User Recommendation in Content Curation Platforms

Jianling Wang, Ziwei Zhu and James Caverlee
Content Creation vs Curation

Content creators generate new digital artifacts such as tweets, blog posts, or photos.
Content Creation vs Curation

Music Streaming platforms allow users to create and share playlists.
Content Creation vs Curation

Goodreads provide a platform for users to curate interesting books via tagging, ratings and reviews.
Our Goal: Recommend Curators

In Content Curation Platforms, users acting as curators, collect and organize existing content via reviews, pins, boards, ratings and other actions.
Our Goal: Recommend Curators

Compared with:

- Item-level recommendation, e.g., recommend music tracks
  
  *There are many new items or items with little feedback.*

- Curation-level recommendation, e.g., recommend playlists
  
  *Curations (e.g. pin boards, playlists) are frequently updated.*
Why Recommend Curators?

- **Curators** can provide a human-powered overlay that can link seemingly unrelated items (e.g., a collection of songs that are thematically related though from different genres).
Why Recommend Curators?

- By receiving updates from whom they follow, users can be exposed to interesting items and curation decisions.
Our Setting

We can collect:

- User-curator following relationships
- Implicit feedback on items
Challenge

How to model these **two aspects** - curator preferences and item preferences - in a unified model?
The Goal

We are motivated to develop a new model for Curator Recommendation that leverages the linkage between user-curator following relationships and the items they are interested in.
The Joint Tasks

Ultimately, the model aims to provide users with recommendation on:

- who to follow (the primary task)
- interesting items (the supplementary task)
CuRe - *Curator Recommendation*

Three components:

• Learning Curator & Item Preferences

• Fusing Latent Representations

• Personalized via Attention
Use **Denoising Autoencoder (DAE)** to uncover the latent representation of user preference on curators.
Uncover the Preferences

Use Denoising Autoencoder (DAE) to uncover the latent representation of user preference on curators.
Uncover the Preferences

We can enrich the preference on curators with preference on items.

Feedback Vector on Curators

Feedback Vector on Items
Uncover the Preferences

We can enrich the preference on curators with preference on items.
Uncover the Preferences

A Joint Curator-Item DAE model

Feedback Vector on Curators

Shared Latent Factors

Feedback Vector on Items
What’s Next?

The element at the same dimension in $h^C$ and $h^I$ may not correspond to the same latent factor.
What’s Next?

How to assign personalized weights on $h^C$ and $h^I$?

Feedback Vector on Curators

$V \rightarrow h^C$

Shared Latent Factors

Feedback Vector on Items

$V^I \rightarrow h^I$
Use a Discriminator to force $h^C$ and $h^I$ to live in a shared space.
Personalized Fusing

Generate the user-dependent weights for $h^C$ and $h^I$ via an attention layer.
Personalized Fusing

Generate the user-dependent weights for $h^C$ and $h^I$ via an attention layer.
CuRe - **Curator Recommendation**

Adversarial loss for distinguishing $h^C$ and $h^I$

Input \rightarrow Discriminator \rightarrow Adversarial loss \rightarrow Feedback Vector on Curators

Feedback Vector on Items

Full-Connected Layers

Isolated Latent Factors

Shared Latent Factors

Attention Layer

Feedback Vector on Items
## Experiment: Data

### Two Datasets:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#User</th>
<th>#Item</th>
<th>#User-User Interactions</th>
<th>#User-Item Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goodreads</strong></td>
<td>48,208</td>
<td>61,848</td>
<td>528,816</td>
<td>10,526,215</td>
</tr>
<tr>
<td><strong>Spotify</strong></td>
<td>25,471</td>
<td>70,107</td>
<td>227,024</td>
<td>4,499,741</td>
</tr>
</tbody>
</table>
Experiment: Metric

- F1@K: combination of recall and precision
- NDCG@K: takes the position of recommendations into consideration
- K=5, 10
Experiment: Baselines

Compare with the widely used recommendation frameworks:

- **MP**: Most Popular
- **UCF**: User-based collaborative filtering
- **BPR**: Matrix Factorization with Bayesian Personalized Ranking
Experiment: Baselines

Compare with recommendation frameworks enhanced with an adversarial component or built on Autoencoder:

- **AMF**: Adversarial Matrix Factorization
- **DAE**: Denoising Autoencoder
- **CDAE**: Collaborative Denoising Autoencoder
- **VAE**: Variational Autoencoder for Collaborative Filtering
Experiment: Baselines

Additional Approaches considering both user-user and user-item interactions:

- **EMJ**: Embedding Factorization odes for Joint Recommendation
- **Joint-DAE**: A simplified version of CuRe without adversarial learning process and the attention layer.
CuRe vs Baselines

• The proposed model outperforms the state-of-the-art in recommending curators (by 18% in Goodreads, 6% in Spotify).

• Simultaneously, it is able to achieve significant improvements in item recommendation compared with the baselines.

• Larger improvements under the cold-start setting.
Impact of each component?

Utilizing feedback on items can help in inferring preferences on curators.

![Graph showing the impact of different components](image-url)
Impact of each component?

The adversarial component enables the model to achieve better performance in less epochs.
Impact of each component?

Providing personalized fusing is important for achieving the improved performance in both tasks.
Conclusion

• New Problem - *Curator Recommendation*

• Joint Recommendation for *a primary and a supplementary task*.

• Experiments prove that the proposed models can outperform the state-of-the-art in both the primary and the supplementary tasks.

• The next step…

  • Can we support various types of interactions between users?
  • How to capture the temporally dynamic patterns of curators?