Collaborative Filtering

Peter Brusilovsky
with slides by Sue Yeon Syn and Danielle Lee

Agenda

- Context
- Concepts
- Uses
- CF vs. CB
- Algorithms
- Practical Issues
- Evaluation Metrics
- Future Issues
Types of Recommender Systems

- Collaborative Filtering Recommender System
  - “Word-of-Mouth” phenomenon.
- Content-based Recommender System
  - Recommendation generated from the content features associated with products and the ratings from a user.
- Case-based Recommender System
  - A kind of content-based recommendation. Information are represented as case and the system recommends the cases that are most similar to a user’s preference.
- Hybrid Recommender System
  - Combination of two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one (Burke, 2002).

Recommendation Procedure

1. Understand and model users
2. Collect candidate items to recommend.
3. Based on your recommendation method, predict target users’ preferences for each candidate item.
4. Sort the candidate items according to the prediction probability and recommend them.
Example: Amazon.com

Amazon’s Source of Wisdom

Customers Who Bought This Item Also Bought

Frequently Bought Together

Price For All Three: $117.36

Additional Items to Explore

Nikon D3100 14.2MP Digital SLR Camera...
Nikon 55-200mm f/4-5.6... $1,236.00
Nikon D7000 16.2MP D- ...
Nikon D7000-05000-D3100-02000-D5100-02000-DSLR...
Nikon D3100 14.2MP Digital SLR Camera... $746.05
Concepts

- Collaborative Filtering
  - “word of mouth”
  - The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given a set of historical preference judgments for a community of users.

- User
  - Any individual who provides ratings to a system

- Items
  - Anything for which a human can provide a rating

User-based CF

The input for the CF prediction algorithms is a matrix of users’ ratings on items, referred as the ratings matrix.

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>??</td>
<td>16/4</td>
</tr>
<tr>
<td>User1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>9/4</td>
</tr>
<tr>
<td>User2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>14/4</td>
</tr>
<tr>
<td>User3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>12/4</td>
</tr>
<tr>
<td>User4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>13/4</td>
</tr>
</tbody>
</table>
User-based CF (2)

Collaborative Filtering Recommender System, Danielle Lee

Algorithms

User-based nearest neighbor
Non-probabilistic Algorithms
Item-based nearest neighbor
Reducing dimensionality

Collaborative Filtering
Probabilistic Algorithms

Bayesian-network models
EM algorithm
User-Based NN Recommendation

1. Select like-minded peer group for a target user
2. Choose candidate items which are not in the list of the target user but in the list of peer group.
3. Score the items by producing a weighted score and predict the ratings for the given items.
4. Select the best candidate items and recommend the m to a target user.

Redo all the procedures through 1 ~ 4 on a timely basis.

User-based NN: User Similarity

- Pearson’s Correlation Coefficient for User a and User b for all rated Products, P.

\[
sim(a, b) = \frac{\sum_{p \in \text{product}(P)} (r_a, p - \bar{r}_a)(r_b, p - \bar{r}_b)}{\sqrt{\sum_{p \in \text{product}(P)} (r_a, p - \bar{r}_a)^2} \sqrt{\sum_{p \in \text{product}(P)} (r_b, p - \bar{r}_b)^2}}
\]

- Pearson correlation takes values from +1 (Perfectly positive correlation) to -1 (Perfectly negative correlation).
User-based NN: Rating Prediction

\[
pred(a, p) = \bar{r}_a + \sum_{b \in \text{neighbors}(n)} \frac{\text{sim}(a, b) \cdot (r_b, p - \bar{r}_b)}{\sum_{b \in \text{neighbors}(n)} \text{sim}(a, b)}
\]

One Typical CF recommendation
Motivations for Collaborative Filtering based Recommendations

- Collaborative filtering systems work by people in the system, and it is expected that people to be better at evaluating information than a computed function.

- CF doesn’t require contents.

- Completely independent of any machine-readable representation of the objects being recommended.
  - Works well for complex objects (or multimedia) such as music, pictures and movies.

- More diverse and serendipitous recommendations.
## CF vs. CB

<table>
<thead>
<tr>
<th></th>
<th>CF</th>
<th>CB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare</td>
<td>Users interest</td>
<td>Item info.</td>
</tr>
<tr>
<td>Similarity</td>
<td>Set of users</td>
<td>Item info.</td>
</tr>
<tr>
<td></td>
<td>User profile</td>
<td>Text document</td>
</tr>
<tr>
<td>Shortcoming</td>
<td>Other users’ feedback matters.</td>
<td>Feature matters.</td>
</tr>
<tr>
<td></td>
<td>Coverage.</td>
<td>Over-specialize.</td>
</tr>
<tr>
<td></td>
<td>Unusual interest.</td>
<td>Eliciting user feedback.</td>
</tr>
</tbody>
</table>

## Uses for CF : Domains

- Many items
- Many ratings
- Many more users than items recommended
- Users rate multiple items
- For each user of the community, there are other users with common needs or tastes
- Item evaluation requires personal taste
- Items persists
- Taste persists
- Items are homogenous
More on User-Based NN

- Adjusted Cosine similarity, Spearman’s rank correlation coefficient, or mean squared different measures.
- Necessity to reduce the relative importance of the agreement on universally liked items: inverse user frequency (Breese, et al., 1998) and variance weighting factor (Herlocker, et al., 1999).
- Skewed neighboring is possible: Significance weighting (Herlocker, et al., 1999).
- Calculating a user’s perfect neighborhood is immensely resource intensive calculations.

Item-based NN Recommendation

<table>
<thead>
<tr>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>???</td>
</tr>
<tr>
<td>User1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Item-based Nearest Neighbor

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>User1</td>
<td>0.6</td>
<td>-1.4</td>
<td>-0.4</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>User2</td>
<td>0.2</td>
<td>-0.8</td>
<td>0.2</td>
<td>-0.8</td>
<td>1.2</td>
</tr>
<tr>
<td>User3</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-2.2</td>
<td>1.8</td>
<td>0.8</td>
</tr>
<tr>
<td>User4</td>
<td>-1.8</td>
<td>2.2</td>
<td>2.2</td>
<td>-0.8</td>
<td>-1.8</td>
</tr>
</tbody>
</table>

Item-Based NN Recommendation

- Generate predictions based on similarities between items
  - Usually a cosine similarity used
- Prediction for a user $u$ and item $i$ is composed of a weighted sum of the user $u$’s ratings for items most similar to $i$.

$$\text{pred}(u, i) = \frac{\sum_{j \in \text{items}(u)} \text{sim}(i, j) \cdot r_{ui}}{\sum_{j \in \text{items}(u)} \text{sim}(i, j)}$$
Item-based Nearest Neighbor

- More computationally efficient than user-based nearest neighbors.
- Compared with user-based approach that is affected by the small change of users’ ratings, item-based approach is more stable.
- Recommendation algorithm used by Amazon.com (Linden et al., 2003).

Uses for CF: User Tasks

- What tasks users may wish to accomplish
  - Help me find new items I might like
  - Advise me on a particular item
  - Help me find a user (or some users) I might like
  - Help our group find something new that we might like
  - Domain-specific tasks
  - Help me find an item, new or not
Uses for CF: System Tasks

- What CF systems support
  - Recommend items
    - Eg. Amazon.com
  - Predict for a given item
  - Constrained recommendations
    - Recommend from a set of items

Other Non-Probabilistic Algorithms

- Dimensionality Reduction
  - Map item space to a smaller number of underlying “dimensions.”
  - Matrix Factorization/Latent Factor models such as Singular Value Decomposition, Principal Component Analysis, Latent Semantic Analysis, etc.
  - Expensive offline computation and mathematical complexity
Dimensionality Reduction Algorithms

- Matrix Factorization got an attention since Netflix Prize competition.

Other Non-Probabilistic Algorithms

- Association Rule Mining
  - Ex) “If a customer purchases baby food then the customer also buys diapers in 70% of the cases.”
  - Build Models based on commonly occurring patterns in the ratings matrix.
  - “If user X liked both item 1 and item 2, then X will most probably also like item 5.”

\[
\text{Support (X→Y)} = \frac{\text{Number of Transactions containing X and Y}}{\text{Number of Transactions}}
\]

\[
\text{Confident(X→Y)} = \frac{\text{Number of Transactions containing X and Y}}{\text{Number of Transactions containing X}}
\]
Algorithms : Probabilistic

- Represent probability distributions
- Given a user \( u \) and a rated item \( i \), the user assigned the item a rating of \( r : p(r|u, i) \).

\[
E(r | u, i) = \sum_r r \cdot p(r | u, i)
\]

- Bayesian-network models, Expectation maximization (EM) algorithm

Practical Issues : Ratings

- Explicit vs. Implicit ratings
  - Explicit ratings
    - Users rate themselves for an item
    - Most accurate descriptions of a user’s preference
    - Challenging in collecting data
  - Implicit ratings
    - Observations of user behavior
    - Can be collected with little or no cost to user
    - Ratings inference may be imprecise.
Practical Issues : Ratings

- Rating Scales
  - Scalar ratings
    - Numerical scales
    - 1-5, 1-7, etc.
  - Binary ratings
    - Agree/Disagree, Good/Bad, etc.
  - Unary ratings
    - Good, Purchase, etc.
    - Absence of rating indicates no information

Practical Issues : Cold Start

- New user
  - Rate some initial items
  - Non-personalized recommendations
  - Describe tastes
  - Demographic info.
- New Item
  - Non-CF : content analysis, metadata
  - Randomly selecting items
- New Community
  - Provide rating incentives to subset of community
  - Initially generate non-CF recommendation
  - Start with other set of ratings from another source outside community
Evaluation Metrics

- **Accuracy**
  - Predict accuracy
    - The ability of a CF system to predict a user’s rating for an item
    - Mean absolute error (MAE)
  - Rank accuracy
    - Precision – percentage of items in a recommendation list that the user would rate as useful
    - Half-life utility – percentage of the maximum utility achieved by the ranked list in question

- **Novelty**
  - The ability of a CF system to recommend items that the user was not already aware of.

- **Serendipity**
  - Users are given recommendations for items that they would not have seen given their existing channels of discovery.

- **Coverage**
  - The percentage of the items known to the CF system for which the CF system can generate predictions.
Evaluation Metrics

- Learning Rate
  - How quickly the CF system becomes an effective predictor of taste as data begins to arrive.
- Confidence
  - Ability to evaluate the likely quality of its predictions.
- User Satisfaction
  - By surveying the users or measuring retention and use statistics.

Additional Issues: Privacy & Trust

- User profiles
  - Personalized information
- Distributed architecture

- Recommender system may break trust when malicious users give ratings that are not representative of their true preferences.
Additional Issues : Interfaces

- **Explanation**
  - Where, how, from whom the recommendations are generated.
  - Do not make it too much!
    - Not showing reasoning process
    - Graphs, key items
    - Reviews

---

Additional Issues : Interfaces

- **Social Navigation**
  - Make the behavior of community visible
  - Leaving “footprints” : read-wear / edit-wear
  - Attempt to mimic more accurately the social process of word-of-mouth recommendations
  - Epinions.com
Additional Issues: Interfaces

Epinions.com (http://www.epinions.com)