Matrix Factorization Techniques
For Recommender Systems

Collaborative Filtering

Markus Freitag, Jan-Felix Schwarz
28 April 2011
1. Paper Backgrounds
2. Latent Factor Models
3. Overfitting & Regularization
4. Learning Algorithms
5. Biases & Temporal Dynamics
6. Paper Evaluation
7. Application For KDD Cup
8. Discussion
Matrix Factorization Techniques For Recommender Systems

Yehuda Koren, Yahoo Research
Robert Bell and Chris Volinsky, AT&T Labs-Research

Paper published in August 2009
Authors won the grand Netflix Prize in September 2009
Latent Factor Models

- Find features that describe the characteristics of rated objects
- Item characteristics and user preferences are described with numerical factor values
- Assumption: Ratings can be inferred from a model put together from a smaller number of parameters
Latent Factor Models

- Items and users are associated with a factor vector
- Dot product captures the user’s estimated interest in the item:

\[ \hat{r}_{ui} = q^T_i p_u \]

- Challenge: How to compute a mapping of items and users to factor vectors?

- Approaches:
  - Singular Value Decomposition (SVD)
  - Matrix Factorization
Singular Value Decomposition

Rating Matrix (N x M)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

User Feature Matrix (F x N)

\[
\begin{bmatrix}
  f_1 & 1 & -4 & 1 \\
  f_2 & -2 & 0 & -3 \\
  f_3 & 0 & -5 & 1 \\
\end{bmatrix}
\]

Movie Feature Matrix (F x M)

\[
\begin{bmatrix}
  f_1 & -1 & 0 & -2 \\
  f_2 & 4 & -4 & 1 \\
  f_3 & 0 & 2 & 2 \\
\end{bmatrix}
\]
Singular Value Decomposition

\[
\begin{pmatrix}
 f_1 & 1 & -4 & 1 \\
 f_2 & -2 & 0 & -3 \\
 f_3 & 0 & -5 & 1 \\
\end{pmatrix}^T
\begin{pmatrix}
 f_1 & -1 & 0 & -2 \\
 f_2 & 4 & -4 & 1 \\
 f_3 & 0 & 2 & 2 \\
\end{pmatrix} =
\begin{pmatrix}
 5 & 3 & 5 \\
 4 & 2 & 1 \\
 0 & 3 & 3 \\
\end{pmatrix}
\]
SVD - Problems

- Conventional SVD is undefined for incomplete matrices!

- Imputation to fill in missing values
  - Increases the amount of data
  - “SVD of ginormous matrices is... well, no fun“ (Simon Funk)

- We need an approach that can simply ignore missing ratings
Matrix Factorization Techniques for Recommender Systems

\[ \min_{q^*, p^*} \sum_{(u, i) \in K} (r_{ui} - q_i^T p_u)^2 \]

\( r_{ui} \): known rating of user \( u \) for item \( i \)

remember:
predicted rating \( \hat{r}_{ui} = q_i^T p_u \)
A model is built to represent the training data – not to reproduce the training data.
Matrix Factorization - Regularization

Idea: penalize complexity

\[
\min_{q^*,p^*} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2)
\]

\[\lambda:\] constant to control the extend of regularization
\rightarrow determined by cross-validation
Learning Algorithms

- **Stochastic gradient descent**
  - Modification of parameters \((q_i, p_u)\) relative to prediction error
  - Recommended algorithm

- **Alternating least squares**
  - Allows massive parallelization
  - Better for densely filled matrices
Learning Algorithms

- Calculation of the prediction error
  - Error = actual rating – predicted rating
  - \( e_{ui} = r_{ui} - q^T_i p_u \)

- Modification
  - By magnitude proportional to \( \gamma \)
  - In the opposite direction of the gradient
  - \( q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \)
Biases

- Item or user specific rating variations are called biases

- Example:
  - Alice rates no movie with more than 2 (out of 5)
  - Movie X is hyped and rated with 5 only

- Matrix factorization allows modeling of biases

- Including bias parameters in the prediction:
  \[
  \hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u
  \]
Temporal Dynamics

- Ratings may be affected by temporal effects
  - Popularity of an item may change
  - User’s identity and preferences may change
- Modeling temporal affects can improve accuracy significantly

- Rating predictions as a function of time:

\[ \hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t) \]
Paper Evaluation

- High-level overview of matrix factorization techniques
- Mathematical foundations are pointed out but not elaborated
- Many useful references to related work
- Authors do not reveal their secret implementation tidbits
Application For KDD Cup

- Different item types
  - We must assume different prediction models!
  - We have explicit dependencies between items

- How to apply Matrix Factorization to the KDD data set?
  - Segment training data in separate sets for each type
  - Consider ratings for dependent items to make a prediction

Matrix Factorization Techniques For Recommender Systems
Application For KDD Cup - Hypotheses

- Users change their taste in music
- Users tend to rate new songs better
  - Users get tired of songs
- Some artists and albums are hyped for a while
- Evergreens
  - Loved by many, but maybe also hated by some
- ...
References


Discussion

Summary
- Matrix factorization is a promising approach for collaborative filtering
- Factor vectors are learned by minimizing the RSME
- Regularization to prevent overfitting
- Addition of bias parameters and temporal dynamics further improve accuracy

Outlook
- Develop strategies for applying matrix factorization on our data set with different item types
- Make use of the available dependencies between items
- Explore biases and rating behaviors specific for our music domain