Convolutional Matrix Factorization for Document Context-Aware Recommendation

Donghyun Kim¹, Chanyoung Park¹, Jinoh Oh¹, Sungyoung Lee², Hwanjo Yu*¹

¹Datamining Lab @ POSTECH
²Ubiquitous Computing Lab @ Kyunghee University

*corresponding author
Matrix Factorization (MF)

- A popular model-based collaborative filtering for recommendation

$$r_{ij} \approx u_i^T v_j$$

$$\begin{array}{c|c|c|c}
\text{Users} & \text{Items} \\
\hline
5 & ? & ? & 3 \\
4 & ? & ? & 2 \\
? & 1 & 3 & 1 \\
\end{array}$$
Matrix Factorization (MF)

• A popular model-based collaborative filtering for recommendation

\[ \mathbf{u}_i^T \mathbf{v}_j = \hat{r}_{ij} \]

# of users × # of items

Users

<table>
<thead>
<tr>
<th>5</th>
<th>?</th>
<th>?</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>?</td>
<td>?</td>
<td>2</td>
</tr>
<tr>
<td>?</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Matrix completion

User latent models

Item latent models

Predicted ratings
Matrix Factorization (MF)

- A popular model-based collaborative filtering for recommendation

\[ u_1^T v_3 = \hat{r}_{1,3} \]
Matrix Factorization (MF)

• A popular model-based collaborative filtering for recommendation

\[ \mathbf{u}_2^T \mathbf{v}_2 = \hat{r}_{2,2} \]
Matrix Factorization (MF)

- A popular model-based collaborative filtering for recommendation

\[ u_3^T v_1 = \hat{r}_{3,1} \]
User and item latent models in 2D space!

However, when the rating matrix becomes extremely sparse...

- **Dark Knight** (action)
- **Inception** (action)
- **A Beautiful Mind** (drama)
- **Interstellar** (drama)
User and item latent models in 2D space!

MF

User

Item

# of users

# of items

Drama

Action

Interstellar (drama)

A Beautiful Mind (drama)

Dark Knight (action)

Inception (action)
User and item latent models in 2D space!
Common approaches

• To handle sparseness of a rating matrix, text information (review, synopsis, abstract, etc.) has been widely used in recent researches. [KDD`15, RecSys`14, RecSys`13, KDD`11]
Common approaches

• Trial to understand description documents for recommendation
Common approaches

• Trial to understand description documents for recommendation
  • Collaborative topic modeling for scientific articles (CTR) [KDD’11]
    • Latent Dirichlet Allocation (LDA)
Common approaches

• Trial to understand description documents for recommendation
  • Collaborative topic modeling for scientific articles (CTR) [KDD’11]
    • Latent Dirichlet Allocation (LDA)
  • Collaborative deep learning for recommender system (CDL) [KDD’15]
    • Stack Denoising AutoEncoder (SDAE)
Drawback of common approaches

• Trial to understand description documents for recommendation
  • Collaborative topic modeling for scientific articles (CTR) [KDD`11]
    • Latent Dirichlet Allocation (LDA)
  • Collaborative deep learning for recommender system (CDL) [KDD`15]
    • Stack Denoising AutoEncoder (SDAE)

• However, LDA and SDAE analyze “bag of words models” of item descriptions to generate latent models.
“Contextual information” in documents

• Considering surrounding words and word order as “contextual information” improves the accuracy of word vectors in the word embedding.
  • Word2Vec [NIPS`13]

• What if recommender systems are able to capture contextual information in documents?
  • Generate more accurate item latent models through a deeper understanding of item descriptions.

• Thus, contextual information should be considered for better recommendation!
Our proposed model

• We develop a novel document context-aware recommendation model, **Convolutional Matrix Factorization** (ConvMF).
  • To consider contextual information
  • To effectively exploit both ratings and description documents
  • To jointly optimize the recommendation model in order to properly predict ratings to items of users
Inspired by Convolutional Neural Network (CNN)

• For the NLP and IR tasks, convolutional neural networks (CNNs) have been mainly developed to consider local contextual information in a document.
  • NLP: [JMLR’11, ACL’14, EMNLP’14], IR: [EMNLP’14, CIKM’14]

• An example of CNN architecture for sentiment classification. [EMNLP 2014]
Overview of our CNN architecture

• Trial to generate more accurate item latent models
Embedding layer – *word embedding*

- Transform a raw description document into a **numeric document matrix**.

[Diagram showing a process involving document latent vector, output layer projection, pooling layer, max pooling, convolution layer, convolution, embedding layer, and an example text: "...people trust the man. people betray his trust finally. ..."]

(pre-trained) word embedding models.
Convolution layer – **contextual information**

- Extract *contextual features* from a document matrix.
Convolution layer – *contextual information*

- For example (window size: 3)

\[ c = [c_1, c_2, \ldots, c_i, \ldots, c_{l-\text{ws}+1}] \]

... people betray his trust finally ...

... people betray his trust finally ...

... people betray his trust finally ...

... people betray his trust finally ...

... people trust the man. people betray his trust finally. ...
Pooling layer – representative information

- Extract **representative features** from the convolutional layer

![Diagram showing the pooling layer process]

deal with variable lengths of documents
Output layer – high level features of documents

• Project representative features to a \( k \)-dimensional space
Then, how to predict ratings?

- However, the direct usage of CNNs is not suitable for a recommendation task.
Probabilistic Matrix Factorization (PMF) [NIPS`08]

• Ratings can be approximated by probabilistic methods.

<The graphical model of PMF>
How about PMF + CNN?

• Overview of ConvMF
  • We integrate CNN into PMF for the recommendation task.
Graphical model of ConvMF

- Overview of ConvMF
  - We integrate CNN into PMF for the recommendation task.
Key of connection – Item variable

• Overview of ConvMF
  • **Item variable** plays a role of the connection between PMF and CNN in order to exploit ratings and description documents.
Optimization Methodology

- Use **maximum a posteriori** to solve $U$, $V$ and $W$
  - $\max_{U,V,W} p(U, V, W | R, X, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_W^2) =$
    $\max_{U,V,W} p(R | U, V, \sigma^2) p(U | \sigma_U^2) p(V | W, X, \sigma_V^2) p(W | \sigma_W^2)$
  - By taking negative logarithm,
    $$
    \mathcal{L}(U, V, W) = \sum_i^{N} \sum_j^{M} \frac{I_{ij}}{2} (r_{ij} - u_i^T v_j)^2 + \frac{\lambda_U}{2} \sum_i^{N} \|u_i\|_2
    $$
    $$
    + \frac{\lambda_V}{2} \sum_j^{M} \|v_j - cnn(W, X_j)\|_2 + \frac{\lambda_W}{2} \sum_k \|w_k\|_2,
    $$
  - Use coordinate descent to update latent models per iteration
    $$
    u_i \leftarrow (V I_i V^T + \lambda_U I_K)^{-1} V R_i
    $$
    $$
    v_j \leftarrow (U I_j U^T + \lambda_V I_K)^{-1} (U R_j + \lambda_V cnn(W, X_j))
    $$
    $\lambda_v$ balances between ratings and documents
Optimization Methodology

• However, \( W \) cannot be solved analytically as we can do for \( U \) and \( V \).
Optimization Methodology

• However, $W$ cannot be solved analytically as we can do for $U$ and $V$.

• Fortunately, when $U, V$ are temporarily fixed, loss function $\mathcal{L}$ becomes an error function with regularized terms of neural net.

\[
\mathcal{E}(W) = \frac{\lambda_V}{2} \sum_j^M \| (v_j - \text{cnn}(W, X_j)) \|^2 \\
+ \frac{\lambda_W}{2} \sum_k |w_k| \|w_k\|^2 + \text{constant}
\]

• To optimize $W$, we use backpropagation algorithm with given target value $v_j$. 
Explicit feedback datasets (range from 1 to 5)

<table>
<thead>
<tr>
<th>Dataset</th>
<th># users</th>
<th># items</th>
<th># ratings</th>
<th>density</th>
<th>documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens-1m (ML-1m)</td>
<td>6,040</td>
<td>3,544</td>
<td>993,482</td>
<td>4.641%</td>
<td>IMDB</td>
</tr>
<tr>
<td>MovieLens-10m (ML-10m)</td>
<td>69,878</td>
<td>10,073</td>
<td>9,945,875</td>
<td>1.413%</td>
<td>IMDB</td>
</tr>
<tr>
<td>Amazon Instant Video (AIV)</td>
<td>29,757</td>
<td>15,149</td>
<td>135,188</td>
<td>0.030%</td>
<td>Amazon Review</td>
</tr>
</tbody>
</table>

AIV is the most skewed and sparse dataset!

In AIV, 50% of items have only one rating!
Experiment Setting

• Competitor
  • PMF [NIPS’08] – conventional MF
  • CTR [KDD’11] – the state-of-the-art LDA-integrated recommendation
  • CDL [KDD’15] – the state-of-the-art SDAE-integrated recommendation
  • ConvMF – our proposed model
  • ConvMF+ – our proposed model with the pre-trained word embedding model (Glove)

• Measure
  • Follow the convention in recommender system.

\[
RMSE = \sqrt{\frac{\sum_{i,j}^{N,M} (r_{ij} - \hat{r}_{ij})^2}{\# \text{ of ratings}}}
\]
Overall performance comparison

- RMSE – training / valid / test dataset (80% / 10% / 10%)

<table>
<thead>
<tr>
<th>Model</th>
<th>PMF</th>
<th>CTR</th>
<th>CDL</th>
<th>ConvMF</th>
<th>ConvMF+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.8971 (0.0020)</td>
<td>0.8311 (0.0010)</td>
<td>1.4118 (0.0105)</td>
<td>0.8969 (0.0027)</td>
<td>0.8275 (0.0004)</td>
</tr>
<tr>
<td></td>
<td>0.8879 (0.0015)</td>
<td>0.8186 (0.0005)</td>
<td>1.3594 (0.0139)</td>
<td><strong>0.8531 (0.0018)</strong></td>
<td><strong>0.7958 (0.0006)</strong></td>
</tr>
<tr>
<td>ConvMF</td>
<td><strong>0.8531 (0.0018)</strong></td>
<td><strong>0.7958 (0.0006)</strong></td>
<td><strong>1.1337 (0.0043)</strong></td>
<td><strong>0.8549 (0.0018)</strong></td>
<td><strong>0.7930 (0.0006)</strong></td>
</tr>
<tr>
<td>ConvMF+</td>
<td><strong>0.8549 (0.0018)</strong></td>
<td><strong>0.7930 (0.0006)</strong></td>
<td><strong>1.1279 (0.0073)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improve</td>
<td>3.92%</td>
<td>2.79%</td>
<td>16.60%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ConvMF and ConvMF+ achieve significant improvements on all the datasets.

*Extremely sparse dataset!*
Best performing parameter analysis – $\lambda_u$ and $\lambda_v$

When considering that $\lambda_v$ balances between ratings and documents, this natural pattern implies that ConvMF is well modeled.

MovieLens-1m
$\lambda_u$: 100
$\lambda_v$: 10

MovieLens-10m
$\lambda_u$: 10
$\lambda_v$: 100

Amazon Instant Video
$\lambda_u$: 1
$\lambda_v$: 100

More skewed and sparse dataset
Impact of pre-trained word embedding model

• On AIV dataset

![Graph showing the relationship between RMSE and the dimension size of word embedding. The graph indicates that as the dimension size increases, the RMSE decreases, suggesting that the model's information becomes richer. The lower the RMSE, the better the model's performance.](image-url)

*Lower is better

*Information contained in the model gets richer*
Case study of subtle contextual differences

The only max feature value affects the performance of ConvMF.
⇒ A higher value has more chance to affect the performance!

<table>
<thead>
<tr>
<th>Phrase captured by $W_c^{11}$</th>
<th>$\max(c^{11})$</th>
<th>Phrase captured by $W_c^{86}$</th>
<th>$\max(c^{86})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>people <strong>trust</strong> the man</td>
<td>0.0704</td>
<td>betray his <strong>trust</strong> finally</td>
<td>0.1009</td>
</tr>
<tr>
<td><strong>Test phrases for $W_c^{11}$</strong></td>
<td>$\max(c_{test}^{11})$</td>
<td><strong>Test phrases for $W_c^{86}$</strong></td>
<td>$\max(c_{test}^{86})$</td>
</tr>
<tr>
<td>people <strong>believe</strong> the man</td>
<td>0.0391</td>
<td>betray his <strong>believe</strong> finally</td>
<td>0.0682</td>
</tr>
<tr>
<td>people <strong>faith</strong> the man</td>
<td>0.0374</td>
<td>betray his <strong>faith</strong> finally</td>
<td>0.0693</td>
</tr>
<tr>
<td>people <strong>tomas</strong> the man</td>
<td>0.0054</td>
<td>betray his <strong>tomas</strong> finally</td>
<td>0.0480</td>
</tr>
</tbody>
</table>

$W_c^{11}$ is more likely to capture “**trust**” as a **verb**  
$W_c^{86}$ is more likely to capture “**trust**” as a **noun**

**ConvMF distinguishes a subtle contextual difference of the term "trust"**
Conclusion

• We demonstrate that considering **contextual information** provides a deeper understanding of description documents.

• We develop a novel document context-aware recommendation model, ConvMF, that seamlessly integrates CNN into PMF in order to capture contextual information for the rating prediction.

• Since ConvMF is based on PMF, ConvMF is able to be extended to combining other MF-based recommendation models such as SVD++.
Thank you

• ConvMF webpage
  • http://dm.postech.ac.kr/ConvMF

• Any question?
Reference

- [KDD`15] Collaborative deep learning for recommender systems
- [RecSys`14] Ratings meet reviews, a combined approach to recommend
- [RecSys`13] Hidden factors and hidden topics: Understanding rating dimensions with review text
- [IJCAI`13] Hierarchical Bayesian matrix factorization with side information
- [NIPS`13] Deep content-based music recommendation
- [ICML`12] Collaborative topic regression with social matrix factorization for recommendation systems
- [KDD`11] Collaborative topic modeling for recommending scientific articles
- [JMLR`11] Natural language processing (almost) from scratch
- [ACL`14] A convolutional neural network for modelling sentences
- [EMNLP`14] Convolutional neural networks for sentence classification
- [EMNLP`14] Modeling interestingness with deep neural networks
- [CIKM`14] A latent semantic model with convolutional-pooling structure for information retrieval