Received October 31, 2018, accepted December 20, 2018, date of publication January 1, 2019, date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890293

Collaborative Additional Variational Autoencoder for Top-N Recommender Systems

MING HE, QIAN MENG, AND SHAOZONG ZHANG
Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Corresponding author: Ming He (heming@bjut.edu.cn)

This work was supported in part by the Beijing Natural Science Foundation under Grant 4192008 and in part by the General Project of Beijing Municipal Education Commission under Grant KM201710005023.

ABSTRACT Collaborative filtering (CF) has been generally used in recommender systems when faced some practical problems. Due to the sparsity of the rating matrix, the traditional CF-based approach has a significant decline in recommendation performance. Gradually, a hybrid method, using side information and rating information, has been widely employed and achieves great performance. Together with side information and rating information, the hybrid method can overcome the data sparsity and cold-start problems. However, they seem to fail to take into consideration the fact that the sparsity of single side information. To solve this problem, we take full advantage of the characteristics of deep learning that can learn effective representation and propose a novel deep learning model named additional variational autoencoder that considers both content and tag information of the item. The model learns effective latent representations from additional side information, including content information and tag information in an unsupervised manner. With the help of graphical models, it can extract the implicit relationships between users and items effectively. A large number of experimental results on two actual datasets show that our proposed model is superior to other methods, and the performance improvement is achieved.

INDEX TERMS Collaborative filtering, deep learning, variational autoencoder, side information.

I. INTRODUCTION

Recommender systems play a major role for many applications, such as social network, recommendation of products such as movies, games, music and articles [5], [9], [14]. Generally, various recommendation methods have been proposed over the past decade. These algorithms can be divided into three categories [1], [11] : content-based methods [10], [12], [13], collaborative filtering (CF) based methods [13], [15] and hybrid methods [7]. However, these approaches tend to suffer from the nature sparsity of the user-item rating data and cold start problem. Moreover, matrix factorization has became a useful method that used in recommendation about large datasets. The latent factors can be learned from matrix factorization from the user-item rating matrix [9]. However, the learned latent factors are not effective because of the sparsity of user-item rating matrix and side information [2]. Therefore, it is essential to find a way to solve the above problems.

A. SIDE INFORMATION FOR RECOMMENDER SYSTEMS

Recently, multiple hybrid methods fused additional information such as user-item information into the CF model to promote recommendation performance [15]. The side information can be obtained from users and items. The item information can contain a brief introduction from abstract and specific tag information. How to integrate multiple information and extracting the latent feature vector is an urgent task to promote recommendation performance.

B. DEEP LEARNING FOR RECOMMENDATION

Nowadays, deep learaning has been proved that can effectively learn the user and item representations to use in recommendation and achieve great performance [8]. However, there are still two critical issues with these algorithms. Firstly, the latent feature vectors are often not valid when the additional information is very sparse. Just like [16], it used the restricted Boltzmann machines to learn the latent vector and together with CF for recommendation. However, it did not fuse additional information and lead a common result. Moreover, collaborative topic regression (CTR) [19] achieves poor results when the additional side information is sparse. Secondly, some works only use single information as input to extract item feature. Consequently, the deep model is difficult to obtain and learn implicit relationships between users (or items) accurately. For example, [11] and [18] have been proposed for learning item feature vector from a stacked

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

denoising autoencoder (SDAE) and a variational autoencoder with collaborative filtering, which show promising performance. However, they only use the abstract information as input to extract the item feature. Addition information is not being used well. There is still room for improvement in performance.

C. MAIN CONTRIBUTIONS

In this paper, to solve the problems mentioned above, we proposed a hybrid deep learning model together with CF named additional variational autoencoder. It integrates additional information as inputs and extracts the feature vector from additional information. We use the probabilistic graphical model to learn the relationships between users (or items) and employ probabilistic matrix factorization for recommendation tasks.

The main contribution of this paper is threefold:

1) To the best of our knowledge, we propose a novel deep learning model named additional variational autoencoder (AVAE), which integrates additional information as inputs and extracts effective latent vector from additional information.
2) We present a hybrid deep learning model called CAVAE, which integrates AVAE model and probabilistic matrix factorization. With the help of probabilistic graphical model, CAVAE model can effectively learn the relationships between users or items.
3) We conduct experiments on two real-world datasets to evaluate the recommendation effectiveness of CAVAE model. Experimental results show that our model significantly outperform baseline methods.

II. PRELIMINARY

In this paper, we begin with elaborating the problem discussed in this paper and give a brief introduction about variational autoencoder (VAE).

A. PROBLEM DEFINITION

Similar to [17], we extract the implicit feedback data [9] from user-item rating matrix. The feedback data are treated as the training and test data. The task of our paper is to recommend suitable articles to every user. The additional information contains rating and side information included item content information that represents by bag of words and its tag information. We define a user-item rating matrix R_{ij}. The value $R_{ij} = 1$ implies that user i show interest in item j and $R_{ij} = 0$ otherwise. Just like the same procedure about rating matrix, the tag matrix T_{jt} are binary matrix. $T_{jt} = 1$ means that tag t is associated with item j and $T_{jt} = 0$ otherwise.

B. VARIATIONAL AUTOENCODER

As shown as in Fig. 1, variational autoencoder contains a recognition network and a generative network. The concrete implementation process is described as follows:

1) For input data x, draw data $x \sim p_\theta(x|z)$.

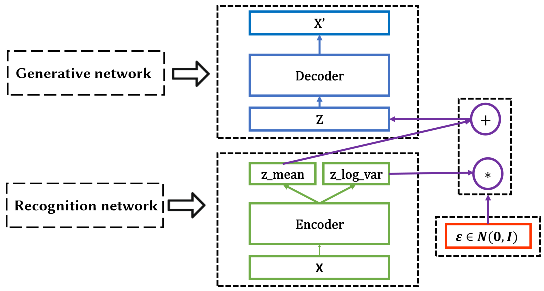


FIGURE 1. The model of VAE.

2) For generative network, draw a latent factor $z \sim p(z) = N(z|0, I)$.

Where x is the input information (user or item information) and z is the latent vector. The objective function of VAE includes the log likelihood about input data x and the KL divergence between the generative and the prior results. The corresponding cost function is as follows:

$$E\left[\log p_\theta\left(x|z\right)\right] - KL\left[q_\theta\left(z|x\right)||p\left(z\right)\right] \tag{1}$$

where $q_\phi(z|x)$ is the variational posterior that used to simulate the true posterior.

III. METHODOLOGY

In this section we first provide our model called AVAE and give a detailed introduction of our hybrid collaborative filtering model named CAVAE.

A. ADDITIONAL VARIATIONAL AUTOENCODER

As shown as in Fig. 2, we proposed a novel deep learning model called additional variational autoencoder, which can be divided into recognition process and generative process. AVAE model integrates the item content information and its concrete tag information as inputs. The blue balls mean the item content information. The corresponding tag information is represented by red balls. Compared with traditional variational autoencoder, we integrate additional information

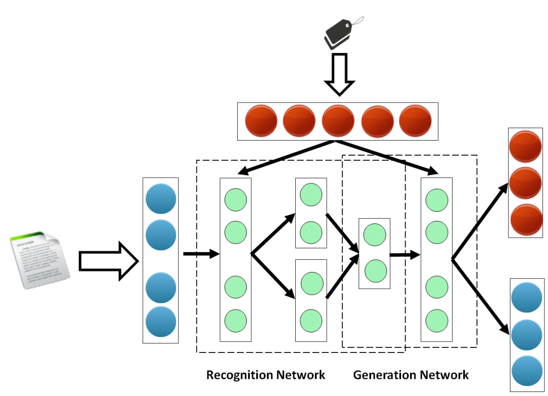


FIGURE 2. The model of AVAE. Our model contains recognition network and generation network.

as inputs and effectively learn the latent vector. Similar to [4], [6], and [13], we can define the two processes as follows:

1) Draw the weight matrix $W_l \sim N(0, \lambda_w^{-1} I)$
2) Draw the additional weight matrix $W_t \sim N(0, \lambda_t^{-1} I)$
3) Draw the bias vector $b_l \sim N(0, \lambda_w^{-1} I)$
4) Draw the additional bias vector $b_t \sim N(0, \lambda_t^{-1} I)$

For recognition network:

1) For each layer L of the recognition network
   a) For each row j of h_l, draw $h_l \sim N(\sigma(h_{l-1}W_l + b_l + T_a W_t + b_t), \lambda_s^{-1} I)$
2) For each item j, draw the hidden vectors:
   a) Draw the latent mean and covariance vector $\mu_j \sim N(h_l W_\mu + b_\mu, \lambda_s^{-1} I)$ $log\sigma_j^2 \sim N(h_l W_\sigma + b_\sigma, \lambda_s^{-1} I)$
   b) Draw the latent content vector $z_j \sim N(\mu_j, diag(\sigma_j))$

For generation network:

1) For each item $v_j \in V$
   a) Draw the content latent vector $\varepsilon_j \sim N(0, \lambda_v^{-1} I)$
2) For each layer L of the Additional Variational Autoencoder except for the output layer
   a) For each row j of h_l, draw $h_l \sim N(\sigma(h_{l-1}W_l + b_l + T_a W_t + b_t), \lambda_s^{-1} I)$
3) For the output layer
   a) For input content information: $x_j \sim N(h_l W_{l+1} + b_{l+1}, \lambda_j^{-1} I)$
   b) For input tag information: $x_t \sim N(h_l W_{t+1} + b_{t+1}, \lambda_t^{-1} I)$

Above, $\lambda_s, \lambda_v, \lambda_t, \lambda_i, \lambda_u, \lambda_j, \lambda_w$ are hyperparameters. Generally, λ_s goes to infinity. $\sigma(\cdot)$ is the sigmoid function. T_a is the tag information. AVAE model will degenerate to be a Bayesian nature. Just as shown as in Fig. 2, the first L/2 layers of the network act as the recognition process and the last L/2 layers of the network as the generation process. In our model, we assume the hidden layer as the latent factor, and the latent factor is generated from z_j. The z_j can be generated from standard norm distribution η. The concrete form is shown as follows $z_j = \mu_j + \sigma_j \odot \eta$.

B. ADDITIONAL VARIATIONAL AUTOENCODER FOR COLLABORATIVE FILTERING

As shown in Fig. 3. We use the additional variational autoencoder as a component that extracts the item latent vector, the generative process of collaborative filtering is shown as follows:

1) Through the AVAE network:
   a) For each item j, draw the corresponding latent vector $z_j \sim N(\mu_j, diag(\sigma_j))$
2) Draw a latent user vector for each user i: $u_i \sim N(0, \lambda_u^{-1} I)$
3) Draw a latent item offset vector for each item j: $\delta_j \sim N(0, \lambda_v^{-1} I)$ Together with the latent vector z_j, the item vector: $v_j = \delta_j + z_j$

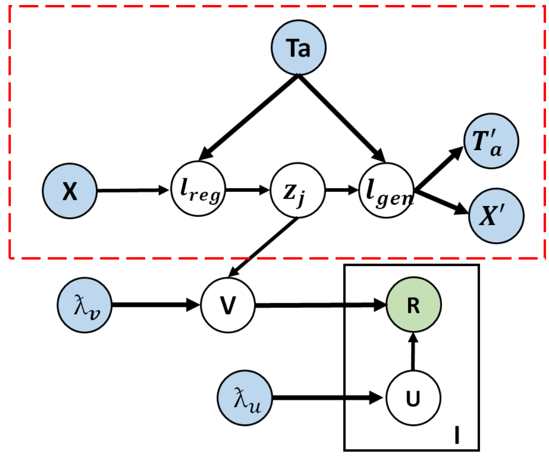


FIGURE 3. The model of CAVAE. The part inside the dashed rectangle represents AVAE model.

4) Draw a rating R_{ij} for each user-item pair (i, j): $R_{ij} \sim N(u_i^T v_j, C_{ij}^{-1})$

Here λ_v, λ_u, λ_t, λ_w are hyperparameters. The variable C_{ij} is a confidence parameter just the same as CDL [18]. The latent vector can be learned effectively from AVAE model. With the help of graphical model, CAVAE model can easily capture the implicit relationship between items. We can easily extend our model to other deep learning models, such as CNN and RNN because of its own Bayesian environment. Certainly, the additional information is not only tag information, but also it can be other binary content information.

C. OPTIMIZATION METHODOLOGY

Through the CAVAE network, we use maximum a poster probability estimator to learn our model parameters. The objective function contains three parts: latent loss, KL loss and regularization loss. Similar to the equation 1 and combine our AVAE model, the objective function thus becomes:

$$\begin{aligned}\ell = &-\sum_{i,j} \frac{C_{ij}}{2}(R_{i,j} - u_i^T V_j) - \frac{\lambda_w}{2}\sum_l (\| W_l \|_F^2 + \| b_l \|_2^2) \\ &-\frac{\lambda_t}{2}\sum_t (\| W_t \|_F^2 + \| b_t \|_2^2) - \frac{\lambda_u}{2}\sum_i \| u_i \|_2^2 \\ &+\frac{1}{L}\sum_{l=1}^{L} logp_\theta(x, t|z^{(l)}) - \frac{\lambda_v}{2}\sum_j E_{q\phi}(z|x,t) \| v_j - z_j \|_2^2 \\ &-KL[q_\phi(z|x,t)||p(z)] \end{aligned} \tag{2}$$

where the z_j is the content latent vector, L is the layer number of AVAE model. We use the stochastic gradient descent to train u_i and v_j. The u_i and v_j thus becomes:

$$u_i = (VI_iV^T + \lambda_U I_K)^{-1} VI_iR_i \tag{3}$$
$$v_j = (UI_jU^T + \lambda_V I_K)^{-1}(UI_iR_i + \lambda_v E_{q\phi}[z_j]) \tag{4}$$

where I_i is a diagonal matrix and I_{ij} are its elements. $R_i \in R_M$ is a vector of R_{ij}. $I_{ij} = R_{ij} = 0$ if user i has not yet rated item j.

For parameters W_l, W_t and biases b_l, b_t for each layer, we use the back-propagation algorithm to train them. For the parameters W_l and b_l, we use the same method to train them just like training W_t and b_t. The implementation process is as follows:

$$W_t = W_t - \delta \frac{\partial L_{(w_t)}}{\partial (w_t)} \tag{5}$$

$$b_t = b_t - \delta \frac{\partial L_{(b_t)}}{\partial (b_t)} \tag{6}$$

As for the mean parameter μ_j and the covariance vector σ_j, the gradient with respect to μ and σ is:

$$\mu_j = -\mu_j + \frac{1}{L}\sum_{l=1}^{L}\lambda_v(v_j - z_j) + \nabla_{z_j} log p_\theta(x_j, t_j|z_j) \tag{7}$$

$$\sigma_j = \frac{1}{\sigma_j} - \sigma_j + \frac{1}{L}\sum_{l=1}^{L}\lambda_v(v_j - z_j) + \nabla_{z_j} log p_\theta(x_j, t_j|z_j) \odot \eta \tag{8}$$

D. PREDICTION

We predict the rating value R_{ij} as follows:

$$E[R_{ij}|U, V, W^+, ...] \approx u_i^T(\delta_j + z_j) = u_i v_j \tag{9}$$

And then generate a sorted list of items for each user based on these predicted values.

IV. EXPERIMENTS

In this section, we will evaluate our model on two actual datasets to compare its performance with the number of newly recommended technologies reported in the literature.

A. DATASETS AND EVALUATION METRICS

We use two public datasets from CiteULike.[1] The first dataset, citeulike-a, is collected by Wang and Blei [17] and the second dataset citeulike-t is collected by Wang *et al.* [19]. See table 1. The text information contains abstract and title information. After preprocessing the content information, the vocabulary number of each dataset is shown in table 1. The text information of each article has been preprocessed using the same procedure as that [17]. We preprocess the content data using the bag of words and get a content vector for each item. All vectors are normalized. Each article has its own tag information. The corresponding tag matrix T_{jt} can be used to represent the tag information for all items. Each matrix entry T_{jt} is a binary value, where $T_{jt} = 1$ means that tag t is associated with item j and $T_{jt} = 0$ otherwise.

In each dataset, similar to [19], we use the same approach for datasets. We select the data of N items from the dataset as the training set and the rest of the dataset is used as the test set. The number of N is 1 and 10 as the sparse setting and dense setting, respectively. The results are obtained under repeated ten times. Following [19], we use recall rates as indicators to evaluate preformance. Like most recommender systems, we rank the predictive ratings of candidate articles and recommend the first N items to the target user. The recall rate is defined as follows:

$$recall@M = \frac{items\ that\ users\ like\ among\ the\ top\ M}{items\ that\ the\ user\ likes} \tag{10}$$

B. BASELINE MODELS

Here, we compare our method, which is named as "CAVAE," against the following baselines:

1) CTR [17]. Collaborative Topic Regression is a model that integrates topic modeling and collaborative filtering for recommendation.
2) CDL [18]. Collaborative Deep Learning is a probabilistic model that combines deep learning model about stacked denoising autoencoder (SDAE) with collaborative filtering. CDL is a classical algorithm and achieves better performance among recommendation algorithms.
3) CVAE [11]. Collaborative Variational Autoencoder is an advanced arithmetic based on CDL and using variational autoencoder (VAE) to replace the stacked densoing autoencoder. It achieves performance enhancements compared with CDL.

C. HYPERPARAMETERS SETTINGS

In the experiment, we use a validation set to find the optimal hyperparameters for CTR, CDL, CVAE, and CAVAE. The concrete statistic is shown in table 2. We also pretrained our model in plain AVAE to train the hyperparameters. For CVAE, the recognition network and generation network are set to be a two-layer network (600-300) for recognition network and (300-600) for generation network. Sigmoid function is used as the activation function. For CAVAE, the settings are the same as the CVAE that are used for fairness.

[1] http://www.citeulike.com

TABLE 1. Post-processed datasets statistics.

	Citeulike-a	citeulike-t
#user	5551	7947
#item	16980	25975
#user-item	204986	134860
#tag	7836	8311
#tag-item	204987	134860
#sparsity	99.87%	99.93%
#vocabulary	8000	20000

TABLE 2. Hyperparameter settings of our framework for the two datasets.

	Parameter Settings
#CAVAE	$a = 1$, $b = 0.01$, $k = 50$, $dim = 4$
#CVAE	$a = 1$, $b = 0.01$, $k = 50$, $dim = 4$, $\lambda_r = 10$
#CDL	$a = 1$, $b = 0.01$, $\lambda_u = 1$, $\lambda_v = 1000$, $k = 50$, $\lambda_w = 0.0001$, $noise = 0.3$, $dim = 4$
#CTR	$a = 1$,$b = 0.01$,$\lambda_u = 0.1$,$\lambda_v = 1$,$k = 50$

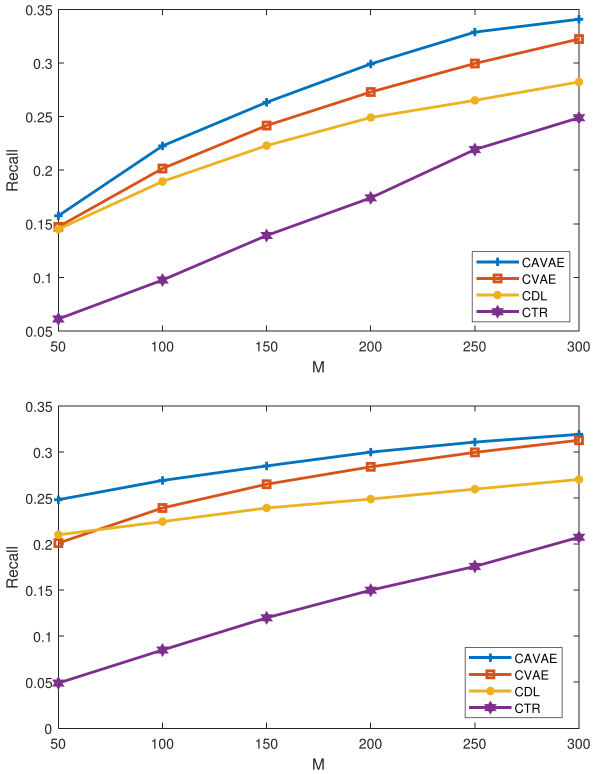


FIGURE 4. **Performance comparison of CAVAE, CVAE, CDL, CTR based on recall@M for datasets citeulike-a, citeulike-t in the sparse setting.**

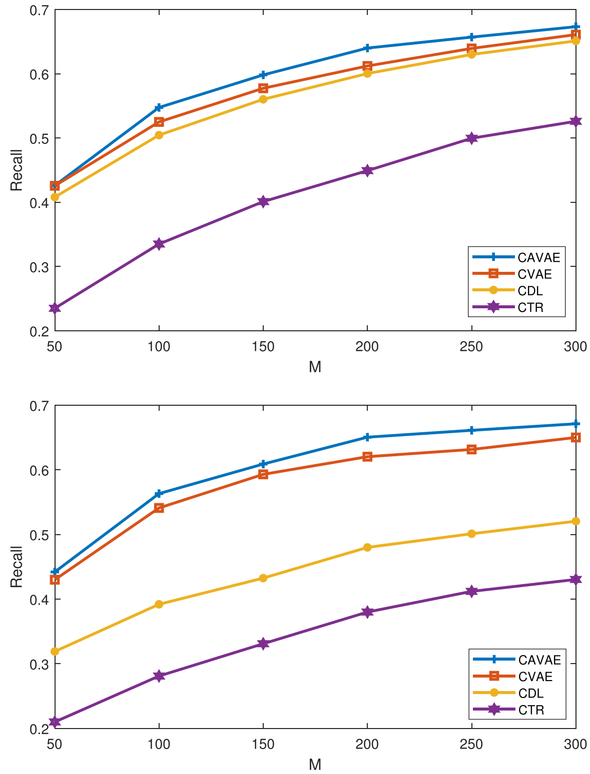


FIGURE 5. **Performance comparison of CAVAE, CVAE, CDL, CTR based on recall@M for datasets citeulike-a, citeulike-t in the dense setting.**

D. EVALUATION RESULTS

The specific evaluation results of all the compared models are presented in Fig. 4 in sparse setting and Fig. 5 in dense setting. We note that our model outperforms all the baseline models across all the two datasets. The evaluation results actually meet our expectations. We can observe from the Fig. 4 and Fig. 5 that CAVAE, CVAE and CDL achieve better performance than CTR. We also find that CDL outperforms better in citeulike-a compare with citeulike-t. It demonstrates two problems: First, CAVAE, CVAE, CDL can learn effective latent vector compare with CDL. Second, we can know that the representation capability of CTR is limited to the topic model and the potential representation learned is often not effective, especially when the additional information is very sparse. Moreover, CVAE and our model outperform CDL. That is, instead of corrupting the input data, our model and CVAE model seeks for a probabilistic latent variable model for content. Corresponding, CDL can easily overfit data. Furthermore, we can see that our proposed model obtains higher recall than CVAE, CDL and CTR, which validates the strength of the latent vectors learned by our CAVAE. It integrates additional information as inputs to learn the latent vector compared with CVAE. CVAE extracts the latent vector from content information and the recall decreases significantly when the single side information is very sparse. Hence, the Recall metric demonstrates the effectiveness of CAVAE model.

We also evaluate the sensitivity to hyperparameters. We use the same procedure like [18] to evaluate the performance of CAVAE with different K values based on recall for datasets in sparse setting and dense setting. The K value is the number of latent factors. From Fig. 3 we know that different values K make a difference between the latent vector of content and the latent variable for the probabilistic matrix factorization model. Fig. 6 shows that our model can not learn a good representation from the input information including content information and tag information when K value is very small such as 10. Meanwhile, CAVAE model can not learn a great representation when the K is enough larger because its representation capability is enough for the input information. However, the opposite situation appears in the dense setting, we can see that the larger K can achieve better performance, this is because denser ratings provide more guidance for reasoning networks to perform variational reasoning, and therefore require greater representation capability to learn. We did the similar experiment to verify our conclusion in citeulike-t just as shown in Fig. 7.

E. DISCUSSION

In order to evaluate the impact of tag information on recommendation effectiveness, we train our model and CVAE in the sparse setting with the dataset citeulike-a. The corresponding recommendatory top 10 results for user are shown in Table 3 and Table 4. The instance article we

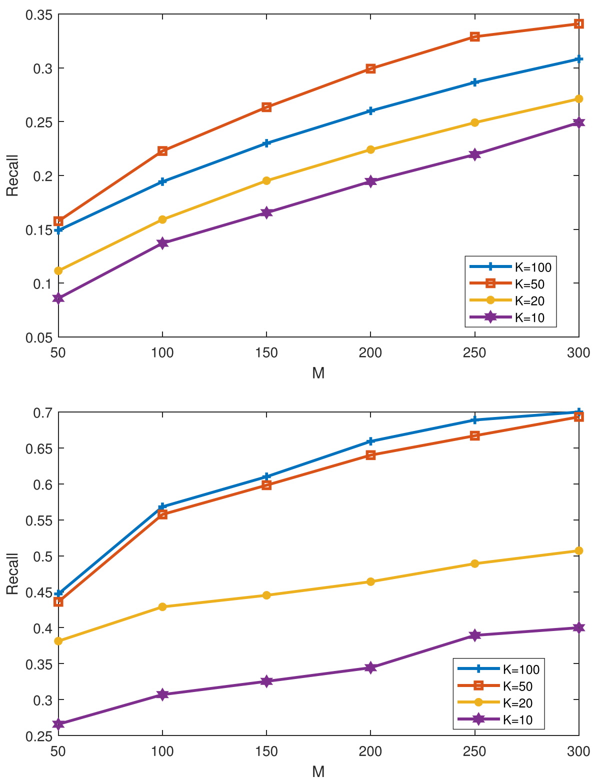


FIGURE 6. Performance comparison of different K values based on the dataset citeulike-a in sparse setting and dense setting.

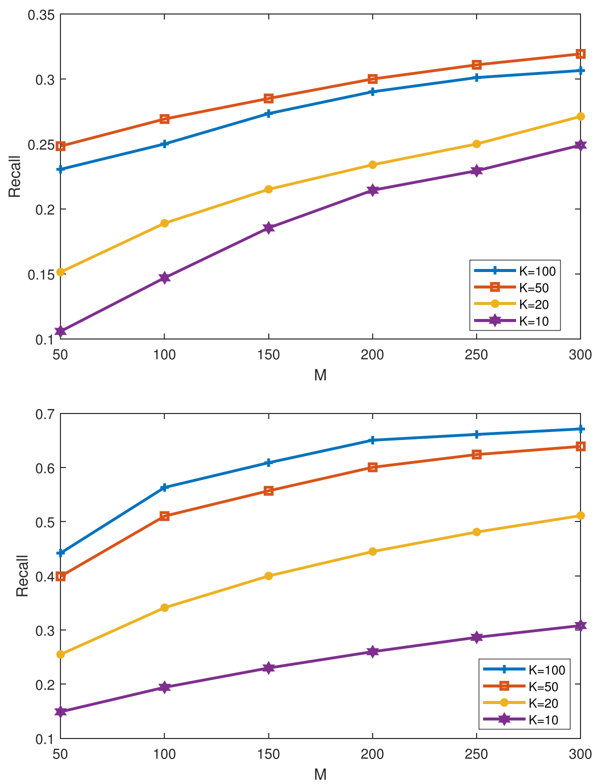


FIGURE 7. Performance comparison of different K values based on the dataset citeulike-t in sparse setting and dense setting.

TABLE 3. Actual recommendation results about CAVAE.

	User I(CAVAE)	is user's lib?
#tag information	Naturalcapital, encode pilot, pulse characterization, text analysis, collaborative	
#article 1	**Reverse engineering of biological complexity.**	**yes**
#article 2	**Diffusion on complex networks a way to probe their large scale topological structures.**	**yes**
#article 3	**Serendipity and information seeking an empirical study.**	**yes**
#article 4	**Collaborating with writing tools an instrumental perspective on the problem of computer supported.**	**yes**
#article 5	**Folksonomies cooperative classification and communication through shared metadata.**	**yes**
#article 6	Information processing with population codes.	no
#article 7	**Creating the gene ontology resource design and implementation.**	**yes**
#article 8	Probabilistic latent semantic indexing.	no
#article 9	**Identification of transcription factor binding sites with variable order Bayesian networks.**	**yes**
#article 10	**Coarsegraining and self dissimilarity of complex networks.**	**yes**

TABLE 4. Actual recommendation results about CVAE.

	User I(CVAE)	is user's lib?
#tag information	Naturalcapital, encode pilot, pulse characterization, text analysis, collaborative	
#article 1	**Reshaping the gut microbiome with bacterial transplantation and antibiotic intake.**	**yes**
#article 2	**High resolution analysis of parent of origin allelic expression in the mouse brain.**	**yes**
#article 3	**From computational science to inter metics integration of science with computer science.**	**yes**
#article 4	Cloud computing for comparative genomics.	no
#article 5	Natural selection on cis and trans regulation in yeasts.	no
#article 6	Bayesian analysis of chip seq data.	no
#article 7	**Local dna topography correlates with functional noncoding regions of the human genome.**	**yes**
#article 8	**Optical deconstruction of parkinsonian neural circuitry.**	**yes**
#article 9	Macromolecular modeling with rosetta.	no
#article 10	**GeNGe: systematic generation of gene regulatory networks.**	**yes**

selected is "Topllogical generalized of network motifs." The results show that our model can provide diversity results because of the additional tag information. AVAE model can automatically learn the user preference together with tag information. Compare with CVAE model, our proposed model integrates additional information as inputs to extract the item latent vector. The experimental results in Fig. 4 And Fig. 5 show that AVAE model can outperform all the baselines and by a margin about 3% $\sim$ 5% compare with CVAE. Our model identified user I as a learner about network. Together with the tag information of item, CAVAE model can extend user preference such as text analysis and related network technologies. However, CVAE can only recommend some information about network because of its single content information.

V. CONCLUSION

In this paper, we proposed a deep learning model called additional variational autoencoder (AVAE) and a hybrid collaborative filtering model named collaborative additional variational autoencoder (CAVAE). Our AVAE model integrates additional information as inputs compared traditional variational autoencoder, which overcomes the sparsity of single side information and learns the latent factor accurately. CAVAE model is a Bayesian probabilistic model that fuses the AVAE model. Together with graphical model, CAVAE model can effectively learn the implicit relationships between users or items. Our experimental results present that our model can achieve great performance compared with the state-of-the-art methods.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] D. Agarwal, B. C. Chen, and B. Long, "Localized factor models for multi-context recommendation," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 609–617.

[3] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[4] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising autoencoders as generative models," in *Proc. NIPS*, 2013, pp. 899–907.

[5] P. G. Campos, F. Díez, and I. Cantador, "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols," *User Model. User-Adapted Interact.*, vol. 24, nos. 1–2, pp. 67–119, 2014.

[6] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. ICML*, 2012, pp. 767–774.

[7] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proc. WWW*, 2013, pp. 595–606

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[10] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[11] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proc. 23th SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 305–314.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[13] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[14] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. UAI*, 2009, pp. 452–461.

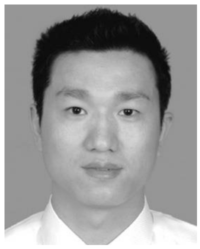
[15] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, 2014.

[16] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 791–798.

[17] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 448–456.

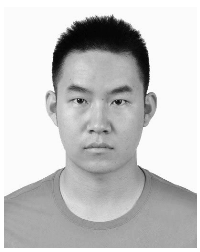
[18] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1235–1244.

[19] H. Wang, B. Chen, and W. J. Li, "Collaborative topic regression with social regularization for tag recommendation," in *Proc. IJCAI*, 2013, pp. 2719–2725.



MING HE received the B.S. degree in computer and application and the M.S. degree in oil–gas well engineering from the Xi'an Petroleum Institute, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, China, in 2005. From 2006 to 2011, he was a Research Assistant with the College of Computer Science, Beijing University of Technology, Beijing, China. Since 2012, he has been an Assistant Professor with the Faculty of Information Technology, Beijing University of Technology. He has authored more than 60 articles. His research interests include recommendation systems, data mining, and machine learning.

Dr. He is a member of the International Association of Computer Science and Information Technology. His awards and honors include the Excellent Teachers Award, in 2015, and the Beijing University of Technology Best Academic Paper Award, in 2010.



QIAN MENG was born in Zhoukou, China, in 1996. He received the B.S. degree from the School of Computer Science and Technology, Qilu University of Technology, in 2017.

He is currently pursuing the master's degree with the Beijing University of Technology, instructed by Prof. He. His main research interests include deep learning and recommender system.

SHAOZONG ZHANG received the B.S. degree in network engineering from Hebei University, China, in 2017. He is currently pursuing the master's degree with the Beijing University of Technology, instructed by Prof. He. His main research interests include deep learning and recommender systems.