Recommender System and Collaborative Filtering

COMP 135 Intro to Machine Learning
Liping Liu & Hao Cui

slides modified from Roni Khardon’s & Liping’s with permission

Outline

• What & Why?
• How?
• Content-Based
• Collaborative Filtering

What is Recommender System?

• A automatic system that recommend items to users, such that ________
  A. Users can find what they need/like ASAP
  B. Users can spend long time on the website
  C. Users can explore as much as possible on my items
  D. etc

Why important?

• Need recommendation everywhere
  Google, Amazon, LinkedIn, eBay, The New York Times, Yelp, Trivago, Facebook, Santander, Airbnb

Long-tail effect

• Online system makes more items available
  - Many items have only a small user population
  - Recommendation has great benefit

Recommender Systems

• What & Why?
• How?
• Content-Based
• Collaborative Filtering
A motivating example

- Items
- Users
- User “ratings” of items

The utility matrix

- The “value” of items to users
  - Only known when ratings happen
  - Very sparse

The utility matrix

<table>
<thead>
<tr>
<th></th>
<th>Chibi (User1)</th>
<th>Tammy (User2)</th>
<th>Hanawa (User3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Rating</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

Methodologies for RecSys

- Content-Based
  - Based on similarity among items
- Collaborative Filtering
  - Based on similarities among users
- Session-Based
  - Based on recent user activities
- MDP-Based
  - Same as Session-Based, but optimizing cumulative reward from users

Content-based item recommendation

- Represent items with feature vectors
- A supervise problem
  - Train a regressor for each user
  - Train a classifier on user’s decision

Item features

- Movie
  - Set of actors, director, genre, year
- Document
  - Bag of words, topic (obtained from topic modeling)
- Product
  - Tags, reviews
Collaborative Filtering

- Recommendation only based on incomplete utility matrix
  - No separate information about items or users

<table>
<thead>
<tr>
<th></th>
<th>i1</th>
<th>i2</th>
<th>i3</th>
<th>i4</th>
</tr>
</thead>
<tbody>
<tr>
<td>u1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>u2</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u3</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Collaborative Filtering

- Principle
  - If two users share similar views toward one item, they tend to share similar views toward another item

Vector representations

- User $i$ represented by vector $\mathbf{u}_i \in \mathbb{R}^k$
- Item $j$ represented by vector $\mathbf{v}_j \in \mathbb{R}^k$
- The inner product $\mathbf{u}_i^T \mathbf{v}_j$ approximates the utility $M_{ij}$
- Intuition:
  - Two items with similar vectors get similar utility scores from the same user;
  - Two users with similar vectors give similar utility scores to the same item

The formal formulation

- Objective
  - $\mathbf{U} = (\mathbf{u}_i; i = 1, ..., N)$, $\mathbf{V} = (\mathbf{v}_j; j = 1, ..., M)$
  - Regularization terms to prevent overfitting
  - Minimization with stochastic gradient
    $$\min_{\mathbf{U}, \mathbf{V}} \sum_{ij} (M_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \lambda \sum_i \|\mathbf{u}_i\|_2^2 + \lambda \sum_j \|\mathbf{v}_j\|_2^2$$

Formulation with intercept term

- Objective
  - Intercept term $s_i$, $t_j$ to indicate popularity
  - (similar to $w_0$ in linear classifier, Why?)
    $$\min_{\mathbf{U}, \mathbf{V}} \sum_{ij} (M_{ij} - \mathbf{u}_i^T \mathbf{v}_j - s_i - t_j)^2 + \lambda \sum_i \|\mathbf{u}_i\|_2^2 + \lambda \sum_j \|\mathbf{v}_j\|_2^2$$

Recommendation

- Populating the utility matrix
- For a user $i$, choose top items to recommend
  $$\hat{M}_{ij} = \sum_k u_{ik} \cdot v_{jk}$$
Evaluation

- Ranking measures
  - Rank items with recommendation scores
  - Calculate ranking measures by user's ratings/usage
  - Usually pay more attention to top ranked items
  - Examples: precision@k, mean precision@k, DCG, NDCG

<table>
<thead>
<tr>
<th>Item ranking</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual usage</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Cold start

- New user entering the system
  - Matching similar users
  - Trial-and-error
- New item entering the system
  - Content-based recommendation
  - Trial-and-error

Implicit feedback

- An item is not rated/used/visited by a user
  - Might be an indication that the user does not like the item
  - Include such item as negative examples

Issues

- Recommendation system and users form a loopy system
  - RS changes user's behavior
  - User generate data for RS
- User groups becoming more homogeneous
  - Youtube recommendation of politic videos: recommend videos from the same camp