Distributed Collaborative Filtering and Adaptive User-to-User Correlation

Francesco Ricci
Faculty of Computer Science
Free University of Bozen-Bolzano, Italy

Joint work with Shlomo Berkovsky (CIRSO), Tsvi Kuflik (University of Haifa), and Linas Baltrunas (University of Bozen)
Content

- Introduction to recommender systems and collaborative filtering
- Motivations:
  - Decentralized collaborative filtering
  - Improve accuracy by partitioning ratings and re-aggregating information
- Domain-based rating partitioning
- Importing user modelling information
- Computing inter-domain correlations
- Evaluation
- Extension: adapting the similarity metric to the prediction problem
What movie should I see?

The Internet Movie Database (IMDb) provides information about actors, films, television shows, television stars, video games and production crew personnel.

Owned by Amazon.com since 1998
September 15, 2008 IMDb featured
1,039,447 titles and 2,723,306 people

More than 57M users per month.
Recommender Systems

- In everyday life **we rely on recommendations** from other people either by word of mouth, recommendation letters, movie and book reviews printed in newspapers ...

- In a typical recommender system **people provide recommendations as inputs**, which the system then aggregates and directs to **appropriate recipients**
  - Aggregation of recommendations
  - Match the recommendations with those searching for recommendations

[Resnick and Varian, 1997]
Examples

- **Amazon.com** – looks in the user past buying history, and recommends product bought by a user with similar buying behavior.
- **Tripadvisor.com** - Quoting product reviews of a community of users.
- **Myproductadvisor.com** – make questions about searched benefits (product features) to reduce the number of candidate products.
- **Yahoo.com** – “Today’s Picks” highlight ten destinations that are highly-relevant to individual users, based on recent online activity and preferences.
- **iTunes Genius** – recommend albums similar to those found in your library.
- **Smarter Kids** – self selection of a user profile – classification of products in user profiles.
Social Filtering
Welcome to MovieLens!

Free, personalized, non-commercial, ad-free, great movie recommendations. Have questions? Take the MovieLens Tour for answers. Not a member? Join MovieLens now.

Need a gift idea? Try MovieLens QuickPick!

New to MovieLens?

Join today!

You get great recommendations for movies while helping us do research. Learn more:

- Try out QuickPick: Our Movie Gift Recommender
- Take the MovieLens Tour
- Read our Privacy Policy
- See our Browser Requirements
- Learn about Our Research

Hello MovieLens Users!

Please log in:

Username: [ ]
Password: [ ]
Save login: [ ]

Log into MovieLens

Forgot your password?

New member? Join now

MovieLens is a free service provided by GroupLens Research at the University of Minnesota. We sometimes study how our members use MovieLens in order to learn how to build better recommendation systems. We promise to never give your personal information to anyone; see our privacy policy for more information.
Welcome to the new MovieLens!

Existing MovieLens users: We'd like to welcome you back to MovieLens, and let you know we have a new MovieLens FAQ you might want to read. We hope you like what you will see!

Take me to MovieLens!

New MovieLens users: Thank you for joining MovieLens! In order to generate personalized movie recommendations, we need to know a little about what movies you have already seen. MovieLens will now display several lists of movies. If you have seen any of the listed movies, please rate them using the rating scale shown below.

Ratings are on a scale of 1 to 5:

- ★★★★★ = Must See
- ★★★★☆ = Will Enjoy
- ★★★☆☆ = It's OK
- ★★☆☆☆ = Fairly Bad
- ★☆☆☆☆ = Awful

Remember: the more movies you rate, the more accurate MovieLens' predictions will be.

To rate a movie, just click on the pulldown next to the title of a movie you have seen. Blue stars will appear to indicate that your rating has been received.

This image shows that the movie 'Dude, Where's My Car?' was rated 1.5 stars.

I'm ready to start rating!
So far you have rated 0 movies. MovieLens needs at least 15 ratings from you to generate predictions for you. Please rate as many movies as you can from the list below.

<table>
<thead>
<tr>
<th>Your Rating</th>
<th>Movie Information</th>
</tr>
</thead>
</table>
| 🌟🌟🌟🌟 | Austin Powers: International Man of Mystery (1997)  
Action, Adventure, Comedy |
| 🌟🌟🌟🌟 | Contact (1997)  
Drama, Sci-Fi |
Action, Adventure, Drama, Fantasy, Romance |
| ??? | Demolition Man (1993)  
Action, Comedy, Sci-Fi |
| ??? | Eraser (1996)  
Action, Drama, Thriller |
| ??? | Maverick (1994)  
Action, Comedy, Western |
| 🌟🌟🌟🌟 | Philadelphia (1993)  
Drama |
| 🌟🌟🌟🌟 | Piano, The (1993)  
Drama, Romance |
| ??? | Toy Story 2 (1999)  
Adventure, Animation, Children, Comedy, Fantasy |
| 🌟🌟🌟🌟 | X-Men (2000)  
Action, Adventure, Sci-Fi |

To get a new set of movies click the next > link.
Welcome fricci@unibz.it [Log Out]
You've rated 47 movies.
You're the 18th visitor in the past hour.

Home | Find Movies | Discussion Forums | Preferences | Help

New Movies
- Simpsons Movie, The (2007)
- Sunshine (2007)
- 3:10 to Yuma (2007)
- Superbad (2007)
- Stardust (2007)
- Hot Rod (2007)
- Death at a Funeral (2007)
- Hairspray (2007)

New DVDs
- I Love You Again (1940)
- Hot Fuzz (2007)
- Ace in the Hole (a.k.a. The Big Carnival) (1951)
- Zodiac (2007)
- Sweet Land (2005)
- They Live by Night (1948)

421 new movies have been added since you last visited. See the newest additions.

News and Updates (archives)

11 Jan 2007: We've added the ability to rate tags.

1 Dec 2006: We've recently launched some new features. You can find more information about movie groups and profiles on the new features page.
Bridge on the River Kwai, The (1957)

Movie Information (edit info)

Starring: Alec Guinness, Jack Hawkins, Bessie Hayakawa, William Holden
Directed by: David Lean
Genres: Adventure, Drama, War
Languages: English, Japanese, Thai
Available on: DVD, VHS
Average rating: ★★★★☆ (4.1 stars)
Rated by: 11,061 users
Links: IMDb, Rotten Tomatoes

Forum Posts
These posts mention Bridge on the River Kwai, The (1957)

Topic | Author
--- | ---
Re: What is everyone's favorite War movie... | (mod)
Re: What is everyone's favorite War movie... | (Yymkata)
Re: What is everyone's favorite War movie... | (Ellipsis)
Re: What is everyone's favorite War movie... | (dispenser)
Re: What is your favorite historical film... | (Dudaman)

Related Forum Posts
These posts mention movies similar to Bridge on the River Kwai, The (1957)

Topic | Author
--- | ---
Re: ML top picks | (momasa)
Re: What's your favorite drama? | (Ryuukuro)
Re: Which classics are you excited about? | (Tongue)
### Matrix of ratings

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| a | 1 | 4 | 5 | 4 | 3 | 2 | 4 | 2 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| b | 4 | 3 | 5 | 1 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| c | 5 | 4 | 4 | 3 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| d | 3 | 5 | 4 | 5 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| e | 5 | 4 | 4 | 3 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| f | 4 | 1 | 3 | 5 | 4 | 1 | 5 | 4 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| g | 2 | 4 | 2 | 5 | 1 | 4 | 5 | 4 | 2 | 4 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| h | 2 | 4 | 1 | 4 | 3 | 5 | 4 | 2 | 5 | 4 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| i | 1 | 3 | 5 | 5 | 4 | 4 | 5 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| j | 4 | 4 | 5 | 1 | 5 | 4 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| k | 5 | 4 | 2 | 5 | 1 | 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| l | 2 | 4 | 2 | 5 | 1 | 4 | 5 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| m | 5 | 3 | 5 | 5 | 3 | 4 | 5 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| n | 1 | 4 | 5 | 4 | 5 | 1 | 5 | 4 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| o | 4 | 4 | 5 | 4 | 5 | 1 | 5 | 4 | 3 | 4 | 4 |    |    |    |    |    |    |    |    |    |    |    |
| p | 4 | 4 | 5 | 4 | 5 | 1 | 5 | 4 | 3 | 4 | 4 | 4 |    |    |    |    |    |    |    |    |    |
| q | 5 | 4 | 2 | 5 | 1 | 5 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| r | 2 | 4 | 4 | 5 | 1 | 4 | 5 | 4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| s | 2 | 4 | 4 | 5 | 1 | 4 | 4 | 2 | 4 | 4 | 5 | 4 |    |    |    |    |    |    |    |    |    |
| t | 2 | 4 | 4 | 5 | 1 | 4 | 4 | 2 | 4 | 4 | 5 | 4 | 3 |    |    |    |    |    |    |    |
| u | 2 | 4 | 4 | 5 | 1 | 4 | 4 | 2 | 4 | 4 | 5 | 4 | 3 | 3 |    |    |    |    |    |    |
| v | 2 | 4 | 4 | 5 | 1 | 4 | 4 | 2 | 4 | 4 | 5 | 4 | 3 | 3 | 3 |    |    |    |    |    |    |
| w | 2 | 2 | 3 | 5 | 4 | 5 | 4 | 2 | 3 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| x | 4 | 5 | 3 | 3 | 4 | 5 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| y | 1 | 3 | 2 | 3 | 4 | 5 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
Collaborative-Based Filtering

- A collection of $n$ user $u_i$ and a collection of $m$ products $p_j$.
- A $n \times m$ matrix of ratings $v_{ij}$, with $v_{ij} = ?$ if user $i$ did not rate product $j$.
- Prediction for user $i$ and product $j$ is computed as

$$v_{ij}^* = v_i + K \sum_{v_{kj} \neq ?} u_{ik} (v_{kj} - v_k)$$

- Where, $v_i$ is the average rating of user $i$, $K$ is a normalization factor such that the sum of $u_{ik}$ is 1, and

$$u_{ik} = \frac{\sum_j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_j (v_{ij} - v_i)^2 \sum_j (v_{kj} - v_k)^2}}$$

- Where the sum (and averages) is over $j$ s.t. $v_{ij}$ and $v_{kj}$ are not “?”. 

[Breese et al., 1998]
Example

Users’ similarities: \( u_{i5} = 0.5, u_{i8} = 0.5, u_{i9} = 0.8 \)

\[
\begin{align*}
    v_{ij}^* &= v_i + K \sum_{v_{kj} \neq v_i} u_{ik} (v_{kj} - v_k) \\
    v_{ij}^* &= 3.2 + \frac{1}{2.5} \times [0.5 \times (-1) + 0.5 \times (-0.5) + 0.8 \times 2] \\
    &= 3.2 + 1/2.5 \times [-0.25 + 1.6] = 3.2 + 0.75 = 3.95
\end{align*}
\]
Distributed Scenario

$q=\langle \text{user} = i \rangle$ recommend $j$

$q=\langle \text{user} = i, \text{item} = j, \text{target} = t \rangle$

reply from a remote system
- User identifiers
- User models
- User identifiers and their similarities
- Rating prediction for $j$
Related Works

Information processing in CF

1. **Similarity computation**: assessing the similarity of all the users to the active user, i.e., the user for whom a recommendation is searched

2. **Neighborhood formation**: selecting the $K$ most similar users to the active user

3. **Computing the active user rating prediction**: for a target item whose rating is unknown
   1. weight the ratings - on the target item - of the $K$ most similar users, found at (2) according to the user-to-user similarity computed at (1)
   2. the predicted rating is the weighted average.
What information can be exchanged

- UMs (rating vectors) stored by the remote systems
- Lists of the neighborhood candidates computed by the remote systems
- Degrees of similarity between the active user and the other users, computed over the data stored by the remote systems
- Complete predictions generated by the remote systems.
Assumptions

- **Users** can be identified uniquely in all the domains
- **Items** can be identified uniquely in all domains
- **Target domain** sends a request to remote domains specifying $q = \langle i, j, t \rangle$
  - $i$ is the identifier of the active user
  - $j$ is the target item identifier (possibly null)
  - $t$ is the target domain.
- Different distributed prediction methods are characterized by "what the remote domains reply".
Rating Matrix and Domains

- Given the assumptions: there is a "centralized" (aggregated) model of the distributed scenario
- $V$ is the overall rating matrix
- $V_a, V_b, V_c$, are rating sub matrices for three domains $R_a, R_b, R_c$
Prediction Methods - Remote Replies

- **Local Prediction:** the remote systems \( \{R_d\}_{d \in D} \) do not return any data.

- **Centralized Prediction:** All the ratings managed by \( \{R_d\}_{d \in D} \) are sent back - we assume that all the domains are related and \( D \) is the full set of domains.

- **Distributed Peer Identification:** The identifiers of users that all the remote systems \( \{R_d\}_{d \in D} \) consider as “similar” to the target user \( i \) are sent back.

- **Distributed Neighborhood Formation:** The identifiers of the users that all the remote systems \( \{R_d\}_{d \in D} \) consider as “similar” to the target user \( i \), **together with their similarities** to the target user \( i \) - similarities are computed by the remote system using only the ratings in \( V_d \).

- **Distributed Prediction:** The rating predictions for item \( j \) computed by the remote systems \( \{R_d\}_{d \in D} \) using the ratings contained in \( V_{d_i} \), \( d \in D \) are sent back.
Distributed Peer Identification

- The identifiers of $K$ users that all the remote systems $\{R_d\}_{d \in D}$ consider as “similar” to the target user $i$ are sent back.
- $D$ is the set of all remote domains.
- In our experiments a domain is identified by a tag (a genre).
- The target domain merge the received peers and make a prediction using the local data (ratings only in the target domain).
- Remote systems provide knowledge as an informed selection of users.
Distributed Peer Identification

- $V_a$ is the target domain
- $V_b$ and $V_c$ are remote domains (containing some ratings of the target user)
Distributed Neighborhood Formation

- The identifiers of some users that the remote systems \( \{R_d\}_{d \in D} \) consider as “similar” to the target user \( i \), \textbf{together with their similarities} to the target user \( i \) - similarities are computed by the remote system using only the ratings in \( V_d \).
- The similarity of a neighbor user \( l \) with the target user \( i \) is a weighted average

\[
\sum_{d \in D} \frac{\text{cor}(d,t) \text{sim}_d(i,l)}{\sum_{d \in D} \text{cor}(d,t)}
\]

- where \( D \) is the set of \textbf{all} the domains (including \( t \)), \( t \) is the target domain and
- \( \text{cor}(d,t) \) is the \textbf{Inter-Domain Correlation} measure between domains.
Inter-Domain Correlations

- **Content-based method**
  - Mining the textual descriptions of the items in the domains from external data sources to obtain tf-idf description (vector $v$) of each domain
  - Computing cosine correlation of the domain representations

$$cor_{content}(d_1, d_2) = \text{sim}(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|}$$

- **Rating-based method**
  - Average correlation of the items in the domains ($J_d$)

$$cor_{ratings}(d_1, d_2) = AVG\{\text{sim}(j, k) : j \neq k, j \in J_{d_1}, k \in J_{d_2}\}$$
Distributed Prediction

- The rating predictions for item $j$ computed by the remote systems $\{R_d\}_{d \in D}$ using the ratings contained in $V_d$, $d \in D$ are sent back to the target domain $R_t$.

- Upon receiving the set of predictions, $R_t$ aggregates all the predictions (including the local one) into a single value by averaging the predictions.

- *We do not use here the Inter-Domain Correlation.*
Evaluation

- EachMovie data set - each movie belongs to 1.366 genres (average)
- We used only 8 genres

<table>
<thead>
<tr>
<th></th>
<th>action</th>
<th>animat.</th>
<th>comedy</th>
<th>drama</th>
<th>family</th>
<th>horror</th>
<th>romance</th>
<th>Thriller</th>
</tr>
</thead>
<tbody>
<tr>
<td>num. Movies</td>
<td>198</td>
<td>43</td>
<td>400</td>
<td>536</td>
<td>145</td>
<td>87</td>
<td>137</td>
<td>177</td>
</tr>
<tr>
<td>num. ratings</td>
<td>1,166K</td>
<td>193K</td>
<td>2,209K</td>
<td>3,056K</td>
<td>800K</td>
<td>433K</td>
<td>681K</td>
<td>991K</td>
</tr>
<tr>
<td>sparsity (%)</td>
<td>91.923</td>
<td>93.852</td>
<td>92.425</td>
<td>92.180</td>
<td>92.432</td>
<td>93.181</td>
<td>93.179</td>
<td>92.321</td>
</tr>
</tbody>
</table>

- Content-based correlation used **IMDb** data
- We used **Cosine** similarity
- At least 6 overlapping movies to make a prediction
- The number of nearest neighbors used to make the prediction is **20**
## Inter-domain correlations: content

<table>
<thead>
<tr>
<th></th>
<th>action</th>
<th>animat.</th>
<th>comedy</th>
<th>drama</th>
<th>family</th>
<th>horror</th>
<th>romance</th>
<th>thriller</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>1.000</td>
<td>0.860</td>
<td>0.935</td>
<td>0.932</td>
<td>0.820</td>
<td>0.902</td>
<td>0.913</td>
<td>0.943</td>
</tr>
<tr>
<td>animat.</td>
<td>0.860</td>
<td><strong>1.000</strong></td>
<td>0.913</td>
<td>0.848</td>
<td><strong>0.914</strong></td>
<td>0.765</td>
<td>0.838</td>
<td>0.787</td>
</tr>
<tr>
<td>comedy</td>
<td>0.935</td>
<td>0.913</td>
<td><strong>1.000</strong></td>
<td>0.965</td>
<td>0.905</td>
<td>0.868</td>
<td>0.957</td>
<td>0.903</td>
</tr>
<tr>
<td>drama</td>
<td>0.932</td>
<td>0.848</td>
<td>0.965</td>
<td>1.000</td>
<td>0.841</td>
<td>0.873</td>
<td>0.987</td>
<td>0.938</td>
</tr>
<tr>
<td>family</td>
<td>0.820</td>
<td>0.914</td>
<td>0.905</td>
<td>0.841</td>
<td>1.000</td>
<td>0.739</td>
<td>0.832</td>
<td>0.772</td>
</tr>
<tr>
<td>horror</td>
<td>0.902</td>
<td>0.765</td>
<td>0.868</td>
<td>0.873</td>
<td>0.739</td>
<td>1.000</td>
<td>0.850</td>
<td>0.939</td>
</tr>
<tr>
<td>romance</td>
<td>0.913</td>
<td>0.838</td>
<td>0.957</td>
<td>0.987</td>
<td>0.832</td>
<td>0.850</td>
<td>1.000</td>
<td>0.913</td>
</tr>
<tr>
<td>thriller</td>
<td>0.943</td>
<td>0.787</td>
<td>0.903</td>
<td>0.938</td>
<td>0.772</td>
<td>0.939</td>
<td>0.913</td>
<td>1.000</td>
</tr>
</tbody>
</table>
# Inter-domain correlations: ratings

<table>
<thead>
<tr>
<th></th>
<th>action</th>
<th>animat.</th>
<th>comedy</th>
<th>drama</th>
<th>family</th>
<th>horror</th>
<th>romance</th>
<th>thriller</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>0.129</td>
<td>0.095</td>
<td>0.078</td>
<td>0.067</td>
<td>0.086</td>
<td>0.093</td>
<td>0.075</td>
<td>0.109</td>
</tr>
<tr>
<td>animat.</td>
<td>0.095</td>
<td>0.167</td>
<td>0.074</td>
<td>0.059</td>
<td>0.125</td>
<td>0.077</td>
<td>0.074</td>
<td>0.082</td>
</tr>
<tr>
<td>comedy</td>
<td>0.078</td>
<td>0.074</td>
<td>0.072</td>
<td>0.058</td>
<td>0.071</td>
<td>0.065</td>
<td>0.070</td>
<td>0.074</td>
</tr>
<tr>
<td>drama</td>
<td>0.067</td>
<td>0.059</td>
<td>0.058</td>
<td>0.063</td>
<td>0.056</td>
<td>0.060</td>
<td>0.065</td>
<td>0.069</td>
</tr>
<tr>
<td>family</td>
<td>0.086</td>
<td>0.125</td>
<td>0.071</td>
<td>0.056</td>
<td>0.119</td>
<td>0.067</td>
<td>0.072</td>
<td>0.076</td>
</tr>
<tr>
<td>horror</td>
<td>0.093</td>
<td>0.077</td>
<td>0.065</td>
<td>0.060</td>
<td>0.067</td>
<td>0.149</td>
<td>0.060</td>
<td>0.098</td>
</tr>
<tr>
<td>romance</td>
<td>0.075</td>
<td>0.074</td>
<td>0.070</td>
<td>0.065</td>
<td>0.072</td>
<td>0.060</td>
<td>0.091</td>
<td>0.074</td>
</tr>
<tr>
<td>thriller</td>
<td>0.109</td>
<td>0.082</td>
<td>0.074</td>
<td>0.069</td>
<td>0.076</td>
<td>0.098</td>
<td>0.074</td>
<td>0.109</td>
</tr>
</tbody>
</table>
**MAE: Centralized - Local - Distributed**

- **Local**: more reliable similarity computation especially for users that rated few movies (below 6% of the movies)
- **Distributed**: very similar to local - there are few cases where the average prediction is actually using information from remote domains.
Distributed peer identification works well for medium-large user models - the neighbors found in the remote systems tend to be the same as in the target.

Distributed peer identification error is bounded (from below) by the local approach.

For small user models the peers identified are different from the local ones and the accuracy decreases.
Using a **uniform weighting** for the similarities computed in the remote domains **does not work**

- Distributed neighbor formation methods - with content-based and rating-based correlation - works in a similar way and **improve the centralized approach**
- But they do not improve the local and distributed approach.
Limitations

- **Local and (all) distributed methods reduce recall** since they exploit smaller user profiles (lack of overlapping items between target user and neighbor)

- **Distributed neighbor formation**: better recall because the user-to-user similarity is computed in all domains and then averaged – if there is enough rating overlap in some domain then ok

- The **genre-based partition** is domain specific - not tested general methods for item partitioning

- Tested on **one dataset only**

- Experiments using "real" distributed recommender systems were not done - privacy issues may prevent UM data exchange.
Adaptive Similarity Metric

- All the previous approaches (except the *Centralized* one) compute the user-to-user similarity using a **subset** of the target user and peers ratings.
- Is it possible to identify - for each user-item pair - the best set of ratings upon which to compute the user-to-user similarity?
- **Example:** when predicting a rating for a movie by Schwarzenegger use only the ratings for other movies by Schwarzy and Stallone.
Item Selection

- In **adaptive** user-to-user similarity select an item if:
  - there is a rating for the item in both users
  - the item has a **great correlation/importance** for the item whose rating has to be predicted

- **Items Correlation Weights**
  - **Variance Weighting**: the larger the variance of the item, the bigger is the weight
  - **IPCC**: the larger is the Pearson correlation, the larger the weight
  - **Mutual Information**: weight is computed as the information that predictive item provides to the knowledge of target item
  - **Genre weighting**: the more genres the two items share the larger is the weight.
Performance of Adaptive Item Selection

- Less data can lead to a better prediction.
- Selecting a small number of overlapping items improves the accuracy.
- Improvement achieved for all our used error measures (also precision and recall).
- Best weighting method depends on the error measure.
Contribution

- A **distributed model** for a **domain-specialized** recommender system - components communicate in a cooperative way with a simple request-response protocol
- **Methods** for exploiting the **additional knowledge** provided by the **classification of an item into a domain** and the proof that the accuracy of CF can be improved
- **Methods** for **adapting the similarity metric** to the target prediction and improve accuracy
- The validation that **accuracy of CF can be improved** by basing the rating prediction on information contained in a subset (carefully selected) of the items.
More info on recommender systems:
Recommended for Ricci Francesco

Your recommendations are based on 3 items you own and more.

1. **Object-Oriented Common LISP [FACSIMILE]**
   - by Stephen Slade
   - Average Customer Review: ★★★★★
   - Publication Date: July 30, 1997
   - Our Price: $16.45 Used & new from $41.40

2. **How Would You Move Mount Fuji? Microsoft's Cult of the Puzzle - How the World's Smartest Company Selects the Most Creative Thinkers**
   - by William Poundstone
   - Average Customer Review: ★★★★★
   - Publication Date: May 1, 2003
   - Our Price: $16.07 Used & new from $9.95

3. **Introduction to Artificial Intelligence**
   - by Philip C. Jackson
   - Average Customer Review: ★★★★★
   - Publication Date: July 1, 1985
   - Our Price: $11.87 Used & new from $5.49
Welcome to MyProductAdvisor.com

Answer some easy optional questions and get unbiased recommendations of products that fit your needs.
We don't promote any products and we don't sell you any products.

What You Do
- Select Your Advisor
- Rate Product Features Important To You

What You Get
- Personalized Product Recommendations
- Latest Product Specs
- Comparisons & Info

Select Your Advisor

New Autos  Digital Cameras

Laptops  Televisions
Bolzano, Italy hotel search results

TripAdvisor Traveler Reviews

Stayed Here? Rate This!
WRITE A REVIEW
POST PHOTOS | UPLOAD VIDEO

Language: English first
Sort by: Date: Newest first

Based on 13 reviews worldwide

Traveler Rating
4.0 Overall

What to expect
4.5 Rooms
4.0 Service
4.0 Value
4.5 Cleanliness
3.5 Dining

Recommendations
An amazing honeymoon
Young singles
Older travelers
Business travelers
Couples/romantics

Specials: Save up to 70% standard rates!
No booking fees & No payment until check-out at over 25,000 Hotels.

Reviews in English (1-10 of 12)

"Nice Modern Hotel"
Aug 8, 2007  Sh. Trav, San Diego

4.0

Love the building, decor & breakfast - very good price as well. The services, after I talked to some European friends, are typical European, not as accommodating as U.S., a bit matter-of-fact. The hotel can also use a bit more maintenance: we stayed in three rooms in 2 nights - story to follow, but the AC in the first room...

"Loved Bolzano and loved the Greif"
Mar 6, 2007  Hawaii Girl, Los Angeles, California

4.0

Based on researching Bolzano hotels on TripAdvisor, I wasn't sure if I should choose the Hotel Figli or the Hotel Greif. Hotel Figli sounded like a quaint and charming classical place but I decided on the Hotel Greif because I enjoy modern/design hotels, but can sometimes regret it when the staff turn out to be pretentious. I'm glad to say...

Hotel Greif
Walther Square
Entrance Raingasse
Bolzano 39100 Italy