VARIATIONAL AUTOENCODERS FOR COLLABORATIVE FILTERING

KAAN OKTAY, ÖZGE BOZAL

INTRODUCTION

In this project variational autoencoders are extended to collaborative filtering with implicit feedback which is a method of making automatic predictions (filtering) about the interests of a user by collecting indirect preferences from many users (collaborating).

MODEL

- $u \in 1, \ldots, U$ and $i \in 1, \ldots, I$
- $\boldsymbol{X} \in \mathbb{N}^{U \times I}$ and $\boldsymbol{x_u} = [x_{u1}, \ldots, x_{uI}]^T \in \mathbb{N}^I$
- $\boldsymbol{z_u}$: K-dimensional hidden factor for user u
- $f_{\boldsymbol{\theta}}(\cdot)$: non-linear function $\in \mathbb{R}^I$
- $\pi(\boldsymbol{z_u})$: the probability function from which $\boldsymbol{x_u}$ vector is assumed to be drawn
- $N_u = \sum_i x_{ui}$: Total number of clicks of the given user u

$$\boldsymbol{z_u} \sim \mathcal{N}(0, I_k)$$

$$\pi(\boldsymbol{z_u}) \propto \exp(\, f_{\boldsymbol{\theta}}(\boldsymbol{z_u})\,)$$

$$\boldsymbol{x_u} \sim \text{Mult}(N_u, \pi(\boldsymbol{z_u}))$$

- Multinomial log-likelihood function:

$$\log\, p_\theta(\boldsymbol{x_u}|\boldsymbol{z_u}) = \sum_i x_{ui}\, \log\, \pi_i(\boldsymbol{z_u})$$

- For comparison, logit log-likelihood with the logit function $\sigma(x)$:

$$\log\, p_\theta(\boldsymbol{x_u}|\boldsymbol{z_u}) = \sum_i [x_{ui} \cdot \log \sigma(f_{ui}) + (1 - x_{ui}) \cdot (1 - \log \sigma(f_{ui})]$$

METHOD

- **Variational inference:** To learn the generative model we need to estimate $\boldsymbol{\theta}$ parameters of $f_\theta(\cdot)$ to calculate the intractable posterior $p(\boldsymbol{z_u}|\boldsymbol{x_u})$ for each data point. Variational inference approximates the true posterior to an instrumental posterior $q(\boldsymbol{z_u})$ with Kullback-Leibler divergence $KL(q(\boldsymbol{z_u})||p(\boldsymbol{z_u}|\boldsymbol{x_u}))$ where,

$$q(\boldsymbol{z_u}) \sim \mathcal{N}(\boldsymbol{\mu_u}, \text{diag}(\boldsymbol{\sigma_u^2}))$$

- **Amortized inference and variational autoencoders:** The number of variational parameters $(\boldsymbol{\mu_u}, \boldsymbol{\sigma_u^2})$ grows with the number of users and items in the dataset. To cope with this issue, variational autoencoders are used which are in fact data-dependent functions. This is commonly called the inference model:

$$g_\phi(\boldsymbol{x_u}) = [\boldsymbol{\mu_\phi}(\boldsymbol{x_u}), \boldsymbol{\sigma_\phi}(\boldsymbol{x_u})] \in \mathbb{R}^{2K}$$

Thus the variational distribution becomes also data-dependent:

$$q_\phi(\boldsymbol{z_u}|\boldsymbol{x_u}) \sim \mathcal{N}(\boldsymbol{\mu_\phi}(\boldsymbol{x_u}), \text{diag}\{\boldsymbol{\sigma_\phi^2}(\boldsymbol{x_u})\})$$

METHOD(CONT'D)

- **Learning VAEs:** The evidence $p(\boldsymbol{x_u})$ is a constant term, and KL divergence is a non negative term, we can rearrange the equation as below:

$$\log\, p_{\boldsymbol{\theta}}(\boldsymbol{x_u}) \geq \mathbb{E}_q[\log\, p_{\boldsymbol{\theta}}(\boldsymbol{x_u}|\boldsymbol{z_u})] - KL(q_{\boldsymbol{\phi}}(\boldsymbol{z_u}|\boldsymbol{x_u})||p(\boldsymbol{z_u}))$$

Right-hand side of the inequality is $\mathcal{L}(\boldsymbol{x_u}; \boldsymbol{\theta}, \boldsymbol{\phi})$ which is called evidence lower bound (ELBO).

- **Reparameterization trick:** ELBO is a function of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. We cannot take gradient w.r.t. $\boldsymbol{\phi}$ when we sample $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}$. Instead; we sampled $\boldsymbol{z_u}$ as follows:

$$\boldsymbol{\epsilon} \sim N(0, \boldsymbol{I_k})$$

$$\boldsymbol{z_u} = \boldsymbol{\mu_\phi}(\boldsymbol{x}) + \epsilon \odot \boldsymbol{\sigma_\phi}(\boldsymbol{x})$$

By doing so, the stochasticity in the sampling process is isolated and the gradient w.r.t. $\boldsymbol{\phi}$ can be backpropagated through the sampled $\boldsymbol{z_u}$.

Algorithm 1: VAE-SGD

```
Input: Click matrix X ∈ N^{I×U}
Initialize φ and θ
while not converged do
    Sample a batch of users U
    for u ∈ U do
        Sample ε ~ N(0, I_k)
        Compute z_u
        Compute noisy gradients ∇_φ L
        and ∇_θ L
    end
    Find the average of noisy gradients
    Update φ and θ
end
```

EVALUATION METRIC

Normalized Discounted Cumulative Gain:

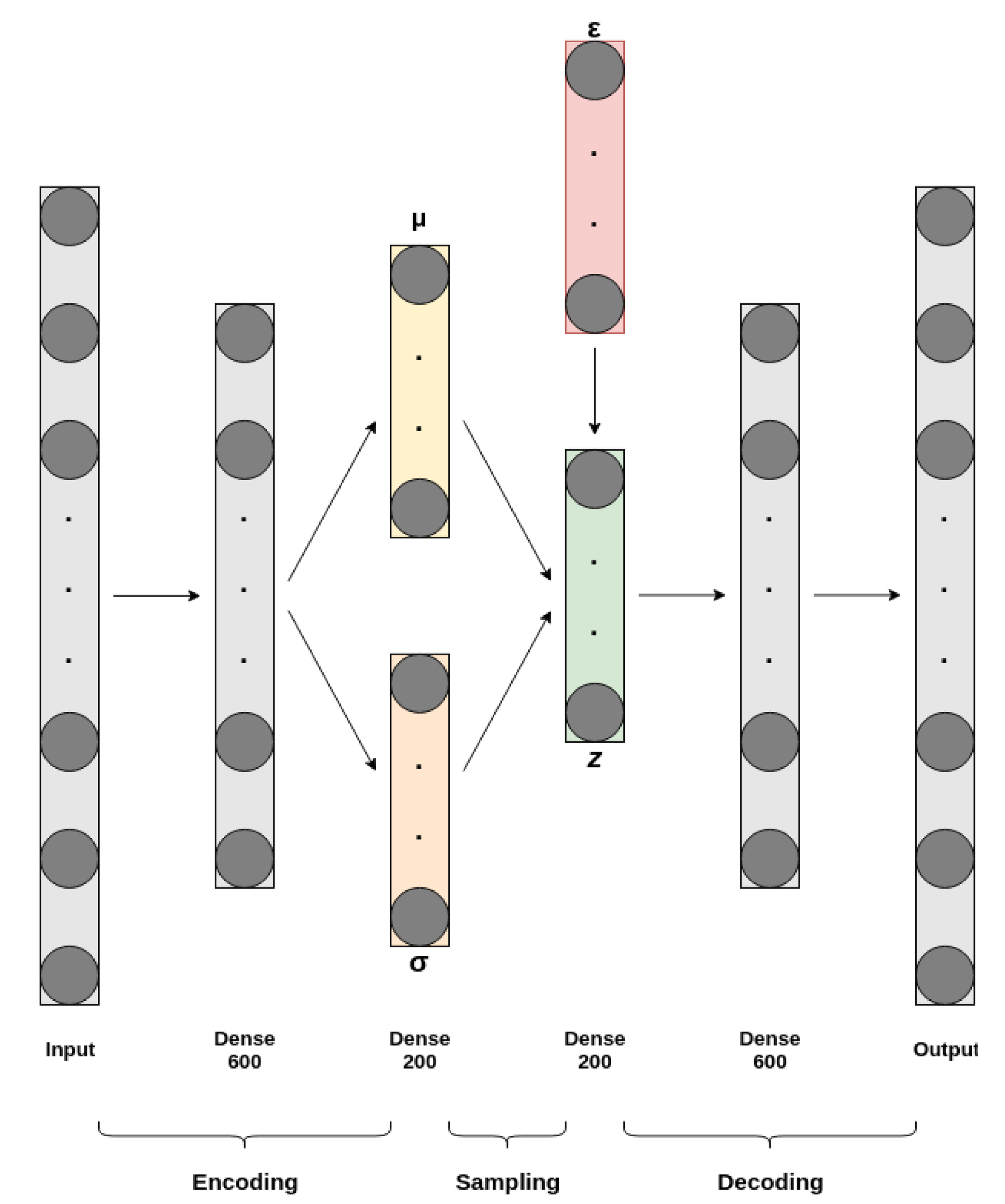
- $w(r)$: Item at rank r
- $I[.]$: Indicator function
- I_u: Set of held-out items user u clicked on

Discounted cumulative gain:

$$DCG@R(u, w) = \sum_{r=1}^{R} \frac{2^{I[w(r) \in I_u]} - 1}{log(r + 1)}$$

$DCG@R$ is linearly normalized to [0, 1] by dividing it by the best possible $DCG@R$, where all the held-out items are ranked at the top, and we end up with $NDCG@R$.

ARCHITECTURE



DATA-RESULTS

- Random 60k users from MovieLens 20M dataset. NDCG is 0.36 for logistic and 0.38 for multinomial likelihood on the test set.

Table 1: Attributes of users and training

# of Training Users	50,000
# of Val and Test Users	5,000
# of Movie Items	17,136
# of Interactions	3.7M
% of Interactions	0.43%
Batch Size	256
# of Epochs	30

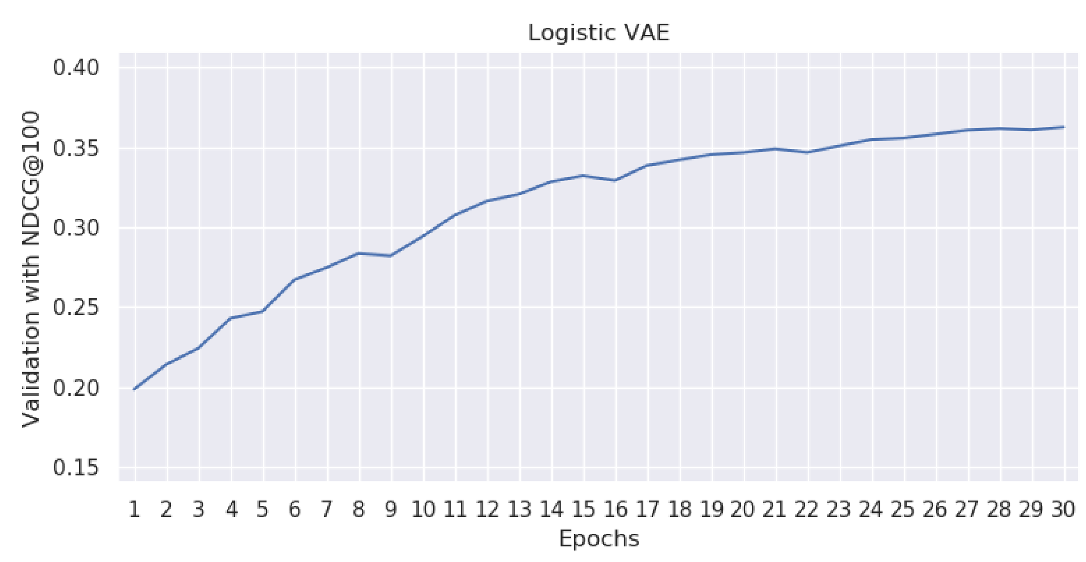


Figure 1: Training results with logistic likelihood

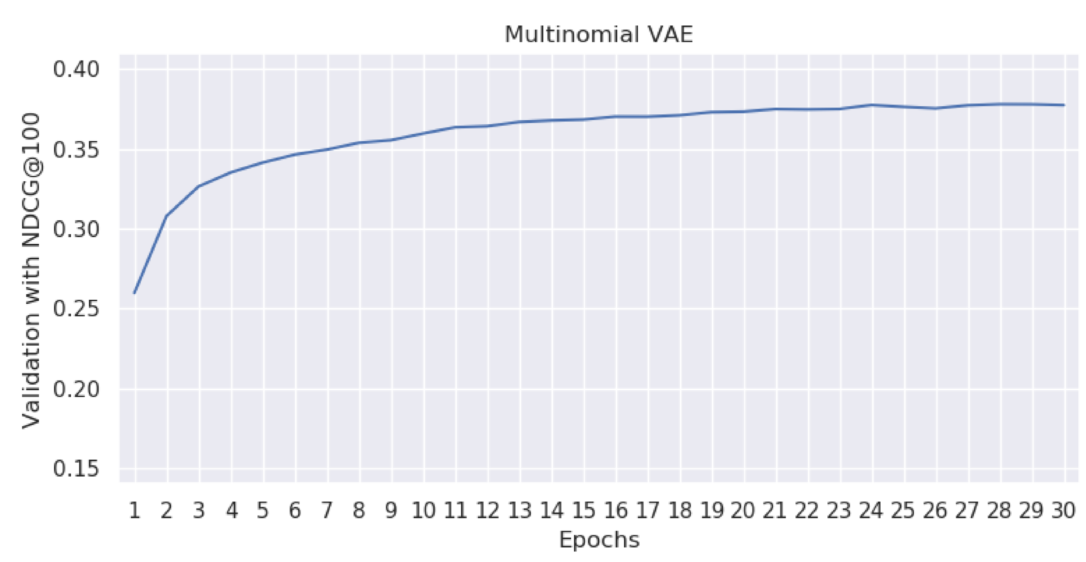


Figure 2: Training results with multinomial likelihood

REFERENCES

[1] Liang, Dawen, Krishnan, Rahul G, Hoffman, Matthew D, and Jebara, Tony. "Variational autoencoders for collaborative filtering." arXiv preprint arXiv:1802.05814, 2018.

[2] Variational Autoencoder: Intuition and Implementation, https://wiseodd.github.io/techblog/2016/12/10/variational-autoencoder/