VAE++: Variational AutoEncoder for Heterogeneous One-Class Collaborative Filtering

Wanqi Ma[1, 2, 3], Xiancong Chen[1, 2, 3], Weike Pan[1, 2, 3*], Zhong Ming[1, 2, 3*]
{mawanqi2019, chenxiancong}@email.szu.edu.cn, {panweike, mingz}@szu.edu.cn

[1]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, China

[2]Guangdong Laboratory of Artficial Intelligence and Digital Economy (SZ), Shenzhen University, China

[3]College of Computer Science and Software Engineering, Shenzhen University, China

Motivation

The limitation of existing recommendation methods:

- Neural network-based models for collaborative filtering have received widespread attention, among which variational autoencoder (VAE) has shown unique advantages in the task of item recommendation.
- However, most existing VAE-based models only focus on one type of user feedback, leading to their performance bottlenecks.

Overall of Our Solution

- We propose a novel VAE-based recommendation model called VAE++, which can effectively utilize heterogeneous feedback to boost recommendation performance.
- Specifically, it combines three different types of signals, i.e., purchase feedback, examination feedback and their mixed feedback, via two well-designed modules, i.e., a target representation enhancement module and a target representation refinement module.

Advantages of Our Solution

- Our VAE++ combines three types of signals, i.e., the purchase feedback, the examination feedback and their mixed feedback, into one model, so that different types of feedback can complement each other to obtain better recommendation performance.
- Extensive experiments on three public datasets show that our VAE++ achieves the best results compared with several state-of-the-art methods.

Related Work (1/2)

- **One-Class Collaborative Filtering (OCCF)**
  - Variational autoencoder (VAE) [Liang et al., 2018] is a generative model with strong generalization, which enables it to accurately predict the users' preferences towards items.

VAE is a very competitive method compared with a variety of state-of-the-art methods [Dacrema et al., 2019]. However, it is designed to solve the OCCF problem, which motivates us to design a novel method based on it to deal with the HOCCF problem.

Related Work (2/2)

- **Heterogeneous One-Class Collaborative Filtering (HOCCF)**
  - Role-based transfer to rank (RoToR) [Pan et al., 2019] is based on the traditional MF model [Koren et al., 2009], which may not be sufficient to capture the complex interactions between users and items.
  - Efficient heterogeneous collaborative filtering (EHCF) [Chen et al., 2020] leverages linear functions to model the relations between multiple behaviors, which may fail to learn the users' behavior patterns well.
  - Staged VAE (SVAE) [Chen et al., 2021] captures the users' examination and purchase preferences through two separate models, which may not be a good way to share knowledge between different types of feedback.

Our VAE++ jointly models the users' purchase, examination and their mixed feedback via two well-designed modules, and improves the reconstruction ability of VAE by sharing knowledge between different data.

Problem Definition

- **Input:** We have a set of users $\mathcal{U} = \{u\} = \{1, 2, ..., n\}$, a set of items $\mathcal{I} = \{i\} = \{1, 2, ..., m\}$, and two different types of user feedback, i.e., the target feedback such as purchases $\mathcal{R}^{\mathcal{P}} = \{(u, i)\}$ and the auxiliary feedback such as examinations $\mathcal{R}^{\mathcal{E}} = \{(u, i)\}$. We keep a (u, i) pair associated with both purchase and examination behaviors only in the purchase data $\mathcal{R}^{\mathcal{P}}$, and thus have $\mathcal{I}_u^{\mathcal{P}} \cap \mathcal{I}_u^{\mathcal{E}} = \emptyset$.
- **Goal:** Our goal is to generate a personalized ranked list of items for each user u from the items that he/she has not purchased, i.e., $\mathcal{I} \backslash \mathcal{I}_u^{\mathcal{P}}$.

VAE++ (1/2)

- Our VAE++ incorporates three types of signals, i.e., the purchase feedback, the examination feedback and their mixed feedback, into one model in a seamless manner.
- It consists of four main components, including a target encoder, a target representation enhancement module, a target representation refinement module and a target decoder.

VAE++ (2/2)

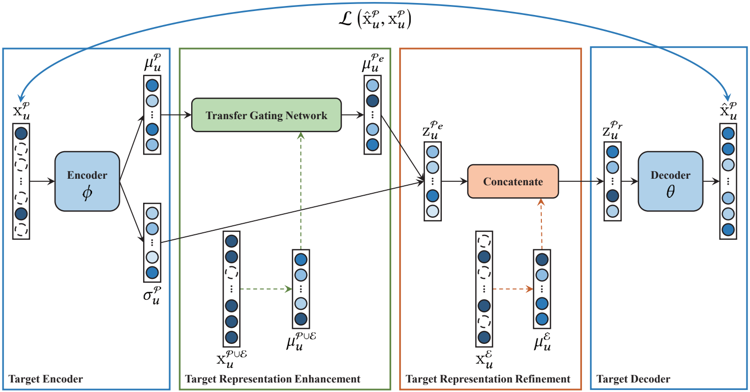


Figure: Illustration of our VAE++ for modeling the target feedback, i.e., purchases ($\mathcal{R}^{\mathcal{P}}$), and the auxiliary feedback, i.e., examinations ($\mathcal{R}^{\mathcal{E}}$), in HOCCF.

Target Encoder (1/2)

- Following [Liang et al., 2018], we use a variational autoencoder to learn the users' latent representations because of its strong feature learning ability.
- Let $\mathbf{z}_u^{\mathcal{P}} \in \mathbb{R}^{1 \times d}$ denote the purchase latent representation of user u with d as the latent dimensionality. The objective of the target encoder is to produce the distribution of $\mathbf{z}_u^{\mathcal{P}}$ according to the purchase data $\mathcal{R}^{\mathcal{P}}$.

Target Encoder (2/2)

- For each user u, the input of the target encoder is a multi-hot encoding purchase vector $\mathbf{x}_u^{\mathcal{P}} \in \{0, 1\}^{1 \times m}$, and the output is the mean $\boldsymbol{\mu}_u^{\mathcal{P}} \in \mathbb{R}^{1 \times d}$ and the standard deviation $\boldsymbol{\sigma}_u^{\mathcal{P}} \in \mathbb{R}^{1 \times d}$ of the latent variable $\mathbf{z}_u^{\mathcal{P}}$, which can be obtained as follows,

$$\boldsymbol{\mu}_u^{\mathcal{P}} = f(\mathbf{x}_u^{\mathcal{P}} \mathbf{W}_{\mu_1} + \mathbf{b}_{\mu_1}), \tag{1}$$

$$\boldsymbol{\sigma}_u^{\mathcal{P}} = \exp^{f(\mathbf{x}_u^{\mathcal{P}} \mathbf{W}_{\sigma_1} + \mathbf{b}_{\sigma_1})}, \tag{2}$$

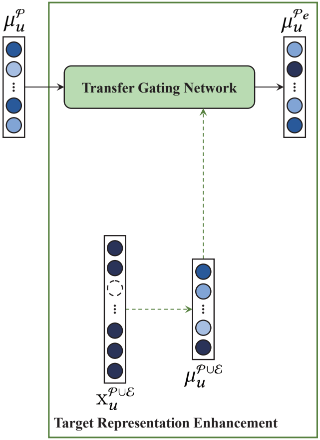
where $\mathbf{W}_{\mu_1}, \mathbf{W}_{\sigma_1} \in \mathbb{R}^{m \times d}$ and $\mathbf{b}_{\mu_1}, \mathbf{b}_{\sigma_1} \in \mathbb{R}^{1 \times d}$ are the weight matrices and bias vectors, respectively, and $f(\cdot)$ is an activation function for the hidden layer.

Target Representation Enhancement (1/7)

- In most real-world applications, the purchase feedback are relatively sparse, which may limit the generalization ability of VAE.
- To overcome this obstacle, we introduce the users' examination feedback to accurately model their latent representations.
- The examination data does not include the purchase data, i.e., for each user u, $\mathcal{I}_u^{\mathcal{P}} \cap \mathcal{I}_u^{\mathcal{E}} = \emptyset$. Therefore, there are two forms of auxiliary data available. One is a mixture of the purchase data and the examination data, denoted as $\mathcal{R}^{\mathcal{P} \cup \mathcal{E}}$, and the other only contains the examination data, denoted as $\mathcal{R}^{\mathcal{E}}$.

Target Representation Enhancement (2/7)

- To enhance the learning of the user purchase representations, we leverage the first type of auxiliary data, i.e., the mixed data of purchases and examinations $\mathcal{R}^{\mathcal{P}\cup\mathcal{E}}$, in the target representation enhancement (TRE) module.
- The purpose of TRE is to connect the purchase feedback and the mixed feedback via a transfer gating network, so that the knowledge in the mixed feedback can be used to learn the users' purchase preferences.



Target Representation Enhancement (3/7)

- We use a multilayer perceptron (MLP) to compress the mixed vector $\mathbf{x}_u^{\mathcal{P}\cup\mathcal{E}} \in \{0,1\}^{1\times m}$, which denotes the overall interactions of user u on the entire item set. The obtained latent feature $\boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}}$ can be seen as the user u's mixed preferences in the purchase data and the examination data,

$$\boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}} = f(\mathbf{x}_u^{\mathcal{P}\cup\mathcal{E}}\mathbf{W}_{\mu_2} + \mathbf{b}_{\mu_2}), \tag{3}$$

where $\mathbf{W}_{\mu_2} \in \mathbb{R}^{m\times d}$ and $\mathbf{b}_{\mu_2} \in \mathbb{R}^{1\times d}$ are the weight matrix and bias vector, respectively, and $f(\cdot)$ is an activation function for the hidden layer.

Target Representation Enhancement (4/7)

- To better fuse the purchase latent feature $\boldsymbol{\mu}_u^{\mathcal{P}}$ and the mixed latent feature $\boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}}$, different weights need to be assigned to them. Inspired by [Lin et al., 2020], we propose a transfer gating network to calculate their weights. The gating network is represented as an MLP as follows,

$$g = \sigma([\boldsymbol{\mu}_u^{\mathcal{P}}, \boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}}]\mathbf{W}_G + \mathbf{b}_G), \tag{4}$$

  where $[\cdot,\cdot]$ is the concatenation operation, $\boldsymbol{W}_G \in \mathbb{R}^{2d\times 1}$ and $b_G \in \mathbb{R}$ are the parameters of the feedforward network, and $\sigma(\cdot)$ is the sigmoid function to restrict g to $(0, 1)$.

Target Representation Enhancement (5/7)

- With the gating value g, the enhanced latent feature $\boldsymbol{\mu}_u^{\mathcal{P}e}$ can be obtained by the weighted sum of the purchase latent feature $\boldsymbol{\mu}_u^{\mathcal{P}}$ and the mixed latent feature $\boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}}$ as follows,

$$\boldsymbol{\mu}_u^{\mathcal{P}e} = \boldsymbol{\mu}_u^{\mathcal{P}} \otimes g + \boldsymbol{\mu}_u^{\mathcal{P}\cup\mathcal{E}} \otimes (1 - g), \tag{5}$$

where $\otimes$ is the element-wise product. Notice that the latent feature $\boldsymbol{\mu}_u^{\mathcal{P}e}$ is encoded with the user u's purchase preferences and mixed preferences, which allows it to leverage the information obtained from the mixed behaviors to enhance the learning of the purchase representations.

Target Representation Enhancement (6/7)

- With the enhanced mean $\boldsymbol{\mu}_u^{\mathcal{P}e}$ and the standard deviation $\boldsymbol{\sigma}_u^{\mathcal{P}}$, the improved latent representation $\mathbf{z}_u^{\mathcal{P}e}$ can be obtained by sampling from a variational distribution with model parameters ϕ,

$$q_\phi(\mathbf{z}_u^{\mathcal{P}e}|\mathbf{x}_u^{\mathcal{P}}) = \mathcal{N}(\boldsymbol{\mu}_u^{\mathcal{P}e}, \mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{u}}^{\mathcal{P}2})). \tag{6}$$

- To allow parameters to be optimized in backpropagation, the reparameterization trick is applied [Kingma and Welling, 2013, Rezende et al., 2014]. Specifically, we approximate the enhanced latent variable $\mathbf{z}_u^{\mathcal{P}e}$ with the normal distribution $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathbf{1}))$, and have $\mathbf{z}_u^{\mathcal{P}e} = \boldsymbol{\mu}_u^{\mathcal{P}e} + \epsilon \otimes \boldsymbol{\sigma}_u^{\mathcal{P}}$.

Target Representation Enhancement (7/7)

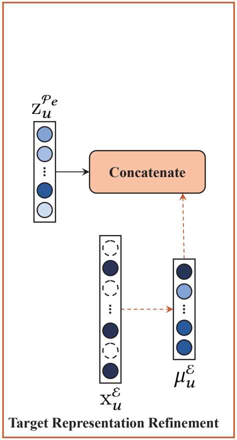
- The enhanced latent variable $\mathbf{z}_u^{\mathcal{P}e}$ needs to be regularized through the Kullback-Leibler (KL) divergence between the variational posterior $q_\phi(\mathbf{z}_u^{\mathcal{P}e}|\mathbf{x}_u^{\mathcal{P}})$ and the prior $p(\mathbf{z}_u^{\mathcal{P}e})$ as follows,

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{z}_u^{\mathcal{P}e}) = \mathrm{KL}(q_\phi(\mathbf{z}_u^{\mathcal{P}e}|\mathbf{x}_u^{\mathcal{P}})||p(\mathbf{z}_u^{\mathcal{P}e})), \tag{7}$$

  which encourages the learned posterior distribution $q_\phi(\mathbf{z}_u^{\mathcal{P}e}|\mathbf{x}_u^{\mathcal{P}})$ to be close to the assumed prior distribution $p(\mathbf{z}_u^{\mathcal{P}e})$, i.e., the commonly used standard normal distribution.

Target Representation Refinement (1/2)

- To further refine the users' purchase representations, we introduce the examination data $\mathcal{R}^{\mathcal{E}}$ into the target representation refinement (TRR) module.
- The intuition is that the users' purchase representations can be more accurately modeled by learning the difference between their purchase preferences and examination preferences.



Target Representation Refinement (2/2)

- The examination latent feature $\boldsymbol{\mu}_u^{\mathcal{E}}$ can be obtained by using an MLP to compress the examination vector $\mathbf{x}_u^{\mathcal{E}} \in \{0, 1\}^{1\times m}$,

$$\boldsymbol{\mu}_u^{\mathcal{E}} = f(\mathbf{x}_u^{\mathcal{E}} \mathbf{W}_{\mu_3} + \mathbf{b}_{\mu_3}), \quad (8)$$

  where $\mathbf{W}_{\mu_3} \in \mathbb{R}^{m\times d}$ and $\mathbf{b}_{\mu_3} \in \mathbb{R}^{1\times d}$ are the weight matrix and bias vector, respectively, and $f(\cdot)$ is an activation function for the hidden layer.
- Then we adopt the concatenation operation to combine the enhanced latent variable $\mathbf{z}_u^{\mathcal{P}e}$ and the examination latent feature $\boldsymbol{\mu}_u^{\mathcal{E}}$, and obtain the final purchase representation $\mathbf{z}_u^{\mathcal{P}r}$ as follows,

$$\mathbf{z}_u^{\mathcal{P}r} = [\mathbf{z}_u^{\mathcal{P}e}, \boldsymbol{\mu}_u^{\mathcal{E}}], \quad (9)$$

  where $\boldsymbol{\mu}_u^{\mathcal{E}}$ serves as a signal to distinguish the most useful part of the enhanced purchase representations, enabling the target decoder to generate high-quality samples.

Target Decoder (1/2)

- The target decoder is a generative model, whose objective is to produce the probability distribution over the user u's purchase history $\mathbf{x}_u^{\mathcal{P}}$.
- For each user u, it takes the final latent variable $\mathbf{z}_u^{\mathcal{P}r}$ as input and outputs the distribution over the entire item set through a softmax function. Then, it reconstructs the input vector from the multinomial distribution [Liang et al., 2018],

$$\pi(\mathbf{z}_u^{\mathcal{P}r}) = \text{softmax}(f_\theta(\mathbf{z}_u^{\mathcal{P}r})), \tag{10}$$

$$\mathbf{x}_u^{\mathcal{P}} \sim \text{Multi}(N_u^{\mathcal{P}}, \pi(\mathbf{z}_u^{\mathcal{P}r})), \tag{11}$$

where $f_\theta(\cdot)$ is an MLP with parameters θ, $\pi(\mathbf{z}_u^{\mathcal{P}r})$ is the distribution function of $f_\theta(\cdot)$, and $N_u^{\mathcal{P}}$ is the total number of purchases of user u.

Target Decoder (2/2)

- Let $\hat{\mathbf{x}}_u^{\mathcal{P}}$ denote the reconstructed purchase vector, which should be close to the input vector $\mathbf{x}_u^{\mathcal{P}}$, so we have the reconstruction loss as follows,

$$\mathcal{L}(\hat{\mathbf{x}}_u^{\mathcal{P}}, \mathbf{x}_u^{\mathcal{P}}) = \mathbb{E}_{q_\phi(\mathbf{z}_u^{\mathcal{P}e}|\mathbf{x}_u^{\mathcal{P}})}[\log p_\theta(\mathbf{x}_u^{\mathcal{P}}|\mathbf{z}_u^{\mathcal{P}r})]. \quad (12)$$

  The examination and purchase preferences are jointly learned by maximizing Eq.(12), which allows the knowledge extracted from the users' examination behaviors to be effectively transferred to learning their purchase interests.
- Following [Liang et al., 2018], the overall loss function of our VAE++ is as follows,

$$\mathcal{L}_{\mathrm{VAE++}} = \mathcal{L}(\hat{\mathbf{x}}_u^{\mathcal{P}}, \mathbf{x}_u^{\mathcal{P}}) - \beta \mathcal{L}_{\mathrm{KL}}(\mathbf{z}_u^{\mathcal{P}e}), \quad (13)$$

  where $\beta \in [0, 1]$ is a parameter to weight the regularization.

Research Questions

RQ1: How does our VAE++ perform compared to some state-of-the-art recommendation methods?

RQ2: What is the impact of different components in our VAE++?

RQ3: How do hyperparameters, such as the dimensionality and the number of recommended items, affect the performance of our VAE++?

RQ4: What is the effect of using different data as input in the target representation enhancement module and the target representation refinement module of our VAE++?

Datasets (1/3)

- We use three widely used datasets, i.e., MovieLens 10M (ML10M), Netflix and RecSys Challenge 2015 (Rec15), to evaluate the performance of the proposed model VAE++.
- ML10M and Netflix are two popular benchmark datasets related to movies. We preprocess these two datasets in the same way as [Pan et al., 2019]. (i) We randomly sample 60 percent of the rating records from each dataset, keep the (user, item) pairs with a score equal to 5 as the purchase data, and discard other records. (ii) We divide the purchase data into three parts equally. One is used as the training set, one is used as the validation set, and the other is used as the test set. (iii) All the remaining 40 percent of the rating records in each dataset are regarded as the examination data. We repeat the above steps three times to obtain three different copies of each dataset.

Datasets (2/3)

- Rec15 is a real dataset released by the RecSys 2015 competition. We process this dataset as follows. (i) For the items that are repeatedly purchased or examined in a session, we only keep the records with the earliest interaction. (ii) For the items purchased fewer than 5 times and the sessions with fewer than 5 purchase records, we remove them. (iii) For each session, we treat the penultimate purchase record as the validation set, the last purchase record as the test set, and the remaining records as the training set. (iv) If the validation set or the test set contains the examined items in the training set, we remove these items in the training set.

Datasets (3/3)

Table: Statistics of the three processed datasets, including the number of purchases (P) and examinations (E) in the training data, the number of purchases (P(val.)) in the validation data, and the number of purchases (P(te.)) in the test data. P/E denotes the ratio between P and E.

Statistics	ML10M	Netflix	Rec15
#Users	71,567	480,189	36,917
#Items	10,681	17,770	9,621
P	309,317	4,554,888	159,429
E	4,000,024	39,628,846	213,332
P(val.)	308,673	4,556,347	36,917
P(te.)	308,702	4,558,506	36,917
P/E	1:12.93	1:8.70	1:1.33
Density	0.56%	0.52%	0.11%

Evaluation Metrics

- To evaluate the performance of item recommendation, we adopt four widely used ranking-oriented metrics, including precision, recall, normalized discounted cumulative gain (NDCG) and 1-call [Valcarce et al., 2018].
- We report these metrics with the number of recommended items $K = 5$, denoted as Prec@5, Rec@5, NDCG@5 and 1-call@5.

Baselines

- Three OCCF algorithms:
  - BPR [Rendle et al., 2009].
  - MFLogLoss [Johnson, 2014].
  - VAE [Liang et al., 2018].
- Four HOCCF algorithms:
  - VALS [Ding et al., 2018] models the pairwise relations among the purchase data, examination data and missing data.
  - RoToR [Pan et al., 2019] combines purchase feedback and examination feedback through two forms of knowledge transfer.
  - EHCF [Chen et al., 2020] is a deep learning method, which considers the relations between different types of behaviors.
  - SVAE [Chen et al., 2021] is a two-stage model based on VAE [Liang et al., 2018], which transfers the knowledge extracted from the examination model to the purchase model.

Parameter Settings (1/2)

- For a fair comparison, we fix the size of latent factor dimension $d = 100$ in all models [Chen et al., 2021].
- For BPR, MFLogLoss and RoToR, we set the learning rate $\gamma = 0.01$, and search the best values of the tradeoff parameters from $\{0.001, 0.01, 0.1\}$ and the iteration number T from $\{100, 500, 1000\}$ [Pan et al., 2019].
- For VALS, we follow the suggested configurations [Ding et al., 2018], and search the weight of the missing data from $\{100, 200, 400, 800, 1600, 3200, 6400\}$, the weight of the examination data from $\{0.25, 0.5, 1, 1.5, 2\}$, and the margin values from $\{0.25, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$.

Parameter Settings (2/2)

- For the four deep learning-based models, i.e., EHCF, VAE, SVAE and our VAE++, they are implemented in TensorFlow[1], where the batch size is set to 500 and the dropout ratio ρ is set to 0.5 to prevent overfitting [Liang et al., 2018].
- For EHCF, we set the learning rate to 0.05 and configure other parameters by following the settings in [Chen et al., 2020].
- For VAE, we follow the settings in [Liang et al., 2018], adopt a structure with 1 hidden layer, and use the identity activation function for the hidden layer. In addition, we select the learning rate from $\{0.0001, 0.001, 0.01\}$, optimize it with mini-batch Adam, and use an early-stop strategy with a threshold 50.
- For SVAE and the proposed method VAE++, the parameter settings are consistent with VAE.

[1] https://www.tensorflow.org

Performance Comparison (RQ1) (1/5)

Table: Recommendation performance of our VAE++ and seven baselines, including three OCCF algorithms and four HOCCF algorithms, on ML10M.

Dataset	Method	Prec@5	Rec@5	NDCG@5	1-call@5
ML10M	BPR	0.0680±0.0002	0.0915±0.0003	0.0933±0.0004	0.2837±0.0023
	MFLogLoss	0.0736±0.0005	0.0995±0.0010	0.1019±0.0004	0.3034±0.0017
	VAE($\mathcal{R}^{\mathcal{P}}$)	0.0744±0.0003	0.0997±0.0003	0.1031±0.0005	0.3067±0.0008
	VAE($\mathcal{R}^{\mathcal{E}}$)	<u>0.0957</u>±0.0010	0.1367±0.0014	<u>0.1396</u>±0.0019	<u>0.3812</u>±0.0035
	VAE($\mathcal{R}^{\mathcal{P}\cup\mathcal{E}}$)	0.0838±0.0003	0.1166±0.0004	0.1172±0.0005	0.3427±0.0011
	VALS	0.0671±0.0006	0.0759±0.0012	0.0901±0.0011	0.2745±0.0024
	RoToR	0.0872±0.0001	0.1239±0.0007	0.1235±0.0006	0.3562±0.0008
	EHCF	0.0704±0.0009	0.0928±0.0023	0.0957±0.0015	0.2914±0.0044
	SVAE	0.0935±0.0004	<u>0.1369</u>±0.0011	0.1372±0.0010	0.3752±0.0023
	VAE++	**0.1071**±0.0003	**0.1524**±0.0003	**0.1567**±0.0003	**0.4153**±0.0014

Performance Comparison (RQ1) (2/5)

Table: Recommendation performance of our VAE++ and seven baselines, including three OCCF algorithms and four HOCCF algorithms, on Netflix.

Dataset	Method	Prec@5	Rec@5	NDCG@5	1-call@5
Netflix	BPR	0.0755±0.0004	0.0503±0.0005	0.0854±0.0004	0.2994±0.0013
	MFLogLoss	0.0785±0.0003	0.0549±0.0006	0.0900±0.0004	0.3103±0.0014
	VAE($\mathcal{R}^{\mathcal{P}}$)	0.0860±0.0001	0.0593±0.0001	0.0996±0.0002	0.3322±0.0004
	VAE($\mathcal{R}^{\mathcal{E}}$)	0.0960±0.0011	0.0738±0.0011	0.1158±0.0015	0.3677±0.0032
	VAE($\mathcal{R}^{\mathcal{P}\cup\mathcal{E}}$)	0.0907±0.0004	0.0689±0.0005	0.1063±0.0005	0.3541±0.0013
	VALS	–	–	–	–
	RoToR	0.0941±0.0003	0.0750±0.0003	0.1119±0.0004	0.3674±0.0010
	EHCF	0.0850±0.0007	0.0609±0.0006	0.0980±0.0008	0.3318±0.0031
	SVAE	<u>0.0986</u>±0.0004	<u>0.0795</u>±0.0005	<u>0.1187</u>±0.0005	<u>0.3797</u>±0.0013
	VAE++	**0.1235**±0.0003	**0.0961**±0.0006	**0.1502**±0.0006	**0.4409**±0.0014

Performance Comparison (RQ1) (3/5)

Table: Recommendation performance of our VAE++ and seven baselines, including three OCCF algorithms and four HOCCF algorithms, on Rec15.

Dataset	Method	Prec@5	Rec@5	NDCG@5	1-call@5
Rec15	BPR	0.0457±0.0004	0.2286±0.0017	0.1473±0.0007	0.2286±0.0017
	MFLogLoss	0.0490±0.0002	0.2451±0.0006	0.1586±0.0002	0.2451±0.0006
	VAE($\mathcal{R}^{\mathcal{P}}$)	0.0511±0.0002	0.2553±0.0006	0.1671±0.0005	0.2553±0.0006
	VAE($\mathcal{R}^{\mathcal{E}}$)	0.0357±0.0002	0.1783±0.0007	0.1185±0.0004	0.1783±0.0007
	VAE($\mathcal{R}^{\mathcal{P}\cup\mathcal{E}}$)	0.0509±0.0001	0.2543±0.0005	0.1615±0.0003	0.2543±0.0005
	VALS	<u>0.0557</u>±0.0001	<u>0.2784</u>±0.0005	<u>0.1858</u>±0.0003	<u>0.2784</u>±0.0005
	RoToR	0.0534±0.0001	0.2669±0.0007	0.1734±0.0007	0.2669±0.0007
	EHCF	0.0512±0.0001	0.2559±0.0005	0.1653±0.0003	0.2559±0.0005
	SVAE	0.0533±0.0001	0.2664±0.0004	0.1769±0.0002	0.2664±0.0004
	VAE++	**0.0558**±0.0000	**0.2792**±0.0001	**0.1861**±0.0003	**0.2792**±0.0001

Performance Comparison (RQ1) (4/5)

We can have the following observations:

- Compared with all the baselines, our VAE++ achieves the best performance across the three datasets, which clearly shows the advantage of our generic solution.
- For the four baselines exploiting heterogeneous one-class feedback, SVAE outperforms the other three methods in most cases, but it is still worse than our VAE++. In addition, the results of VALS and our VAE++ are comparable on Rec15, but on ML10M, VALS does not perform well compared to all the methods.
- For most HOCCF methods, including RoToR, SVAE and our VAE++, they achieve better results compared to the three methods that only use the purchase data, i.e., BPR, MFLogLoss and VAE($\mathcal{R}^{\mathcal{P}}$), which indicates that heterogeneous behavior information can help improve the recommendation accuracy.

Performance Comparison (RQ1) (5/5)

For the models exploiting homogeneous one-class feedback, we can have the following observations:

- VAE performs better than the other two methods, which illustrates the advantage of learning the distribution of user representations.
- For the three VAE-based methods with different sources of data, VAE($\mathcal{R}^{\mathcal{E}}$) performs better on ML10M and Netflix, while VAE($\mathcal{R}^{\mathcal{P}}$) achieves better performance on Rec15.
- VAE($\mathcal{R}^{\mathcal{P}} \cup \mathcal{R}^{\mathcal{E}}$) performs relatively poorly on the three datasets, indicating that simply merging two different types of data cannot capture the users' preferences well. Therefore, it is necessary to design an effective method to fuse them like our VAE++.

Ablation Study (RQ2) (1/2)

Table: Recommendation performance of our VAE++ by removing different components, i.e., target representation enhancement (TRE), target representation refinement (TRR) and TRE & TRR, respectively, for ablation studies on three datasets.

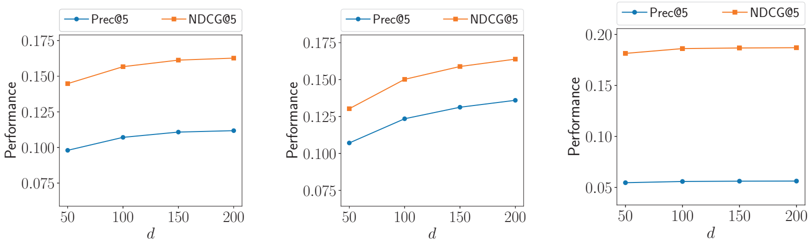
Dataset	Method	Prec@5	NDCG@5
ML10M	-TRE	**0.1072**$\pm$0.0005	0.1566$\pm$0.0001
	-TRR	0.0845$\pm$0.0007	0.1187$\pm$0.0006
	-TRE & TRR	0.0744$\pm$0.0003	0.1031$\pm$0.0005
	VAE++	0.1071$\pm$0.0003	**0.1567**$\pm$0.0003
Netflix	-TRE	0.1231$\pm$0.0005	0.1491$\pm$0.0006
	-TRR	0.0952$\pm$0.0003	0.1119$\pm$0.0004
	-TRE & TRR	0.0860$\pm$0.0001	0.0996$\pm$0.0002
	VAE++	**0.1235**$\pm$0.0003	**0.1502**$\pm$0.0006
Rec15	-TRE	0.0545$\pm$0.0003	0.1812$\pm$0.0009
	-TRR	0.0528$\pm$0.0001	0.1698$\pm$0.0003
	-TRE & TRR	0.0511$\pm$0.0002	0.1671$\pm$0.0005
	VAE++	**0.0558**$\pm$0.0000	**0.1861**$\pm$0.0003

Ablation Study (RQ2) (2/2)

We have the following observations:

- **"-TRE".** The performance of our VAE++ without TRE declines on Rec15, which demonstrates the usefulness of the transfer gating network in combining the purchase and mixed behaviors. Besides, it achieves comparable results with our VAE++ on ML10M and Netflix. The reason is that there are relatively more purchase data on these two datasets.
- **"-TRR".** Our VAE++ without TRR performs much worse, which shows the significance of learning the difference between the purchase and examination preferences to reconstruct the input purchase samples.
- **"-TRE & TRR".** The performance of our VAE++ without TRE and TRR further decreases, which shows that these two modules are critical to the performance of our VAE++.

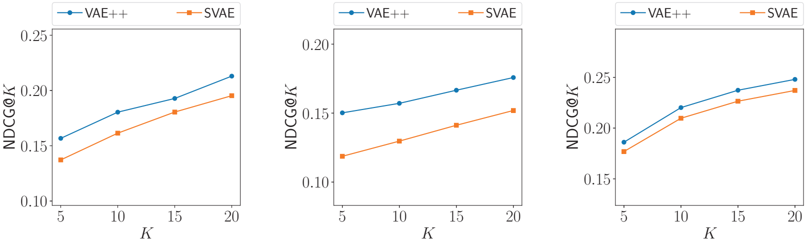
Hyperparameter Sensitivity (RQ3) (1/2)



(a) ML10M (b) Netflix (c) Rec15

Figure: Recommendation performance of our VAE++ with different numbers of latent dimensions on three datasets.

Hyperparameter Sensitivity (RQ3) (2/2)



(a) ML10M (b) Netflix (c) Rec15

Figure: Recommendation performance of our VAE++ and SVAE with different numbers of recommended items on three datasets.

Effect of Input Data (RQ4) (1/2)

Table: Recommendation performance of our VAE++ by using different input data in TRE and TRR, i.e., EE, EM, MM and ME, respectively, on three datasets. Notice that the default configuration of our VAE++ is ME, i.e., using the mixed data $\mathcal{R}^{\mathcal{P}\cup\mathcal{E}}$ in TRE and the examination data $\mathcal{R}^{\mathcal{E}}$ in TRR.

Dataset	Method	Prec@5	NDCG@5
ML10M	EE	**0.1072**$\pm$0.0002	<u>0.1564</u>$\pm$0.0005
	EM	0.0796$\pm$0.0007	0.1107$\pm$0.0008
	MM	0.0804$\pm$0.0001	0.1124$\pm$0.0001
	ME	<u>0.1071</u>$\pm$0.0003	**0.1567**$\pm$0.0003
Netflix	EE	<u>0.1228</u>$\pm$0.0004	<u>0.1487</u>$\pm$0.0004
	EM	0.0879$\pm$0.0004	0.1027$\pm$0.0005
	MM	0.0907$\pm$0.0001	0.1065$\pm$0.0001
	ME	**0.1235**$\pm$0.0003	**0.1502**$\pm$0.0006
Rec15	EE	<u>0.0548</u>$\pm$0.0001	<u>0.1825</u>$\pm$0.0004
	EM	0.0520$\pm$0.0003	0.1661$\pm$0.0003
	MM	0.0517$\pm$0.0003	0.1659$\pm$0.0001
	ME	**0.0558**$\pm$0.0000	**0.1861**$\pm$0.0003

Effect of Input Data (RQ4) (2/2)

We have the following observations:

- ME, i.e., our VAE++, outperforms the other three in most cases, which showcases that using the mixed feedback in TRE and the examination feedback in TRR can obtain excellent recommendation results.
- The performance of EE, i.e., using the examination data as the input of TRE and TRR, is comparable to that of ME, i.e., our VAE++, on ML10M and Netflix. The reason is that these two datasets are relatively dense, so only using the examination data can also help learn the users' purchase preferences.

Conclusions

- We propose a novel and generic VAE-based recommendation framework, i.e., VAE++, for dealing with the HOCCF problem.
- It utilizes three types of signals, including the purchase behaviors, the examination behaviors and their mixed behaviors, via two well-designed modules, i.e., a target representation enhancement module and a target representation refinement module.
- Extensive empirical studies on three public datasets show that our VAE++ achieves very promising performance compared with some highly competitive baseline methods.

Future Work

For future works, we will consider introducing some additional information into our VAE++, such as temporal dynamics [Bian et al., 2021] and social networks [Chen and Wong, 2021], to better transfer knowledge between different types of data.

Thank you!

- We thank the anonymous reviewers for constructive and expert comments, and the support of National Natural Science Foundation of China Nos. 61836005 and 62172283. We thank Mr. Dugang Liu for his helpful discussions and assistance.
- If you have any questions, please feel free to contact us.

Bian, Z., Zhou, S., Fu, H., Yang, Q., Sun, Z., Tang, J., Liu, G., Liu, K., and Li, X. (2021).
Denoising user-aware memory network for recommendation.
In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys'21, pages 400–410.

Chen, C., Zhang, M., Ma, W., Zhang, Y., Liu, Y., and Ma, S. (2020).
Efficient heterogeneous collaborative filtering without negative sampling for recommendation.
In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, AAAI'20, pages 19–26.

Chen, T. and Wong, R. C.-W. (2021).
An efficient and effective framework for session-based social recommendation.
In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM'21, page 400–408.

Chen, X., Pan, W., and Ming, Z. (2021).
Staged variational autoencoder for heterogeneous one-class collaborative filtering (in chinese).
In *Proceedings of the 18th China Conference on Machine Learning*, CCML'21.

Dacrema, M. F., Cremonesi, P., and Jannach, D. (2019).
Are we really making much progress? a worrying analysis of recent neural recommendation approaches.
In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys'19, pages 101–109.

Ding, J., Yu, G., He, X., Quan, Y., Li, Y., Chua, T.-S., Jin, D., and Yu, J. (2018).
Improving implicit recommender systems with view data.
In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 3343–3349.

Johnson, C. C. (2014).
Logistic matrix factorization for implicit feedback data.
In *Proceedings of the Workshop on Distributed Machine Learning and Matrix Computations at NeurIPS 2014*.

Kingma, D. P. and Welling, M. (2013).
Auto-encoding variational Bayes.
arXiv preprint arXiv:1312.6114.

Koren, Y., Bell, R., and Volinsky, C. (2009).

Matrix factorization techniques for recommender systems.
Computer, 42(8):30–37.

Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018).
Variational autoencoders for collaborative filtering.
In *Proceedings of the 27th International Conference on World Wide Web*, WWW'18, pages 689–698.

Lin, J., Pan, W., and Ming, Z. (2020).
FISSA: Fusing item similarity models with self-attention networks for sequential recommendation.
In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 130–139.

Pan, W., Yang, Q., Cai, W., Chen, Y., Zhang, Q., Peng, X., and Ming, Z. (2019).
Transfer to rank for heterogeneous one-class collaborative filtering.
ACM Transactions on Information Systems, 37(1):10:1–10:20.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009).
BPR: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI'09, pages 452–461.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014).
Stochastic backpropagation and approximate inference in deep generative models.
In *Proceedings of the 31st International Conference on Machine Learning*, ICML'14, pages 1278–1286.

Valcarce, D., Bellogín, A., Parapar, J., and Castells, P. (2018).
On the robustness and discriminative power of information retrieval metrics for top-n recommendation.
In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys'18, pages 260–268.