Variational Autoencoders for Top-K Recommendation with Implicit Feedback

Bahare Askari
Ontario Tech University
Canada
bahare.askarifiroozjayi@ontariotechu.ca

Jaroslaw Szlichta
Ontario Tech University
Canada
jarek@ontariotechu.ca

Amirali Salehi-Abari
Ontario Tech University
Canada
abari@ontariotechu.ca

ABSTRACT

Variational Autoencoders (VAEs) have shown to be effective for recommender systems with implicit feedback (e.g., browsing history, purchasing patterns, etc.). However, a little attention is given to ensembles of VAEs, that can learn user and item representations jointly. We introduce Joint Variational Autoencoder (JoVA), an ensemble of two VAEs, which jointly learns both user and item representations to predict user preferences. This design allows JoVA to capture user-user and item-item correlations simultaneously. We also introduce JoVA-Hinge, a JoVA's extension with a hinge-based pairwise loss function, to further specialize it in recommendation with implicit feedback. Our extensive experiments on four real-world datasets demonstrate that JoVA-Hinge outperforms a broad set of state-of-the-art methods under a variety of commonly-used metrics. Our empirical results also illustrate the effectiveness of JoVA-Hinge for handling users with limited training data.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Learning to rank**.

KEYWORDS

Recommender Systems, Deep Learning, Variational Autoencoders

ACM Reference Format:
Bahare Askari, Jaroslaw Szlichta, and Amirali Salehi-Abari. 2021. Variational Autoencoders for Top-K Recommendation with Implicit Feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3462986

1 INTRODUCTION

The information overload and abundance of choices on the Web have made recommendation systems indispensable in facilitating user decision-making. Recommender systems provide personalized user experience by filtering relevant items (e.g., books, music, or movies) or information (e.g., news). Many efforts have been devoted to developing effective recommender systems [1, 19].

Collaborative filtering (CF)—a well-recognized approach in recommender systems—is based on the idea that users with similar revealed preferences are likely to have similar preferences in the future [19]. User preferences in CF techniques are in the form of either *explicit feedback* (e.g., ratings, reviews, etc.) or *implicit feedback* (e.g., browsing history, purchasing history, search patterns, etc.). While explicit feedback is more informative than its implicit alternative, it imposes more cognitive burden on users through elicitation, is subject to noisy self-reporting [2], and suffers from interpersonal comparison or *calibration* issues [3]. In contrast, implicit feedback naturally originates from user behavior when an interaction with an item is a signal of interest in the item.

The implicit feedback has made collaborative filtering more intriguing at the cost of some practical challenges. The implicit feedback lacks negative examples, as the absence of a user-item interaction is not necessarily indicative of user disinterest (e.g., the user is unaware of the item). Also, the user-item interaction data for implicit feedback is large, yet *sparse*. It is even more sparse than explicit feedback data, since the unobserved user-item interactions are a mixture of both missing values and real negative feedback. Many attempts have been made to address these challenges by deep learning [24]. Multilayer perceptron networks were arguably the first class of neural networks successfully applied for collaborative filtering [6, 9]. Recent interest is in deploying the variants of autoencoders, such as classical [25], denoising [21], and variational [14, 15]. However, these solutions either do not capture uncertainty of the latent representations [21, 25], or solely focus on latent representation of users [14, 15].

We present the *joint variational autoencoder (JoVA)* model, an ensemble of two variational autoencoders (VAEs), that jointly learns both user and item representations under uncertainty, and then collectively predicts user preferences. This design enables JoVA to encapsulate user-user and item-item correlations simultaneously. We also introduce *JoVA-Hinge*, a variant of JoVA, which extends the JoVA's objective function with a pairwise ranking loss, to additionally specialize it for top-k recommendation with implicit feedback. Through extensive experiments over four real-world datasets, we show the accuracy improvements of our proposed solutions over a variety of state-of-the-art methods. Our JoVA-Hinge significantly outperforms other methods in the sparse datasets (up to 34% accuracy improvement). Our extensive ablation study on JoVA-Hinge confirms that its success originates from all of its integral components (i.e., ensemble of VAEs and hinge loss).

2 RECOMMENDATION AND IMPLICIT DATA

We assume that a set of n users U can interact with the set of m items I (e.g., users click ads, purchase products, watch movies, or

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462986

listen to musics). We consider user-item interactions are binary (e.g., a user watched a specific movie or not), and represent them with the user *implicit feedback matrix* $\mathbf{R} \in \{0,1\}^{m \times n}$, where $\mathbf{R}_{ui} = 1$, if the interaction of a user u with an item i is observed. As each column (or row) of the matrix corresponds to a specific item (or user), $\mathbf{R}_u$ and $\mathbf{R}_i^T$ denote the user u's and item i's interaction vector, respectively. We consider that user u has interacted with items $I_u^+ = \{i \in I \mid \mathbf{R}_{ui} = 1\}$ and has not interacted with $I_u^- = I \setminus I_u^+$.

Our goal in top-k recommendation is to suggest k most preferred (or likely) items to the user u from I_u^-. We predict the likelihood of interaction between the user u and I_u^- , and then select a rank-list of k items with the highest prediction score to recommend to the user u. Our learning task is to find a *scoring (or likelihood) function* f that predicts an *interaction score* $\hat{r}_{ui}$ for each user u and an unobserved item $i \in I_u^-$. If $\hat{r}_{ui} \in [0,1]$, it can be interpreted as the likelihood of user u's interaction with item i. The function f is formulated as $\hat{r}_{ui} = f(u, i|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the model parameters.

Most of model-based CF methods [19] differentiate from each other on the scoring function f formulation or the objective functions used for parameter learning. Some notable examples for modeling the function f are deep networks [24] and matrix factorization [13]. The objective functions fall into two categories. *Pointwise loss* [9, 11], by assuming an unobserved user-item interaction as a negative example, minimizes the error between predicted score $\hat{r}_{ui}$ and its actual value r_{ui}. *Pairwise loss* [8, 17] directly optimizes the ranking of the user-item interaction while assuming that users prefer observed items to unobserved items.

Related Work. Many CF methods have been developed for recommendation with implicit feedback [10, 11]. Deep learning has been promising [24] by capturing more complex user-item interactions (e.g., [6, 8, 9]). Of the most relevant to our work are recommender systems built based on autoencoders. *Collaborative deep ranking (CDR)* [23] jointly implements representation learning and collaborative ranking by employing stacked denoising autoencoders. *Joint collaborative autoencoder (JCA)* [25] deploys two separate classical autoencoders jointly optimized only by a single hinge loss function. Mult-VAE [15] and its variant RecVAE [18] are collaborative filtering models based on just one variational autoencoder. Our proposed work differentiate from both JCA and Mult-VAE with regards to both architecture and loss function. While JCA optimizes two classical autoencoders, it does not capture the uncertainty of latent representations. Mult-VAE models this uncertainty with just one variational autoencoder. Putting their strengths together, we optimize two separate variational autoencoders by our proposed loss function, which well tunes them for dealing with implicit feedback.

3 JOINT VARIATIONAL AUTONECODER

We describe variational autonecoder and detail how *Joint Variational Autoencoder (JoVA)* extends its architecture and loss function.

Variational Autoencoder. Our model uses the variational autoencoder (VAE) [7] as a building block. VAE, similar to classical autoencoders, consists of encoder and decoder. The encoder first encodes the inputs to latent representations, and then the decoder reconstructs the original inputs from latent representations. The VAE differentiates from classical autoencoders by encoding an input as a distribution over latent representations (rather than a single point).

The encoder network of VAE encodes the input $\mathbf{x}$ to a d-dimensional latent representation $\mathbf{z}$, with a prior distribution $p(\mathbf{z})$. One can view the encoder as the posterior distribution $p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ parametrized by $\boldsymbol{\phi}$. As this posterior distribution is intractable, it is commonly approximated by a variational distribution [4]:

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_{\boldsymbol{\phi}}(\mathbf{x}), \sigma^2_{\boldsymbol{\phi}}(\mathbf{x})\mathbf{I}), \tag{1}$$

where two multivariate functions $\mu_{\boldsymbol{\phi}}(\mathbf{x})$ and $\sigma_{\boldsymbol{\phi}}(\mathbf{x})$ map the input $\mathbf{x}$ to the mean and standard deviation vectors. In VAE, $\mu_{\boldsymbol{\phi}}(\mathbf{x})$ and $\sigma_{\boldsymbol{\phi}}(\mathbf{x})$ are formulated by the *inference network* $f_{\boldsymbol{\phi}}(\mathbf{x}) = [\mu_{\boldsymbol{\phi}}(\mathbf{x}), \sigma_{\boldsymbol{\phi}}(\mathbf{x})]$.

The decoder network $p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z})$, also known as a *generative network*, takes $\mathbf{z}$ and outputs the probability distribution over (reconstructed) input data $\mathbf{x}$. Putting together the encoder and decoder networks, one can lower bound the log-likelihood of the input $\mathbf{x}$ by

$$\log p(\mathbf{x}) \geq E_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z})\right] - KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \mid\mid p(\mathbf{z})), \tag{2}$$

where KL is Kullback-Leibler distance measuring the difference between the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the prior distribution $p(\mathbf{z})$. This lower bound, known as *evidence lower bound (ELBO)*, is maximized for learning the parameters of encoder and decoder, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, respectively. Equivalently, for learning VAE parameters, one can minimize the negation of the ELBO as a loss function (see Eq. 3) by stochastic gradient decent with the reparameterization trick [7].

$$L_{\text{VAE}}(\mathbf{x}|\boldsymbol{\theta}, \alpha) = -E_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z})] + \alpha KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \mid\mid p(\mathbf{z})), \tag{3}$$

where $\boldsymbol{\theta} = [\boldsymbol{\psi}, \boldsymbol{\phi}]$. This loss function can be viewed as a linear combination of *reconstruction loss* and KL divergence, which serves as a regularization term. Recent research [15, 18] has introduced the regularization hyperparameter α for controlling the trade-off between regularization term (i.e., KL loss) and reconstruction loss.

As our input data $\mathbf{x}$ is a binary vector (i.e., implicit feedback), we consider logistic likelihood for the output of the VAE decoder. Defining $f_{\boldsymbol{\psi}}(\mathbf{z}) = [o_i]$ as the output of generative network, the logistic log-likelihood for input $\mathbf{x}$ is

$$\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}) = \sum_i x_i \log \sigma(o_i) + (1 - x_i) \log(1 - \sigma(o_i)). \tag{4}$$

Here, $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function. This logistic likelihood renders the reconstruction loss to the cross-entropy loss.

JoVA Model. Our model consists of two separate variational autoencoders: the *user VAE* and the *item VAE* (see Figure 1). Given the implicit feedback matrix $\mathbf{R}$, the user VAE reconstructs the matrix row-by-row (i.e., reconstructs user vector $\mathbf{R}_u$), whereas the item VAE reconstructs it column-by-column (i.e., reconstructs item vector $\mathbf{R}_i^T$). These two VAEs independently and simultaneously complete the implicit feedback matrix. The final output is the average of two predicted implicit matrices: $\hat{\mathbf{R}} = \frac{1}{2}(\hat{\mathbf{R}}^{user} + \hat{\mathbf{R}}^{item})$, where $\hat{\mathbf{R}}^{user}$ and $\hat{\mathbf{R}}^{item}$ are implicit matrices predicted (or completed) by the user VAE and the item VAE, respectively. We note that $\hat{\mathbf{R}} \in [0,1]^{m \times n}$, where each $\hat{r}_{ui}$ represents the predicted likelihood that user u interacts with item i. This natural probabilistic interpretation originates from our choice of logistic likelihood for the output of VAEs (see Eq. 4). The parameters of the user VAE and the item VAE are learned with a joint loss function (see below).

We carefully designed JoVA model to capture both user-user and item-item correlations. The item VAE embeds similar items close

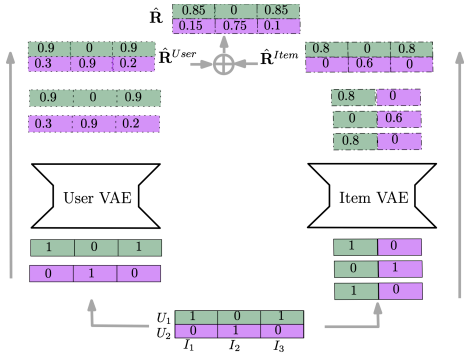


Figure 1: Illustration of the JoVA model. User and item VAEs recover the input matrix independently. The final output is the average of these two reconstructed matrices.

to each other in its latent representations to preserve their correlations, while the user VAE does the same for similar users. The joint optimization of these two VAEs helps their fine-tune calibration, so that they can complement each other in their predictions. The item and user VAEs together can learn complementary information from user-item interactions beyond what each could separately learn. This richer learning is a valuable asset for sparse datasets (as confirmed by our experiments in Section 4).

Similar to ensemble learning, JoVA aggregates the predictions of user and item VAEs into the final prediction. The aggregation in JoVA is with unweighted averaging, shown to be a reliable aggregation method in the ensemble of deep learning models [12]. One can use the weighed averaging at the cost of tuning more hyper-parameters, but with the promise of increased accuracy.

Loss functions. We consider two variants of loss functions for JoVA. One naturally arises from the combination of two user and item variational autoencoders:

$$L_{\text{JoVA}}(\mathbf{R}|\boldsymbol{\theta},\alpha) = \sum_{u\in U} L_{\text{VAE}}(\mathbf{R}_u|\boldsymbol{\theta}_U,\alpha) + \sum_{i\in I} L_{\text{VAE}}(\mathbf{R}_i^T|\boldsymbol{\theta}_I,\alpha) \quad (5)$$

Here, $\boldsymbol{\theta}_U$ and $\boldsymbol{\theta}_I$ represent the model parameters of user and item VAEs respectively, and L_{VAE} is computed by Eq. 3 with the logistic likelihood of Eq. 4. To further specialize JoVA model for the top-k recommendation with implicit feedback, we incorporate a pairwise ranking loss in its joint loss function. Specifically, we introduce the *JoVA-Hinge (JoVA-H)* loss function:

$$L_{\text{JoVA-H}}(\mathbf{R}|\boldsymbol{\theta},\alpha,\beta,\lambda) = L_{\text{JoVA}}(\mathbf{R}|\boldsymbol{\theta},\alpha) + \beta L_{\text{H}}(\mathbf{R}|\boldsymbol{\theta},\lambda), \quad (6)$$

where $L_{\text{H}}(\mathbf{R}|\boldsymbol{\theta},\lambda) = \sum_{u\in U}\sum_{i\in I_u^+}\sum_{j\in I_u^-}\max(0,\hat{r}_{uj}-\hat{r}_{ui}+\lambda)$ is a *hinge loss function*, widely and successfully used as a pairwise ranking loss [20, 22, 25] for recommendation with implicit feedback. Here, $\hat{r}_{ui}$ is the predicted interaction score (or likelihood) of a user u for an item i, and λ is the margin hyperparameter.

The hinge loss is built upon the assumption that a user u prefers his interacted item $i \in I_u^+$ over an uninteracted item (or negative example) $j \in I_u^-$ with the margin of at least λ. We have introduced the hyperparameter β for controlling the influence of hinge loss to the JoVA's objective function. The JoVA-Hinge loss function in Eq. 6, by combining pointwise losses of variational autoencoders and the hinge pairwise loss, extends the standard approaches of deploying either pointwise or pairwise loss functions.

Table 1: The summary statistics of datasets.

Dataset	#User	#Item	#Interaction	Sparsity
MovieLens	6,027	3,062	574,026	96.89%
Yelp	12,705	9,245	318,314	99.73%
Pinterest	55,187	9,911	1,500,806	99.73%
Netflix	70,000	17,769	8,623,831	99.31%

4 EXPERIMENTS

Our empirical experiments assess the effectiveness of our proposed methods against a set of state-of-the-art methods.[1]

Evaluation Datasets. We report results obtained on four datasets: MovieLens-1M (ML1M)[2], Pinterest[3], Yelp[4], and Netflix.[5] Pinterest is a dataset with implicit feedback. Following the previous work [9], we kept only users with at least 20 interactions (pins). ML1M, Yelp, and Netflix originally include five-star ratings. As with [15, 16, 25], we converted the user-item rating R_{ui} to 1, if $R_{ui} \geq 4$ and to 0 otherwise. For Netflix, we have randomly selected 70,000 users with all their user-item interactions from the original dataset. Table 1 provides the summary statistics of our datasets after pre-processing. For each dataset, the user-item interactions are randomly split into 80% training, 10% validation, and 10% testing datasets.

Evaluation Metrics. We utilize four commonly-used metrics to assess the quality of predicted ranked list for each user u: *Precision@k (P@k)* ; *Recall@k (R@k)F1-score@k (F1@k)* ; and *NDCG@k*. We report the average of these metrics (over testing users).

Baselines. We compare our methods against state-of-the-art methods: *BPR* [17] optimizes a matrix factorization (MF) model with a pair-wise ranking loss. *CDAE* [21] uses denoising auto-encoder to user-interaction predictions. *Mult-VAE* [15] uses only a single VAE with multinomial distribution for the output of the decoder. *NCF* [9] learns user-item interaction by combining MF and multi-layer perceptrons. *JCA* [25] deploys two classical autoencoders. *FAWMF* [5] is an adaptive weighted matrix factorization method based on a VAE. For all these baselines, we have used the implementations and optimal parameter settings reported by the original papers.

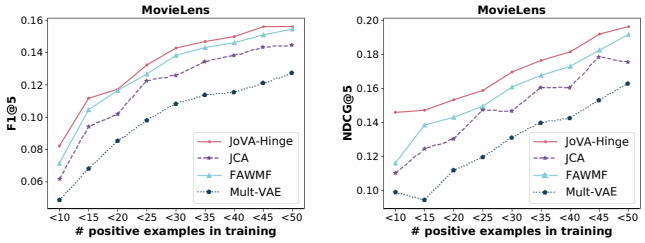
Experimental Setup. The models are trained by Adam with a learning rate of 0.003. For our models, as with [25], we decomposed the training matrix into 1500x1500 mini-batch matrices. The hyperparameters are set by the grid search on the validation sets: $\lambda = 0.15$, $\alpha = 0.01$, and $\beta = 0.001$ (except for Yelp with $\beta = 0.01$). Similar to [25], we randomly sampled one negative instance per a positive instance in each epoch. For each encoder and decoder, we had two hidden layers each with 320 dimensions and tanh activation functions, while the sigmoid activation function was used for the output layers. We set the dimension of the latent space d to 80.

Exp-1: Acuracy Comparison. We compare the accuracy of the top-k recommendation of our models and baselines with various $k \in \{1, 5, 10\}$. Table 2 reports F1-Score and NDCG for all datasets and methods. The results for precision and recall were qualitatively similar. Our JoVA-Hinge outperforms others for F1 measure on

[1] Source code available at https://github.com/bahareAskari/JoVA-Hinge.git
[2] http://files.grouplens.org/datasets/movielens/ml-1m.zip.
[3] https://sites.google.com/site/xueatalphabeta/academic-projects
[4] https://www.yelp.com/dataset/challenge.
[5] https://www.kaggle.com/netflix-inc/netflix-prize-data

Table 2: Acuracy comparions of the baselines and JoVA-Hinge. The best and second best are in purple and grey respectively.

	ML1M						Yelp						Pinterest						Netflix					
	F1-score			NDCG			F1-score			NDCG			F1-score			NDCG			F1-score			NDCG		
	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
BPR	.041	.129	.170	.284	.255	.243	.007	.018	.022	.017	.022	.030	.012	.029	.033	.033	.031	.041	.001	.003	.005	.009	.010	.010
NCF	.051	.149	.188	.296	.273	.271	.015	.033	.035	.037	.039	.050	.012	.031	.038	.038	.035	.048	.001	.006	.008	.013	.012	.013
CDAE	.052	.147	.187	.343	.290	.273	.016	.032	.036	.038	.039	.047	.015	.035	.040	.042	.039	.051	.001	.005	.007	.011	.011	.013
Mult-VAE	.052	.142	.180	.343	.289	.270	.015	.032	.034	.035	.038	.047	.015	.035	.040	.047	.04	.050	.001	.004	.01	.011	.011	.011
FAWMF	.060	.166	.207	.378	.318	.299	.015	.029	.031	.036	.036	.043	.013	.031	.036	.042	.036	.045	.002	.006	.009	.019	.017	.017
JCA	.060	.163	.208	.370	.313	.298	.016	.035	.038	.041	.044	.054	.015	.038	.046	.045	.042	.056	.002	.007	.011	.017	.016	.017
JoVA-H	.062	.167	.212	.372	.314	.301	.020	.039	.040	.045	.048	.058	.020	.047	.054	.060	.053	.068	.003	.008	.012	.020	.019	.019
% improve	3.333	0.600	1.923	-1.58	-1.25	0.668	25	11.43	5.26	9.76	9.10	7.41	33.33	23.68	17.39	33.33	26.19	21.43	50	14.29	9.10	5.26	11.76	11.76



(a) F1-Score@5 (b) NDCG@5

Figure 2: The avg. accuracy of users with the varying number of training data, MovieLens.

all datasets and various k. Compared with the best baseline, F1-score@k is improved by up to 3.33% in ML1M, 25% in Yelp, 33.33% in Pinterest, and 50% in Netflix. For NDCG, JoVA-Hinge also outperforms others significantly in three datasets of Yelp, Pinterest, and Netflix. In Yelp, the mimimum improvement is 7.41% (for $k = 10$) and the maximum improvement is 9.76% (for $k = 1$). In Netflix, the mimimum improvement is 5.26% (for $k = 1$) and the maximum improvement is 11.76% (for $k = 5$). The JoVA-Hinge has even higher improvement for Pinterest with the mimimum of 21.72% (for $k = 10$) and the maximum of 33.33% (for $k = 1$). Cross-examination of Tables 1 and 2 suggest that our JoVA-H model significantly improves the accuracy of the state-of-the-art methods in terms of both F1 and NDCG for sparser datasets (i.e., Yelp, Pinterest, and Netflix). Our results also suggest that JoVA-H offer more improvement for smaller k, which is of special practical interest for reducing cognitive burden on users, when the recommendation slate is small.

Exp-2: Users with Limited Data. We aim to understand how the prediction accuracy changes for users with a different number of user-item interactions (i.e., positive examples). For the previous experiments, rather than computing the average accuracy over all users, we compute the average accuracy over users with at most L user-item interactions in training data (while increasing L). This setting allows us to study how more availability of user-item interactions affect the accuracy of recommendation. Fig. 2 shows the performance of the top four methods of previous experiments when L increases. JoVA-Hinge outperforms other methods not only for users with the low number of user-item interactions (i.e., cold-start users), but also for well-established users. This suggests that the success of JoVA-Hinge is not limited to a specific class of users.

Exp-3: Ablation Study. Our JoVA-Hinge encompass three integral components of User VAE, Item VAE, and Hinge loss. To understand the extent of which each component has contributed in the success

Table 3: Ablation study of JoVA-Hinge. The purple and grey shows the best and second best results, respectively.

	ML1M		Yelp		Pinterest	
	F1@1	NDCG@1	F1@1	NDCG@1	F1@1	NDCG@1
User VAE	.0510	.3191	.0150	.0352	.0168	.0508
User VAE-H	.0486	.3043	.0154	.0344	.0127	.0383
Item VAE	.0555	.3423	.0156	.0352	.0178	.0538
Item VAE-H	.0573	.3479	.0181	.0407	.0200	.0597
JoVA	.0605	.3730	.0180	.0433	.0189	.0571
JoVA-H	.0624	.3718	.0201	.0449	.0200	.0604
	F1@5	**NDCG@5**	**F1@5**	**NDCG@5**	**F1@5**	**NDCG@5**
User VAE	.1379	.2683	.0328	.0397	.0430	.0477
User VAE-H	.1360	.2596	.0323	.0388	.0330	.0364
Item VAE	.1556	.2933	.0308	.0376	.0435	.0489
Item VAE-H	.1558	.2932	.0362	.0442	.0469	.0520
JoVA	.1657	.3135	.0360	.0449	.0461	.0516
JoVA-H	.1665	.3143	.0391	.0483	.0471	.0532
	F1@10	**NDCG@10**	**F1@10**	**NDCG@10**	**F1@10**	**NDCG@10**
User VAE	.1750	.254	.0365	.0495	.0512	.0625
User VAE-H	.1728	.2482	.0346	.0474	.0401	.0486
Item VAE	.1980	.2816	.0340	.0472	.0498	.0621
Item VAE-H	.1984	.2816	.0385	.0540	.0538	.0663
JoVA	.2092	.2990	.0395	.0553	.0538	.0666
JoVA-H	.2115	.3013	.0401	.0581	.0542	.0678

of JoVA-Hinge, we conduct an ablation study on JoVA-Hinge, by removing some of its components and evaluating the resulting models. Table 3 shows the results of our ablation studies. We notice that JoVA always outperforms both User VAE and Item VAE (for all datasets and metrics), suggesting that the ensemble of VAEs is more effective than individual VAEs. Hinge loss always improves Item VAE, surprisingly downgrades User VAE, and improves JoVA (except for NDCG@1 on MovieLens). This finding suggests that (i) hinge loss not necessarily can improve the performance of each individual VAE; however, (ii) it usually improves the performance of ensemble of VAEs.

5 CONCLUDING REMARKS

We have introduced joint variational autoencoder (JoVA) for top-k recommendation with implicit feedback and its variant JoVA-Hinge. Our empirical experiments on four real-world datasets show that JoVA-Hinge significantly advances the recommendation accuracy compared to state-of-the-art methods, under various evaluation metrics. In future work, we plan to explore extending JoVA-Hinge to incorporate user and item features (e.g., descriptions, demographics, etc.), side information (e.g., social networks), context (e.g., time, location, etc.), or non-stationary user preferences.

REFERENCES

[1] Charu C. Aggarwal. 2016. *Recommender Systems.* Springer.

[2] Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. 2009. I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems. In *UMAP'09*. 247–258.

[3] Suhrid Balakrishnan and Sumit Chopra. 2012. Collaborative Ranking. In *WSDM'12* (Seattle, Washington, USA). 143–152.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational Inference: A Review for Statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[5] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast Adaptively Weighted Matrix Factorization for Recommendation with Implicit Feedback.. In *AAAI'20*. New York, USA, 3470–3477.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (Boston, USA). 7–10.

[7] P Kingma Diederik, Max Welling, et al. 2014. Auto-encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)* (Banff, Canada), Vol. 1.

[8] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI'16* (Phoenix, Arizona USA).

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW'17* (Perth, Australia). 173–182.

[10] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *SIGIR'16* (Pisa, Tuscany, Italy). 549–558.

[11] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM'08* (Washington, USA). 263–272.

[12] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. 2018. The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification. *Journal of Applied Statistics* 45, 15 (2018), 2800–2818.

[13] Yehuda Koren. 2008. Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. In *ICDM'08* (Las Vegas, Nevada, USA). 426–434.

[14] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *KDD'17* (Halifax, Canada). 305–314.

[15] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW'18* (Lyon, France). 689–698.

[16] Huafeng Liu, Jingxuan Wen, Liping Jing, and Jian Yu. 2019. Deep Generative Ranking for Personalized recommendation. In *RecSys'19* (Copenhagen, Denmark). 34–42.

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized Ranking from Implicit Feedback. In *UAI'09* (Montreal, Canada). 452–461.

[18] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. 528–536.

[19] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of Collaborative Filtering Techniques. *Advances in artificial intelligence* 2009, Article 4 (Jan. 2009).

[20] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to Large Vocabulary Image Annotation. In *IJCAI'11* (Barcelona, Spain). 2764–2770.

[21] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-encoders for Top-n Recommender Systems. In *WSDM'16* (San Francisco, USA). 153–162.

[22] Ting Yao, Tao Mei, and Yong Rui. 2016. Highlight Detection with Pairwise Deep Ranking for First-person Video Summarization. In *CVPR'16* (Las Vegas, USA). 982–990.

[23] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. Collaborative deep ranking: A Hybrid Pair-wise Recommendation Algorithm with Implicit Feedback. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Auckland, New Zealand). 555–567.

[24] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[25] Ziwei Zhu, Jianling Wang, and James Caverlee. 2019. Improving Top-K Recommendation via Joint Collaborative Autoencoders. In *WWW'19* (San Francisco, USA). 3483–3482.