represents that the users have not rated the corresponding movies
- the recommendation engine should be able to estimate (predict) the ratings of the nonrated movie/user combinations
classification of recommender systems

• content-based recommendations
  • the user will be recommended items similar to the ones the user preferred in the past
  • IR based heuristics, model based

• collaborative recommendations
  • the user will be recommended items that people with similar tastes and preferences liked in the past
  • memory based (heuristic based), model based

• hybrid approaches
  • combine collaborative and content-based methods
**content-based methods (1)**

- $u(c, s)$ is estimated based on $u(c, s_i)$, where $s_i \in S$ is similar to $s$
- **Content($s$):** item profile
  - a set of features characterizing item $s$
  - features usually correspond to keywords $k_j$ contained in $s$
  - that is, for document $d_j$ (i.e., item), $Content(d_j) = (w_{1j}, ..., w_{kj})$, where $w_{ij}$ represents a weight of $k_i$ in $d_j$
- estimation of weights by TF-IDF measure
  - $N$: total # of documents
  - $n_i$: # of occurrences of $k_i$ in $N$ documents
  - $f_{ij}$: # of times $k_i$ appears in $d_j$
  
  $$TF_{ij} = \frac{f_{ij}}{\max_z f_{zj}} \quad IDF_i = \log \frac{N}{n_i}$$
  
  $$w_{ij} = TF_{ij} \times IDF_i$$
content-based methods (2)

- **ContentBasedProfile(c):** user profile
  - profile of user c containing tastes and preferences of c
  - obtained by analyzing the content of the items previously seen and rated by c
  - usually constructed using keyword analysis techniques
  - that is, ContentBasedProfile(c) = (w_{c1}, ..., w_{ck}), where w_{ci} denotes the importance of k_i to user c

- \( u(c, s) = score(ContentBasedProfile(c), Content(s)) \)
  - cosine similarity measure based utility estimation

\[
u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{||\vec{w}_c||_2 \times ||\vec{w}_s||_2} = \frac{\sum_{i=1}^{K} w_{i,c}w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2} \sqrt{\sum_{i=1}^{K} w_{i,s}^2}} ,
\]
limitations of content-based methods

- limited content analysis
  - domains other than text documents: inherent problem with automatic feature extraction
  - if 2 different items are represented by the same set of features, they are indistinguishable (regardless of the item quality)
- overspecialization
  - no serendipity: the user is limited to being recommended items that are similar to those already rated
  - diversity of recommendations is needed: items should not be recommended if they are too similar to something the user has already seen
- new user problem
  - user has to rate a sufficient # of items before a content-based recommender system can really understand the user’s preferences
memory-based collaborative methods (1)

- \( u(c, s) \) is estimated based on \( u(c_j, s) \), where \( c_j \in C \) is similar to \( c \)
  - e.g., those who rate the same movies similarly -> the movies that are highly rated by the peers of \( c \) would be recommended
- the unknown rating \( r_{c,s} \) is usually computed by

\[
\begin{align*}
  r_{c,s} &= \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s} \\
  \bar{r}_c &= \bar{r}_c + k \sum_{c' \in \hat{C}} \text{sim}(c,c') \times (r_{c',s} - \bar{r}_{c'})
\end{align*}
\]

\( \hat{C} \) : the set of \( N \) users that are the most similar to \( c \) and who have rated item \( s \)

\[
k = \frac{1}{\sum_{c' \in \hat{C}} |\text{sim}(c,c')|} : \text{normalizing factor}
\]

\[
\bar{r}_c = \left( \frac{1}{|S_c|} \right) \sum_{s \in S_c} r_{c,s}, \text{ where } S_c = \{ s \in S | r_{c,s} \neq \emptyset \}
\]
memory-based collaborative methods (2)

- $sim(x, y)$ for users $x$ and $y$
  - usually based on their ratings of $S_{xy}$, where $S_{xy}$ is the set of all items co-rated by both users $x$ and $y$
  - i.e., $S_{xy} = \{ s \in S \mid r_{x,s} \neq \emptyset \text{ and } r_{y,s} \neq \emptyset \}$
  - correlation-based vs. cosine-based approach

$$
sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}
$$

$$
sim(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2} \sqrt{\sum_{s \in S_{xy}} r_{y,s}^2}}
$$
limitations of memory-based collaborative methods

- not satisfactory when the size of $S_{xy}$ is small
  - *item-based* collaborative algorithms provide better performance and comparable or better quality than user-based algorithms
- new user problem
  - the system must first learn the user’s preferences from the ratings that the user gives
- new item problem
  - until the new item is rated by a substantial number of users, the recommender system would not be able to recommend it
- sparsity
  - # of ratings already obtained is usually very small compared to # of ratings that need to be predicted
model-based collaborative methods

• use the collection of ratings to learn a model, which is then used to make rating predictions
• existing approaches: cluster models, Bayesian networks, MDP, probabilistic latent semantic analysis
• in some applications, model-based methods outperform memory-based approaches in terms of accuracy
  • yet, no underlying theoretical evidence supporting this claim is provided
• a probabilistic approach based on cluster model
  • $r_{c,s}$ is predicted as
    \[
    r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)
    \]
  • limitation: each user can be clustered into a single cluster of interest
hybrid methods

- implementing collaborative and content-based methods separately and combining their predictions
  - linear combination of ratings, voting scheme, selection of best
- incorporating some content-based characteristics into a collaborative approach
  - overcoming the sparsity problem by use of content-based profiles
- incorporating some collaborative characteristics into a content-based approach
  - use of LSI to create a collaborative view of a collection of user profiles
- constructing a general unifying model that incorporates both content-based and collaborative characteristics
# Classification of Recommender Systems Research

<table>
<thead>
<tr>
<th>Recommendation Approach</th>
<th>Recommendation Technique</th>
<th>Heuristic-based</th>
<th>Model-based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content-based</strong></td>
<td>Commonly used techniques:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- TF-IDF (information retrieval)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Clustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Representative research examples:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Lang 1995</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Balabanovic &amp; Shoham 1997</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Pazzani &amp; Billsus 1997</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Collaborative</strong></td>
<td>Commonly used techniques:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Nearest neighbor (cosine, correlation)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Clustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Graph theory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Representative research examples:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Resnick et al. 1994</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Hill et al. 1995</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Shardanan &amp; Maes 1995</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Breese et al. 1998</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Nakamura &amp; Abe 1998</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Aggarwal et al. 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Delgado &amp; Ishii 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Pennock &amp; Horvitz 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Sarwar et al. 2001</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hybrid</strong></td>
<td>Combining content-based and collaborative components using:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Linear combination of predicted ratings</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Various voting schemes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Incorporating one component as a part of the heuristic for the other</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Representative research examples:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Balabanovic &amp; Shoham 1997</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Claypool et al. 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Good et al. 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Pazzani 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Billsus &amp; Pazzani 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Tran &amp; Cohen 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Melville et al. 2002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Combining content-based and collaborative components by:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Incorporating one component as a part of the model for the other</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Building one unifying model</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Representative research examples:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Basu et al. 1998</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Condlliff et al. 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Soboroff &amp; Nicholas 1999</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Ansari et al. 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Popescul et al. 2001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Schein et al. 2002</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
extending capabilities of recommender systems

- more comprehensive understanding of users and items
- extensions for model-based recommendation techniques
- flexibility
- nonintrusiveness
  - requiring explicit feedback from users are intrusive
  - e.g., use of click streams, time spent for reading an article
- effectiveness of recommendations
  - beyond the traditional measures such as MAE and precision / recall
  - avoidance of recommending too obvious items (such as milk or bread)
- multicriteria ratings
  - e.g., Zagat’s Guide: food, decor, and service for restaurant ratings
- multidimensionality of recommendations
multidimensionality of recommendations

- extension of two-dimensional User × Item space
- takes into consideration additional contextual information
  - e.g., the utility of an item may depend on time, person with the product will be shared, ...
- utility function needs to be extended to \( u: D_1 \times \ldots \times D_n \to R \)
- then, a recommendation problem is defined by selecting certain “what” dimensions \( D_{i_1}, \ldots, D_{i_k} (k < n) \) and certain “for whom” dimensions \( D_{j_1}, \ldots, D_{j_l} (l < n) \) that do not overlap, i.e., \( \{D_{i_1}, \ldots, D_{i_k}\} \cap \{D_{j_1}, \ldots, D_{j_l}\} = \emptyset \), and recommending, for each tuple \( (d_{j_1}, \ldots, d_{j_l}) \in D_{j_1} \times \ldots \times D_{j_l} \), the tuple \( (d_{i_1}, \ldots, d_{i_k}) \in D_{i_1} \times \ldots \times D_{i_k} \) that maximizes the utility \( u(d_1, \ldots, d_n) \), i.e.,

\[
\forall (d_{j_1}, \ldots, d_{j_l}) \in D_{j_1} \times \ldots \times D_{j_l},
\]

\[
(d_{i_1}, \ldots, d_{i_k}) = \arg \max_{(d_{i_1}', \ldots, d_{i_k}') \in D_{i_1} \times \ldots \times D_{i_k}} u(d_{i_1}', \ldots, d_{i_k}')
\]

\[
= (d_{j_1}', \ldots, d_{j_l}') = (d_{j_1}, \ldots, d_{j_l})
\]