Personalized Employee Training Course Recommendation with Career Development Awareness

Chao Wang
Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
Baidu Talent Intelligence Center, Baidu Inc.
wdyx2012@mail.ustc.edu.cn

Hengshu Zhu*
Baidu Talent Intelligence Center, Baidu Inc.
zhuhengshu@gmail.com

Chen Zhu
Baidu Talent Intelligence Center, Baidu Inc.
zc3930155@gmail.com

Xi Zhang
College of Management and Economics, Tianjin University
jackyzhang@tju.edu.cn

Enhong Chen
Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
cheneh@ustc.edu.cn

Hui Xiong*
Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China
Baidu Talent Intelligence Center, Baidu Inc.
Business Intelligence Lab, Baidu Research
xionghui@gmail.com

ABSTRACT

As a major component of strategic talent management, learning and development (L&D) aims at improving the individual and organization performances through planning tailored training for employees to increase and improve their skills and knowledge. While many companies have developed the learning management systems (LMSs) for facilitating the online training of employees, a long-standing important issue is how to achieve personalized training recommendations with the consideration of their needs for future career development. To this end, in this paper, we propose an explainable personalized online course recommender system for enhancing employee training and development. A unique perspective of our system is to jointly model both the employees' current competencies and their career development preferences in an explainable way. Specifically, the recommender system is based on a novel end-to-end hierarchical framework, namely Demand-aware Collaborative Bayesian Variational Network (DCBVN). In DCBVN, we first extract the latent interpretable representations of the employees' competencies from their skill profiles with autoencoding variational inference based topic modeling. Then, we develop an effective demand recognition mechanism for learning the personal demands of career development for employees. In particular, all the above processes are integrated into a unified Bayesian inference view for obtaining both accurate and explainable recommendations. Finally, extensive experimental results on real-world data clearly demonstrate the effectiveness and the interpretability of DCBVN, as well as its robustness on sparse and cold-start scenarios.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

Employee training course recommendation, Recommender system, Intelligent education

ACM Reference Format:
Chao Wang, Hengshu Zhu, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2020. Personalized Employee Training Course Recommendation with Career Development Awareness. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3366423.3380236

1 INTRODUCTION

In strategic talent management, learning and development (L&D) aims at improving the individual and organization performance through planning tailored training for employees to increase and hone their skills and knowledge, which is of great importance for companies to maintain their competitive edges in the fast-pace business environments [40]. According to the research reports from the *Association for Talent Development*[1], U.S. organizations spent $1,296 per employee on L&D in 2018, with an average of 34.1 learning hours. Therefore, in recent years, more and more companies have built the learning management systems (LMSs) for facilitating the online training of employees, which can provide not only large cost savings, but also an effective way to deliver engaging development for talents due to the benefits of reach, scale, and timeliness [8].

*Hui Xiong and Hengshu Zhu are corresponding authors.

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

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380236

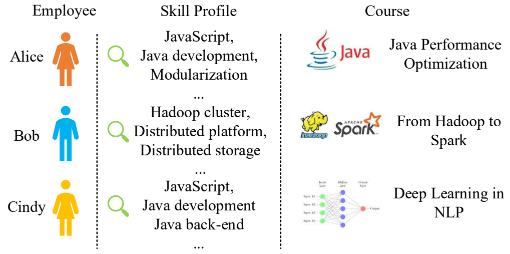
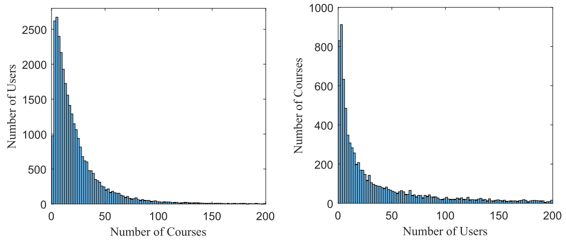


Figure 1: A motivating example of personalized training recommendations for employees.

However, along this line, a long-standing challenge is how to offer the talents with personalized training recommendations.

Different from traditional recommendation scenarios (e.g., movies or product recommendations), the learning motivation of employees heavily depends on not only their current competencies but also their goals of future career development. Figure 1 shows a motivating example of building a personalized employee training recommender system. Specifically, based on the skill profiles and historical course records, we can guess that Alice studied the course *Java Performance Optimization* due to the demand of honing existing skills, while Bob studied *From Hadoop to Spark* for increasing his current skill set. Meanwhile, although Cindy has a similar skill profile to Alice, she made a very different choice of training course, i.e., *Deep Learning in NLP*, which is far away from her existing competencies. This might be because she wanted to master skills in a new field for achieving her future career goal. Therefore, it is critically important to simultaneously consider the current competencies and the career development demands of employees for the recommendation. Also, in the educational domain, interpretable models play an important role in learning and progress, designing of better instruction, and possibly intervention to address individual and group needs [9, 14, 21]. Consequently, it is vital to provide explanations on the recommendation results. Moreover, we usually have to face extreme data sparseness and cold-start problems on the LMS. Meanwhile, the skill profiles may not completely reflect employees' skills and contain both fine-grained and coarse-level skill labels. It is very challenging to learn true competencies and the demands of employees from such noisy and sparse data.

In order to solve the above challenges, in this paper, we propose a personalized online course recommender system for enhancing employee training, which jointly models both the current competencies and the career development demands of employees. Specifically, the recommender system is based on a novelly-designed end-to-end hierarchical framework, namely *D*emand-aware *C*ollaborative *B*ayesian *V*ariational *N*etwork (DCBVN). Considering that the original skill profiles may be sparse and ambiguous, they could not be readily utilized. Thus, in DCBVN, we first extract the latent interpretable representations of employees' competencies from their skill profiles with autoencoding variational inference based topic modeling [34]. In this way, the auxiliary skill information helps us deepen the understanding of employees and hence, alleviate the sparsity and cold-start problem. Then, by exploiting the observed course records and collaborative learning behaviors, we develop an effective demand recognition mechanism for learning the personal demands of career development. Each dimension of both the latent competence and demand variable represents an interpretable skill topic in reality. Finally, the most appropriate training courses are recommended through an adapted collaborative filtering algorithm. All the above processes are integrated into a unified Bayesian collaborative filtering way to make sure both the recommendation accuracy and explainability at the same time. Extensive experimental results on real-world data clearly demonstrate the effectiveness and the interpretation power of DCBVN framework, as well as its robustness on sparse and cold-start scenarios.



(a) User perspective (b) Course perspective

Figure 2: Distributions of the number of learning records.

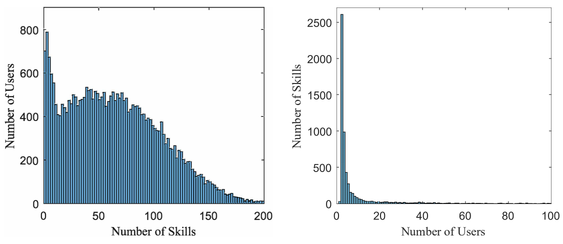
2 DATA DESCRIPTION

In this section, we will introduce the real-world dataset exploited in this paper. Specifically, our dataset *DLearner* consists of two main components, namely learning records and skill profiles of employees, which are provided by a major high tech company in China. Note that, all of the sensitive information in the dataset has been removed or anonymized for privacy prevention purpose.

Learning records. In the dataset, the learning records were collected from the LMS for employee training. Most courses are in the form of videos and slides. Different from student education, courses on the LMS usually focus on specific job skills. Thus, there is usually no clear sequential relationship (i.e., without course dependency) among the courses on the LMS. One can choose and study any course on the website of LMS without special restriction.

Specifically, the learning records were collected from May 2016 to August 2019, containing 30,662 users and 8,693 courses. Firstly, we need to exclude the noisy records that users only click the online course but not spend time learning. Consequently, only when a user had studied more than half part of the video or slides, we considered it as a valid record. In this way, we can represent the entire learning records in the form of 0/1 rating matrix, where 1 means the valid record and 0 otherwise. There are totally 714,091 valid records in the *DLearner* dataset. Therefore, we can find that learning records are quite sparse as only 0.27% of the rating matrix entries contain valid records. Figure 2 shows the distributions of the learning records from the user and course perspectives, respectively. It can be observed from Figure 2 that most courses were studied by a small number of users while the long tail effect is quite obvious in the distribution from course perspective.

Skill profiles. In the dataset, we also have a detailed skill profile for each employee, which indicates the professional job skills the employee has already mastered before online training. Specifically,



(a) User perspective (b) Skill perspective

Figure 3: Distributions of the number of different job skills.

there are totally 6,504 unique job skills in the *DLearner* dataset. Besides, the employees are classified into different groups according to their departments and job positions in our dataset. There are totally 557 departments and 5 job positions (i.e., Technology, Product, User Interface, Manager and Others). To show more statistical characteristics of the skill data, Figure 3 presents the distributions of the number of the employees' skills from user and skill perspectives, respectively. We can see that the numbers of users with respect to the different numbers of skills show the double crest variation. The average number of skills per user mastered is 63.94. However, on the other hand, the data is quite sparse from the skill perspective. It is mainly because that the profile data contains both quite fine-grained and coarse-grained skill labels at the same time. As a result, some of the skill labels may have similar meanings but different names. Other skill labels may belong to contain relationships. Moreover, it is likely that the skills of each employee are not completely recorded in the profile data. Consequently, about 75% skills can only be found in less than 10 users' profiles in our dataset.

Based on the above, data sparsity and skill ambiguity raise great challenges to the design of the recommendation algorithm. Directly using the primary skill labels is improper. It is necessary to propose an appropriate approach to extract the effective competency representations from the high-dimensional and intensive noise data.

3 TECHNICAL DETAILS

In this section, we will introduce the technical details of our proposed DCBVN framework.

3.1 DCBVN Framework

Following the famous latent factor models (LFMs) [27, 42], we suppose that the process of users selecting courses is influenced by two perspectives, i.e., user demand perspective and course property perspective. Hence, we represent the user i by a latent variable $u_i \in \mathbb{R}^K$ and course j by a latent variable $v_j \in \mathbb{R}^K$ in a shared low-dimensional space with dimension K. Then the rating r_{ij} of user i on course j is drawn from the Normal distribution centered at the inner product of the two latent variables:

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}), \tag{1}$$

where c_{ij} is the precision parameter. It is usually set higher when $r_{ij} = 1$ than when $r_{ij} = 0$ ($c_{ij} = a$ if $r_{ij} = 1, c_{ij} = c$ if $r_{ij} = 0$, $a >> c$) [42, 44], indicting we have more confidence on the rating when $r_{ij} = 1$, since $r_{ij} = 0$ means the user may either be uninterested or unaware of the course.

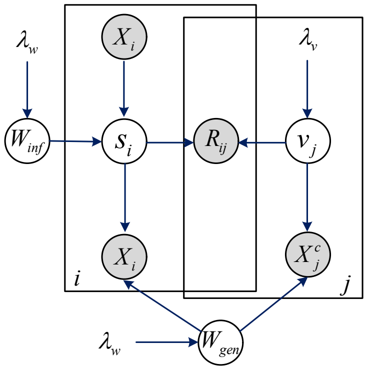


Figure 4: The graphical model of DCBVN.

Then, we put the Normal distribution as the prior of each latent course variable v_j:

$$v_j \sim \mathcal{N}(0, \lambda_v^{-1} I). \tag{2}$$

In traditional LFMs, the latent user demand variable u_i is also drawn from the Normal prior like v_j. However, according to common sense, the user skill backgrounds have dominated influences in the course selection procedure. Comprehending users' competency is of great benefit to model their demands. Along this line, we utilize the latent competency variable $s_i \in \mathbb{R}^K$ to represent the competency of user i by building a probabilistic generative progress using VAE-LDA [34]. Thus, the skill profile x_i of user i is generated through a generative network parameterized by ψ:

$$x_i \sim p_\psi(x_i | s_i). \tag{3}$$

On the other hand, as discussed before, users who have analogous employee skills and hence have similar latent competency variables may differ greatly in personal learning preferences depending on their goals of career development. It is inappropriate to only consider the competency for training course recommending. We can draw support from historical records to recognize users' learning goals. Hence, in this paper, we propose the demand recognition mechanism $G(\cdot)$ to transform the original competency variable s_i into the latent user demand variable u_i by considering both collaborative learning information and personalized employee competency information:

$$u_i = G(s_i). \tag{4}$$

In summary, Figure 4 illustrates the graphical model of DCBVN which comprehensively combines the conventional LFM based collaborative filtering method with autoencoding variational inference for topic modeling.

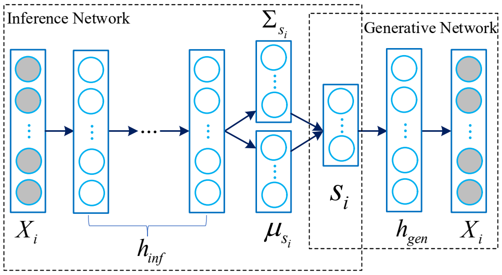


Figure 5: The network architecture of autoencoding variational inference based topic modeling.

3.2 Variational Autoencoder Network

In this subsection, we discuss how to extract semantic representations from the employee skills and build the collaborative network in Figure 5 for personalized training course recommendation.

3.2.1 *Topic modeling of skill profiles.*

In the literature, topic modeling algorithm is a quite popular probabilistic generative model for learning latent and interpretable representations [3, 33, 48]. However, traditional topic models may not perform well on the sparse data [51]. On the other hand, Variational autoencoder (VAE) has demonstrated its great inference abilities in many content embedding tasks, e.g., texts, labels and images [15, 29]. Thus, we employ VAE-LDA in our framework for constructing a hierarchical and explainable course recommender system.

Following the classic topic modeling algorithm [3], we assume every skill profile is generated from a mixture of K topics $\beta = (\beta_1, ..., \beta_K)$. Each topic $\beta_k \in \mathbb{R}^M$ represents a probability distribution over the entire M skills. For example, in the topic related to machine learning, skills like "SVM" and "random forest" would have high probabilities while the probabilities of "product design" and "market research" would be very small. The sum of the probabilities of all the skills should be equal to 1 for each topic.

Then the topic proportions $\theta_i \in \mathbb{R}^K$ of user i over the skills are supposed to obey Dirichlet distribution: $\theta_i \sim Dirichlet(\alpha)$, where each dimension represents the proportion of the corresponding topic. For example, $\theta_i = (0.1, 0.3, 0.6)$ means that the skill profile is related to the three topics with the proportions (10%, 30%, 60%), respectively.

Consequently, the skill profile $x_i = (x_{i1}, ..., x_{iN_i})$ with length N_i for user i can be drawn from the Multinomial distribution: $x_{in} \sim Multinomial(1, \beta\theta_i)$. Under this assumption, the marginal likelihood of the skill profile x_i can be given by:

$$p(x_i|\alpha, \beta) = \int_{\theta_i} \left(\prod_{n=1}^{N_i} p(x_{in}|\beta, \theta_i) \right) p(\theta_i|\alpha) d\theta_i. \quad (5)$$

The Dirichlet prior in LDA is significant for obtaining interpretable topics [41]. Nevertheless, it is problematic to implement the reparameterization trick for the Dirichlet distribution so that incapable to take gradients through the sampling process in VAE. To solve the problem, we utilize a Laplace approximation to the softmax basis of Dirichlet prior, which supports unconstrained optimization of the cost function [24].

Hence, we use s_i to denote the competency variable and we have $\theta_i = \sigma(s_i)$, where $\sigma(\cdot)$ is the softmax function. Each dimension of s_i is related to a specific topic in full accord with θ_i.

Then following the Laplace approximation of the softmax basis in [12], the off-diagonal elements of the covariance matrix are suppressed with $O(1/K)$, leading to approximately diagonal covariance matrix for large K. Accordingly, the Laplace approximation $p(s_i)$ over the competency variable s_i can be given as a multivariate Normal with mean μ and covariance Σ where:

$$\begin{aligned} \mu_k &= \log \alpha_k - \frac{1}{K} \sum_{l=1}^{K} \log \alpha_l, \\ \Sigma_{kk} &= \frac{1}{\alpha_k}(1 - \frac{2}{K}) + \frac{1}{K^2} \sum_{l=1}^{K} \frac{1}{\alpha_l}. \end{aligned} \quad (6)$$

Recalling that competency variable s_i is the softmax basis of Dirichlet prior, we can thereupon approximate the simplex basis with the logistic normal distribution $p(\theta_i|\alpha) \approx \mathcal{LN}(\theta_i|\mu, \Sigma)$ [17].

3.2.2 *Generative process.*

Now we can present the generative process for user skill profiles as follows:

(1) For the layer h_{gen} of the generative network:

a. Draw the m_{th} column of weight matrix W_{gen}:

$$W_{gen,*m} \sim \mathcal{N}(0, \lambda_w^{-1} I_K).$$

b. The topic matrix $\beta = \sigma(W_{gen})$.

c. Obtain the i_{th} row of hidden state h_{gen} by:

$$h_{gen,i*} = (\beta\theta_i)^T.$$

(2) For each user i, draw x_i from:

$$x_i \sim Multinomial(N_i, h_{gen,i*}).$$

Here, we also take advantage of the softmax basis W_{gen} of topic matrix β for the ease of unconstrained optimization. We use $\sigma(W_{gen})$ to denote the softmax transformation separately on every column of weight matrix W_{gen}.

It is noticed that there are two kinds of data collection modes for skill profile x_i in real scenes, depending on whether one skill label could appear once or multiple times in a list. For the situation that all the skills only hold at most once in the list per user, the skill generation follows a multivariate hypergeometric distribution instead of a multinomial distribution. Fortunately, the multivariate hypergeometric distribution converges to the multinomial distribution with large skill size [4]. Accordingly, we can still apply the above generative process for a large dataset in such a situation.

Owing to the above process, we can obtain the explainable competency variable s_i. However, the course property variable v_j still remains unexplainable. Due to the various sources of online courses, it may be quite manpower-consuming to collect the labels for each course. Moreover, it would be much more beneficial for helping users make decisions if the courses could be labeled with the same topics as user competency variables. Therefore, we can make use of the skill profiles of users who have learned the course to represent the property of this course.

Let x_j^c and s_j^c denote the skill profile and latent skill variable of course j, respectively. To construct x_j^c, we first count the numbers of occurrences of each skill among users who have learned course j.

Then we only keep the top e (e is a constant) frequent skills so that the popular courses would not have much noise skills. As for the latent skill variable s_j^c, we can transform the original unexplainable course property variable v_j into a suitable latent space to obtain the explainable vector. Here, we employ the 1-layer MLP for the transformation process:

$$s_j^c = \sigma(W^c v_j + b^c). \tag{7}$$

Then the generative process of course skill profile x_j^c with length N_j^c can be given by:

$$x_j^c \sim Multinomial(N_j^c, \beta s_j^c). \tag{8}$$

Here we share the same topic matrix β with the generative process of user skill profile to make sure that the topics in both user and course perspectives have exactly the same meanings.

3.2.3 *Demand Recognition*.

The learned competency variables are very useful for understanding the employees' skill backgrounds. Intuitively, they are also highly relevant to the career development demands of employees. However, as discussed before, it is obviously inappropriate and limited to only utilize s_i for generating the final demand u_i. Supposing there are two technical employees with similar skill profiles using LMS for online learning. One wants to become a senior Java programmer, and the other one intends to transfer to an AI developer. Certainly, they should accept totally different recommendations to fit their actual demands.

Therefore, we propose a demand recognition mechanism as shown in Figure 6 to bridge the competency variable s_i and demand variable u_i by exploiting the historical course records and collaborative information. Since each dimension of s_i represents one of the K topics, we can model the pattern of demand transformation on each topic by combining transfer variable $d_t \in \mathbb{R}^K$ with s_i. Here d_t reflects the changes in the proportions of topics. Specifically, each transfer pattern can be interpreted as one kind of learning trend. Then we have transformed matrix $Z_i = (z_{i1}, ..., z_{iT}) \in \mathbb{R}^{K \times T}$, where $z_{it} = s_i + d_t$, supposing there are T patterns.

In order to comprehensively analyze users' true demands, we need to measure the relations between each transfer pattern and users' real demand by calculating the importance score of every transfer pattern. Larger importance score ω_{it} would indicate more critical influence of pattern t on user i. In this way, we are able to capture the most relevant transfer patterns and pay more attention to them. In this paper, we propose two ways to automatically compute the importance score $\omega_i = (\omega_{i1}, ..., \omega_{iT}) \in \mathbb{R}^T$, namely individual-based and group-based importance score.

For individual-based importance score ω_i, we only employ the historical course records of user i to scoop out his/her demands:

$$\omega_i = \sigma(Z_i^T V R_{i*}^T), \tag{9}$$

where R_{i*} is the i_{th} row of the course record matrix R. Here we take advantage of the softmax function to make sure that the sum of the T patterns' score is equal to 1.

However, the course records of a single user may be too few to find his/her real demand, since the data are quite sparse. Therefore, we further take the influence of user groups into consideration due to the common sense that employees doing similar work may share similar demands of career development. To this end, we first divide the users into different groups based on their departments and job positions. Let g_i denote the group that employee i belongs to. Then we compute the group-based record vector $\tilde{R}_{i*}$ as follows:

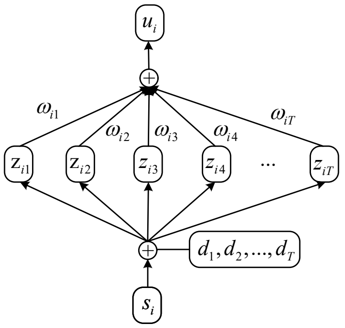


Figure 6: The demand recognition mechanism.

$$\tilde{R}_{i*} = \frac{\sum_f R_{f*} I(g_f = g_i)}{sum\left(\sum_f R_{f*} I(g_f = g_i)\right)}, \tag{10}$$

where $sum(\cdot)$ denotes the sum of every column of the vector and $I(\cdot)$ is the indicator function. Here we employ the element-wise summation function $sum(\cdot)$ to prevent the record vectors of large groups being far greater than those of small groups. Hence, we could calculate the grouped-based importance score ω_i as follows:

$$\omega_i = \rho \cdot \sigma(Z_i^T V R_{i*}^T) + (1 - \rho) \cdot \sigma(Z_i^T V \tilde{R}_{i*}^T), \tag{11}$$

where ρ is a balance parameter to adjust the influence from individual and user group perspectives.

Finally, by combing the influence of T patterns with the importance score $\omega_i \in \mathbb{R}^T$, we are able to comprehensively analyze users' true demands. Specifically, the final latent demand variable u_i can be obtained by the weighted summation:

$$u_i = Z_i \omega_i = \sum_{t=1}^{T} \omega_{it} z_{it}. \tag{12}$$

3.2.4 *Inference process*.

After building the generative model, we could provide the joint probability of the full data:

$$p(R, X, X^c, V, S) = \prod_{i,j} p(r_{ij}|u_i, v_j) p_\psi(x_i|s_i) p(s_i) p(x_j^c|v_j) p(v_j).$$

We need to obtain the posterior distribution of the latent variables to inference the model. Nevertheless, it is difficult to get an analytical solution. To address this problem, we utilize a popular approximation, i.e., Stochastic Gradient Variational Bayes (SGVB) estimator, to construct a stochastic gradient ascent solution for our proposed framework. Specifically, we realize an inference network parameterized by ϕ for efficient posterior inference of competency variable s_i. The multi-layer perception network (MLP) with L layers is chosen as the inference network structure. Thus, the variational distribution q is parameterized by ϕ and x_i:

$$q(S) = \prod_i q_\phi(s_i|x_i) = \prod_i \mathcal{N}(\mu_\phi(x_i), \Sigma_\phi(x_i)). \tag{13}$$

Then the inference process can be defined as follows:
(1) For each layer l of the inference network:
a. Draw the m_{th} column of weight matrix W_l:

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

b. Draw the bias vector b_l from $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l}))$.
c. Draw the i_{th} row of hidden state h_l by:

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

(2) Draw mean and covariance of latent variable by:

$$\begin{aligned} \mu_{s_i} = \mu_\phi(x_i) \quad &\sim \quad \mathcal{N}(h_L W_\mu + b_\mu, \lambda_s^{-1} I_K), \\ \Sigma_{s_i} = \Sigma_\phi(x_i) \quad &\sim \quad diag(\mathcal{N}(h_L W_\Sigma + b_\Sigma, \lambda_s^{-1} I_K)). \end{aligned}$$

Draw latent variable s_i from:

$$s_i \sim \mathcal{N}(\mu_{s_i}, \Sigma_{s_i}),$$

where λ_w, λ_s are the hyperparameters and K_l is the number of columns of the hidden state h_l. Besides, λ_s is often supposed to be infinite in VAE so that the Normal distribution would degrade to Dirac delta function [36]. In such a case, VAE could work like other common neural networks and improve computing efficiency.

With the help of reparameterization trick [31], we could easily gain a differentiable sampling process of s_i. Specifically, we draw sample by $\epsilon \sim \mathcal{N}(0, I)$ and let $s_i = \mu_{s_i} + \epsilon\Sigma_{s_i}$.

Thus, the Evidence Lower Bound (ELBO) of our DCBVN framework is formulated as:

$$\begin{aligned} \mathcal{L}(q) = \log p(R|U,V) + \mathbb{E}_q[\log p_\psi(X|S)] + \log p(X^c|V) \\ + \log p(V) - \mathbb{KL}(q_\phi(S|X)||p(S)). \end{aligned} \tag{14}$$

It can be seen that the ELBO is separated into three pieces: prediction loss, reconstruction loss, and prior loss.

First, the prediction loss is frequently utilized in LFMs to maximize the log-likelihood of records:

$$\log p(R|U,V) = -\sum_{i,j} \frac{C_{ij}}{2}(r_{ij} - u_i^T v_j)^2.$$

Second, the reconstruction loss measures the generative quality of skill profiles of users and courses. For user generative process, in consideration of the intractability of computing expected values, we employ Monte Carlo sampling from ϵ following Law of the Unconscious Statistician:

$$\log p(X^c|V) = \sum_j p(x_j^c|v_j),$$

$$E_q[\log p(X|S)] = \sum_i \frac{1}{D} \sum_d p_\psi(x_i|s_i^{(d)}).$$

Finally, the prior loss can be viewed as a regularization term:

$$\log p(V) = -\frac{\lambda_v}{2} \sum_j \|v_j\|^2,$$

$$\mathbb{KL}(q_\phi||p) = \frac{1}{2} \sum_i \left[(\mu - \mu_{s_i})^T \Sigma^{-1} (\mu - \mu_{s_i}) \ + tr(\Sigma^{-1}\Sigma_{s_i}) + \log \frac{|\Sigma|}{|\Sigma_{s_i}|} - K\right].$$

Thus, stochastic optimization methods such as Adam can be used to operate the ELBO.

3.3 Prediction

Let Y denote the observed data. On the basis of the trained model, we can make the estimation by:

$$\mathbb{E}[r_{ij}|Y] = \mathbb{E}[u_i|Y]^T \mathbb{E}[v_j|Y] = \mathbb{E}[G(s_i)|Y]^T \mathbb{E}[v_j|Y]. \tag{15}$$

For the point estimation, we approximate the prediction as:

$$r_{ij}^* = G(\mathbb{E}[s_i])^T v_j, \tag{16}$$

where $\mathbb{E}[s_i] = \mu_{s_i}$, i.e., the mean variable in inference network.

For the cold-start situation, we have no course records for the new users. However, we can still make recommendations based on the competency variable s_i. Moreover, we can also draw help from user group information to solve the cold-start problem. Specifically, we adopt the group-based importance score and set the balance parameter ρ as 0. In this way, we can still make recommendations like in the normal position. As a result, our DCBVN framework can greatly alleviate the cold-start problem.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed DCBVN framework from the following aspects: (1) the overall recommendation performances compared with several state-of-the-art baselines on normal and sparse scenarios; (2) the analysis on cold-start scenarios; (3) the parameter analysis and (4) some case studies to visualize the interpretability of DCBVN.

4.1 Baseline Approaches

To verify the effectiveness of DCBVN, we compare it with several baseline methods including two traditional models (i.e., WMF and CTR), four state-of-the-art models (i.e., NeuMF, DeepMF, CDL and CVAE) and a variant of DCBVN:

- **WMF** [13]: This is a classic collaborative model using linear low-rank factorization for recommendation.
- **CTR** [42]: This is a classic hybrid model, which leverages LDA for modeling content information and combines it with traditional collaborative LFM.
- **NeuMF** [11]: This is a state-of-the-art collaborative model for implicit feedback, which leverages a multi-layer perceptron to learn the user-item interaction function with cross entropy loss.
- **DeepMF** [47]: This is a state-of-the-art collaborative model, which implements matrix factorization with multi-layer perception network.
- **CDL** [44]: This is a state-of-the-art hybrid recommender utilizing stacked denoising autoencoder for extracting deep latent representations.

Table 1: The overall recommendation performance of different approaches in the normal setting.

Methods	R@20	R@50	R@100	R@300
WMF	0.1577	0.2026	0.2735	0.4430
CTR	0.2150	0.2900	0.3598	0.4903
NeuMF	0.2178	0.2944	0.3676	0.5154
DeepMF	0.2228	0.2982	0.3632	0.5033
CDL	0.2311	0.3132	0.3867	0.5217
CVAE	0.2401	0.3214	0.3943	0.5273
DCBVN-0	0.2571	0.3478	0.4197	0.5554
DCBVN	**0.2668**	**0.3519**	**0.4341**	**0.5840**

- **CVAE** [18]: This is a state-of-the-art hybrid method and can be viewed as the improved version of CDL, which improves the content representations by applying variational autoencoder with Bayesian inference.
- **DCBVN-O**: This is a variant of our proposed DCBVN framework by ignoring the group information of users and only adopt the individual-based importance score in the modeling process. By comparing DCBVN-O with DCBVN, which adopts the group-based importance score, we can verify the usefulness of group information.

Among the baselines, WMF, NeuMF and DeepMF only make use of course records for matrix factorization while all the other methods employ both record information and skill profile information for collaborative filtering.

4.2 Evaluation Metric

To measure the performances of recommendation results, we adopt the widely used evaluation metrics in recommender systems, i.e., Recall@P [18, 19, 44]. Recall@P counts the ratio of successfully predicted items among top-P items to all positive items for each user as follows:

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

The final Recall result reported is the average value of all users. Generally, the larger the values of Recall are, the better results we have. It is noticed that the zero entry in the course record matrix does not necessarily mean the user dislikes the course, but may be indeed unaware of the course. Consequently, precision is not so suitable for measuring performance here [44].

4.3 Experimental Settings

In the experiments, we evaluated and compared the models under both the normal setting and sparse setting. In the normal setting, we selected 70% and 10% course records for each user to construct the training and validation set respectively according to the chronological order, since it might be inappropriate to use an employee's future course selection for training and then recommending during the testing process. The rest course records composed the test set. The dataset partitioning in the sparse setting was similar, but we only chose 30% courses for each user to form the training set.

Table 2: The overall recommendation performance of different approaches in the sparse setting.

Methods	R@20	R@50	R@100	R@300
WMF	0.1620	0.2034	0.2629	0.4080
CTR	0.1888	0.2551	0.3173	0.4394
NeuMF	0.2045	0.2654	0.3295	0.4740
DeepMF	0.2168	0.2685	0.3329	0.4676
CDL	0.2214	0.2940	0.3603	0.4804
CVAE	0.2201	0.2933	0.3631	0.5080
DCBVN-0	0.2290	0.3208	0.3979	0.5318
DCBVN	**0.2449**	**0.3293**	**0.4053**	**0.5435**

For all the baseline methods, we set the dimension of latent space K as 50 for a fair comparison. Following the settings in their papers, we set the precision parameters c_{ij} as $a = 1, c = 0.01$ and pretrained CTR, CDL and CVAE with an LDA, a two-layer SDAE network, and a two-layer VAE network respectively to get the parameter initialization. The noise level in CDL was set to be 0.3. Then we explored the corresponding parameters of all the baselines, such as regularization parameters, learning rates and other parameters.

In our DCBVN framework, we also set $K = 50$ and chose the 2-layer MLP network as the inference network architecture for a fair comparison with baselines. The dimensions of the two layer were set as 200 and 100 respectively, which is the same setting in CDL and CVAE. Similar to Li and She [18], we added a balance parameter λ_r to adjust the penalty of reconstruction loss with respect to prediction terms. Then the precision parameter $\tilde{c}_{ij}$ could be set as $a = \lambda_r, b = 0.1\lambda_r, c = 0.01\lambda_r$. Thus, tuning the parameter λ_r is equivalent to tuning the precision parameter. Besides, the value of α_k in Equation 6 was set as $1/K$ for each k and the maximum number e of skills for each course as 200. Finally, we tuned the value of λ_v from the candidate set $\{0.01, 0.1, 1, 10, 100\}$.

Though using the same user inference network architecture with CDL and CVAE, we found that DCBVN could still work well without the pre-trained network parameters for content embedding. This might because we performed the stochastic optimization methods for all the parameters synchronously, while CDL and CVAE optimized the content variables and collaborative variables alternately. Thus, DCBVN does not necessarily need the pre-training process.

4.4 Experimental Results

4.4.1 ***Recommendation performance.*** In this part, we investigate the performance of DCBVN and baselines in the *DLearner* dataset. We evaluate all the baselines in two positions, i.e., normal and sparse settings.

Table 1 and Table 2 show the recommendation performance results of all models in the normal and sparse settings, respectively. It can be observed that our proposed DCBVN method achieves the best performances in both settings. By comprehensively modeling the skill backgrounds and personal demands of employees, DCBVN could outperform CVAE by a margin of 2.67% (a relative boost of 11.12%) in the normal setting and 2.48% (a relative boost of 11.27%) in the sparse setting when $P = 20$. Besides, DCBVN-O, i.e., the variant of DCBVN, also has better performance than other baselines.

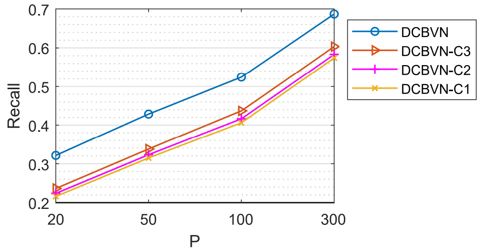


Figure 7: The performance results of DCBVN on cold-start scenarios.

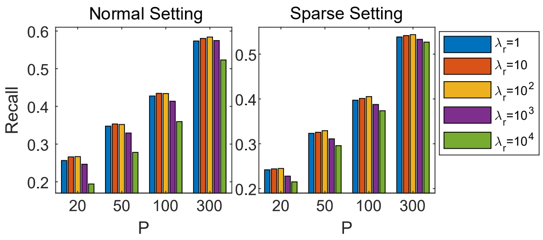


Figure 8: The performance of Recall@P with different different values of λ_r.

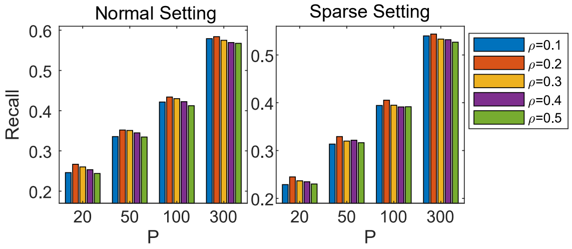


Figure 9: The performance of Recall@P with different values of ρ.

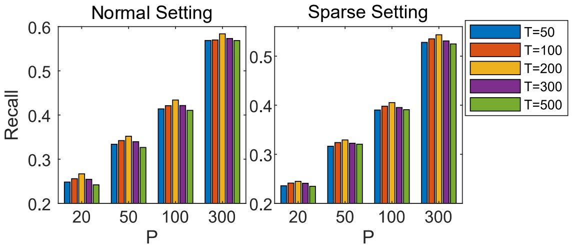


Figure 10: The performance of Recall@P with different values of T.

DCBVN-O is a little worse than DCBVN, which demonstrates that user group information is actually beneficial in the modeling process. We can also find that DCBVN outperforms DCBVN-O more in the sparse setting when $p = 20$ than in the normal setting, since employing group information is somewhat equivalent to increasing the data size of each employee's course records, which alleviate the sparse problem to some extent. Meanwhile, we can find that the performances of CVAE are only lower than DCBVN and DCBVN-O, which demonstrate the embedding ability of variational autoencoder. By comparison, CTR, which leverages LDA in its model, performs not well due to the sparsity of skills. Besides, NeuMF and DeepMF perform better than CTR and WMF due to the superior performance of neural networks.

4.4.2 ***Cold-start scenarios.*** Cold-start problem is a common hard problem in recommender systems that new users have no historical record. It becomes even more serious on the LMS due to the massive new employees in many modern fast-paced companiese [30]. Without the user-item interactions, many CF methods would fail to make predictions. In this part, we provide some analysis of our proposed framework on the cold-start scenarios.

In this subsection, we propose three different approaches to handle the cold-start problem under our framework, namely DCBVN-C1, DCBVN-C2 and DCBVN-C3. In DCBVN-C1, we only use the employees' competency variables for the recommendation, that is, the latent user variables are exactly the same as the skill variables of new users. In DCBVN-C2, we place no assumption on different transfer patterns to calculate the demand variables of each new user, that is, we let every transfer pattern have the same importance score. In DCBVN-C3, for each new user, we only employ the user group information to calculate the importance scores of the employees, that is, we adopt the group-based importance score and set the balance parameter ρ as 0 in Equation 11. For comparison, we also provide the result of DCBVN without cold-start restriction.

First, we selected 50% users in the training stage and the rest users were assumed to be new users. In order to be consistent with the normal setting, we randomly chose 70% course records of training users to construct the training set for DCBVN and the three variants. The rest 30% course records were used for validation. Then we select 30% course records of new users as the training set only for DCBVN. Noticed that for DCBVN-C1, DCBVN-C2 and DCBVN-C3, these data would not be used. The rest 70% course records of new users compose the test set for all the four methods. The performance results are shown in Figure 7.

From Figure 7 we can easily find that DCBVN outperforms all the variants a lot, which clearly demonstrates the necessity of considering the personal demands of employees for the recommendation. Even DCBVN-C3 has utilized user group information, it still performs much worse than DCBVN, showing the dominated effect of individual historical records.

Among the three cold-start variants, it can be observed that DCBVN-C1 still performs well, which verifies the effectiveness of our framework even on the cold-start scenario. Besides, DCBVN-C2 performs better than DCBVN-C1, which demonstrates again that it is improper to directly treat skill variable as the user demand variable. Finally, we can observe that DCBVN-C3 outperforms the other two variants. This shows the effectiveness of user group information even on the cold-start scenario.

4.4.3 ***Parameter analysis.*** First we evaluate the parameter λ_r, which represents whether we concentrate more on prediction accuracy or reconstruction error. Figure 8 presents the results for different values of λ_r. When λ_r is small, the penalty of prediction

Table 3: Case studies on DCBVN based user understanding.

top 3 topics of skills
a. distributed, hadoop, system design, apache, spark, hive, scheduling, storage, operation
b. products design, product operation, promotion, impact, planning, scheme, disassembly, data analysis
c. retrieval, data flow, trigger, capture packets, stability, query, automation, building database
selected courses
a. *Information security awareness training*
b. *From technological backbone to manager*
top 3 topics of demands
1. distributed, hadoop, system design, apache, spark, hive, scheduling, storage, operation
2. business, service, management, guidance, strategy, organize, examine, implement
3. achieve, dispose, service, system, solution, develop, platform, workflow, leadership
suggested courses
1. *Office network security training*
2. *Application cases of big data on predictions*
3. *What are the success factors of a high-tech company*
4. *Engineering leadership talk: platform governance*
5. *How to use the tailor-made data storage system*
6. *Introduction to safety specification*
7. *The technology platforms in the company*
8. *Engineering leadership and strategy*

loss declines. Thus, the DCBVN framework tends to reduce the effectiveness of collaborative filtering. Moreover, it requires a larger iterative number for DCBVN to convergence with small λ_r. On the other hand, when λ_r grows larger, the quality of latent representations could not remain good due to the overfitting problem.

Then we discuss the influence of balance parameter ρ, which shows the demand recognition mechanism focusing more on individual/group information. Figure 9 presents the results for different values of ρ. We can observe that when $\rho = 0.2$, DCBVN gets the best performance results. It is noticed that $\rho = 0.2$ does not imply we pay much more attention to group information than individual information because we have normalized the group course records $\tilde{R}_{i*}$ so that the norm of $\tilde{R}_{i*}$ is much smaller than the norm of individual records R_{i*}.

Lastly, we investigate the influence of the number of demand transfer patterns. We assume that there are T chief patterns in the practical scenes. With small T, we could not model the demand transferring process much well. Nevertheless, if T is too large, it would lead to overfitting. As shown in Figure 10, $T = 200$ seems to be an appropriate choice. We can find that our DCBVN framework is robust for overfitting since DCBVN is inherently a Bayesian generative model learning the latent distributions rather than the point estimates of variables.

4.4.4 *Case study.*

In this subsection, we provide the interpretable insights of the recommendations obtained by our DCBVN framework from both employee and course perspectives.

Employee perspective. Table 3 shows a real user case study of DCBVN in *DLearner* dataset. We first present the top 3 topics of the user's skills. From Table 3, we can speculate that she might be an R&D engineer since her current skills are mostly related to data processing and programming. Meanwhile, from the historical learning records, we can find that she preferred to learn about how to grow into a manager of the team (*course* b). Besides, she was also interested in office privacy and security (*course* a).

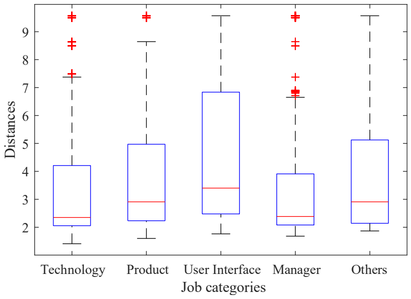


Figure 11: The Euclidean distances between competency and demand variables of employees in different job categories.

Then we provide the top 3 topics of the demands learned by our model and the top recommended courses in Table 3. It can be directly found from the *topic* 2 and *topic* 3 that the user demand variable correctly captures the differences of user's current competencies and personal demands. In this way, DCBVN successfully understands her career development goal, i.e., become a good manager. Thus, DCBVN recommends some courses to enhance her business horizons (*course* 2 and 3) and develop her leadership abilities (*course* 4 and 8). DCBVN also suggests some technical courses (*course* 5 and 7) to help her improve the current competencies. Besides, *course* 1 and 6 are recommended due to her demand for information safety awareness training.

Next, we present some interesting results on the differences in employees' skill backgrounds and personal career development demands. Figure 11 shows the Euclidean distances between competency and demand variables of employees in different job categories. The small distance means they mainly choose courses based on their current competencies while large distance implies the high impact of various career development demands. The distances of employees in Technology and Manager categories are quite smaller than employees in other categories. This may because their career development goals are tightly related to their current career directions. Meanwhile, employees in User Interface and Product categories usually have a wide variety of learning preferences, which indicates they are more likely to seek different career developments.

Course perspective. DCBVN is also able to label the courses with the topics of skills, which is very helpful for users to make decisions. Especially when the course titles cannot directly reflect the content, users would be quite confused and most likely miss the suitable courses. For example, Table 4 presents the top 3 topics of two real courses in *DLearner* dataset. If one does not know what Selenium is, he/she may feel puzzled about the course *Exploring Selenium*. However, with the help of our labels learned by DCBVN, he/she could guess that the course is about an automated testing Tool (In fact, Selenium is an open-source Web automated testing tool). Similarly, if one does not know who is Mr. Wu, he/she would absolutely have no idea about the course *Mr. Wu's experience sharing*.

Table 4: Case studies on DCBVN based course labeling.

Exploring Selenium
a. function test, test tool, continuous integration, quality assurance, code, coverage rate, review, jenkins
b. debug, compile, shell, programming, sdk, encapsulation, match, hadoop, open source, reconfiguration
c. product design, redis, encapsulation, http, cache, spark, com, reviewing, cooperation, sdk, offline
Mr. Wu's experience sharing
1. management, product, data, business, business requirements, scheme, service, design
2. plan, innovate, management, data analysis, put on market, project management, reposition
3. mining, orientation, data analysis, reposition, trans-department, solution, investigation & research

Actually, Mr. Wu is an excellent grassroots manager in the company and we can easily infer this from the given topics.

Furthermore, we provide an overall view of the courses in *DLearner* dataset to show the interpretability of DCBVN. We first performed k-means clustering [1] to partition all the 8, 693 courses into 8 clusters according to their latent skill variables $\{s_j^c, j = 1, ..., 8693\}$ obtained by DCBVN. Then we utilize the t-SNE algorithm [23] to transform the original 50-dimensional vector into a 2-dimensional space for visualization, as shown in Figure 12. We can observe that *cluster* 5, 6 and 7 are quite close. Actually, courses in these three clusters are all technical courses while *cluster* 5 focuses on underlying architecture and database; *cluster* 6 focuses on application and algorithm; *cluster* 7 focuses on web and mobile devices. Moreover, *cluster* 2 and 3 are also very close to and interlocked with each other in the low-dimensional space. In fact, the courses in these two clusters are both relevant to product and marketing. Besides, *cluster* 1 mainly contains courses about management and *cluster* 8 is mainly composed of courses helping improve the employees' personal qualities, such as communication and leadership. Lastly, *cluster* 4, which is far away from the other clusters, consists of some miscellaneous courses without a clear theme.

5 RELATED WORK

In this section, we first introduce some general recommendation algorithms and then explainable recommendation systems. Lastly, we will focus on course recommenders.

General Recommenders. In general recommender, Collaborative Filtering (CF) methods have been regarded as the most popular and successful technique for mining the relevancy between users and items from the historical interactions [20, 22]. Among various CF methods, the latent factor models (LFMs) [27, 43] are the most widely used approach due to their advanced recommendation performance compared with traditional neighborhood based methods [32]. For example, the probabilistic matrix factorization (PMF) [27], as a representative LFM, aims to factorize the rating matrix into the product of user and item latent matrices in a low-rank space. Furthermore, Wang and Blei proposed an approach to combine LFM with classic topic models [3] for integrating content information. Recently, with the successful adoption of deep learning methods in many fields [14, 50], some researchers focused on building advanced hybrid recommendation models to blend LFM and neural network for collaborative information modeling [45]. For example, Wang et al. [44] exploited the Stacked Denoising Autoencoders for achieving collaboratively content embeddings with latent content variables. Furthermore, Li and She [18] utilized the Variational Autoencoder for modeling content information to construct a Bayesian hybrid recommendation model.

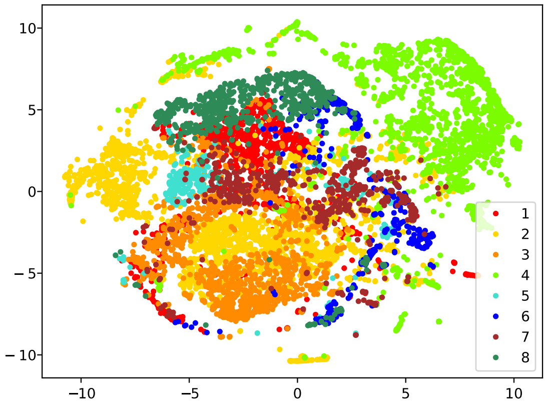


Figure 12: The visualization of DCBVN based course clustering. (The clusters are distinguished by different colors.)

Explainable Recommenders. There are mainly two types of methods for constructing explainable recommender systems in the literature. One is post-hoc method, which chooses to separate the recommending and interpreting processes and the explanations are picked from a group of pre-defined templates [39]. The other is the embedded method, which tried to build a unified model to integrate both the recommending and interpreting processes [5, 26, 46]. The existing embedded methods are mainly based on reviews. For example, McAuley and Leskovec [26] obtained interpretable textual labels for latent rating dimensions from product reviews. Besides, Chen et al. [5] introduced an attention mechanism to explore the usefulness of reviews and produce review-level explanations. Furthermore, Gao et al. [10] built an explainable deep hierarchy network with an attentive multi-view learning framework. Chen et al. [6] designed a hierarchical co-attentive selector to optimize accuracy and explainability in a joint way.

Course Recommenders. Current course recommender systems mostly focus on the scenario of student education [2, 49]. For example, Parameswaran et al. [28] studied the problem of constraint based course recommendations for students; Vialardi et al. [38] studied to recommend how many and which courses to study on the basis of previous students; Thai-Nghe et al. [37] proposed to predict the student performance with LFM for recommendation; and Chu et al. [7] proposed to recommend courses on the web with rule based methods. Recently, some efforts were made to enhance the learning practices of individuals and organizations in talent management [25]. For example, Klašnja-Milićević et al. [16] recognized different patterns of learning style and utilized neighborhood methods for online training recommendations. Srivastava et al. [35] studied the scenario of industrial training in organizations with sequence matching and mining.

Different from the above works, we studied the problem of explainable personalized training course recommendation with employees' career development awareness

6 CONCLUSIONS

In this paper, we proposed a personalized online course recommender system for improving employees' training and development. A unique perspective of this system is that we jointly model the employees' current competencies and their sustainable career development. Specifically, we developed a novel end-to-end Demand-aware Collaborative Bayesian Variational Network (DCBVN) framework, which could extract latent interpretable representations from their skill profiles and then learn the personal demands of career development for different employees. Furthermore, we designed an adapted collaborative filtering algorithm for recommending the most appropriate training courses for employees. Moreover, all the above processes are integrated into a unified Bayesian inference view. Finally, we conducted extensive experiments on real-world data to demonstrated the effectiveness and the interpretation power of DCBVN, as well as its robustness on sparse and cold-start scenarios.

ACKNOWLEDGMENTS

This work was supported by grants from the National Natural Science Foundation of China (No.91746301, 61836013, U1605251, 71722005, 71571133)), and Tianjin Natural Science Foundation (No.18JCJQJC45900).

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

[2] Narimel Bendakir and Esma Aïmeur. 2006. Using association rules for course recommendation. In *Proceedings of the AAAI Workshop on Educational Data Mining*, Vol. 3.

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

[4] Youngchul Cha and Junghoo Cho. 2012. Social-network analysis using topic models. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 565–574.

[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1583–1592.

[6] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. IJCAI.

[7] Ko-Kang Chu, Maiga Chang, and Yen-Teh Hsia. 2003. Designing a course recommendation system on web based on the students' course selection records. In *EdMedia: World Conference on Educational Media and Technology*. Association for the Advancement of Computing in Education (AACE), 14–21.

[8] Renée E Derouin, Barbara A Fritzsche, and Eduardo Salas. 2005. E-learning in organizations. *Journal of Management* 31, 6 (2005), 920–940.

[9] Louis V DiBello, Louis A Roussos, and William Stout. 2006. 31a review of cognitively diagnostic assessment and a summary of psychometric models. *Handbook of statistics* 26 (2006), 979–1030.

[10] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation Through Attentive Multi-View Learning. AAAI.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 173–182.

[12] Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. 2012. Kernel topic models. In *Artificial Intelligence and Statistics*. 511–519.

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.

[14] Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question Difficulty Prediction for READING Problems in Standard Tests. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[16] Aleksandra Klašnja-Milićević, Boban Vesin, Mirjana Ivanović, and Zoran Budimac. 2011. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education* 56, 3 (2011), 885–899.

[17] John D Lafferty and David M Blei. 2006. Correlated topic models. In *Advances in neural information processing systems*. 147–154.

[18] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 305–314.

[19] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1734–1743.

[20] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 689–698.

[21] Qi Liu, Runze Wu, Enhong Chen, Guandong Xu, Yu Su, Zhigang Chen, and Guoping Hu. 2018. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 4 (2018), 48.

[22] Zheng Liu, Xing Xie, and Lei Chen. 2018. Context-aware Academic Collaborator Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 1870–1879.

[23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[24] David JC MacKay. 1998. Choice of basis for Laplace approximation. *Machine learning* 33, 1 (1998), 77–86.

[25] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. 2011. Recommender systems in technology enhanced learning. In *Recommender systems handbook*. Springer, 387–415.

[26] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[27] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[28] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. 2011. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)* 29, 4 (2011), 20.

[29] 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.

[30] Chuan Qin, Hengshu Zhu, Chen Zhu, Tong Xu, Fuzhen Zhuang, Chao Ma, Jingshuai Zhang, and Hui Xiong. 2019. DuerQuiz: A Personalized Question Recommender System for Intelligent Job Interview. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2165–2173.

[31] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).

[32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[33] Dazhong Shen, Hengshu Zhu, Chen Zhu, Tong Xu, Chao Ma, and Hui Xiong. 2018. A joint learning approach to intelligent job interview assessment.. In *IJCAI*. 3542–3548.

[34] Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488* (2017).

[35] Rajiv Srivastava, Girish Keshav Palshikar, Saheb Chaurasia, and Arati Dixit. 2018. What's Next? A Recommendation System for Industrial Training. *Data Science and Engineering* 3, 3 (2018), 232–247.

[36] Robert S Strichartz. 2003. *A guide to distribution theory and Fourier transforms*. World Scientific Publishing Company.

[37] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. 2010. Recommender system for predicting student performance. *Procedia Computer Science* 1, 2 (2010), 2811–2819.

[38] Cesar Vialardi, Javier Bravo, Leila Shafti, and Alvaro Ortigosa. 2009. Recommendation in Higher Education Using Data Mining Techniques. *International Working Group on Educational Data Mining* (2009).

[39] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 47–56.

[40] Richard A Voorhees. 2001. Competency-Based learning models: A necessary future. *New directions for institutional research* 2001, 110 (2001), 5–13.

[41] Hanna M Wallach, David M Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. In *Advances in neural information processing systems*. 1973–1981.

[42] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[43] Chao Wang, Qi Liu, Runze Wu, Enhong Chen, Chuanren Liu, Xunpeng Huang, and Zhenya Huang. 2018. Confidence-aware matrix factorization for recommender systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[44] 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.

[45] 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.

[46] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 587–596.

[47] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[48] Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)* 9, 3 (2018), 1–17.

[49] Fan Zhu, Horace HS Ip, Apple WP Fok, and Jiaheng Cao. 2007. PeRES: A personalized recommendation education system based on multi-agents & SCORM. In *International Conference on Web-Based Learning*. Springer, 31–42.

[50] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*. 465–476.

[51] Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2105–2114.