Variational AutoEncoder for Collaborative Filtering

Jixiong Liu (revised by Weike Pan)

College of Computer Science and Software Engineering
Shenzhen University

Reference: Variational AutoEncoder for Collaborative Filtering (WWW 2018) by Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman and Tony Jebara.

Problem Definition

One-Class Collaborative Filtering (OCCF)

- Input: One-class feedback matrix with positive feedback and missing entries.
- Goal: generate a personalized ranked list of items for each user u from the set of items that user u has not seen before, i.e., $\mathcal{I} \backslash \mathcal{I}_u$, where $\mathcal{I}_u$ denotes the set of interacted items of user u.

Notations (1/2)

Table: Some notations and explanations.

$\mathcal{U}$	the whole set of users, $u \in \mathcal{U}$, $\lvert\mathcal{U}\rvert = n$
$\mathcal{I}$	the whole set of items, $i \in \mathcal{I}$, $\lvert\mathcal{I}\rvert = m$
$\mathcal{I}_u$	a set of items interacted by user u
$\boldsymbol{x}_u \in \{0,1\}^{1\times m}$	user-specific vector obtained by multi-hot transformation of $\mathcal{I}_u$
$\boldsymbol{z}_u \in \mathbb{R}^{1\times k}$	latent representation of user u, where k is the dimension
$\boldsymbol{\mu}_u \in \mathbb{R}^{1\times k}$	mean value vector of $\boldsymbol{z}_u$
$\boldsymbol{\sigma}_u \in \mathbb{R}^{1\times k}$	standard deviation vector of $\boldsymbol{z}_u$
$\hat{\boldsymbol{x}}_u \in \mathbb{R}^{1\times m}$	predicted preference vector of user u to all items
$\hat{x}_{ui} \in \mathbb{R}$	predicted preference of user u to item i

Notations (2/2)

Table: Some notations and explanations.

$W^{\boldsymbol{\mu}}_{i\cdot}, W^{\boldsymbol{\sigma}}_{i\cdot} \in \mathbb{R}^{1\times k}$	item-specific latent feature vector of item i to get $\boldsymbol{\mu}$. and $\boldsymbol{\sigma}$.
$V_{i\cdot} \in \mathbb{R}^{1\times k}$	item-specific latent feature vector of item i
ϕ	parameters of the inference model, i.e., encoder
θ	parameters of the generation model, i.e., decoder
$q_\phi(\boldsymbol{z}_u \| \boldsymbol{x}_u)$	distribution (function) learned by ϕ
$p_\theta(\boldsymbol{x}_u \| \boldsymbol{z}_u)$	distribution (function) learned by θ
$p(\boldsymbol{z}_u)$	pre-defined prior distribution of $\boldsymbol{z}_u$, e.g., standard Gaussian distribution in Mult-VAE

Framework

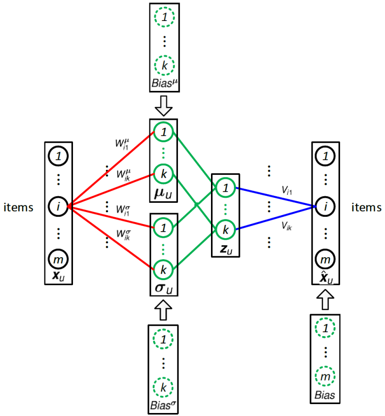

Figure: Structure of Mult-VAE

Assumptions

For each user u,

- $\boldsymbol{z}_u$ should obey the standard Gaussian distribution;
- $\boldsymbol{x}_u$ should obey the multinomial distribution.

Evidence Lower Bound (ELBO)[1] via Jensen's Inequality

$$
\begin{aligned}
\log p(x) &= \log \int_z p(x,z)dz = \log \int_z p(x,z)\frac{q(z|x)}{q(z|x)}dz \\
&= \log\left(\mathbb{E}_{q(z|x)}\frac{p(x,z)}{q(z|x)}\right) \geq \mathbb{E}_{q(z|x)}\left(\log\frac{p(x,z)}{q(z|x)}\right) \\
&= \mathbb{E}_{q(z|x)}\left(\log\frac{p(x|z)p(z)}{q(z|x)}\right) \\
&= \mathbb{E}_{q(z|x)}(\log p(x|z)) + \mathbb{E}_{q(z|x)}\left(\log\frac{p(z)}{q(z|x)}\right) \\
&= \mathbb{E}_{q(z|x)}(\log p(x|z)) + \int_z q(z|x)\log\frac{p(z)}{q(z|x)}dz \\
&= \mathbb{E}_{q(z|x)}(\log p(x|z)) - \mathrm{KL}(q(z|x)||p(z))
\end{aligned}
$$

[1] https://fangdahan.medium.com/derivation-of-elbo-in-vae-25ad7991fdf7

Objective Function

The objective function to be maximized in Mult-VAE,

$$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)}\{\log p_\theta(\hat{\boldsymbol{x}}_u|\boldsymbol{z}_u) - \beta \mathrm{KL}(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)||p(\boldsymbol{z}_u))\}, \tag{1}$$

where $q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)$ is the function that the encoder learns $\boldsymbol{z}_u$ with the input $\boldsymbol{x}_u$, and $p_\theta(\hat{\boldsymbol{x}}_u|\boldsymbol{z}_u)$ is the function that the decoder reconstructs $\boldsymbol{x}_u$ with the input $\boldsymbol{z}_u$. Meanwhile, the first term above is the multinomial likelihood, while the second term is to constrain the learned posterior distribution $q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)$ to obey the assumed prior distribution $p(\boldsymbol{z}_u)$, i.e., the standard multivariate Gaussian distribution $p(\boldsymbol{z}_u) = N(\boldsymbol{z}_u; \mathbf{0}, \mathbf{I})$.

Derivation (1/4)

For the first term of the objective function,

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)}[\log p_\theta(\hat{\boldsymbol{x}}_u|\boldsymbol{z}_u)] = \frac{1}{n}\sum_{u=1}^{n}\sum_{i\in\mathcal{I}_u}\log\frac{\exp(\hat{x}_{ui})}{\sum_{j=1}^{m}\exp(\hat{x}_{uj})}, \tag{2}$$

where $\hat{x}_{ui} = g_o(\boldsymbol{z}_u V_{i\cdot}^T + Bias_i)$ with $g_o(\cdot)$ as an activation function for the output layer.

Derivation (2/4)

For the second term of the objective function,

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)}\mathrm{KL}(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)||p(\boldsymbol{z}_u)) = \frac{1}{n}\sum_{u=1}^{n}\sum_{d=1}^{k}\mathrm{KL}(N(z_{ud};\mu_{ud},\sigma_{ud})||N(z_{ud};0,1))$$
$$= \frac{1}{n}\sum_{u=1}^{n}\sum_{d=1}^{k}\frac{1}{2}(-\log\sigma_{ud}^2+\mu_{ud}^2+\sigma_{ud}^2-1), \quad (3)$$

where $\mu_{ud} = g_h(\sum_{i\in\mathcal{I}_u} W_{id}^{\mu} + Bias_d^{\mu}) \in \mathbb{R}$, $\sigma_{ud} = g_h(\sum_{i\in\mathcal{I}_u} W_{id}^{\sigma} + Bias_d^{\sigma}) \in \mathbb{R}$ with $g_h(\cdot)$ as an activation function for the hidden layer, and $N(z_{ud};\mu_{ud},\sigma_{ud})$ is a normal distribution[2].

[2] https://en.wikipedia.org/wiki/Normal_distribution. Notice that the KL divergence between two Gaussian has an analytic solution.

Derivation (3/4)

$\mathrm{KL}(N(z_{ud}; \mu_{ud}, \sigma_{ud}) || N(z_{ud}; 0, 1))$

$$= \int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \left(\log \frac{\frac{\exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}}}{\sqrt{2\pi\sigma_{ud}^2}}}{\frac{\exp^{\frac{-z_{ud}^2}{2}}}{\sqrt{2\pi}}} \right) dz_{ud}$$

$$= \int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \log \left(\frac{1}{\sqrt{\sigma_{ud}^2}} \exp^{\frac{1}{2}\left(z_{ud}^2 - \frac{(z_{ud}-\mu_{ud})^2}{\sigma_{ud}^2}\right)} \right) dz_{ud}$$

$$= \frac{1}{2} \int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \left(-\log \sigma_{ud}^2 + z_{ud}^2 - \frac{(z_{ud}-\mu_{ud})^2}{\sigma_{ud}^2} \right) dz_{ud}$$

Derivation (4/4)

$$\int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \left(-\log \sigma_{ud}^2\right) dz_{ud} = -\log \sigma_{ud}^2$$

$$\int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \left(z_{ud}^2\right) dz_{ud} = \mu_{ud}^2 + \sigma_{ud}^2$$

$$\int \frac{1}{\sqrt{2\pi\sigma_{ud}^2}} \exp^{\frac{-(z_{ud}-\mu_{ud})^2}{2\sigma_{ud}^2}} \left(-\frac{(z_{ud}-\mu_{ud})^2}{\sigma_{ud}^2}\right) dz_{ud} = -1$$

We thus have

$$\mathrm{KL}(N(z_{ud}; \mu_{ud}, \sigma_{ud}) || N(z_{ud}; 0, 1)) = \frac{1}{2}\left(-\log \sigma_{ud}^2 + \mu_{ud}^2 + \sigma_{ud}^2 - 1\right).$$

Some Details

- **Reparametrization trick**: To support the gradient back-propagation to ϕ, the model samples $\epsilon \in N(0, I_k)$ and reparametrizes $\boldsymbol{z}_u = \boldsymbol{\mu}_u + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_u$ in the training phase. And in the test phase, the model adopts the mean value vector as the latent representation for prediction, i.e., $\boldsymbol{z}_u = \boldsymbol{\mu}_u$.
- In fact, for each user u, Mult-VAE learns $\boldsymbol{\mu}_u$ and $\log \boldsymbol{\sigma}_u^2$ by the encoder, thus $\boldsymbol{\sigma}_u = \exp^{\frac{1}{2} \log \boldsymbol{\sigma}_u^2}$.

Dataset

Table: Statistics of the MovieLens 1M (ML1M for short). Notice that $|\mathcal{R}^{tr}|$, $|\mathcal{R}^{vad}|$ and $|\mathcal{R}^{te}|$ represent the numbers of records of the training data, validation data and test data, respectively.

Dataset	n	m	$\|\mathcal{R}^{tr}\|$	$\|\mathcal{R}^{vad}\|$	$\|\mathcal{R}^{te}\|$
ML1M	6,040	3,648	604,897	197,604	197,616

- We treat all ratings ($\geq$ 1) as positive implicit feedback;
- We randomly split all records into three parts, i.e., 60% for training, 20% for validation and the remaining 20% for test;
- We remove the records of the validation data and test data where the items do not appear in the training data.

Baseline

- Autoencoders Meet Collaborative Filtering (AutoRec) [Sedhain et al., 2015]

Experiment Settings (1/3)

- For strong generalization, there is no overlap between users in the training data, users in the validation data and users in the test data, while the setup for weak generalization is the opposite. We check the performance under weak generalization.
- In our experiments, we use the trained model obtained in the validation step with the best NDCG@5 to check the performance on the test data.

Experiment Settings (2/3)

- For Mult-VAE, we basically follow the settings in the original paper [Liang et al., 2018]
  - We set the value of β as 0.2, which is found via linearly increasing its value from 0 to 1 with 200000 gradient updates
  - We set the dropout ratio at the input layer as 0.5
  - We use one hidden layer with 200 nodes, and identity activation functions for both the hidden layer and the output layer
  - We set the batch size (i.e., number of users) as 100
  - We adopt an early-stop strategy with a threshold of 50
  - We adopt the Adam optimizer with learning rate 1e-3

Experiment Settings (3/3)

- For AutoRec, we adopt a structure with one hidden layer and 200 nodes, the activation function for the hidden layer (i.e., $g_h(\cdot)$) and the output layer (i.e., $g_o(\cdot)$) is the sigmoid function, and batch size is 100, a threshold of early-stop is 50, and we use Adam optimizer with learning rate 1e-3. We choose the regularization coefficient λ from $\{$1e-4, 1e-3, 1e-2, 1e-1$\}$.

Evaluation Metrics

- Precision@5
- Recall@5
- NDCG@5
- MRR@5

Results

Method	Precision@5	Recall@5	NDCG@5	MRR@5
AutoRec	0.2707	0.0724	0.2807	0.4715
Mult-VAE	**0.2718**	**0.0771**	**0.2819**	**0.4747**

Conclusion

- Mult-VAE is empirically shown to be both effective and efficient.

Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018).
Variational autoencoders for collaborative filtering.
In Champin, P., Gandon, F. L., Lalmas, M., and Ipeirotis, P. G., editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 689–698. ACM.

Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015).
Autorec: Autoencoders meet collaborative filtering.
In Gangemi, A., Leonardi, S., and Panconesi, A., editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 111–112. ACM.