Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis

Yongfeng Zhang,
Guokun Lai, Min Zhang, Yiqun Liu, Shaoping Ma, Tsinghua University

Yi Zhang, University of California, Santa Cruz
When I am Choosing a Whitening Cream

A whitening cream that my GF used to buy

The recommended items that other customers also viewed
But I have made a WRONG choice

The recommended item is suitable for DRY skins while she has OIL skins

⭐⭐⭐⭐⭐ nice scent
By Y. Xiong on December 16, 2013
Verified Purchase
Unlike other cream I have tried, Nivea has a really great scent. It is greasy and fits for my dry skin well. I also bought Serum so I used the Serum first before applying the cream. I used it as a daily lotion. It's hard to tell if my skin has lightened since I've used it for over a month and I haven't really seen a difference yet. Maybe a slight change but nothing dramatic to where you can tell my skin has whitened.

Can the recommender system give me more detailed EXPLANATIONS about WHY an item is recommended?

Customers Who Viewed This Item Also Viewed
Challenges in Generating Explanations

- Factorization models are hard to explain
  - The ability to recommend without clear content information
  - High rating prediction accuracy
  - Latent Factor Models (LFM) have achieved significant success

- The latent features make it difficult to explain the recommendation results to users

Can we have a solution that is both highly **accurate** and easily **explainable**?
However

One of the underlying reason

How users compose the different attributes of a product into a single numerical rating.
Textual Reviews Could be Helpful

Numerical Star Rating

Review Text

By Anand (Chennai) - See all my reviews
Exceeds the expectation, June 18, 2013
I am very happy to have bought this phone from Amazon and the service rendered from the seller is excellent. Phone quality is perfect as new though I bought an used one. Care to their customers is something a key strategy the seller has followed. I would like to deal again with the same group in near future and recommend to others highly. Thank you.

Service – Excellent
Phone quality – Perfect
The Role of Textual Reviews

- Phrase-level Sentiment Analysis
  - To extract product features and user opinions from reviews

The service from the seller is excellent, but the battery life is short.
Construct a sentiment lexicon from large amount of textual user reviews.

Service, Battery life, Quality …

Feature Word Set

Opinion Word Set

Feature-Opinion Pairs

Labeled Feature-Opinion Pairs

Review Corpus

Excellent, Short, High …

(Service, Excellent)
(Batter life, Short)
(Quality, High) …

(Service, Excellent, +1)
(Batter life, Short, -1)
(Quality, High, +1) …
Two basic properties to note

The sentiment lexicon is *domain specific*
- Different product domain may have different product feature words and user opinion words

The sentiment lexicon is *contextual*
- The same opinion word may exhibit different sentiment with different feature word
- (Quality, High, +1) vs (Noise, High, -1)
Our Approach: the Intuition

- To recommend a product that performs well on the features that a user concerns.
Our Approach: the Intuition

- To recommend a product that performs well on the features that a user concerns.

Users pay attention to different features

Review Corpus → Sentiment Lexicon

Items perform well on different features

Battery, OS, Color
Memory, Earphone, Price
Screen, Service
Brand
Our Approach: the Intuition

➢ To recommend a product that performs well on the features that a user concerns.
## Structure the Textual Reviews

<table>
<thead>
<tr>
<th>Feature</th>
<th>Opinion</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>perfect</td>
<td>1</td>
</tr>
<tr>
<td>Earphone</td>
<td>good</td>
<td>1</td>
</tr>
</tbody>
</table>

**Star Rating:** 4 stars  
**Review Text:** Screen is perfect, but earphone is not that good.

- (screen, perfect, 1) [normal]  
- (earphone, good, 1) [reversed]  
- (screen, 1), (earphone, -1)

- Extract the Feature-Opinion pairs contained therein  
- Detect whether the sentiment is reversed by negation words  
- Calculate the real sentiment expressed on each feature
User-Feature Attention Matrix

\[ X_{ij} = \begin{cases} 
0, & \text{if user } u_i \text{ did not mention feature } F_j \\
1 + (N - 1) \left( \frac{2}{1 + e^{-t_{ij}}} - 1 \right), & \text{else}
\end{cases} \]

\( t_{ij} \) is the frequency that user \( u_i \) mentions feature \( F_j \)
### Item-Feature Quality Matrix

<table>
<thead>
<tr>
<th>Battery</th>
<th>Price</th>
<th>OS</th>
<th>Memory</th>
<th>Color</th>
<th>Screen</th>
<th>Service</th>
<th>Brand</th>
</tr>
</thead>
</table>

$Y_{ij} = \begin{cases} 
0, & \text{if item } p_i \text{ is not reviewed on feature } F_j \\
1 + \frac{N - 1}{1 + e^{-k \cdot s_{ij}}}, & \text{else} 
\end{cases}$

- $k$ is the frequency feature $j$ is mentioned on item $i$
- $s_{ij}$ is the average sentiment of these mentions
Multi-Matrix Factorization

- Integrating the Explicit and Implicit Features

\[ \text{minimize} \quad \left\{ \| PQ^T - A \|^2_F + \lambda_x \| U_1 V^T - X \|^2_F + \lambda_y \| U_2 V^T - Y \|^2_F \right. \\
+ \left. \lambda_u (\| U_1 \|_F^2 + \| U_2 \|_F^2) + \lambda_h (\| H_1 \|_F^2 + \| H_2 \|_F^2) + \lambda_v \| V \|_F^2 \right\} \]

\[ P = [U_1 \quad H_1], \quad Q = [U_2 \quad H_2] \]
To select the hyper-parameters, we first randomly initialize the five parameters, and tune them one-by-one with the remaining four fixed. This procedure is conducted several times and we select the best choice.
How to Generate Recommended List

User-based feature selection: select the top-k most cared features (with the highest predicted values) to conduct vector multiplication.

For each user $i$, rank the items with the ranking score:

$$R_{ij} = \alpha \cdot \frac{\sum_{c \in C_i} \tilde{X}_{ic} \tilde{Y}_{jc}}{kN} + (1 - \alpha) \tilde{A}_{ij}$$
How to Generate Recommended List

User-based feature selection: select the top-k most cared features (with the highest predicted values) to conduct vector multiplication.

For each user $i$, rank the items with the ranking score:

$$R_{ij} = \alpha \cdot \frac{\sum_{c \in C_i} \tilde{X}_{ic} \tilde{Y}_{jc}}{kN} + (1 - \alpha) \tilde{A}_{ij}$$
Explanations Could be Very Helpful

- **Scrutability**: Make the system more transparent and easier to understand.

- **Effectiveness**: Increase users’ confidence or trust in the system, help users make better decisions.

- **Efficiency**: Help users to make decisions faster.

- **Persuasiveness**: Convince users to try or buy.

- **Satisfaction**: Increase the ease of the user enjoyment.
How to Generate Explanations

Feature-level explanation for a recommended item

You might be interested in [feature], on which this product performs well.

For each user $u_i$ and a recommended item $p_j$, the feature used for explanation construction is $F_c$, where:

$$c = \arg\max_{c \in C_i} \tilde{Y}_{jc},$$

Provide disrecommendations by telling the user why the current browsing item is disrecommended

You might be interested in [feature], on which this product performs poorly.

$$c = \arg\min_{c \in C_i} \tilde{Y}_{jc}$$
Experiments: Setup

- Offline experiment to evaluate recommendation accuracy
  - Rating Prediction & Top-K Recommendation
  - Yelp (English) and Dianping (Chinese) user review datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#users</th>
<th>#items</th>
<th>#reviews</th>
<th>#reviews/ #user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>45,981</td>
<td>11,537</td>
<td>229,907</td>
<td>5.00</td>
</tr>
<tr>
<td>Dianping</td>
<td>11,857</td>
<td>22,365</td>
<td>510,551</td>
<td>43.06</td>
</tr>
<tr>
<td>Yelp10</td>
<td>4,393</td>
<td>10,801</td>
<td>138,301</td>
<td>31.48</td>
</tr>
</tbody>
</table>

- Online A/B test to evaluate explanation effectiveness
  - Recommendation explanation on a major e-commerce web site
  - Focus on the persuasiveness of explanation
Results: Rating Prediction is Improved

- Ratio of Explicit ($U_1 \ U_2$) and Hidden Factors ($H_1 \ H_2$)
  - Fix $r+r'=100$ and tune their ratio
  - Set $r=100$ in comparable algorithms for equal model complexity

When an appropriate number of explicit factors is used, our EFM algorithm is better.
Results: Top-K Recommendation is Improved

- **Comparative Algorithms**
  - **MostPopular**: Rank items by popularity
  - **SlopeOne**: Neighborhood-based algorithm [Lemire 2005]
  - **NMF**: Non-negative Matrix Factorization [Ding and Lee 2001]
  - **BPRMF**: Bayesian Personalized Raking (BPR) optimization for Matrix Factorization (MF) [Rendle 2009]
  - **HFT**: Hidden Factors as Topics [McAuley 2013, Recsys]
Results: Top-K Recommendation is Improved

Number of Most Cared Features $k$

$$R_{ij} = \alpha \cdot \frac{\sum_{c \in C_i} \tilde{X}_{ic} \tilde{Y}_{jc}}{kn} + (1 - \alpha) \tilde{A}_{ij}$$

NDCG of EFM rises with the increase of $k$ until about 15

Tends to be stable before it begins to drop when $k = 45$

However, results on AUC is better consistently

(a) NDCG vs $k$
(b) AUC vs $k$

AUC evaluates only the pairwise rankings rather than the positions
Further Analysis of Explicit Features

It’s beyond expectation that a user considers tens of features.

Coverage in term frequency of the top-k most cared features

\[
\text{Coverage@}k = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{\sum_{j \in C_i} t_{ij}}{\sum_{j=1}^P t_{ij}}
\]

A small number of explicit features could dominate the term frequency in textual reviews.

This verifies our assumption to use the most cared features for recommendation.

Results: Top-K Recommendation is Improved
Further Analysis of Explicit Features

Why users consider tens of explicit features?

We group the explicit features into synonym clusters

WordNet for English and HowNet for Dianping

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Feature</th>
<th>#Cluster</th>
<th>#F/#C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp10</td>
<td>96</td>
<td>31</td>
<td>3.10</td>
</tr>
<tr>
<td>Dianping</td>
<td>113</td>
<td>26</td>
<td>4.35</td>
</tr>
</tbody>
</table>

Each synonym cluster has 3~4 explicit features on average.

Users just use different words to express similar concepts!
Further Analysis of Explicit Features

The top 15 features can be fully included in the top 7 clusters:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Features</th>
<th>Rank</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>place, restaurant, location, area, way</td>
<td>2</td>
<td>food, menu, lunch, pizza, dinner</td>
</tr>
<tr>
<td>3</td>
<td>service, time, staff, order</td>
<td>4</td>
<td>experience, quality</td>
</tr>
<tr>
<td>5</td>
<td>room, atmosphere, decor</td>
<td>6</td>
<td>price, cost</td>
</tr>
<tr>
<td>7</td>
<td>beer, wine, drink, water, coffee</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relations with previous work

Consistent with the Hidden Factors as Topics (HFT) model [McAuley 2013, Recsys]

Where the authors find that the performance would not improve with more than 10 topics.

They could be ‘long tail’, redundant topics.
Online Experiment for Explanations: Setup

- Provide mobile phone recommendation by a popular commercial web browser in an e-commerce website.

![Image showing recommended items in an e-commerce website]

**List:** Recommended Items

**Indicator:** Whether the current browsing item is recommended or not
Online Experiment for Explanations: Setup

- The explanations are displayed when user hover the mouse on an recommended item.
  - To ensure that the users examined the explanations
  - Word cloud to show the detailed performance on features
Click Through Rate on Recommendation List

- Design 3 user groups
  - A (experimental group): Receive our feature-level explanations
  - B (comparison group): Receive the ‘people also viewed’ explanation
  - C (control group): Receive no explanation

- Only consider the records that hovered the mouse on the recommendations
  - As an indication of examining the explanations.

<table>
<thead>
<tr>
<th>User Set</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records</td>
<td>#Record</td>
<td>#Click</td>
<td>#Record</td>
</tr>
<tr>
<td></td>
<td>15,933</td>
<td>691</td>
<td>11,483</td>
</tr>
<tr>
<td>CTR</td>
<td>4.34%</td>
<td>3.22%</td>
<td>3.20%</td>
</tr>
</tbody>
</table>

- Click through rate is significantly higher in group A than B and C.
(Dis)Recommendation with Additional Explanation is More Influential on User Buying Decision

- A group: receives the feature-level explanations
- B group: receives no explanation

We didn’t assign other comparison groups because there is no previous work presenting disrecommendation explanations.

Explanations help persuade a user to add a recommended product to shopping cart or to ignore a disrecommended product.
Attempt to bring new insights into the problem of recommendation explanation

Incorporate phrase-level sentiment analysis into recommender systems

Propose the Explicit Factor Models for both accurate recommendation and intuitional explanations

Good performance on both offline and online A/B tests