Collaborative Variational Autoencoder for Recommender Systems

Xiaopeng Li
HKUST-NIE Social Media Lab
The Hong Kong University of Science and Technology
xlibo@connect.ust.hk

James She
HKUST-NIE Social Media Lab
The Hong Kong University of Science and Technology
eejames@ust.hk

ABSTRACT

Modern recommender systems usually employ collaborative filtering with rating information to recommend items to users due to its successful performance. However, because of the drawbacks of collaborative-based methods such as sparsity, cold start, etc., more attention has been drawn to hybrid methods that consider both the rating and content information. Most of the previous works in this area cannot learn a good representation from content for recommendation task or consider only text modality of the content, thus their methods are very limited in current multimedia scenario. This paper proposes a Bayesian generative model called collaborative variational autoencoder (CVAE) that considers both rating and content for recommendation in multimedia scenario. The model learns deep latent representations from content data in an unsupervised manner and also learns implicit relationships between items and users from both content and rating. Unlike previous works with denoising criteria, the proposed CVAE learns a latent distribution for content in latent space instead of observation space through an inference network and can be easily extended to other multimedia modalities other than text. Experiments show that CVAE is able to significantly outperform the state-of-the-art recommendation methods with more robust performance.

CCS CONCEPTS

•**Human-centered computing** →**Collaborative filtering; Social recommendation;** •**Theory of computation** →*Bayesian analysis; Social networks;*

KEYWORDS

Recommender systems; deep learning; generative models; Bayesian; variational inference; autoencoder

1 INTRODUCTION

With the rapid growth and high prevalence of Internet services and applications, people have access to large amounts of online multimedia content, such as movies, music, news and articles. While this growth has allowed users to be able to consume a huge number of resources with only one click, it has also made it more difficult for users to find information relevant to their interests. For example, users might be not aware of the existance of interesting movies they would like and researchers might find it difficult to search for important scientific articles related to their area of research. Therefore, recommender systems are becoming increasingly important to attract users, and make effective use of the information available. An application example of recommender systems is shown in Fig. 1. Generally, in recommendation applications, there are two types of information available: the rating and the item content, e.g., the posters of the movies or the plot descriptions. Existing methods for recommender systems can be roughly categorized into three classes [1]: content-based methods, collaborative-based methods, and hybrid methods. Content-based methods [12, 15, 17] make use of user profiles or item descriptions and the user will be recommended items similar to those the user has liked in the past. Collaborative-based methods [3, 21, 22] make use of usage or history data, such as user ratings on items, without using item content information, and the user will be recommended items that people with similar tastes and preferences have liked in the past. Collaborative-based methods generally achieve a better recommendation performance than content-based methods. Nevertheless, collaborative-based methods do have their limitations. The recommendation performance often drops significantly when the ratings are very sparse; moreover, they cannot be used for recommending new items that have not received any ratings from users, which is so-called cold-start problem. Consequently, hybrid methods [2, 14, 27] that use both content and collaborative information seek to get the best of both worlds.

Based on how tightly the rating information and auxilliary information are integrated, the hybrid methods may be further divided into two sub-categories: loosely coupled and tightly coupled methods [29]. Loosely coupled methods include the heuristic methods that implement separate collaborative and content-based systems and then combine the outputs obtained from individual recommender systems into final recommendations using a linear combination [6] or a voting scheme [18]. Yet, over the past years, researchers have been seeking to develop a unified model to incorporate both collaborative and content information in a tightly coupled way. Latent factor models have been proposed as such tightly coupled methods like [2, 14, 24], where auxilliary information is processed and regarded as a feature for the collaborative methods. However, all these methods assume that the features are good representations of the content considered, which is usually not the case. Even if the features are good representations of the content in general, they are not necessarily good representations for the recommendation tasks. Thus, the improvement often has to rely on a manual and tedious feature engineering process. Especially if the content of the items is multimedia content, e.g. texts

Rating

Content

Figure 1: An example of recommendation application. Two types of information could exist: the rating and the item content, e.g., posters of movies or plot descriptions.

and images, a good representation for the recommendation tasks is hard to engineer but very crucial. Collaborative topic regression (CTR) [27] is a recently proposed method that explicitly links the learning of content to the recommendation task. It is a probabilistic graphical model that seamlessly integrates latent Dirichlet allocation (LDA) [4] and probabilistic matrix factorization (PMF) [22], and it produces promising and interpretable results. However, the representation capability of the model is limited to the topic model and the latent representation learned is often not effective enough especially when the auxiliary information is very sparse.

On the other hand, deep learning models have recently shown great potential for learning effective representations and achieved state-of-the-art performance in computer vision [13], natural language processing applications, etc. Though representation learning is appealing, few attempts have been made to develop deep learning models for recommendation. This is due partially to the fact that deep learning models, like many machine learning models, assume i.i.d. inputs. Thus it is difficult to develop deep learning models to capture and learn the implicit relationship between items (and users), which is, on the contrary, the strengh of probabilistic graphical models [10, 16]. This calls for the integration of Bayesian graphical models and deep learning models to benefit from the best of both worlds. [7, 23] use restricted Boltzmann machines instead of the conventional matrix factorization to perform collaborative filtering (CF). Although these models involve both deep learning models and CF, they actually belong to collaborative-based methods. There have also been some models that use a convolutional neural network (CNN) or deep belief network (DBN) for content-based recommendation [25, 31]. However, they use the latent factors learned through matrix factorization (MF) as the ground truth, instead of jointly learning representations and latent factors. Thus, it suffers greatly when the ratings are sparse and MF fails. Very recently, collaborative deep learning (CDL) [29] and collaborative recurrent autoencoder [30] have been proposed for joint learning a stacked denoising autoencoder (SDAE) [26] (or denoising recurrent autoencoder) and collaborative filtering, and they shows promising performance. Conceptually, both of the models try to learn a representation from content through some denoising criteria, either

feedforwardly or recurrently. They first corrupt the input by masking out some parts or replacing them with wildcards, and then use neural networks to reconstruct the original input. The responses of the bottleneck layer are thus regarded as features to feed in the CTR model, and the neural network is optimized with additional finetuning. However, denoising autoencoders (DAEs) have in fact no Bayesian nature and the denoising scheme of the DAEs is in fact not from a probabilistic perspective but rather frequentist perspective. Thus, these models are difficult for Bayesian inference or require high computational cost. Furthermore, corrupting the input in observation space requires data specific corruption schemes, for example a corruption scheme with text content might not be a good corruption scheme for image content. The fixed noise level also degrades the robustness of representation learning.

To address above challenges, we propose a Bayesian deep generative model called collaborative variational autoencoder (CVAE) to jointly model the generation of content and the rating information in a collaborative filtering setting. The model learns deep latent representations from content data in an unsupervised manner and also learns implicit relationships between items and users from both content and rating. Unlike CTR, with the representation power of deep neural network, the CVAE model learns more effective representations for content than topic models. Unlike denoising autoencoders [26, 29], the CVAE model does not corrupt the input, but rather seeks for a probabilistic latent variable model for the content. It infers the stochastic distribution of the latent variable through an inference network, instead of point estimates, and it leads to more robust performance. Furthermore, because of the stochasticity in latent space instead of observation space, the CVAE model is not limited to the content data type. Different multimedia modalities, e.g. images and texts, are unified in the framework due to no data-specific corruption schemes in observation space. Note that although CVAE is presented with feedforward neural networks as the inference and generation networks, CVAE is also suitable for other deep learning models such as convolutional neural networks [11] and recurrent neural networks [8], depending on the data type of the content. The main contribution of this paper is summarized as follows:

- By formulating the recommendation problem in a probabilistic generative model with neural networks, the proposed CVAE model can simultaneously learn an effective latent representation for content and implicit relationships between items and users for recommendation tasks.
- By stochastically inferring the latent variable distribution in latent space instead of observation space, to the best of our knowledge, the proposed CVAE model is the first Bayesian generative model that provides a unified framework for recommendation with multimedia content.
- Unlike previous deep learning models, the proposed CVAE model is inherently a Bayesian probabilistic model. Besides maximum a posteriori (MAP) estimates, effecient variational inference is derived with Stochastic Gradient Variational Bayes, leading to efficient Bayesian learning with back-propagation.
- Experiments with two real-world datasets show that the proposed CVAE model significantly outperforms the state-of-the-art methods for recommendation with content.
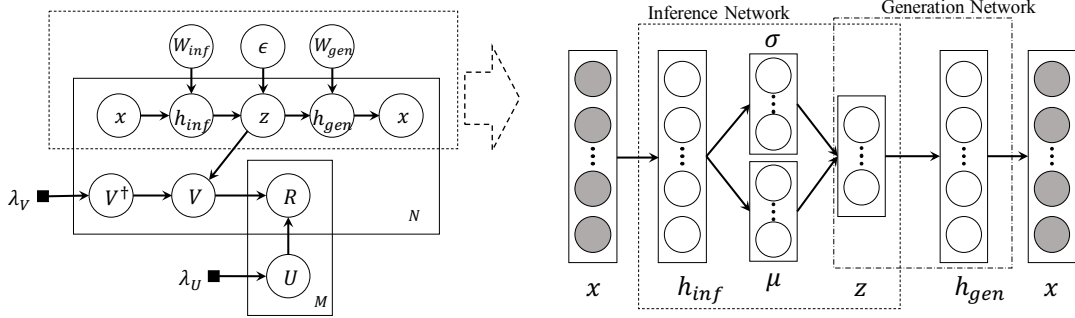
Figure 2: On the left is the proposed Collaborative Variational Autoencoder (CVAE); on the right is the zoom-in of the inference network and generation network in CVAE.

2 THE PROPOSED COLLABORATIVE VARIATIONAL AUTOENCODER

In this section, we describe the proposed collaborative variational autoencoder model (CVAE) for recommendation with content. CVAE, as shown in Fig. 2, is inherently a generative latent variable model, where the contents of the items are generated by their latent content variables and the ratings of the items by the users are generated through latent item variables. Latent item variables embed both content information through latent content variables and collaborative information through latent collaborative variables, bridging hybrid information together with deep architecture.

2.1 Collaborative Variational Autoencoder

In this paper, we represent users and items in a shared latent low-dimensional space of dimension K, where user i is represented by a latent variable $u_i \in \mathbb{R}^K$ and item j is represented by a latent variable $v_j \in \mathbb{R}^K$. Like probabilistic matrix factorization (PMF), we put Normal distribution as the prior. The latent variable for each user i is drawn from

$$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K).$$

As with PMF, the collaborative information is important for predicting the ratings. For each item j, the collaborative information is embedded in the collaborative latent variable and is drawn from Normal distribution,

$$v_j^\dagger \sim \mathcal{N}(0, \lambda_v^{-1} I_K).$$

Traditional PMF only considers collaborative information for predicting rating and does not use the content of the items. The proposed CVAE model constructs a generative latent variable model for the content and assigns a latent content variable z_j to each item j. The content of the item x_j is generated from its latent variable z_j through generation neural network parameterized by θ:

$$x_j \sim p_\theta(x_j|z_j).$$

The generation network is illustrated in Fig. 2, where, given the latent variable z, x is generated through a multi-layer perception network (MLP). For example, if x is binary data, it can be generated from a multivariate Bernoulli distribution parameterized by the output of the generation network:

$$p_\theta(x|z) = \text{Ber}(f_\theta(z)).$$

Or if x is real-value data, it can be generated from Normal distribution $\mathcal{N}(f_\theta(z), \lambda^{-1} I)$. The generation process of the content through the generation network is defined as follows:

(1) For each layer l of the generation network
    (a) For each column n of the weight matrix W_l, draw

$$W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l}).$$

    (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
    (c) For each row j of h_l, draw

$$h_{l,j*} \sim \mathcal{N}(\sigma(h_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_K).$$

(2) For binary data, draw

$$x_j \sim \text{Ber}(\sigma(h_L W_{L+1} + b_{L+1}));$$

for real-value data, draw

$$x_j \sim \mathcal{N}(h_L W_{L+1} + b_{L+1}, \lambda_x^{-1} I).$$

Above, λ_w, λ_s and λ_x are hyperparameters. And generally, λ_s is taken to infinity so that normal neural network can be employed for computational efficiency. Beyond MLP, a deconvolutional network [19, 20] can be applied to model the generation of images, etc.

The prior distribution of the content variable is chosen to be a unit Normal distribution:

$$z_j \sim \mathcal{N}(0, I_K).$$

The item latent variables thus are composed with both the collaborative latent variable and the latent content variable:

$$v_j = v_j^\dagger + z_j.$$

The rating is thus drawn from the Normal distribution centered at the inner product between their latent representations,

$$R_{ij} \sim \mathcal{N}(u_i^T v_j, C_{ij}^{-1}),$$

where C_{ij} is the precision parameter. The precision parameter C_{ij} serves as confidence for R_{ij} similar to that in CTR [27] ($C_{ij} = a$ if $R_{ij} = 1$ and $C_{ij} = b$ otherwise) due to the fact that $R_{ij} = 0$ means the user i is either not interested in item j or not aware of it.

With the generative model constructed, the joint probability of both observations and latent variables is given by

$$p(R, X, U, V, Z) = \prod_{i,j} p(R_{ij}|u_i, v_j) p(u_i) p(v_j|z_j) p(x_j|z_j) p(z_j)$$

The inference of the model is not only apparently intractable but also difficult to perform normal variational inference, especially since $p(x_j|z_j)$ is determined by the generation network with non-linear units. To see that, we can use variational inference to approximate the posterior distribution with mean-field approximation:

$$q(U, V, Z) = \prod_{i=1}^{M} q(u_i|\theta_{u_i}) \prod_{j=1}^{N} q(v_j|\theta_{v_j})q(z_j|\theta_{z_j})$$

However, the common mean-field approach requires an analytical solution of expectations with respect to the approximate posterior, which is not the case for the generation network since no conjugate probability distribution can be found. Instead, we can use the Stochastic Gradient Variational Bayes (SGVB) estimator for efficient approximate posterior inference of the latent content variable z by introducing an inference network parameterized by ϕ. The inference network is also an MLP corresponding to the MLP in the generation network, or convolutional network corresponding to the deconvolutional network. The variational distribution is thus chosen to be conjugate to the prior Normal distribution of z_j and is parameterized by ϕ and x_j, in replacement of θ_{z_j}, leading to per datapoint variational distributions with shared network parameters:

$$q(z_j|\theta_{z_j}) = q_\phi(z_j|x_j) = \mathcal{N}(\mu_\phi(x_j), \text{diag}(\sigma_\phi^2(x_j))).$$

With the reparameterization trick similar to [9], it is straightforward to get a differentiable unbiased estimator of the lower bound and optimize it using standard stochastic gradient ascent techniques. The inference process of the latent content variable z through the inference network is defined as follows:

(1) For each layer l of the inference network
  (a) For each column n of the weight matrix W_l, draw

$$W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l}).$$

  (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
  (c) For each row j of h_l, draw

$$h_{l,j*} \sim \mathcal{N}(\sigma(h_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l}).$$

(2) For each item j
  (a) Draw latent mean and covariance vector

$$\mu_j \sim \mathcal{N}(h_L W_\mu + b_\mu, \lambda_s^{-1} I_K)$$
$$\log \sigma_j^2 \sim \mathcal{N}(h_L W_\sigma + b_\sigma, \lambda_s^{-1} I_K).$$

  (b) Draw latent content vector

$$z_j \sim \mathcal{N}(\mu_j, \text{diag}(\sigma_j)).$$

The evidence lower bound (ELBO) can be obtained by

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(R, X, U, V, Z) - \log q(U, V, Z)]$$
$$= \mathbb{E}_q[\log p(R|U, V) + \log p(U) + \log p(V|Z) + \log p(X|Z)]$$
$$- \mathbb{KL}(q_\phi(Z|X)\|p(Z))$$

In terms of $q_\phi(z|x)$, isolating the terms related to z, we have

$$\mathcal{L}(\theta, \phi; x_j) = \mathbb{E}_{q_\phi(z_j|x_j)}[\log p_\theta(x_j|z_j)] + \mathbb{E}_{q_\phi(z_j|x_j)}[\log p(v_j|z_j)]$$
$$- \mathbb{KL}(q_\phi(z_j|x_j)\|p(z_j)) + \text{const}.$$

It is problematic to compute the two expectations above analytically, as well as the gradient of $\mathcal{L}$ with respect to ϕ. Instead, we can form Monte Carlo estimates of the two expectations by drawing

(a)

(b)
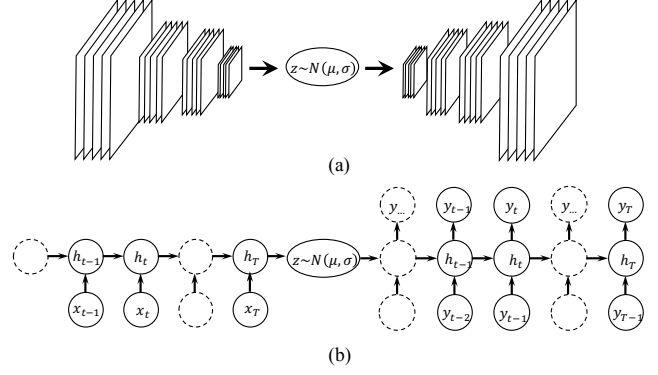
Figure 3: Extensions of inference networks and generation networks under the framework of CVAE for (a) image data and (b) sequential data for recommender systems.

samples from the variational distribution: $z_j \sim q_\phi(z_j|x_j)$. With the reparameterization trick[9], we can instead draw samples from $\epsilon \sim \mathcal{N}(0, I)$ and form the samples of z_j by $z_j = \mu_j + \sigma_j \odot \epsilon$. Thus, the above objective can be estimated using the SGVB estimator:

$$\mathcal{L}(\theta, \phi; x_j) \simeq -\mathbb{KL}(q_\phi(z_j|x_j)\|p(z_j)) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x_j|z_j^{(l)})$$
$$+ \log p(v_j|z_j^{(l)}) + \text{const}.$$

where $z_j^{(l)} = \mu_j + \sigma_j \odot \epsilon^{(l)}$ and $\epsilon \sim \mathcal{N}(0, I)$.

Another interpretation of the proposed CVAE model is from the point of view of denoising. The inference network produces the posterior variational distribution for the latent content variable z. This can also be viewed as as corrupting the latent content vector with Gaussian noise. Unlike the stacked denoising autoencoder (SDAE) as proposed in [26, 29], where the input observations are first corrupted with various noise (masking noise, Gaussian noise, salt-and-peper noise, etc), the corruption of data in latent space avoids the diverse interpretation and corruption scheme of the observation space and provides a unified way of corruption. Also, the noise level is automatically learned through the inference network, instead of manually tuned like in SDAE. This leads to more robust and systematic learning of representation regardless of the data type it is trying to model.

For completeness, Fig. 3 shows the extensions of inference networks and generation networks under the framework of CVAE for other types of content data associated with items. For image data, a convolutional network and deconvolutional network pair can be used for inference network and generation network [19, 20]. For sequential data, such as text data in sequential form instead of bag-of-words representation, a recurrent neural network pair can be used [5]. Due to the stochasticity in latent space instead of the observation space, content data with different modalities are unified in the framework of CVAE for recommender systems.

2.2 Maximum A Posteriori Estimates

The MAP estimation can be performed by only considerring the variational distribution of $q_\phi(Z|X)$ and maximizing the objective

with respect to $\{u_i\}$ and $\{v_j\}$ using block coordinate ascent. The objective thus becomes

$$
\begin{aligned}
\mathcal{L}^{\text{MAP}}(U, V, \theta, \phi) = & -\sum_{i,j} \frac{C_{ij}}{2}(R_{ij} - u_i^T v_j)^2 - \frac{\lambda_u}{2}\sum_i \|u_i\|_2^2 \\
& - \frac{\lambda_v}{2}\sum_j \mathbb{E}_{q_\phi(Z|X)}\|v_j - z_j\|_2^2 \\
& + \mathbb{E}_{q_\phi(Z|X)}\log p(X|Z) - \mathbb{KL}(q_\phi(Z|X)\|p(Z)) \\
& - \frac{\lambda_w}{2}\sum_l (\|W_l\|_F^2 + \|b_l\|_2^2)
\end{aligned}
\tag{1}
$$

The block coordinate ascent for $\{u_i\}$ and $\{v_j\}$ thus becomes

$$
u_i \leftarrow (VC_iV^T + \lambda_U I_K)^{-1}VC_iR_i
$$
$$
v_j \leftarrow (UC_jU^T + \lambda_V I_K)^{-1}(UC_iR_i + \lambda_v \mathbb{E}_{q_{\phi_Z}}[z_j])
$$

where it should be noted that $\mathbb{E}_{q_{\phi_Z}}[z_j]$ equals to μ_j produced by the inference network. And given U and V, the gradient with respect to μ and σ of z is

$$
\begin{aligned}
\nabla_{\mu_j}\mathcal{L}^{\text{MAP}}(\theta, \phi; x_j) \simeq & -\mu_j + \frac{1}{L}\sum_{l=1}^{L}\lambda_v(v_j - z_j^{(l)}) \\
& + \nabla_{z_j^{(l)}}\log p_\theta(x_j|z_j^{(l)})
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\sigma_j}\mathcal{L}^{\text{MAP}}(\theta, \phi; x_j) \simeq & \frac{1}{\sigma_j} - \sigma_j + \frac{1}{L}\sum_{l=1}^{L}[\lambda_v(v_j - z_j^{(l)}) \\
& + \nabla_{z_j^{(l)}}\log p_\theta(x_j|z_j^{(l)})] \odot \epsilon^{(l)}
\end{aligned}
$$

The last term $\log p_\theta(x_j|z_j^{(l)})$ depends on what the data type of x_j is. For binary data, $p_\theta(x|z)$ is a Bernoulli distribution. For categorical data, $p_\theta(x|z)$ could be a Categorical distribution. And for real-value data $p_\theta(x|z)$ could be a Normal distribution. Thus, the gradient of the generation network can be evaluated through backpropagation and so too the gradient of weights of the inference network. Furthermore, the number of samples L per datapoint can be set to 1 as long as the minibatch size M is large enough, e.g. $M = 128$. The weights of the inference network and generation network thus can be performed through backpropagation in a conventional manner.

2.3 Bayesian Inference

Previous works [23, 25, 29, 30] that also employ deep networks are difficult to conduct Bayesian inference, or they have to resort to computationally intensive Markov Chain Monte Carlo (MCMC). However, with the Bayesian nature of the proposed model, it is straightforward to perform efficient variational inference. The complete conditional of u_i is given by

$$
p(u_i|R, V, X, Z) = \mathcal{N}(\mu_{u_in}, \lambda_{u_in}^{-1}I_K)
$$

where

$$
\Lambda_{u_in} = \sum_{j=1}^{N} C_{ij}v_jv_j^T + \lambda_u I_K
$$

$$
\mu_{u_in} = \Lambda_{u_in}^{-1}(\sum_{j=1}^{N} C_{ij}R_{ij}v_j)
$$

The complete conditional of v_j is given by

$$
p(v_j|R, U, X, Z) = \mathcal{N}(\mu_{v_jn}, \lambda_{v_jn}^{-1}I_K)
$$

where

$$
\Lambda_{v_jn} = \sum_{i=1}^{M} C_{ij}u_iu_i^T + \lambda_v I_K
$$

$$
\mu_{v_jn} = \Lambda_{v_jn}^{-1}(\sum_{i=1}^{M} C_{ij}R_{ij}u_i + \lambda_v z_j)
$$

Taking the derivative of $\mathcal{L}$ with respect to θ_{u_i} and setting the gradient to zero, we have

$$
\mu_{u_i} \leftarrow (\mathbb{E}_{q_{\theta_V}}[VC_iV^T] + \lambda_U I_K)^{-1}(\mathbb{E}_{q_{\theta_V}}[V]C_iR_i)
$$

$$
\Lambda_{u_i} \leftarrow (\mathbb{E}_{q_{\theta_V}}[VC_iV^T] + \lambda_U I_K)
$$

where $\mathbb{E}_{q_{\theta_V}}[VC_iV^T] = \mathbb{E}_{q_{\theta_V}}[V]C_i\mathbb{E}_{q_{\theta_V}}[V]^T + \sum_j C_{ij}\Lambda_{v_j}^{-1}$. Likewise, for θ_{v_j}, we have

$$
\mu_{v_j} \leftarrow (\mathbb{E}_{q_{\theta_U}}[UC_jU^T] + \lambda_V I_K)^{-1}(\mathbb{E}_{q_{\theta_U}}[U]C_iR_i + \lambda_v\mathbb{E}_{q_{\phi_Z}}[z_j])
$$

$$
\Lambda_{v_j} \leftarrow (\mathbb{E}_{q_{\theta_U}}[UC_iU^T] + \lambda_V I_K)
$$

where $\mathbb{E}_{q_{\theta_U}}[UC_jU^T] = \mathbb{E}_{q_{\theta_U}}[U]C_i\mathbb{E}_{q_{\theta_U}}[U]^T + \sum_i C_{ij}\Lambda_{u_j}^{-1}$. Compared with the MAP estimates, it can be seen that the precision matrix further gauges our belief of the updates by introducing an additional term. The gradient of $\mathcal{L}$ with respect to μ and σ thus can be obtained:

$$
\begin{aligned}
\nabla_{\mu_j}\mathcal{L}(\theta, \phi; x_j) \simeq & -\mu_j + \frac{1}{L}\sum_{l=1}^{L}\Lambda_{v_j}(\mathbb{E}_{q_{\theta_V}}[v_j] - z_j^{(l)}) \\
& + \nabla_{z_j^{(l)}}\log p_\theta(x_j|z_j^{(l)})
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\sigma_j}\mathcal{L}(\theta, \phi; x_j) \simeq & \frac{1}{\sigma_j} - \sigma_j + \frac{1}{L}\sum_{l=1}^{L}[\Lambda_{v_j}(\mathbb{E}_{q_{\theta_V}}[v_j] - z_j^{(l)}) \\
& + \nabla_{z_j^{(l)}}\log p_\theta(x_j|z_j^{(l)})] \odot \epsilon^{(l)}.
\end{aligned}
$$

2.4 Prediction

Let D be the observed data. After all the parameters U, V (or μ, Λ), and the weights of the inference network and generation network are learned, the predictions can be made by

$$
\mathbb{E}[R_{ij}|D] = \mathbb{E}[u_i|D]^T(\mathbb{E}[v_j^\dagger|D] + \mathbb{E}[z_j|D]).
$$

For point estimation, the prediction can simplified as

$$
R_{ij}^* = u_i^T(v_j^\dagger + \mathbb{E}[z_j])
$$

where $\mathbb{E}[z_j] = \mu_j$, the mean vector obtained through the inference network for each item j. The items that have never been seen before

will have no $v_j^\dagger$ term, but the $\mathbb{E}[z_j]$ can be inferred through the content. In this way, both the sparsity problem and cold start problem are alleviated, leading to robust recommendation performance.

3 EXPERIMENTS

In this section, we present both quantitative and qualitative experiment results on two real-word datasets to demonstrate the effectiveness of the proposed CVAE model for recommender systems[1].

3.1 Datasets

The two datasets are users and their libraries of articles with different scales and degrees of sparsity obtained from CiteULike[2], where users can create their own collections of articles. The first dataset, *citeulike-a*, is collected by [27], and there are 5,551 users and 16,980 articles with 204,986 observed user-item pairs. Users with fewer than 10 articles are not included in the dataset. The sparsity of the rating matrix is 99.78% (only 0.22% of the rating matrix have one entries). The second dataset, *citeulike-t*, is independently collected by [28] and is even larger and sparser. They manually selected 273 seed tags and collected all articles with at least one of the tags. There are 7,947 users and 25,975 articles with 134,860 observed user-item pairs. The sparsity of the rating matrix is 99.93% (only 0.07% of the rating matrix have one entries). Users with fewer than 3 artiles are not included in the dataset. Each article in the two datasets has a title and abstract. The content information of the articles is the concatenation of the titles and abstracts. We follow the same procedure as that in [27] to preprocess the text content information. After removing stop words, the vocabulary for each dataset is selected according to the tf-idf value of each word. The *citeulike-a* has a vocabulary size of 8,000 and the *citeulike-t* has a vocabulary size of 20,000. Each article is represented with a bag-of-words histogram vector and all the content vectors are then normalized over the maximum occurrences of each word in all articles.

3.2 Evaluation Scheme

For the recommendation task, we randomly select P items in the user's collection for each user to form the training set and use the rest of the items as the testing set. For the training, we consider both sparse setting ($P = 1$) and dense setting ($P = 10$) for each dataset. For the testing, each user is presented with M articles sorted by their predicted rating and evaluated based on which of these articles are actually in each users' collection. For each value of P, the evaluation is conducted five times with different random splits and the average performance is reported.

As in [27, 29], we use the recall as the evaluation metric since the zero ratings are in the form of implicit feedback. They may indicate that a user does not like an article or the user simply is not aware of it. This makes it difficult to accurately compute precision. However, since ratings of one explicitly mean true positive, we focus on recall. For each user, the definition of *recall@M* is

$$\text{recall@}M = \frac{\text{number of items that the user likes among the top } M}{\text{total number of items the user likes}}.$$

The recall for the entire system can be summarized using the average recall over all users.

3.3 Baselines and Experimental Settings

As already demonstrated in previous works [27, 29], hybrid recommendation with content performs significantly better than collaborative based methods, only hybrid models are used for comparison. The models included in our comparison are listed as follows:

- **CTR:** Collaborative Topic Regression [27] is a model performing latent dirichlet allocation (LDA) on the content and collaborative filtering on the rating with tightly coupled modeling.
- **DeepMusic:** DeepMusic [25] is a feedforward neural network model for music recommendation, where the objective is a linear regression and the latent factor vectors obtained by applying weighted matrix factorization (WMF) to the training rating data are used as groundtruth. We use the loosely coupled variant as our baseline.
- **CDL:** Collaborative Deep Learning [29] is a probabilistic feedforward model for joint learning of stacked denoising autoencoder (SDAE) and collaborative filtering. CDL is a very strong baseline and achieves the best performance among our baseline methods.
- **CVAE:** Collaborative Variational Autoencoder is our proposed model. The MAP estimates are conducted for fair comparison. It is a generative latent variable model that jointly models the generation of content and rating and uses variational Bayes with inference network for variational inference.

In the experiments, we first use a validation set to find the optimal hyperparameters for CTR, DeepMusic and CDL. For CTR, we find that it can achieve good performance when $a = 1$, $b = 0.01$, $\lambda_u = 0.1$, $\lambda_v = 1$, and $K = 50$, where a and b are the confidence value for C_{ij} under different ratings. The model is first pretrained with LDA to get the initial topic proportions and CTR is performed to jointly learn U, V and topic proportions iteratively. For DeepMusic, a two convolutional layers architecture is used. We tried our best to tune other hyperparameters. For CDL, we also set $a = 1$, $b = 0.01$, $K = 50$ and find that the best performance is achieved with $\lambda_u = 1$, $\lambda_v = 10$, $\lambda_n = 1000$ and $\lambda_w = 0.0001$. A two-layer SDAE network architecture is adopted to be the same as [29]. The weights of the network are first pretrained in a plain SDAE manner with the content of items and then fed into CDL for finetuning for another 200 epochs. The noise is set to be masking noise with a noise level of 0.3. For CVAE, the model with MAP estimation is used for fairness. The parameters of $a = 1$, $b = 0.01$, $K = 50$ are the same. The other hyperparameters are found through cross-validation on the validation set. The model is also pretrained in plain VAE manner to first learn initial starting points for the network weights. In terms of network architecture, it is found that the best performance is achieved by tying the weights of the inference network and the generation network (the weights of the inference network are set to be the transpose of the generation network). It is probably because by tying their weights together, it avoids many bad local minima, leading to more robust representation learning in the generation of the content part. This can also be verified through the result that
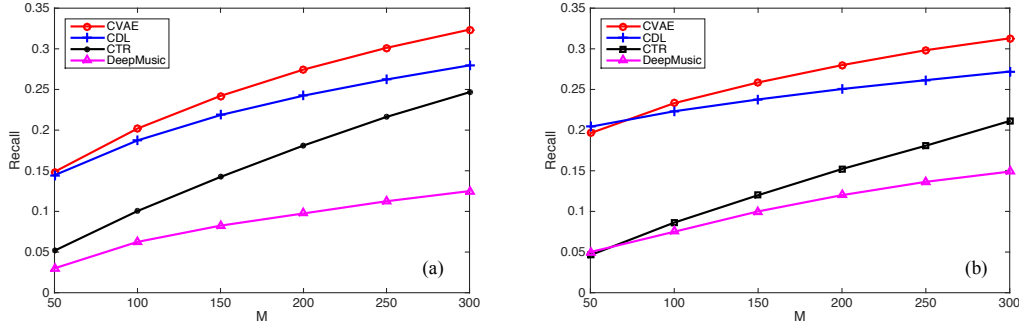
---

[1]Code is available at http://eelxpeng.github.io/research/
[2]http://www.citeulike.org

Figure 4: Performance comparison of CVAE, CDL, CTR and DeepMusic based on recall in the sparse setting for dataset (a) *citeulike-a* **and (b)** *citeulike-t.*
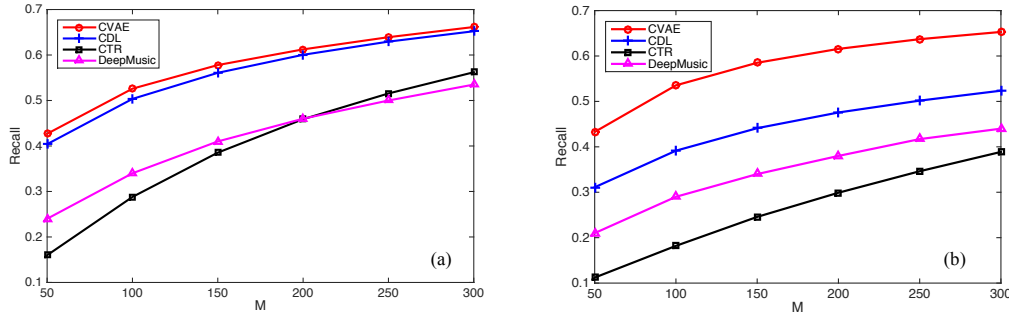

Figure 5: Performance comparison of CVAE, CDL, CTR and DeepMusic based on recall in the dense setting for dataset (a) *citeulike-a* **and (b)** *citeulike-t.*

tying weights together leads to lower reconstructure error. Both the inference network and the generation network are chosen to be a two-layer network architecture ('200-100' for inference network and '100-200' for generation network) with sigmoid activation function.
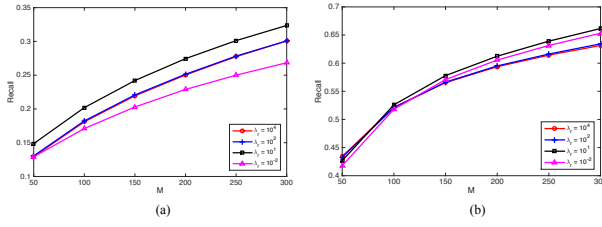
3.4   Performance Comparison

Fig. 4 and 5 show the results that compare CTR, DeepMusic, CDL and CVAE using the two datasets under both the sparse ($P = 1$) and dense ($P = 10$) settings. As it can be seen, CDL is a very strong baseline and outperforms the other baselines by a large margin. Compared with CTR, it can be seen that the deep learning structure of CDL learns better representations than topic modeling, leading to better performance. Although DeepMusic is also a deep learning model, it suffers badly from overfitting when the rating matrix is extremely sparse ($P = 1$) and achieves a comparable performance to CTR when the rating matrix is dense ($P = 10$). This is because Deep-Music uses the latent vector generated by WMF as the regression groundtruth, thus its performance highly relies on the WMF which fails when the rating matrix is sparse. By modeling the generation of content and rating simultaneously in a probabilistic generative latent variable model, CVAE is able to outperform all the baselines by a margin of 4.4%~20% (a relative boost of 15.7%~160%) in the sparse setting and 0.9%~10.8% (a relative boost of 1.4%~37.4%) in

the dense setting for *citeulike-a*. For *citeulike-t*, since the dataset is even sparser, the improvement is more significant, 4.1%~16.39% (a relative boost of 15.1%~110%) in the sparse setting and 12.9%~26.4% (a relative boost of 24.7%~68.0%) in the dense setting. To focus more specifically on the comparison of CDL and CVAE, we can see that although both CDL and CVAE use deep learning models, the proposed CVAE achieves better and more robust recommendation, especially for larger M. This is because CVAE is inherent a Bayesian generative model and the distribution of latent content variable z is estimated, instead of point estimation. Thus it will lead to more robust performance. On the other hand, CDL is using some denoising criteria in observation space with a fixed noise level to learn a point estimate of latent representation, thus it is still easy to overfit the data.

To investigate the difference between focusing more on content and focusing more on ratings, we can add a factor of λ_r to the third line of Eq. 1 to adjust the penalty of reconstruction error with respect to rating-related terms. Fig. 6 shows the results of CVAE for different values of λ_r in *citeulike-a* under the sparse and dense setting while fixing $\lambda_u = 0.1$ and $\lambda_v = 10$. When λ_r is very small, e.g., 0.01, the penalty of reconstruction of the content is greatly reduced. It leads to degraded performance of representation learning of the content, thus the performance of the recommendation. While when λ_r is large, e.g., 100, the CVAE model would tend to degenerate into two independent models, one

Figure 6: Performance comparison of CVAE for different values of λ_r based on recall for dataset *citeulike-a* in the sparse setting (a) and dense setting (b).
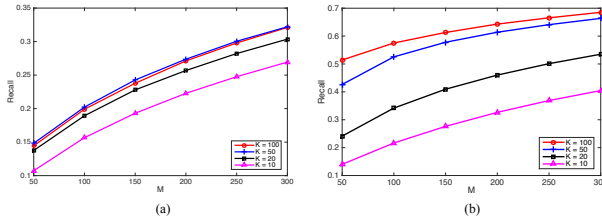
Figure 7: Performance comparison of CVAE for different values of K based on recall for dataset *citeulike-a* in the sparse setting (a) and dense setting (b).

for representation learning of the content, the other for probablistic matrix factorization. Yet, it can be seen that, $\lambda_r = 100$ and $\lambda_r = 10000$ have similar performance, showing that CVAE model is rather robust for preventing separate learning. It is because the CVAE is a generative model that learns the latent distribution, instead of point estimate. The Bayesian nature of the CVAE model makes it able to model the data better, leading to more robust performance. By comparing the influence of λ_r in the sparse setting and dense setting, it can be seen that joint learning content and rating is more important when dealing with very sparse data, and crucial for solving the cold-start problem.

Fig. 7 shows the results of CVAE for different values of the number of latent factors K. The influce of K depends on two parts, the latent variable for the representation of content and the latent variable for the matrix factorization model. It can be seen that when K is very small, e.g., 10, the CVAE is unable to learn a good representation from the content, thus it leads to degraded performance. On the other hand, when K is large enough, the influence of K is trivial since the representation capability is enough for the content it is trying to model. By comparing the influence of K in the sparse setting and dense setting, we can also see that the larger K has more influence in the dense setting. This is mainly because denser ratings offer more guidance for inference network to make variational inference, thus they need a larger representation capability to learn.

3.5 Qualitative Results

In order to gain better insight into CVAE, we train CVAE and CDL in the sparse setting ($P = 1$) with the dataset *citeulike-a* and use

them to recommend articles for two example users. The corresponding rated articles for the target users and the top 10 recommended articles are shown in Table 1. As we can see, CVAE successfully identified user I as a researcher working on information retrieval with an interest in using language model. Consequently, CVAE achieves a high precision of 90% by focusing its recommendations on articles about language model based information retrieval, etc. On the other hand, the topics of articles recommended by CDL covers some purely natural language processing (Article 2, 5), and pure information retrieval (Article 6, 7, 8), instead of the interaction of both. For example, Article 6 recommended by CDL is content based 'image retrieval', which apparently has nothing to do with language models for information retrieval. A similar situation happens for user II. From the rated article, CVAE identifies the user as someone who is doing an overview of the area and is more interested in general methods and evaluation metrics for recommendation. Thus, the articles recommended by CVAE focus on some simpler methods and evaluation articles. Whereas, CDL recommends many articles focusing on some in-depth models for recommender systems, such as content-based or hybrid models (Article 1-4), which contradicts the intention of the user.

From these two users, we can see that the generative nature of the proposed CVAE model can capture the key points of articles and user preferences more accurately, due to its modeling of latent random variable distribution. From the other point of view, unlike CDL, which corrupts the input with fixed noise level at observation space, CVAE is more robust since it adaptively learns the stochastic noise to corrupt the data in latent space. Thus, generally CVAE achieves better recommendation performance.

4 CONCLUSIONS

This paper proposes the collaborative variational autoencoder that can jointly model the generation of item content while extracting the implicit relationships between items and users collaboratively. It is a Bayesian probabilistic generative model that unifies the collaborative and content information through stochastic deep learning models and graphical models, leading to robust recommendation performance. By its Bayesian nature, efficient variational inference is derived with stochastic gradient variational Bayes. To the best of our knowledge, CVAE is the first model that unifies the different modalities of multimedia for recommendation due to its inference of stochastic distribution in latent space instead of observation space. Experiments have shown that the proposed CVAE can significantly outperform the state-of-the-art methods for recommendation with content with more robust performance.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.
[2] Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 19–28.

Table 1: Qualitative comparison between CVAE and CDL.

the rated article	Information retrieval as statistical translation	
	user I (CVAE)	in user's lib?
top 10 articles	**1. Linear discriminant model for information retrieval**	**yes**
	2. Probabilistic latent semantic indexing	**yes**
	3. Latent dirichlet allocation	**yes**
	4. Latent concept expansion using markov random fields	**yes**
	5. Document language models, query models, and risk minimization for information retrieval	**yes**
	6. Two-stage language models for information retrieval	**yes**
	7. Latent semantic indexing: a probabilistic analysis	no
	8. Relevance feedback for best match term weighting algorithms in information retrieval	**yes**
	9. Dependence language model for information retrieval	**yes**
	10. Probabilistic relevance models based on document and query generation	**yes**
	user I (CDL)	in user's lib?
top 10 articles	**1. Bayesian extension to the language model for ad hoc information retrieval**	**yes**
	2. PubMed related articles: a probabilistic topic-based model for content similarity	no
	3. Document language models, query models, and risk minimization for information retrieval	**yes**
	4. A probability ranking principle for interactive information retrieval	no
	5. Discriminative reranking for natural language parsing	no
	6. Relevance feedback: a power tool for interactive content-based image retrieval	no
	7. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval	no
	8. Combining approaches to information retrieval	no
	9. Language models for relevance feedback	**yes**
	10. Using language models for information retrieval	**yes**
the rated article	Methods and metrics for cold-start recommendations	
	user II (CVAE)	in user's lib?
top 10 articles	**1. Amazon.com recommendations: item-to-item collaborative filtering**	**yes**
	2. Evaluating collaborative filtering recommender systems	**yes**
	3. Empirical analysis of predictive algorithms for collaborative filtering	**yes**
	4. Item-based collaborative filtering recommendation algorithms	**yes**
	5. Trust in recommender systems	no
	6. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web	**yes**
	7. Collaborative filtering with temporal dynamics	**yes**
	8. Taxonomy-driven computation of product recommendations	no
	9. Solving the apparent diversity-accuracy dilemma of recommender systems	no
	10. Modeling relationships at multiple scales to improve accuracy of large recommender systems	**yes**
	user II (CDL)	in user's lib?
top 10 articles	1. Content-based book recommending using learning for text categorization	no
	2. Unifying collaborative and content-based filtering	no
	3. Content-boosted collaborative filtering	no
	4. TrustWalker: a random walk model for combining trust-based and item-based recommendation	no
	5. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis.	no
	6. Evaluating collaborative filtering recommender systems	**yes**
	7. AdaRank: a boosting algorithm for information retrieval	no
	8. Learning distance functions for image retrieval	no
	9. Opinion observer: analyzing and comparing opinions on the Web	yes
	10. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments	**yes**

[3] Daniel Billsus and Michael J Pazzani. 1998. Learning collaborative information filters.. In *Proceedings of the 15th International Conference on Machine Learning*, Vol. 98. 46–54.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.

[5] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CONLL*. 10–21.

[6] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, Vol. 60. Citeseer.

[7] Kostadin Georgiev and Preslav Nakov. 2013. A non-IID framework for collaborative filtering with restricted Boltzmann machines.. In *Proceedings of the 30th International Conference on Machine Learning*. 1148–1156.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 369–376.

[9] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*.

[10] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[12] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*. 331–339.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[14] Wu-Jun Li, Dit-Yan Yeung, and Zhihua Zhang. 2011. Generalized latent factor models for social network analysis. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Spain*. 1705.

[15] Xiaopeng Li, Ming Cheung, and James She. 2016. Connection Discovery using Shared Images by Gaussian Relational Topic Model. In *International Conference on Big Data*. IEEE, 931–936.

[16] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

[17] Michael Pazzani and Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning* 27, 3 (1997), 313–331.

[18] Michael J Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13, 5-6 (1999), 393–408.

[19] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems*. 2352–2360.

[20] Yunchen Pu, Xin Yuan, Andrew Stevens, Chunyuan Li, and Lawrence Carin. 2016. A deep generative deconvolutional image model. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. 741–750.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[22] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization.. In *Advances in Neural Information Processing Systems*, Vol. 1. 2–1.

[23] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltz-mann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*. ACM, 791–798.

[24] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 650–658.

[25] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*. 2643–2651.

[26] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[27] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recom-mending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 448–456.

[28] Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative topic regression with social regularization for tag recommendation.. In *International Joint Conference on Artificial Intelligence*.

[29] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[30] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *Advances in Neural Information Processing Systems*. 415–423.

[31] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music rec-ommendation using deep learning. In *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 627–636.