Autoencoders for Collaborative Filtering

WSDM 2020 paper "RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback"

Ilya Shenbin

Samsung-PDMI AI Center

February 19, 2020

Background: Collaborative filtering

- Linear models
  - user-item collaborative filtering:
    - probabilistic matrix factorization (PMF) [Salakhutdinov and Mnih, 2008]
    - weighted matrix factorization (WMF) [Hu et al., 2008]
  - item-item collaborative filtering:
    - sparse linear methods (SLIM) [Ning and Karypis, 2011]
    - embarrassingly shallow autoencoders (EASE) [Steck, 2019]

Background: Collaborative filtering

- Linear models
  - user-item collaborative filtering:
    - probabilistic matrix factorization (PMF) [Salakhutdinov and Mnih, 2008]
    - weighted matrix factorization (WMF) [Hu et al., 2008]
  - item-item collaborative filtering:
    - sparse linear methods (SLIM) [Ning and Karypis, 2011]
    - embarrassingly shallow autoencoders (EASE) [Steck, 2019]
- Deep learning-based models
  - autoencoder-based:
    - AutoRec [Sedhain et al., 2015]
    - collaborative denoising autoencoder (CDAE) [Wu et al., 2016]
    - multinomial VAE (Mult-VAE) [Liang et al., 2018]
    - ranking-critical training (RaCT) [Lobel et al., 2019]
  - ...

Background: SLIM

- Sparse Linear Methods (SLIM) [Ning and Karypis, 2011]:

$$\arg\min_W \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1$$

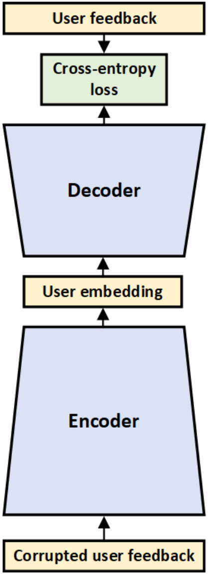
Background: SLIM

- Sparse Linear Methods (SLIM) [Ning and Karypis, 2011]:

$$\arg\min_W \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1$$

- subject to $\mathrm{diag}(W) = 0$

Background: Autoencoders for Collaborative Filtering



$$\tilde{\boldsymbol{x}}_u = \text{noise}(\boldsymbol{x}_u),$$
$$\tilde{\boldsymbol{z}}_u = \text{encoder}(\tilde{\boldsymbol{x}}_u),$$
$$\tilde{\boldsymbol{x}}_u^{\text{pred}} = \text{decoder}(\tilde{\boldsymbol{z}}_u),$$

where $\boldsymbol{x}_u$ is a user feedback vector with $\boldsymbol{x}_{ui} = 1$ iff the uth user has positively interacted with the ith item

Background: Variational Autoencoders

- Variational autoencoders (VAE) [Kingma and Welling, 2013]:

$$\log p(\boldsymbol{x}) = \log \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} =$$

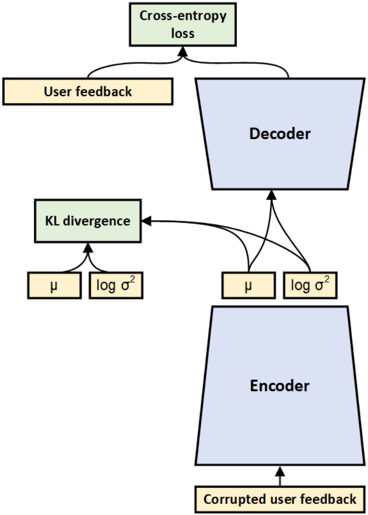
$$= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log \frac{p(\boldsymbol{z}, \boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})} + \mathrm{KL}\left(q(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x})\right) \geq$$

$$\geq \text{ELBO} = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathrm{KL}\left(q(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})\right)$$

Background: Variational Autoencoders for Collaborative Filtering

- Multinomial VAE (Mult-VAE) [Liang et al., 2018]:
  - partially regularized VAE with multinomial likelihood:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)} \log \mathrm{Mult}(\boldsymbol{x}_u|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \\ - \beta \mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u) \| p(\boldsymbol{z}_u)\right)$$



Our model

- Most works that develop further developments of VAE for collaborative filtering introduce alternative loss functions:
  - Wasserstein autoencoders (aWAE) [Zhong and Zhang, 2018]
  - ranking-critical training (RaCT) [Lobel et al., 2019]
  - negative-binomial VAE (NBVAE) [Zhao et al., 2019]
- Instead, we propose several new regularization techniques for Mult-VAE

Background: Variational Autoencoder with Arbitrary Conditioning

- Variational Autoencoder with Arbitrary Conditioning (VAEAC) [Ivanov et al., 2018]:

$$\log p_{\theta,b}(\boldsymbol{x}_b|\boldsymbol{x}_{1-b}, b) \geq \mathcal{L}_{VAEAC} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},b)} \log p_\theta(\boldsymbol{x}_b|\boldsymbol{z}, \boldsymbol{x}_{1-b}, b) - - \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}, b) \| p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_{1-b}, b)\right); \quad (1)$$

Composite prior

- Standard normal prior is important for Mult-VAE

Composite prior

- Standard normal prior is important for Mult-VAE
- Inspired by TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017] from reinforcement learning, we propose to add a regularizer that brings current variational parameters closer to variational parameters on the previous epoch

Composite prior

- Standard normal prior is important for Mult-VAE
- Inspired by TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017] from reinforcement learning, we propose to add a regularizer that brings current variational parameters closer to variational parameters on the previous epoch
- We combine them together as a conditional prior:

$$\tilde{p}(\boldsymbol{z}|\phi_{old}, \boldsymbol{x}) = \alpha\mathcal{N}(\boldsymbol{z}|0, \mathbf{I}) + (1-\alpha)q_{\phi_{old}}(\boldsymbol{z}|\boldsymbol{x})$$

Composite prior

- Standard normal prior is important for Mult-VAE
- Inspired by TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017] from reinforcement learning, we propose to add a regularizer that brings current variational parameters closer to variational parameters on the previous epoch
- We combine them together as a conditional prior:

$$\tilde{p}(\boldsymbol{z}|\phi_{old}, \boldsymbol{x}) = \alpha \mathcal{N}(\boldsymbol{z}|0, \mathbf{I}) + (1-\alpha) q_{\phi_{old}}(\boldsymbol{z}|\boldsymbol{x})$$

- This improves both stability and performance
- Serves as an auxiliary loss function

Background: Trust Region Policy Optimization

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$$

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}} \mathrm{KL}\left(\pi_{\theta_{\text{old}}}(\cdot|s) \| \pi_\theta(\cdot|s)\right) \leq \delta.$$

Rescaling KL-divergence

- $\boldsymbol{x}_{ui} = 0$ means that the uth user either does not like the ith item or has not seen the ith item at all

Rescaling KL-divergence

- $\boldsymbol{x}_{ui} = 0$ means that the uth user either does not like the ith item or has not seen the ith item at all
- We denote by $\mathbf{X}_u^o$ the set of items that the uth user likes according to the training set and by $\mathbf{X}_u^f$ the set of items that the uth user actually likes

Rescaling KL-divergence

- $\boldsymbol{x}_{ui} = 0$ means that the uth user either does not like the ith item or has not seen the ith item at all
- We denote by $\mathbf{X}_u^o$ the set of items that the uth user likes according to the training set and by $\mathbf{X}_u^f$ the set of items that the uth user actually likes
- Then we can derive that $\mathcal{L}$ can be approximated with

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)} \log \mathrm{Mult}(\boldsymbol{x}_u|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{|\mathbf{X}_u^o|}{|\mathbf{X}_u^f|} \mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)\|p(\boldsymbol{z}_u)\right),$$

where $|\mathbf{X}_u^f|$ is unknown, so we let it be equal to some constant

Rescaling KL-divergence

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)} \log \mathrm{Mult}(\boldsymbol{x}_u^f|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)\middle\|p(\boldsymbol{z}_u)\right) = \\
&\sum_{a\in\mathbf{X}_u^f} \mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)} \log \mathrm{Cat}(\mathbf{1}_a|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)\middle\|p(\boldsymbol{z}_u)\right) + C_u = \\
&\sum_{a\in\mathbf{X}_u^f} \left[\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)} \log \mathrm{Cat}(\mathbf{1}_a|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{1}{|\mathbf{X}_u^f|}\mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)\middle\|p(\boldsymbol{z}_u)\right)\right] + C_u \approx \\
&\frac{|\mathbf{X}_u^f|}{|\mathbf{X}_u^o|} \sum_{a\in\mathbf{X}_u^o} \left[\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)} \log \mathrm{Cat}(\mathbf{1}_a|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{1}{|\mathbf{X}_u^f|}\mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u^f)\middle\|p(\boldsymbol{z}_u)\right)\right] + C_u' \approx \\
&\frac{|\mathbf{X}_u^f|}{|\mathbf{X}_u^o|} \sum_{a\in\mathbf{X}_u^o} \left[\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)} \log \mathrm{Cat}(\mathbf{1}_a|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{1}{|\mathbf{X}_u^f|}\mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)\|p(\boldsymbol{z}_u)\right)\right] + C_u' = \\
&\frac{|\mathbf{X}_u^f|}{|\mathbf{X}_u^o|} \left[\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)} \sum_{a\in\mathbf{X}_u^o} \log \mathrm{Cat}(\mathbf{1}_a|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{|\mathbf{X}_u^o|}{|\mathbf{X}_u^f|}\mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)\|p(\boldsymbol{z}_u)\right)\right] + C_u' = \\
&\frac{|\mathbf{X}_u^f|}{|\mathbf{X}_u^o|} \left[\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)} \log \mathrm{Mult}(\boldsymbol{x}_u|\boldsymbol{\pi}(\boldsymbol{z}_u)) - \frac{|\mathbf{X}_u^o|}{|\mathbf{X}_u^f|}\mathrm{KL}\left(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)\|p(\boldsymbol{z}_u)\right)\right] + C_u''
\end{aligned}
\tag{2}
$$

Complementary improvements

- Updated architecture
  - Deep encoder
  - Linear decoder (item embeddings matrix + bias vector)

Complementary improvements

- Updated architecture
  - Deep encoder
  - Linear decoder (item embeddings matrix + bias vector)
- Alternating Training
  - Encoder and decoder are trained alternately
  - More iterations are required to train the encoder

Complementary improvements

- Updated architecture
  - Deep encoder
  - Linear decoder (item embeddings matrix + bias vector)
- Alternating Training
  - Encoder and decoder are trained alternately
  - More iterations are required to train the encoder
- Regularization by denoising
  - It appears that the decoder is overregularized
  - Therefore, we do not use denoising during decoder training

Results

	ML-20M	Netflix	MSD
WMF [Hu et al., 2008]	0.386	0.351	0.257
Mult-VAE [Liang et al., 2018]	0.426	0.386	0.316
RaCT [Lobel et al., 2019]	<u>0.434</u>	0.392	0.319
EASE [Steck, 2019]	0.420	<u>0.393</u>	**0.389**
RecVAE (ours)	**0.442**	**0.394**	<u>0.326</u>

- NDCG@100 scores, best results highlighted in bold, second best ones underlined

Conclusion

- We have proposed several improvements for Mult-VAE
- Combined together, they significantly improve the performance, making RecVAE the new state of the art in deep learning-based autoencoders for collaborative filtering

References

- Mnih, Andriy, and Russ R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural information processing systems. 2008.
- Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." 2008 Eighth IEEE International Conference on Data Mining. Ieee, 2008.
- Ning, Xia, and George Karypis. "Slim: Sparse linear methods for top-n recommender systems." 2011 IEEE 11th International Conference on Data Mining. IEEE, 2011.
- Steck, Harald. "Embarrassingly shallow autoencoders for sparse data." The World Wide Web Conference. 2019.
- Sedhain, Suvash, et al. "Autorec: Autoencoders meet collaborative filtering." Proceedings of the 24th international conference on World Wide Web. 2015.
- Wu, Yao, et al. "Collaborative denoising auto-encoders for top-n recommender systems." Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. 2016.
- Liang, Dawen, et al. "Variational autoencoders for collaborative filtering." Proceedings of the 2018 World Wide Web Conference. 2018.
- Lobel, Sam, et al. "Towards Amortized Ranking-Critical Training for Collaborative Filtering." arXiv preprint arXiv:1906.04281 (2019).
- Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).
- Zhong, Jingbin, and Xiaofeng Zhang. "Wasserstein autoencoders for collaborative filtering." arXiv preprint arXiv:1809.05662 (2018).
- Zhao, He, et al. "Variational Autoencoders for Sparse and Overdispersed Discrete Data." arXiv preprint arXiv:1905.00616 (2019).
- Schulman, John, et al. "Trust region policy optimization." International conference on machine learning. 2015.
- Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).
- Ivanov, Oleg, Michael Figurnov, and Dmitry Vetrov. "Variational autoencoder with arbitrary conditioning." arXiv preprint arXiv:1806.02382 (2018).