A Deep Latent Recommender System based on User Ratings and Reviews

Dingge LIANG
(Joint work with M. Corneli, C. Bouveyron and P. Latouche)

Université Côte d’Azur, INRIA MaasAI
Introduction: towards a topic-aware recommender system

Consider a dataset with $M$ users scoring/reviewing $P$ products, encoded by two matrices:

- $Y$ is an ordinal matrix in $\mathbb{N}^{M \times P}$, with $Y_{ij}$ the rating of the object $j$ by user $i$.
- $W$ is a document-term matrix (DTM), with $W^{(i,j)}$ encoding a review about object $j$ by user $i$.

![Figure: Two encoded matrices.](image)

(a) Ordinal matrix  (b) Document-term matrix
The generative process and VAE inference

We assume the following generative process for rating and review:

- The rating $Y_{ij}$ is:
  \[ Y_{ij} = \langle R_i, C_j \rangle + \epsilon_{ij}, \]  
  where $R_i \sim \mathcal{N}(0, I_D)$, $C_j \sim \mathcal{N}(0, I_D)$, $\epsilon_{ij} \sim \mathcal{N}(0, \eta^2)$.  
- For the reviews, we assume that:
  \[ p(W^{(i,j)} | \theta_{ij}) \sim \text{Multinomial}(L_{ij}, \beta \theta_{ij}), \]  
  where
  \begin{itemize}
  \item $L_{ij}$ is the number of words in $W^{(i,j)}$,  
  \item $\beta \in \mathbb{R}^{V \times K}$ is the matrix whose entry $\beta_{vk}$ is the probability that word $v$ occurs in topic $k$,  
  \item $\theta_{ij} = \sigma(f_{\gamma}(R_i, C_j))$ is the topic proportion, where $f_{\gamma} : \mathbb{R}^{2D} \rightarrow \mathbb{R}^K$ is a continuous function approximated by a neural network parametrized by $\gamma$, $\sigma(\cdot)$ denotes the Softmax function.
  \end{itemize}

A Variational auto-encoder is used for the inference:
\[
\log p(Y, W | \eta^2, \gamma, \beta) \geq \mathbb{E}_{q(R,C)} \left[ \log \frac{p(Y, W, R, C | \eta^2, \gamma, \beta)}{q(R,C)} \right],
\]
where
\[
q(R_i) = g(\mu^R_i := h_1, \phi(Y_i, W^{(i, \cdot)}), S^R_i := h_2, \phi(Y_i, W^{(i, \cdot)})),
\]
\[
q(C_j) = g(\mu^C_j := l_1, \iota(Y_j, W^{(\cdot, j)}), S^C_j := l_2, \iota(Y_j, W^{(\cdot, j)})).
\]

**Figure:** A deep learning view of deepLTRS.
Numerical experiments

- Benchmark on simulated data:

```
0.5 0.6 0.7 0.8 0.9 1.0
Effect of the sparsity for deepLTRS and deepLTRS without texts
0.75
1.00
1.25
1.50
1.75
2.00
2.25
Test RMSE
deepLTRS
deepLTRS (no texts)
```

**Figure:** Comparison of deepLTRS with and without texts.

- Amazon Fine Food data:

**Table:** Test RMSE on Amazon Fine Food data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFT</td>
<td>1.4241</td>
<td>1.5327</td>
<td>1.4737</td>
<td>1.4228</td>
<td>1.3850</td>
<td>1.4477 (±0.0510)</td>
</tr>
<tr>
<td>HPF</td>
<td>2.9486</td>
<td>2.9682</td>
<td>2.9311</td>
<td>2.9428</td>
<td>2.9734</td>
<td>2.9528 (±0.0158)</td>
</tr>
<tr>
<td>CCPF-PMF</td>
<td>1.2695</td>
<td>1.2964</td>
<td>1.3035</td>
<td>1.2923</td>
<td>1.2950</td>
<td>1.2913 (±0.0115)</td>
</tr>
<tr>
<td>deepLTRS</td>
<td>1.1364</td>
<td>1.2595</td>
<td>1.2445</td>
<td>1.1710</td>
<td>1.2475</td>
<td>1.2518 (±0.0489)</td>
</tr>
</tbody>
</table>

**Figure:** Test RMSE of models: HFT, HPF, CCPF and deepLTRS with different sparsity level on simulated data.