Deep Language-based Critiquing for Recommender Systems (Appendix)

Ga Wu*†
University of Toronto
wuga@mie.utoronto.ca

Kai Luo†
University of Toronto
kluo@mie.utoronto.ca

Scott Sanner*
University of Toronto
ssanner@mie.utoronto.ca

Harold Soh
National University of Singapore
harold@comp.nus.edu.sg

ACM Reference Format:
Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-based Critiquing for Recommender Systems (Appendix). In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3298689.3347009

1 APPENDIX

1.1 Full Derivation of the Proposed Variational Probabilistic Model

We show the full derivation of the log marginal likelihood of Equation (7) as follows:

$$
\begin{aligned}
&\log \prod_{i,j} p(r_{i,j}, \mathbf{s}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j}) \\
&= \log \prod_{i,j} \int_{\mathbf{z}_{i,j}} p(r_{i,j}, \mathbf{s}_{i,j} | \mathbf{z}_{i,j}) p(\mathbf{z}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j}) d\mathbf{z}_{i,j} \\
&= \log \prod_{i,j} \int_{\mathbf{z}_{i,j}} q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j) \frac{p(r_{i,j}, \mathbf{s}_{i,j} | \mathbf{z}_{i,j}) p(\mathbf{z}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j})}{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)} d\mathbf{z}_{i,j} \\
&\geq \sum_{i,j} \int_{\mathbf{z}_{i,j}} q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j) \log \frac{p(r_{i,j}, \mathbf{s}_{i,j} | \mathbf{z}_{i,j}) p(\mathbf{z}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j})}{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)} d\mathbf{z}_{i,j} \\
&= \sum_{i,j} \underbrace{E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)}[\log p(r_{i,j} | \mathbf{z}_{\mathbf{i,j}})]}_{\mathcal{L}_0} + \sum_{i,j} \underbrace{E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)}[\log p(\mathbf{s}_{i,j} | \mathbf{z}_{\mathbf{i,j}})]}_{\mathcal{L}_1} \\
&\quad + \sum_{i,j} \underbrace{E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)}[\log p(\mathbf{z}_{\mathbf{i,j}} | \tilde{\mathbf{s}}_{i,j})]}_{\mathcal{L}_2} - \sum_{i,j} KL[q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j) || p(\mathbf{z}_{\mathbf{i,j}})],
\end{aligned}
\tag{1}
$$

where we have exploited conditional independence of $r_{i,j}$ and $\mathbf{s}_{i,j}$ given $\mathbf{z}_{\mathbf{i,j}}$.

*Affiliate to Vector Institute of Artificial Intelligence, Toronto
†Both authors contributed equally to this work

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347009

Table 1: Hyperparameters tuned on the experiments.

name	Range	Functionality	Algorithms affected
r	{50, 100, 200}	Latent Dimension	VNCF, E-VNCF, CE-VNCF, NCF, E-NCF CE-NCF, CDAE, BPR, PureSVD
λ	{5e-5, 1e-5, 5e-4, $\cdots$, 1}	Regularization	VNCF, E-VNCF, CE-VNCF, NCF, E-NCF CE-NCF, CDAE, BPR, PureSVD
α	{1e-4, 5e-4, 1e-3, 5e-3}	Learning Rate	VNCF, E-VNCF, CE-VNCF, NCF E-NCF, CE-NCF, CDAE
β	{0.0, 0.1, 0.2, $\cdots$ 0.5}	Corruption Rate	VNCF, E-VNCF, CE-VNCF, CDAE
η	{1,2,3,4,5}	Negative Samples	VNCF, E-VNCF, CE-VNCF, NCF, E-NCF CE-NCF, BPR

We remark that the above derivation drops conditions and approximates $p(\mathbf{z}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j}) = p(\mathbf{z}_{\mathbf{i,j}} | \tilde{\mathbf{s}}_{i,j})$ since we need only learn how to embed critiques given that we can conveniently use the learned encoder $q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)$ to infer the initial user and item embedding when no critiques $\tilde{\mathbf{s}}_{i,j}$ are present at the start. We did try learning a full personalized critique embedding $p(\mathbf{z}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j})$ as an alternative to this approximation but found that it did not significantly change the results on our datasets.

During the training stage, the model is not exposed to the critiquing environment. In order to train the model, we use the predicted explanation $\hat{\mathbf{s}}_{i,j}$ to replace the value of $\tilde{\mathbf{s}}_{i,j}$ without any critiquing. Thus we can rewrite the $\mathcal{L}_2$ term in the following manner:

$$
\begin{aligned}
\mathcal{L}_2 &= \sum_{i,j} E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)}[\log p(\mathbf{z}_{\mathbf{i,j}} | \tilde{\mathbf{s}}_{i,j})] \\
&\approx \sum_{i,j} E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)}[\log p(\mathbf{z}_{\mathbf{i,j}} | \hat{\mathbf{s}}_{i,j})] \\
&= \sum_{i,j} E_{q(\mathbf{z}_{\mathbf{i,j}} | \mathbf{u}_i, \mathbf{v}_j)} \underbrace{[\log p(\mathbf{z}_{\mathbf{i,j}} | f_s(\mathbf{z}_{i,j}))]}_{\textit{Auto-encoder}}
\end{aligned}
\tag{2}
$$

In short, this representation of the $\mathcal{L}_2$ term tries to ensure self-consistency of the latent space with the explanation *and* the critique with the latent space, i.e., *if* the critique was the same as the explanation then we should recover the same embedding that produced the explanation to begin with.

Finally we remark that during learning, all parameters of distributions p and q are learned as well as the embeddings $\mathbf{u}_i$ and $\mathbf{v}_j$, which are left as free parameters to optimize. Further, we use a SVD on the rating matrix as a warm start initialization for $\mathbf{u}_i$ and $\mathbf{v}_j$ that is further fine-tuned during learning.

1.2 Hyperparameter Sweeps

Table 1 presents our hyperparameter sweeps for architecture and algorithm tuning on a held-out validation set. See the paper for an explanation of the hyperparameters.

In the case of the non-variational NCF approaches, $\lambda_1, \lambda_2, \lambda_3$ were first fixed to 1, all other hyperparameters were tuned then fixed. We note that further fine-tuning of $\lambda_1, \lambda_2, \lambda_3$ did not appreciably change the results so we left them at 1.