FlickOh : Personalized Movie Recommendation and Rating System

Natth Bejraburnin • Naehee Kim • Seongtaek Lim • Mentor: Brian Guaracci
What is FlickOh?

• Movie rating and recommendation system based on Twitter data
  – Provide general movie rankings
  – Suggest movie recommendations to individual users
General Movie Rating

• Provide ranking of movies based on Twitter data
  – 86 movies
  – 132M tweets collected (Oct. 26 – Dec. 2)
General Movie Rating

• Considering
  – movie preference (based on sentiment analysis) and popularity (the number of movie-relevant tweets)

• Formula: \( \frac{P}{P + N + 1} \times T \)
  – P: the number of positive tweets
  – N: the number of negative tweets
  – T: total number of tweets
Personalized Recommendation

Twitter Interest Graph

DF = direct friend, IDF = indirect friend
Personalized Recommendation

The User supplies Twitter's screen_name, say 'gilad'.

Construct gilad's interest graph up to 2 degree of separation.
~ 17000 nodes

Extract k-core graph for suitable k.
~ 4000-7000 nodes

Get tweets (with sentiment) of friends in the k-core graph.

Apply recommendation algorithms
- Attention level-based approach (heuristic)
- Model-based approach

List of recommended movies for Gilad.
Attention level-based approach

- Attention Level – Based Approach
  - Using two-level interest graph & sentiment analysis
- Considering
  - preference (based on sentiment analysis)
  - popularity (the number of a movie relevant tweets )
  - Influential power of friend (level and degree of a friend node)

- Formula: \[ \sum \left( \frac{S_i R_i D_i}{L_i} \right) \]
  - S: Sentiment Polarity (0:negative, 2:neutral, 4:positive)
  - R: Reference of movie (the number of movie tweets)
  - D: Degree of a friend node
  - L : Level of a friend( direct friend:1, indirect friend:2)
Model-based approach

- Use collaborative filtering with naïve Bayes classifier
- Aim to classify whether the user will like or dislike a movie.
- Input: rating matrix, i.e. users’ rating on movies, k-core interest graph centered at the user.
- Data sparsity problem

|       | MV1  | MV2 | MV3  | MV4 | ...
|-------|------|-----|------|-----|-----
| User 1| dislike | x  | x    | x   |     
| User 2| x    | x   | like | x   |     
| ...   |      |     |      |     |     
| The user |     |     |      |     | x   |
# Model-based approach

## Training Data

<table>
<thead>
<tr>
<th></th>
<th>DF 1</th>
<th>DF 2</th>
<th>DF 3</th>
<th>DF 4</th>
<th>...</th>
<th>DF N</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDF 1 like</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>IDF 1 dislike</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>IDF 2 like</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>IDF 2 dislike</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the user</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

## Prediction

<table>
<thead>
<tr>
<th></th>
<th>MV 1</th>
<th>MV 2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDF 1 like</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IDF 1 dislike</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>IDF 2 like</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IDF 2 dislike</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

*Common interest*
Demo

- [http://people.ischool.berkeley.edu/~stlim/flickoh/](http://people.ischool.berkeley.edu/~stlim/flickoh/)
Thank You

• Questions or Comments?