VARIATIONAL BAYESIAN GRAPH CONVOLUTIONAL NETWORK FOR ROBUST COLLABORATIVE FILTERING

Nozomu Onodera[†], *Keisuke Maeda*[††], *Takahiro Ogawa*[†††], *Miki Haseyama*[†††]

[†]School of Engineering, Hokkaido University, Japan
[††]Office of Institutional Research, Hokkaido University, Japan
[†††]Faculty of Information Science and Technology, Hokkaido University, Japan
E-mail:{onodera, maeda, ogawa}@lmd.ist.hokudai.ac.jp, miki@ist.hokudai.ac.jp

ABSTRACT

This paper presents a variational Bayesian graph convolutional network for robust collaborative filtering (VBGCF). Conventional graph convolutional network (GCN)-based recommendation models fully trust the observed interaction graph. However, the data used in real-world applications (e.g., video streaming services) are often incomplete and unreliable. To deal with this realistic situation, we newly introduce the probabilistic model based on variational Bayesian inference to GCN-based recommendation. VBGCF can use various generated graphs instead of the observed interaction graph to learn users' preferences. Therefore, VBGCF is not affected by the incompleteness and the unreliability and can provide robust recommendation. The results of experiments conducted under the realistic situation show the effectiveness of VBGCF.

***Index Terms*—** Recommender systems, collaborative filtering, graph convolutional networks, variational Bayesian inference.

1. INTRODUCTION

The increasing amount of information on the Internet makes it difficult to find items that users truly like. To deal with this problem, the construction of personalized recommender systems is an effective solution [1]. Its core is modeling users' preferences from their historical interaction data (e.g., purchase, click and view logs). Collaborative filtering (CF) models tackle it by assuming that users who share similar interactions have similar preferences [2]. The basic paradigm of implementing CF is learning representations of users and items and predicting recommendation items for each user based on the similarity of the representations [3, 4, 5]. One of the most popular methods of CF is matrix factorization (MF) [6], which decomposes the user-item interaction matrix into two low-rank matrices as the user and item representations. Many subsequent models extending MF have enhanced the performance of recommendation by using neural networks [5, 7], the tensor factorization [8], the autoencoder [9] and the variational autoencoder [10].

Recently, graph convolutional network (GCN)-based recommendation models have realized considerable success for accuracy. GCN [11] is the deep learning model for graph data and has attracted a lot of attention due to its high representation ability [12]. Since user-item interactions can be considered as the bipartite graph (i.e., interaction graph), GCN, which can capture high-order interactions of graphs, is suitable for the recommendation. Thus, it has been demonstrated that GCN is effective for recommendation [13, 14, 15, 16].

Although GCN-based recommendation models have achieved remarkable success, most existing models disregard the following two important issues.

- **Incompleteness:** The observed graph cannot capture all interactions. For example, many users may use multiple platforms, and their interactions with others are not recorded. It is difficult to truly capture users' preferences from such incomplete data.
- **Unreliability:** Since the recommendation service is available to any users, malicious users can easily add fake interactions to control the recommender system (i.e., shilling attack) [17]. Therefore, the observed data have many spurious interactions to degrade recommendation accuracy.

Although it is essential to consider these uncertainty issues under the realistic situation, existing models use the observed interaction graph with complete confidence and do not consider them.

The Bayesian approach is an effective method to deal with the above uncertainty issues, and this approach has been recently introduced to various tasks using the deep learning [18]. Bayesian graph collaborative filtering (BGCF) [19] is one of GCN-based recommendation models introducing the Bayesian approach. This model allows considering the uncertainty in the observed interaction graph by using a Bayesian graph convolutional network [20], which is a probabilistic GCN model that introduces a random graph generative model. Since the computation of a random graph posterior is intractable, the graph generative model of BGCF is based on a *node copying* [21], which generates graphs by replacing the one-hop neighborhoods of some nodes with those of other nodes in the observed graph. However, despite the probabilistic model, the variety of graphs generated by this model is limited due to the simple node copying. To solve this problem, we need a more flexible graph generative model.

In this paper, we propose a variational Bayesian graph convolutional network for robust collaborative filtering (VBGCF). To solve the above problems, we use variational Bayesian inference (VBI) for calculating a graph posterior. VBI is one of the most effective methods for the approximate calculation of the intractable posterior. The posterior optimized by VBI can eliminate the limitations of BGCF and can generate more flexible graphs. This is our main contribution in this paper. When we introduce VBI, we cannot directly adopt the gradient method for optimizing the posterior due to its discrete distribution. Hence, inspired by [22], we introduce the continuous relaxation based on the concrete distribution [23]. The support of the relaxed distribution is continuous, and it becomes feasible to ap-

This work was partly supported by JSPS KAKENHI Grant Numbers JP17H01744 and JP20K19856.

ply the gradient method. Consequently, since VBGCF uses various graphs generated by the optimized posterior instead of the observed graph, this model can provide robust recommendation without being affected by the incompleteness and the unreliability. The experimental results show that our method is more robust compared to recent and state-of-the-art recommendation models.

2. PRELIMINARIES

2.1. Graph Convolutional Network for Recommendation

In the recommendation using implicit feedbacks, we can observe user-item interaction data, i.e., the set of all users, $\mathcal{U}$, the set of all items, $\mathcal{I}$, and sets of items interacted by user u, $\mathcal{I}_u^+$. These data can be regarded as the user-item interaction graph $G_{\rm obs} = (\mathcal{V}_{\rm obs}, \mathcal{E}_{\rm obs})$, where $\mathcal{V}_{\rm obs} = \mathcal{U} \cup \mathcal{I}$ and $\mathcal{E}_{\rm obs} = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}_u^+\}$. In other words, $G_{\rm obs}$ is the bipartite graph whose nodes mean users and items, and edges mean interactions in the observed data. The main challenge of CF is how to get the user and item representations, and many recommendation models using the graph learn these representations based on node embeddings.

Node-level GCN [11] is a deep neural network model to embed nodes. The GCN layer's operation is equivalent to the aggregation of features from 1-hop neighborhoods and the update of node representations. By stacking multiple layers, the nodes can aggregate features from their multi-hop neighborhoods. Nowadays, various GCN models exist depending on how to implement the aggregation method.

LightGCN [16] is a state-of-the-art model for GCN-based recommendation. The aggregation method of this model is implemented simply without an affine transformation and a nonlinear transformation. In each l-th layer, the node embedding of user u is calculated by using its 1-hop neighborhood embeddings as follows:

$$\boldsymbol{e}_u^{(l)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \boldsymbol{e}_i^{(l-1)}, \tag{1}$$

where $\boldsymbol{e}_u^{(l)}$ and $\boldsymbol{e}_i^{(l)}$ respectively denote node embeddings of user u and item i from l-th layer, and $\mathcal{N}_u$ and $\mathcal{N}_i$ respectively denote the neighborhood sets of the nodes of user u and item i. Similarly, the node embedding of item i is calculated as follows:

$$\boldsymbol{e}_i^{(l)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_u|}} \boldsymbol{e}_u^{(l-1)}. \tag{2}$$

To aggregate from L-hop neighborhoods, Eqs. (1) and (2) are iterated L times.

After computing the node embeddings in all layers, the user and item representations $\boldsymbol{h}_u$ and $\boldsymbol{h}_i$ are obtained by using all embeddings from each layer as follows:

$$\boldsymbol{h}_u = \frac{1}{L+1} \sum_{l=0}^{L} \boldsymbol{e}_u^{(l)}, \tag{3}$$

$$\boldsymbol{h}_i = \frac{1}{L+1} \sum_{l=0}^{L} \boldsymbol{e}_i^{(l)}. \tag{4}$$

It is beneficial to use not only embeddings from the final (L-th) layer but also those from other layers since they have different semantics reflecting users' preferences. Finally, a total item ranking for all users, $\{>_{\mathcal{U}}\}$, is generated based on the similarity of the user and item representations, where $\{>_{\mathcal{U}}\} = \prod_{u \in \mathcal{U}} \{>_u\}$, and $\{>_u\}$ is a ranking set of all items for user u.

When the observed interaction graph is highly reliable, LightGCN can provide the effective recommendation. However, we need to suppose the scenarios where interaction graphs are incomplete and unreliable. To deal with the uncertainty issues, we focus on the Bayesian approach, which regards the parameters of the model as random variables and derives joint probability distributions of these parameters.

2.2. Bayesian Graph Collaborative Filtering

BGCF [19] is the GCN-based recommendation model introducing the Bayesian approach to incorporate the uncertainty in the interaction graph. Specifically, a set of GCN parameters θ, a graph $G = (\mathcal{V}_{\rm obs}, \mathcal{E})$ and the item ranking $\{>_{\mathcal{U}}\}$ are regarded as random variables. The idea of BGCF is to focus on the predictive distribution of the item ranking given the interaction graph, which is obtained by marginalizing over θ and G as follows:

$$p(\{>_{\mathcal{U}}\}|G_{\rm obs}) \propto \sum_G \int p(\{>_{\mathcal{U}}\}|G,\theta) p(\theta|G_{\rm obs}, G) d\theta \cdot p(G|G_{\rm obs}). \tag{5}$$

Since this summation with respect to G is intractable, the predictive distribution is approximated by a simple Monte Carlo approximation as follows:

$$p(\{>_{\mathcal{U}}\}|G_{\rm obs}) \approx \frac{1}{S} \sum_{s=1}^{S} \int p(\{>_{\mathcal{U}}\}|G^{(s)},\theta) p(\theta|G_{\rm obs}, G^{(s)}) d\theta, \tag{6}$$

where $G^{(s)}$ is generated from $p(G|G_{\rm obs})$, and S denotes the number of sampled graphs. Equation (6) implies that the sampled graphs directly affect the models to generate the parameters of GCN and the ranking. To maximize this predictive distribution, the integral with respect to θ is approximated by using the maximum a posteriori (MAP) estimation as follows:

$$p(\{>_{\mathcal{U}}\}|G_{\rm obs}) \approx \frac{1}{S} \sum_{s=1}^{S} p(\{>_{\mathcal{U}}\}|G^{(s)}, \widehat{\theta}^{(s)}), \tag{7}$$

where $\widehat{\theta}^{(s)} = \arg\max_\theta p(\theta|G_{\rm obs}, G^{(s)})$. Thus, recommendation items can be predicted from this predictive distribution.

In order to generate graphs, it is crucial to design the graph posterior $p(G|G_{\rm obs})$. In BGCF, this posterior is based on the *node copying*, which generates graphs by replacing the one-hop neighborhoods of some nodes with those of other nodes in the observed graph. In other words, the one-hop neighborhood set of user u, $\mathcal{N}_u$ is probabilistically replaced with that of another user u', $\mathcal{N}_{u'}$. However, this generative model can only generate a limited variety of graphs due to its simple node copy algorithm. In addition, this model does not work well for personalized recommendations since it is difficult to individualize preferences by copying the one-hop neighborhoods, which means copying interactions.

3. PROPOSED MODEL : VBGCF

VBGCF is the novel recommendation model that calculates the approximate graph posterior by VBI and predicts the item ranking based on graphs generated from the posterior. This model is robust to

the incompleteness and the unreliability of the data since it does not fully trust the observed graph. Generally, the main procedure of VBI is the maximization of the evidence lower bound (ELBO) instead of the minimization of Kullback-Leibler (KL) divergence between the true and approximate posterior distributions. Thus, we formulate the predictive distribution of the item ranking and the ELBO in Sec. **3.1**. We then provide a specific methodology to maximize the ELBO in Sec. **3.2**.

3.1. Predictive Distribution and Evidence Lower Bound

In this subsection, we describe the definition of the predictive distribution and the ELBO in VBGCF. We focus on the predictive distribution of the item ranking as follows:

$$p(\{>_u\}|G_{\rm obs}) = \sum_G p_\theta(\{>_u\}|G)p(G|G_{\rm obs}). \quad (8)$$

Note that this equation corresponds to Eq. (5) in the previous section, however, we regard likelihood parameters as GCN parameters θ. This is because GCN parameters require the point estimation as BGCF adopts the MAP estimation. As the graph posterior $p(G|G_{\rm obs})$, BGCF uses the node copying. Instead of this inflexible generative model, we introduce VBI to calculate the approximate posterior. Let $q_\phi(G)$ be the approximate posterior and ϕ be a set of variational parameters, and the predictive distribution is approximated as follows:

$$\begin{aligned} p(\{>_u\}|G_{\rm obs}) &\approx \sum_G p_\theta(\{>_u\}|G)q_\phi(G) \\ &\approx \frac{1}{S}\sum_{s=1}^{S} p_\theta(\{>_u\}|G^{(s)}), \end{aligned} \quad (9)$$

where $G^{(s)}$ is generated from $q_\phi(G)$. While $G^{(s)}$ in Eq. (7) is generated by simply copying some user nodes on the observed graph $G_{\rm obs}$, $G^{(s)}$ in Eq. (9) is edged individually for each user node. Therefore, VBGCF can provide more flexible probabilistic interpretation. Obviously, since our approach is based on the Bayesian probabilistic model, VBGCF can consider the uncertainty issues which conventional deterministic models cannot deal with.

We assume the graph prior as follows:

$$\begin{aligned} p(G) &= \prod_{u\in\mathcal{U}}\prod_{i\in\mathcal{I}} p\left(a_{ui}\right), \\ p\left(a_{ui}\right) &= {\rm Bern}(a_{ui}|\mu_{ui}), \end{aligned} \quad (10)$$

where $a_{ui} = \mathbf{1}_{\mathcal{E}}[(u,i)]$, and $\mathbf{1}_{\mathcal{E}}[\cdot]$ is the indicator function. Specifically, a_{ui} is 1 if G has the edge (u,i), otherwise 0. Then ${\rm Bern}(a_{ui}|\mu_{ui})$ is the Bernoulli distribution over a_{ui} with a parameter μ_{ui}. We can set this parameter μ_{ui} in various ways to encode the confidence in the observed graph. In our model, by following [22], it is defined as $\mu_{ui} = \mu_0(1-\bar{a}_{ui}) + \mu_1\bar{a}_{ui}$, where $\bar{a}_{ui} = \mathbf{1}_{\mathcal{E}_{\rm obs}}[(u,i)]$ and $\mu_0, \mu_1 \in (0,1)$ are hyper-parameters. Similar to the prior, we assume the approximate posterior as follows:

$$\begin{aligned} q_\phi(G) &= \prod_{u\in\mathcal{U}}\prod_{i\in\mathcal{I}} q_\phi\left(a_{ui}\right), \\ q_\phi\left(a_{ui}\right) &= {\rm Bern}(a_{ui}|\rho_{ui}), \end{aligned} \quad (11)$$

where $\rho_{ui} > 0$ is a parameter that denotes the probability of $a_{ui} = 1$. We can write a set of variational parameters as $\phi = \{\rho_{ui}\}$.

Another important component in Eq. (9) is the likelihood $p_\theta(\{>_u\}|G)$. We define the likelihood based on the idea of the Bayesian personalized ranking (BPR) loss [24], which is a typical ranking loss, as follows:

$$\begin{aligned} p_\theta(\{>_u\}|G) &= \prod_{u\in\mathcal{U}} p_\theta(\{>_u\}|G) \\ &= \prod_{(u,i,j)\in\mathcal{D}} \sigma(x_{uij}(G;\theta)), \end{aligned} \quad (12)$$

where $\mathcal{D} = \{(u,i,j)|u\in\mathcal{U}, i\in\mathcal{I}_u^+ \wedge j\in\mathcal{I}\backslash\mathcal{I}_u^+\}$ and $\sigma(\cdot)$ is the sigmoid function. Then $x_{uij}(G;\theta)$ denotes the relationship between item i interacted by user u and that not interacted by u and is defined as follows:

$$x_{uij}(G;\theta) = \boldsymbol{h}_{u,G}\cdot\boldsymbol{h}_{i,G} - \boldsymbol{h}_{u,G}\cdot\boldsymbol{h}_{j,G}, \quad (13)$$

where $\boldsymbol{h}_{u,G}$ is the representation of user u obtained by GCN with G. Similarly, $\boldsymbol{h}_{i,G}$ and $\boldsymbol{h}_{j,G}$ are the item representations. $\sigma(x_{uij}(G;\theta))$ in Eq. (12) denotes the probability that user u prefers item i over item j. Therefore, maximizing this likelihood is equivalent to estimate the parameters that increase the user u's preference for item i when G has the edge (u,i).

Originally, for joint inference over the graph G and the GCN parameters θ, we need to find the true posterior distribution of the graph, i.e., $p(G|\{>_u\}) \propto p(\{>_u\}|G)p(G)$. However, since this true posterior is intractable, we need to rely on VBI. The ELBO $\mathcal{L}_{\rm ELBO}$ is defined with the prior $p(G)$ in Eq. (10) and the approximate posterior $q_\phi(G)$ in Eq. (11) instead of using the true posterior as follows:

$$\begin{aligned} \ln p(\{>_u\}|G_{\rm obs}) &= \ln\sum_G p_\theta(\{>_u\}|G)p(G) \\ &\geq \sum_G q_\phi(G)\ln\frac{p_\theta(\{>_u\}|G)p(G)}{q_\phi(G)} \\ &= -{\rm KL}[q_\phi(G)|p(G)] \\ &\quad +\mathbb{E}_{q_\phi(G)}[\ln p_\theta(\{>_u\}|G)] \\ &=: \mathcal{L}_{\rm ELBO}, \end{aligned} \quad (14)$$

where the second line is based on the Jensen inequality. We estimate the GCN parameters θ and the variational parameters ϕ by maximizing this ELBO. Since this ELBO has discrete (Bernoulli) distributions, we cannot directly use the gradient method. Hence, we provide the alternative methodology to maximize ELBO in the following subsection.

3.2. Continuous Relaxation

The support of the Bernoulli distribution shown in Eqs. (10) and (11) is $a_{ui}\in\{0,1\}$. The previous work [22] introduced the continuous relaxation of the discrete distribution to maximize the ELBO for the Bernoulli distribution by the gradient method. Specifically, they used the Binconcrete distribution, whose support of the density function is (0,1). Then we redefine the posterior of Eq. (11) as $q_\phi(\tilde{a}_{ui}) = {\rm Binconcrete}(\tilde{a}_{ui}|\lambda_{ui},\tau)$, where $\tilde{a}_{ui}\in(0,1)$ and

$$\begin{aligned} &{\rm Binconcrete}(\tilde{a}_{ui}|\lambda_{ui},\tau) \\ &= \frac{\tau\exp\left(-\tau\left(\tilde{a}_{ui} - \frac{\ln\lambda_{ui}}{\tau}\right)\right)}{\left(1+\exp\left(-\tau\left(\tilde{a}_{ui} - \frac{\ln\lambda_{ui}}{\tau}\right)\right)\right)^2}. \end{aligned} \quad (15)$$

Here, τ is a temperature parameter that denotes how much the distribution is relaxed, and λ_{ui} is an alternative variational parameter to

Table 1. Experimental results. %Malicious denotes the number of injected malicious users. R and N respectively denote Recall and NDCG.

%Malicious	0%				5%				10%			
Metric	R@20	R@50	N@20	N@50	R@20	R@50	N@20	N@50	R@20	R@50	N@20	N@50
BPR-MF [24]	0.0154	0.0340	0.0354	0.0386	0.0146	0.0326	0.0337	0.0370	0.0167	0.0383	0.0412	0.0446
LightGCN [16]	0.1449	0.2605	0.3038	0.3005	0.1311	0.2401	0.2780	0.2763	0.1272	0.2306	0.2680	0.2652
BGCF [19]	**0.1803**	**0.3167**	0.1486	0.2003	**0.1575**	0.2707	0.1205	0.1624	0.1324	0.2333	0.0979	0.1354
VBGCF	0.1673	0.2909	**0.3431**	**0.3365**	0.1527	**0.2715**	**0.3199**	**0.3159**	**0.1437**	**0.2546**	**0.3000**	**0.2971**
%Improv.	-	-	12.9%	12.0%	-	0.3%	15.1%	14.3%	8.5%	9.1%	11.9%	12.0%

Algorithm 1 Calculation of the predictive distribution in VBGCF

Require: Interaction graph $G_{\rm obs} = (\mathcal{V}_{\rm obs}, \mathcal{E}_{\rm obs})$
Ensure: Predictive distribution $p(\{>_u\}|G_{\rm obs})$
1: Construct the prior distribution $p(G)$ shown in Eq. (10).
2: Find θ^* and ϕ^* to maximize the ELBO shown in Eq. (16).
3: Sample graphs from the approximate posterior $q_{\phi^*}(\tilde{G}) = \prod_{u\in\mathcal{U}}\prod_{i\in\mathcal{I}} q_{\phi^*}(\tilde{a}_{ui})$, where $q_{\phi^*}(\tilde{a}_{ui})$ is shown in Eq. (15).
4: Calculate the predictive distribution $p_{\theta^*}(\{>_u\}|G_{\rm obs}) \approx \frac{1}{S}\sum_{s=1}^{S} p_{\theta^*}(\{>_u\}|\tilde{G}^{(s)})$ shown in Eq. (9).

ρ_{ui} in Eq. (11). As a result, the ELBO is written as follows:

$$\begin{aligned}\mathcal{L}_{\rm ELBO} &= -{\rm KL}[q_\phi(\tilde{G})|p(G)] + \mathbb{E}_{q_\phi(\tilde{G})}[\ln p_\theta(\{>_u\}|\tilde{G})] \\ &\approx -{\rm KL}[q_\phi(\tilde{G})|p(G)] + \frac{1}{S}\sum_{s=1}^{S}\ln p_\theta(\{>_u\}|\tilde{G}^{(s)}), \quad (16)\end{aligned}$$

where $q_\phi(\tilde{G}) = \prod_{u\in\mathcal{U}}\prod_{i\in\mathcal{I}} q_\phi(\tilde{a}_{ui})$, and $\tilde{G}^{(s)}$ is generated from $q_\phi(\tilde{G})$. Since $\tilde{a}_{ui}$ is the continuous value, $\tilde{G}^{(s)}$ is the weighted graph. Note that we apply the gradient approximation of the likelihood in the second line. Consequently, the derivative of the ELBO becomes continuous values, which allow us to use the gradient method. Thus, we can use an Adam [25] optimizer. Finally, we summarize the procedure for calculating the predictive distribution of VBGCF in Algorithm 1.

4. EXPERIMENTAL RESULTS

We performed experiments under the realistic situation with Movielens1M [26] as the recommendation benchmark. This dataset has 6,040 users, 3,900 items, and 1,000,209 ratings in five grades from one to five. For using implicit feedbacks, we defined $i \in \mathcal{I}_u^+$ if user u gives item i a rating of 4 or 5, otherwise $i \notin \mathcal{I}_u^+$. We used only 50% of all interactions for each user as the training data, while the previous works [15, 16, 19] used more interactions. This setting considers the incompleteness in the real-world applications. The rest is used as the test data. In addition, to consider the existence of users that make fake feedbacks for their own benefit, we injected malicious users to the dataset by push random attacks. Numbers of malicious users are set to $\{0, 5, 10\}$% of the number of all users in the original dataset according to [27, 28]. The push random attack is one of the shilling attack methods, where malicious users give the highest rating to items they want to make easier to recommend for many users (i.e., target items), and then give random ratings to other items (i.e., filler items).

In VBGCF, we adopted LightGCN to calculate the user and item representations in Eq. (13) by following the two reasons. Firstly, LightGCN is the state-of-the-art method for GCN-based recommendation, and secondly, LightGCN has no weight parameter. Thus, it is valid for any generated graph structures. Here, since $\tilde{G}$ is the weighted graph, the calculation of node embeddings is rewritten instead of Eqs. (1) and (2) as follows:

$$\boldsymbol{e}_u^{(l)} = \sum_{i\in\mathcal{I}} \frac{\tilde{a}_{ui}}{p_{ui}} \boldsymbol{e}_i^{(l-1)}, \quad (17)$$

$$\boldsymbol{e}_i^{(l)} = \sum_{u\in\mathcal{U}} \frac{\tilde{a}_{ui}}{p_{ui}} \boldsymbol{e}_u^{(l-1)}, \quad (18)$$

where $p_{ui} = \sqrt{(\sum_{u\in\mathcal{U}}\tilde{a}_{ui})(\sum_{i\in\mathcal{I}}\tilde{a}_{ui})}$.

To verify the effectiveness of VBGCF, we adopted the following three comparative methods.

- BPR-MF [24] : This is the most popular collaborative filtering method, which decomposes the user-item interaction matrix into two low-rank matrices as the user and item representations. This method uses the BPR loss for the optimization.
- LightGCN [16] : This is the state-of-the-art method for GCN-based recommendation. The detail is explained in Sec. **2.1**.
- BGCF [19] : This is the GCN-based recommendation method introducing the Bayesian approach. This method can incorporate the uncertainty issues. The detail is explained in Sec. **2.2**.

Then we used Recall@k and NDCG@k ($k = 20, 50$) as evaluation metrics according to [16, 19]. In all models, the number of GCN layers L (excluding BPR-MF), the number of epochs, the learning rate, and the dimension of the user and item representations respectively were set to 3, 200, 0.0001, and 64. For VBGCF, we set $\mu_0 = 0.25$, $\mu_1 = 0.5$, $\tau = 0.5$, and $S = 3$.

Table 1 shows the experimental results. As shown in this table, our VBGCF outperforms the conventional methods in many metrics. Since we consider the realistic situation that the observed data are incomplete and unreliable, VBGCF significantly outperforms BPR-MF and LightGCN, which cannot deal with the uncertainty issues. In addition, although Recall of VBGCF is almost equal to that of BGCF, NDCG of VBGCF significantly outperforms that of BGCF. Especially, under the setting of 10% malicious users, which is the most unreliable situation, VBGCF achieves the best performance among all the methods. From the above, it is demonstrated that VBGCF can consider the incompleteness and unreliability issues, and more robust recommendation can be realized.

5. CONCLUSION

In this paper, we have presented the novel recommendation model, VBGCF. This model can provide robust recommendation to the incompleteness and the unreliability of the observed interaction graph by introducing the Bayesian approach to GCN-based recommendation. Furthermore, we introduce VBI to obtain the posterior distribution of the graph as the random variable, and it becomes feasible to directly adopt the gradient method by using the continuous relaxation. As a result, VBGCF can easily optimize the parameters. The results of the experiment under the realistic situation have shown the effectiveness of VBGCF for robust recommendation.

6. REFERENCES

[1] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, 2019.

[2] Charu Aggarwal, *Recommender Systems: The Textbook*, vol. 1, Springer, 2016.

[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proceedings of the International Conference on World Wide Web*, 2019, pp. 151–161.

[4] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang, "A survey on neural recommendation: From collaborative filtering to content and context enriched recommendation," *arXiv preprint arXiv:2104.13030*, 2021.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *Proceedings of the International Conference on World Wide Web*, 2017, pp. 173–182.

[6] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[7] Hao Wang, Naiyan Wang, and Dit-Yan Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1235–1244.

[8] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proceedings of the International Conference on Data Mining*, 2010, pp. 211–222.

[9] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proceedings of the International Conference on World Wide Web*, 2015, pp. 111–112.

[10] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the International Conference on World Wide Web*, 2018, pp. 689–698.

[11] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[12] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui, "Graph neural networks in recommender systems: A survey," *arXiv preprint arXiv:2011.02260*, 2020.

[13] Rianne van den Berg, Thomas N Kipf, and Max Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[14] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 974–983.

[15] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua, "Neural graph collaborative filtering," in *Proceedings of the International Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.

[16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the International Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[17] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 767–799, 2014.

[18] Hao Wang and Dit-Yan Yeung, "Towards Bayesian deep learning: A framework and some existing methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3395–3408, 2016.

[19] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al., "A framework for recommending accurate and diverse items using Bayesian graph convolutional neural networks," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 2030–2039.

[20] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay, "Bayesian graph convolutional neural networks for semi-supervised classification," in *Proceedings of the Conference on Artificial Intelligence*, 2019, pp. 5829–5836.

[21] Soumyasundar Pal, Florence Regol, and Mark Coates, "Bayesian graph convolutional neural networks using node copying," *arXiv preprint arXiv:1911.04965*, 2019.

[22] Pantelis Elinas, Edwin V Bonilla, and Louis Tiao, "Variational inference for graph convolutional networks in the absence of graph data and adversarial settings," in *Proceedings of the Advances in Neural Information Processing Systems*, 2020, pp. 18648–18660.

[23] Chris J Maddison, Andriy Mnih, and Yee Whye Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] F Maxwell Harper and Joseph A Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.

[27] Wei Zhou, Junhao Wen, Qingyu Xiong, Min Gao, and Jun Zeng, "SVM-TIA a shilling attack detection method based on svm and target item analysis in recommender systems," *Neurocomputing*, vol. 210, pp. 197–205, 2016.

[28] Fuguo Zhang, "Analysis of love-hate shilling attack against e-commerce recommender system," in *Proceedings of the International Conference of Information Science and Management Engineering*, 2010, pp. 318–321.