Key Opinion Leaders in Recommendation Systems: Opinion Elicitation and Diffusion

Jianling Wang*
Texas A&M University
jlwang@tamu.edu

Kaize Ding*
Arizona State University
kaize.ding@asu.edu

Ziwei Zhu, Yin Zhang
Texas A&M University
{zhuziwei,zhan13679}@tamu.edu

James Caverlee
Texas A&M University
caverlee@tamu.edu

ABSTRACT

Recommendation systems typically rely on the interactions between a crowd of ordinary users and items, ignoring the fact that many real-world communities are notably influenced by a small group of key opinion leaders, whose feedback on items wields outsize influence. With important positions in the community (e.g. have a large number of followers), their elite opinions are able to diffuse to the community and further impact what items we buy, what media we consume, and how we interact with online platforms. Hence, this paper investigates how to develop a novel recommendation system by explicitly capturing the influence from key opinion leaders to the whole community. Centering around opinion elicitation and diffusion, we propose an end-to-end Graph-based neural model - GoRec. Specifically, to preserve the multi-relations between key opinion leaders and items, GoRec elicits the opinions from key opinion leaders with a translation-based embedding method. Moreover, GoRec adopts the idea of Graph Neural Networks to model the elite opinion diffusion process for improved recommendation. Through experiments on Goodreads and Epinions, the proposed model outperforms state-of-the-art approaches by 10.75% and 9.28% on average in Top-K item recommendation.

KEYWORDS

Recommendation; Key Opinion Leaders; Graph Neural Networks

ACM Reference Format:
Jianling Wang*, Kaize Ding*, Ziwei Zhu, Yin Zhang, and James Caverlee. 2020. Key Opinion Leaders in Recommendation Systems: Opinion Elicitation and Diffusion. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371826

1 INTRODUCTION

To alleviate the severe information overload issue, recommendation systems act as essential components in many online platforms helping users find items of interest (e.g. movies, books, or music tracks). Generally, users leave implicit feedback on items through different interactions such as views, clicks or purchases [14, 19], which can be distilled to reveal their preferences. By leveraging such implicit feedback, a variety of recommendation systems [14, 19, 24, 25] have been proposed and shown great success in providing personalized item recommendation.

In fact, within many real-world platforms, there also exists a small group of well-known individuals - Key Opinion Leaders (KOLs), who can shape our views, and further impact what items we buy, what media we consume, and how we interact with online platforms [9, 27]. For example, KOLs on Instagram and Pinterest could influence shopping decisions by highlighting new fashion trends [4, 6, 48], while KOLs on Yelp and TripAdvisor could guide customer restaurant selection by providing explanatory information (like photos and reviews) for restaurants [33, 49]. Previous research has shown the effectiveness of modeling KOLs in different learning tasks, such as public sentiment analysis [22] and social event detection [13]. However, the effect of KOLs in recommendation systems remains largely unexplored, which motivates us to develop a novel recommendation system by explicitly capturing the influence from KOLs to the whole platform.

Despite the importance of investigating the influence of KOLs in recommendation systems, however, it is a non-trivial task due to two major challenges: (i) **Elicitation**: Compared to regular users, KOLs tend to express their opinions on items explicitly rather than leave implicit feedback. More important, such explicit interactions are inherently multi-relational: on the one hand, KOLs are able to express their opinions via different ways (e.g., review, rating or tagging); On the other hand, the opinions from KOLs could have distinct meanings (e.g., tag "fantastic" and tag "terrible" are semantically different). However, it is unclear that how to extract the elite opinions of KOLs from such multi-relational data. (ii) **Diffusion**: In online communities, KOLs are able to guide their followers' preferences and shape how users view the items. For example, users tend to purchase makeup products with the recommendation of Beauty-KOLs they are following; a book which was tagged as "For Teens" by KOLs could attract many teenager readers. Meanwhile, previous research [12, 37, 44] has shown that user preferences on items could diffuse through high-order connectivity (e.g., in Figure 1, the latent preference of user A can possibly diffuse via the transitive path $A \rightarrow q \rightarrow B \rightarrow w$, to items that he/she hasn't interacted with). Therefore, the influence from KOLs will also be propagated

* indicates Equal Contribution.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371826

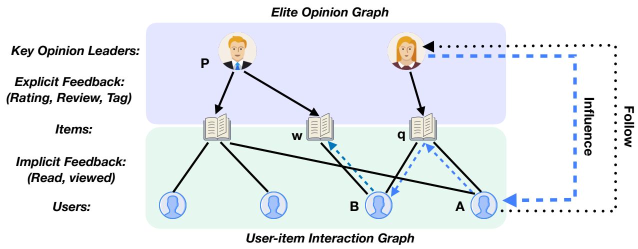

Figure 1: The opinions from Key Opinion Leaders (KOLs) can diffuse to their followers and items they comment. Furthermore, these opinions *diffuse* in the community via both direct and multi-hop connections between users and items.

to those non-direct followers in the community. In this regard, another challenge centers around how to model this elite opinion diffusion process for improved recommendation?

To tackle the aforementioned challenges, in this work, we propose *GoRec*: a novel *end-to-end* **G**raph-based neural model to incorporate the influence of K**O**Ls for **Rec**ommendation. Specifically, we introduce a translation-based embedding method to elicit the opinions of KOLs, in which the elite opinions are regarded as different types of translations from KOLs to items. In this way, we are able to extract the embeddings for both KOLs and items, and use them to enrich the initial user/item embeddings in the user-item interaction graph. In order to model the diffusion process of elite opinions, our model employs multiple Graph Neural Network (GNN) layers to learn the final user/item embeddings following the neighborhood aggregation strategy [8, 17]. As such, the final embeddings enriched with the elite opinions from KOLs can be decoded to accurately infer users' preferences on all the items. The main contributions of this work can be summarized as follows:

- We explore the relationships among users, items and key opinion leaders within real-world online platforms, and uncover the importance of explicitly modeling the influence of KOLs in recommendation systems.
- We develop a novel end-to-end item recommendation framework - *GoRec*, which is able elicit elite opinions from KOLs and model their diffusion in the community.
- We conduct extensive experiments on two real-world scenarios including Goodreads (a book sharing community) and Epinions (an ecommerce review sharing platform). We find that the proposed *GoRec* model outperforms the state-of-the-art by 10.75% and 9.28% on average in Top-K recommendation. Meanwhile, we find that a small set of KOLs is sufficient to hint on the preferences of a huge amount of users in the community and thus benefit the recommendation system.

2 MOTIVATION: OPINION LEADERS

To gain insight into the relationships among users, items, and key opinion leaders (KOLs), we start our discussion with an initial exploration into the Goodreads community. Goodreads is a book-based platform with about 80 million registered users in which users can manage their own book reading habits and also connect with other readers [32]. Users on Goodreads leave lots of *implicit* feedback, which can be treated as positive signals while inferring users preferences [14, 19] (e.g. a user read both "Harry Potter" and "Twilight" probably likes paranormal fiction). However, a user who has not read a book (and hence, left no implicit feedback) may not because she doesn't like it but just not knows about the book. Goodreads are also shaped by *explicit* feedback, which conveys positive or negative opinions on items explicitly, e.g., text reviews, numerical ratings, or semantic tagging. Aside from the bidirectional "friendship", users on Goodreads can unidirectionally follow other users from whom they can receive activity updates. One of the key features of Goodreads is that some users are experts who provide detailed reviews and highlight new releases, and thus are followed by many other users. In our initial analysis, naturally, we ranked all the accounts based on their numbers of followers and treat the top accounts as KOLs [41, 45].

A small number of key opinion leaders (KOLs) can provide sufficient coverage. Before exploiting the influence of KOLs, we want to examine whether the top accounts that we treat as KOLs can provide sufficient coverage for the regular users in Goodreads. In Figure 2(a), we check the percentage of users following at least one of the KOLs (what we refer to as coverage) by changing the number of top accounts that we consider as KOLs. We find that while considering only the top-500 KOLs, there are more than 95% of the users following at least one of these KOLs. In other words, the patterns and behaviors of just the top accounts (the KOLs) can potentially have wide-reaching impact on the community. We can conclude that *a small number of key opinion leaders (KOLs) can provide sufficient coverage.* Next, focusing on the top-500 KOLs, we explore whether there are patterns in their opinions and how their opinions can diffuse to the community.

Users are shifted by the KOLs they are following. To examine whether the explicit opinions from KOLs can influence what their followers read, we represent each user with a simple binary vector over all books, in which a "1" indicates that the user has left implicit feedback on this book. We use five different binary vectors for each KOL to represent books with different ratings (1 to 5) from the KOL. In Figure 2(b), we list the similarities between books read by users and books with different ratings from KOLs they are following. We find that the set of books a user read are more similar to the books receiving high ratings from KOLs they follow, while having little overlapping (similarity) with books with low ratings (1 or 2) from the KOLs they follow. We conclude that the explicit opinions of KOLs could directly influence what their followers consume.

Compared to ordinary users, KOLs tend to express opinions on items explicitly. In Figure 2(c), we compare the numbers of different kinds of feedback from regular users and KOLs in Goodreads. We find that ordinary users and KOLs leave implicit feedback on a similar number of books (that is, they mark a book as "read" or "to read"), indicating that both are active in their use of Goodreads, presumably for managing their own book collections. However, we do find that KOLs tend to leave more reviews, ratings and tags; that is, KOLs are more engaged in explicitly sharing their opinions on books. KOLs appear to be capable of providing specialized expertise and high-quality opinions in the community. These *elite opinions* (including reviews, ratings and tags) are public on the item pages and can influence how the community views or defines

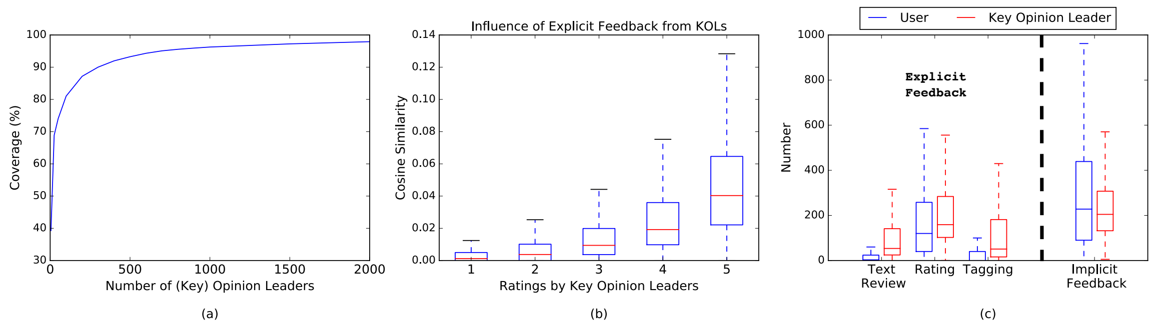

Figure 2: (a) Coverage: The percentage of users following at least one of the top (key) opinion leaders. More than 95% of users follow at least one of the Top-500 accounts. (b) Books read by users are more similar to books with higher ratings from key opinion leaders they are following. (c) While leaving a similar number of implicit feedback, key opinion leaders prefer to show their opinions on items via explicit interactions (reviews, ratings, self-defined tags).

the items, illustrating a possible way of how these elite opinions diffuse in the community. In this work, we will focus on explicitly modeling the influence of KOLs in recommendation system with opinion elicitation and diffusion.

3 GOREC: MODEL

With these observations in mind, we propose *GoRec*, a novel graph-based recommendation system enhanced with the influence of KOLs in the community. Our design is structured around the challenges we are faced with: (i) **Elicitation:** How can we elicit the elite opinions of KOLs from the multi-relational data? (ii) **Diffusion:** How to model the diffusion process of elite opinions in the community for improved recommendation?

3.1 Problem Setting and Notation

Task. In this work, we aim to provide Top-K recommendation from a candidate set of M items $\mathbf{I} = \{i_1, i_2, ..., i_M\}$ to a set of N users $\mathbf{U} = \{u_1, u_2, ..., u_N\}$. For each user u, we use a binary vector $\mathbf{y}_u = \{y_{u1}, y_{u2}, ..., y_{uM}\}$ to indicate the implicit feedback u left on all the items. That is, if u interacted with item i, then $y_{ui} = 1$. And $y_{ui} = 0$ means u has not left any feedback on i.

User-item Interaction Graph. Based on the (implicit) interactions between users and items, we can construct a bipartite graph $\mathbf{G} = (\mathcal{V}, \mathcal{W})$ in which the set of nodes $\mathcal{V} = \mathbf{U} \cup \mathbf{I}$ consists of all the users and items. The edge $(u, i) \in \mathcal{W}$ denotes that user u has implicit feedback on item i. Similarly, we can construct an adjacency matrix $\mathbf{A} \in \{0, 1\}^{N \times M}$ for graph $\mathbf{G}$ by concatenating the feedback vector of each user, that is $\mathbf{A} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]^T$.

Elite Opinion Graph. We use $\boldsymbol{L} = \{l_1, l_2, ..., l_P\}$ to represent the set of key opinion leaders (KOLs) we investigate while constructing the elite opinion graph $\mathbf{G}_o$. The explicit opinions can be different rating levels, words mentioned in the reviews or tags. We consider Q different types of explicit opinions $\boldsymbol{O} = \{o_1, o_2, ..., o_Q\}$ for the graph. Thus based on the explicit feedback from KOLs to items, we can harvest many opinion triplets. Each triplet is denoted as (l, o, i) representing kol l left opinion o on item i. And we construct a directed graph $\mathbf{G}_o$ composed of these *kol-opinion-item* triplets.

User-KOL Following. To explore how the elite opinions from KOLs directly influence their followers, we use $\mathbf{F}_u \subset \boldsymbol{L}$ to represent the set of KOLs followed by user $u \in \boldsymbol{U}$. And we let $\boldsymbol{U} \bigcap \boldsymbol{L} = \emptyset$.

3.2 Translation-based Opinion Elicitation

First, we start by eliciting the opinions from KOLs toward improving the quality of recommendation. Recall that KOLs leave explicit opinions on items via reviews, ratings and tags. These opinions constitute a large scale of multi-relations from KOLs to items. As analogous to the data structure of knowledge graph, the resulted elite opinion graph $\mathbf{G}_o$ consists of many valid opinion triplets. For example, a triplet (l_1, *Review: wizard*, *Harry Potter*) denotes that KOL l_1 mentions the word *wizard* in a review for item *Harry Potter*. As shown in Figure 3, we can also construct these opinion triplets based on ratings or tags provided by KOLs, and get triplets like (l_1, *Rate: 5*, *Harry Potter*) or (l_1, *Tag: fiction*, *Harry Potter*).

Our goal is to generate effective embedding for both items and KOLs in a continuous vector space while preserving the multi-relations (opinions) between them. In the below, we will list three features of $\mathbf{G}_o$ followed by the corresponding design we propose in the opinion elicitation process:

Feature 1. Multiple relations: Opinions come with distinct meaning, e.g., tag "fantastic" and "terrible" are semantically different.

Translation from KOL to Item. Adopting the similar idea in multi-relational graph embedding [2, 15, 20, 39], we treat opinions as translations from KOLs to items. That is, given a valid opinion triplet (k_h, o_r, i_t), we want to ensure that the embedding of item i_t is close to the embedding of KOL k_h plus the embedding of opinion o_r. Let $s(k_h, o_r, i_t)$ denote the scoring function for the translation operation, with which larger value means better translation. Given all the valid (positive) and negative opinion triplets, the objective is to maximize the translation score for all the positive triplets while minimizing that for the negative triplets. We formalize this objective into a task of minimizing the marginal loss below:

$$L_{op} = \sum_{(k_h, o_r, i_t) \in \mathbf{G}_o} \sum_{(k_h, o_r, i_t') \in \mathbf{G}_o^-} [\gamma + s(k_h, o_r, i_t') - s(k_h, o_r, i_t)]_+ \tag{1}$$

in which $[\cdot]_+ \triangleq max(0, \cdot)$ and γ denotes the margin the model used to separate the valid (positive) triplets and negative triplets. Here the

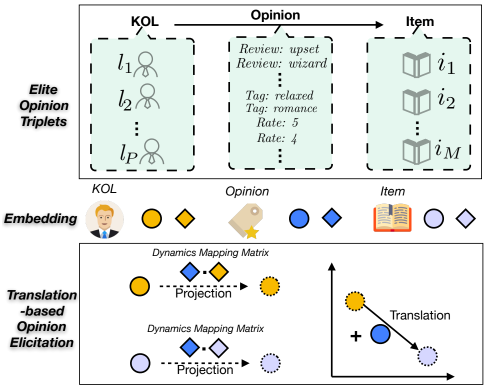

Figure 3: Translation-based Embedding with Elite Opinions.

negative triplet $(k_h, o_r, i_t') \in \mathbf{G}_o^-$ indicates that k_h wouldn't attach opinion o_r on i_t'. Thus while generating the negative samples, we randomly select i_t' from the subset of items that k_h has left feedback on excluding o_r.

Feature 2. Many-to-Many relations: On one hand, the connections between KOLs and items are not always one-to-one. On the other hand, KOLs can endow opinions with their personal attitudes. For example, each KOL has his/her own criteria for tagging a book with "BestOf2019".

Dynamic Mapping Matrix. To handle the *Many-to-Many* relations, a common strategy is to project KOLs and items to an opinion-specific space before the translation operation. Additionally, to cope with the various meanings of the same opinion, while doing projection, we adopt a dynamic mapping matrix [15] which is determined by both the opinion and the KOL (or item). In our case, each kol, item and opinion is represented by two vectors. One vector acts as its latent representations, while the other vector is used to construct the mapping matrix (as in Figure 3). Given a triple (k_h, o_r, i_t), we will initialize dense vectors $\mathbf{k}_h^e$, $\mathbf{k}_h^t$, $\mathbf{o}_r^e$, $\mathbf{o}_r^t$, $\mathbf{i}_t^e$, $\mathbf{i}_t^t$. First we will construct the mapping matrices for k_h and i_t on opinion o_r with vectors $\mathbf{k}_h^t$, $\mathbf{i}_t^t$ and $\mathbf{o}_r^t$:

$$\mathbf{M}_{rh} = \mathbf{o}_r^t {\mathbf{k}_h^t}^T + \mathbf{I} \quad \mathbf{M}_{rt} = \mathbf{o}_r^t {\mathbf{i}_t^t}^T + \mathbf{I}$$

in which $\mathbf{I}$ denotes the identity matrix. $\mathbf{M}_{rh}$ is used to transfer $\mathbf{k}_h^e$ to the space of o_r and $\mathbf{M}_{rt}$ is for transferring $\mathbf{i}_t^e$. Thus we get the projected representation of k_h and i_t under opinion o_r with:

$$\mathbf{k}_h^\perp = \mathbf{M}_{rh}\mathbf{k}_h^e \quad \mathbf{i}_t^\perp = \mathbf{M}_{rt}\mathbf{i}_t^e$$

The score function $s(\cdot)$ used to evaluate the translation distance for triple (k_h, o_r, i_t) is represented as:

$$s(k_h, o_r, i_t) = -\|\mathbf{k}_h^\perp + \mathbf{o}_r^e - \mathbf{i}_t^\perp\|^2 \tag{2}$$

in which we use L2-norm to calculate the distance empirically. Larger $s(k_h, o_r, i_t)$ means k_h and i_t are close to each other with translation o_r, i.e, it is more likely that k_h attaches opinion o_r to i_t.

Feature 3. Preference Signals: KOLs have preferences on the items they would interact with, e.g., a romantic book lover may seldom leave any feedback on horror novels.

Personalized Ranking Model A typical assumption is that the items with feedback from the user are preferred than those without. We also want to capture these (implicit) preference signals while modeling both KOLs and items. Following the basic idea in matrix factorization, we use the multiplication between $\mathbf{k}_h^e$ and $\mathbf{i}_t^e$, that is $p(k_h, i_t) = {\mathbf{k}_h^e}^T \mathbf{i}_t^e$, to capture the preference of k_h on i_t. Then given the positive pair (k_h, i_t) representing k_h has left feedback on i_t and negative pair (k_h, i_t') meaning k_h has not left feedback on i_t', we adopt Bayesian Personalized Ranking (BPR) [25] to maximize the difference of preference scores between the positive pair and the negative pair. With $\delta(\cdot)$ denoting the Sigmoid function, the objective function to model these preference signals is:

$$L_{BPR} = \sum_{(k_h, i_t, i_t') \in \mathbf{S}} -\ln \delta(p(k_h, i_t) - p(k_h, i_t')) \tag{3}$$

Each element in the training data set $\mathbf{S}$ is generated by combining the ground truth interaction pair (k_h, i_t) with item i_t' that KOL k_h hasn't left any feedback on.

Joint Tasks. During the opinion elicitation, we combine the task of modeling explicit opinions and extracting preference signals from the elite opinions graph $\mathbf{G}_o$, leading to the following loss function:

$$L_{\mathbf{G}_o} = L_{op} + \beta L_{BPR} \tag{4}$$

Here β is used to adjust the weight of pairwise loss in capturing the preference signals. By minimizing this joint loss $L_{\mathbf{G}_o}$, we will get the set of embeddings $\mathbf{K}_e = \{\mathbf{k}_1^e, \mathbf{k}_2^e, ..., \mathbf{k}_P^e\}$ for KOLs and $\mathbf{I}^e = \{\mathbf{i}_1^e, \mathbf{k}_2^e, ..., \mathbf{i}_M^e\}$ for items, which inherit both the explicit information and preference signals in the elite opinion graph $\mathbf{G}_o$.

3.3 Neural Graph-based Opinion Diffusion

As explained in Section 2, the opinions from KOLs can influence their followers and items they comment, and thus make up part of their features. Besides the implicit user-item interactions, these elite opinions should also be exploited while modeling users preferences. In what follows, we will start from enriching the initial user/item embeddings with elite opinions. Then we will explain how to model the elite opinion diffusion process with graph neural networks.

Fusing Layer (Users). Each user is associated with an embedding $\mathbf{e}_u^U \in \mathbf{R}^d$ to represent the initial interest, which can be derived from his/her one-hot index with a fully-connected dense layer. Since users are directly influenced by whom they follow, aggregating the embeddings of KOLs whom the user is following can hint on the user preferences on items. However, we know that a particular KOL can have different levels of influence on different users. Building on the recent development of attention mechanisms [5, 35], we can model the dynamic (personalized) linkage between users and KOLs. We have the set of embeddings from Section 3.2 for the set of P KOLs $\mathbf{k} = \{\mathbf{k}_1^e, \mathbf{k}_2^e, ..., \mathbf{k}_P^e\}$. Given that $\mathbf{F}_u$ is the set of KOLs that u is following and $\mathbf{e}_u^U$ is a trainable dense representation for u, the weight of KOL p's influence on user u can be calculated as:

$$\alpha_{up} = \frac{e^{d_{up}}}{\sum_{j \in \mathbf{F}_u} e^{d_{uj}}}, \qquad d_{up} = \mathbf{z}^T ReLu(\mathbf{W}_A[\mathbf{k}_p^e \| \mathbf{e}_u^U] + \mathbf{b}_A)$$

Here || represents the concatenation operation. $\mathbf{W}_A$ and $\mathbf{b}_A$ is the weight matrix and bias for the attention layer. $\mathbf{z}$ is a transformation vector. Then we aggregate the embeddings of all KOLs the user

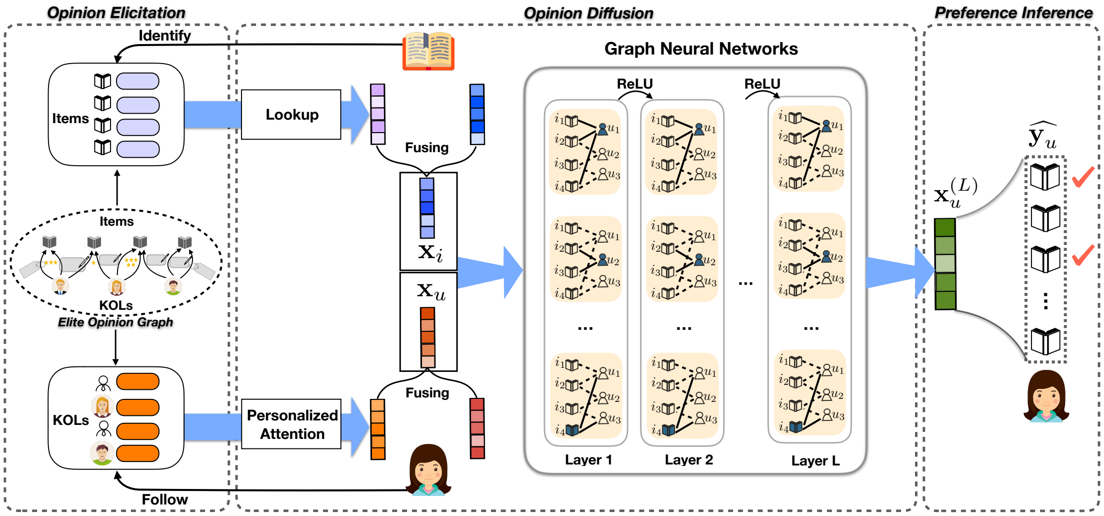

Figure 4: Graph Neural Recommendation Incorporating the Influence of Key Opinion Leaders (GoRec).

follows with the attentive weights:

$$\mathbf{n}_u = \sum_{j \in \mathbf{F}_u} \alpha_{uj} \mathbf{k}_j^e$$

Thus $\mathbf{n}_u$ can be used to characterize influence of elite opinions to user u from whom he/she follows. Lastly, we fuse $\mathbf{n}_u$ with the initial embedding of u with the following operation:

$$\mathbf{x}_u = ReLu(\mathbf{W}_U[\mathbf{n}_u \| \mathbf{e}_u^U])$$

where $\mathbf{W}_U$ is a transformation matrix, and the output $\mathbf{x}_u$ will be treated as cornerstone for the opinion diffusion.

Fusing Layer (Items). Similarly, each item will start with a trainable dense representation $\mathbf{e}_i^I \in \mathbf{R}^d$, which is associated with its index. Since the KOLs can influence how the whole community view an item, we want to complement $\mathbf{e}_i^I$ with the KOL-defined features $\mathbf{i}_i^e$ of item i which we elicit from the the opinions of KOLs. Thus we adopt the similar fusion operation to generate the enriched representation of item i:

$$\mathbf{x}_i = ReLu(\mathbf{W}_I[\mathbf{i}_i^e \| \mathbf{e}_i^I])$$

in which $\mathbf{W}_I$ is a transformation matrix and $\mathbf{i}_i^e$ is the embedding gained from Section 3.2 for item i.

Opinion Diffusion with GNNs. As suggested by [12, 44], user preferences on items could diffuse through high-order connectivity, thus the elite opinions from KOLs will also be propagated to those non-direct followers in the community. In this paper, we propose to model this opinion diffusion process by virtue of Graph neural networks (GNNs) [1, 17, 29, 31].

The core idea of GNNs is that each layer learns the node embeddings by aggregating the features of neighbors. At the initial GNN layer of our model, for user u and item i, given the sets of neighbors $\mathcal{N}_u$ and $\mathcal{N}_i$ which are directly connected with u and i correspondingly, we formulate the message passing on the edge (u, i) from i to u as:

$$\mathbf{c}_{i \to u}^{(1)} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{W}_U^{(0)} \mathbf{x}_i \quad (5)$$

Here, $\mathbf{x}_i$ is the representation of i with influnce from KOLs and $\mathbf{W}_U^{(0)}$ denotes a trainable transforming matrix for users at layer 0. The term $1/\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}$ is a normalization constant between u and i. And then we need to sum up all the message passed to u to generate its representation $\mathbf{x}_u^{(1)}$:

$$\mathbf{x}_u^{(1)} = \tau(\sum_{i \in \mathcal{N}_u} \mathbf{c}_{i \to u}^{(1)})$$

where $\tau(\cdot)$ is the activation function and we choose *ReLu* in this work empirically. Similarly, we can generate the representation of item i at this layer with:

$$\mathbf{x}_i^{(1)} = \tau(\sum_{u \in \mathcal{N}_i} \mathbf{c}_{u \to i}^{(1)}) = \tau(\sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{W}_I^{(0)} \mathbf{x}_u)$$

After generating the $\mathbf{x}_u^{(1)}$ and $\mathbf{x}_i^{(1)}$ from the first GNN layer, we can further capture the high-order diffusion by stacking multiple GNN layers. Specifically, at the L^{th} layer, we will have:

$$\mathbf{x}_u^{(L)} = \tau(\sum_{i \in \mathcal{N}_u} \mathbf{c}_{i \to u}^{(L)}) = \tau(\sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{W}_U^{(L-1)} \mathbf{x}_i^{(L-1)})$$

$$\mathbf{x}_i^{(L)} = \tau(\sum_{u \in \mathcal{N}_i} \mathbf{c}_{u \to i}^{(L)}) = \tau(\sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{W}_I^{(L-1)} \mathbf{x}_u^{(L-1)})$$

Note that $\mathbf{x}_u^{(L)}$ at layer L inherits embeddings of users and items from previous layers. That is how we *capture the diffusion of opinions in multiple-order user-item connectivity with GNNs.*

3.4 Preference Inference

With what we have reached so far, our final step is to infer user u's preference on all the items. That is the probability that these items are connected with u. Following the similar idea as in Graph Auto-encoder [7, 18] and Autorec [26] - a basic autoencoder for recommendation, we use a fully-connected layer to recover the graph structure (user-item interaction graph) from the output of

	#User	#Item	User-item Feedback	User-KOL Feedback	Opinion Triplet
Goodreads	15,324	36,645	1.831,826	167,054	2.8M
Epinions	6,334	8,015	81,965	63,939	0.1M

Table 1: Dataset Statistics.

the encoder (the stack of GNN layers). That is, for u, we will decode $\mathbf{x}_u^{(L)}$ to reconstruct his/her feedback vector $\mathbf{y}_u$:

$$\widehat{\mathbf{y}_u} = \delta(\mathbf{V}\mathbf{x}_u^{(L)} + \mathbf{b}')$$

where $\mathbf{V}$ and $\mathbf{b}'$ are the weight matrix and bias term correspondingly. And $\delta(\cdot)$ represents the Sigmoid function. The objective is to minimize the reconstruction loss L_{re} between $\widehat{\mathbf{y}_u}$ and $\mathbf{y}_u$:

$$L_{re} = \sum_{u \in \mathbf{U}} ||\mathbf{y}_u - \mathbf{S}_u \cdot \widehat{\mathbf{y}_u}||^2 \tag{6}$$

where $\mathbf{S}_u$ is a binary masking vector with 1 indicating items that we want to consider while calculating the reconstruction loss for u. Since the feedback usually is extremely sparse, as in [42, 50], we don't consider all the 0s in $\mathbf{y}_u$ while calculating the loss. We allocate 1 on all the items that u has left implicit feedback on (positive) and also on some randomly selected items without feedback (negative) in $\mathbf{S}_u$. And we combine the tasks of opinion elicitation and diffusion jointly, then the objective function of our final model (GoRec) becomes:

$$L = L_{re} + \lambda L_{\mathrm{G}_o}$$

Thus we reach our GoRec model (in Figure 4) which combines both tasks end-to-end with a hyper-parameter λ to balance the tasks.

Prediction: The reconstructed vector $\widehat{\mathbf{y}_u}$ will be used to infer user's preference on all the items, in which larger value means higher probability that the user is interested in the item. We will rank those predictions to generate the Top-K recommendations to users.

4 EXPERIMENTS

In this section, we will evaluate the performance of the proposed GoRec model on two real-world datasets:

4.1 Datasets

We test GoRec and the baselines on both Goodreads and Epinions (summarized in Table 1). Empirically, we select the Top-500 accounts in the communities as KOLs. There is no overlapping between ordinary users and KOLs. We split the user-item interaction data with ratio 6:1:3 for training, validation and testing.

Goodreads. We randomly sample 2 million user IDs and crawl all their interactions with books and their following information until November 2018. We filter out inactive users with fewer than 5 interactions on books. While constructing the opinion triplets, we utilize the reviews, ratings and tags provided by KOLs. For each review, we handle it with preprocessing, tokenization, and stop word removal to extract the words. Each unique word, rating level (1 to 5), or tag is treated as one type of opinion, based on which we construct the opinion triplets.

Epinions. This is a public dataset with user reviews and unidirectional user-user relations [30]. Epinions is a review site on which users can write and read reviews for products. In Epinions, a user can "trust" another user, which is treated as the "follow" signal in this platform. By analyzing the "trust" relationships between users, we can see similar patterns as shown in Figure 2 for Goodreads. We keep active users leaving no less than 5 feedback. We ranked all the accounts based on the numbers of their followers and select the top accounts as KOLs. For KOLs, we treat all the reviews and ratings as explicit opinions. And we use the same method as in Goodreads to construct the opinion triplets. For users, we treat all their interactions with items as implicit feedback.

4.2 Experimental Setup

Metrics. To better examine how the personalized recommendation system works under a real-world scenario, we adopt the Precision (Pre), Recall (Re), F-1 score (F1) and NDCG of Top-K recommendation as metrics. Pre@k represents the percentage of correctly predicted items among the Top-k recommendations, and Re@k represents the fraction of relevant items which are discovered by the Top-k recommendations. We also consider both recall and precision with their harmonic mean $F1@k = \frac{2 \cdot Pre@k \cdot Re@k}{Pre@k + Re@k}$.

NDCG takes the positions of recommendations into consideration. It is the ratio between discounted cumulative gain (DCG) and ideal discounted cumulative Gain (IDCG): $DCG@K = \sum_{i=1}^{K} \frac{rel_i}{\log_2(i+1)}$ and $IDCG@K = \sum_{i=1}^{Min(|L|,K)} \frac{1}{\log_2(i+1)}$, where $|L|$ is the size of the test set. The relevance score rel_i equals to 1 if the recommendation with rank i is in the test set, otherwise, $rel_i = 0$. Then NDCG is calculated as: $NDCG@K = \frac{DCG@K}{IDCG@K}$.

Baselines.

- **ItemPop:** This model ranks items based on their popularity and recommends the most popular items.
- **BPRMF:** *Bayesian Personalized Ranking* [25]. It estimates user's preference on an item with the multiplication between their latent factors (MF). It is optimized with the Bayesian personalized ranking (BPR) loss [25] based on user-item interactions.
- **CDAE:** *Collaborative Denoising Autoencoder* [42]. This model is a generalization of collaborative filtering and matrix factorization. It models user-item interactions with the basic Autoencoder structure and an additional user node.
- **NGCF:** *Neural Graph Collaborative Filtering* [37]. It models user-item interactions with GNNs and *concatenates* the embeddings from different GNN layers to balance the multi-order connectivity in a bipartite graph. It is optimized with BPR loss.

Somewhat similar to our idea of incorporating the influence of KOLs is exploiting semantic knowledge for recommendation. Below are methods originally proposed to enhance recommendation with knowledge graph (KG). By treating the opinion triplets in the same way as the fact triplets in a KG, they can also consider the interactions between user, KOLs and items for recommendation:

- **MKR:** *Multi-Task Feature Learning* [36]. This model proposes to utilize the cross&compress unit to combine recommendation with the task of KG embedding. It aims to optimize AUC.

Model	Goodreads								Epinions							
	k=5				k=10				k=5				k=10			
	Pre	Re	F1	NDCG	Pre	Re	F1	NDCG	Pre	Re	F1	NDCG	Pre	Re	F1	NDCG
ItemPop	16.95	2.31	4.07	17.78	13.82	3.54	5.64	15.61	2.90	2.62	2.75	3.58	2.36	4.15	3.01	3.94
BPRMF	27.58	4.27	6.13	28.50	24.99	7.27	8.99	27.03	4.33	3.57	3.48	5.04	3.64	6.02	4.01	5.64
NGCF	29.02	4.53	6.41	30.23	26.02	7.57	9.28	28.41	4.73	4.04	3.86	5.53	3.82	6.33	4.21	6.03
CDAE	30.52	4.67	6.69	31.85	27.58	7.94	9.82	30.04	5.02	4.15	4.04	5.94	4.16	6.91	4.60	6.57
MKR	21.80	2.70	4.29	21.43	19.50	4.32	6.28	20.11	4.80	3.94	3.92	4.86	3.10	5.46	3.55	4.97
KTUP	28.70	4.22	6.14	29.76	26.08	7.23	9.14	28.16	4.51	3.79	3.67	5.26	3.87	6.38	4.29	5.94
CKE	30.99	4.43	6.50	32.38	27.82	7.46	9.58	30.28	4.98	4.27	4.08	5.96	4.13	6.86	4.57	6.58
GoRec	**34.61***	**5.06***	**7.50***	**35.88***	**31.09***	**8.58***	**11.02***	**33.61***	**5.45***	**4.68***	**4.47***	**6.50***	**4.43***	**7.62***	**4.97***	**7.16***
Δ(%)	11.68	8.35	12.11	10.81	11.75	8.06	12.21	10.99	8.56	12.77	9.55	9.06	6.49	10.27	8.75	8.81

Table 2: Comparing Models on top-K Recommendation. ***All the results are in percentage.*** **$*$ indicates that the improvement of the best result is statistically significant compared with other methods for $p < 0.05$.**

- **KTUP:** *Unifying KG Learning and Recommendation* [3]. It performs item recommendation and knowledge completion simultaneously. It enhances the basic BPRMF by transferring embeddings for relations and entities learned from KG completion.
- **CKE:** *Collaborative Knowledge Base Embedding for Recommender Systems* [47]. It proposes to combine knowledge of items from multiple resources to enhance recommendation. It uses TransR [20] to construct embeddings for the structural knowledge.

Parameters. All of the experiments were conducted with a 12 GB Nvidia TITAN GPU. For CDAE, NGCF, MKR and KTUP, we use the implementations provided in their original papers. We use the implementation provided in [3] for CKE.

For all the baseline models, we did a grid search for the sizes of embeddings over {5, 10, 20, 50, 100, 200, 250, 500}. We also fine-tune their learning rates over {0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1} and the coefficient of $L2$ regularization over {10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}}. For NGCF, following the paper, we set its depth to be 3 to capture the 3-order connectivity in the graph. And the embedding size is 100 for each layer. We follow the settings as in the original paper for the other parameters.

We use Adam to optimize the proposed model. The batch size is set to be 256 for both datasets. We fine-tune the parameter λ and β over {0.01, 0.05, 0.1, 0.5, 1, 5} for both datasets. Empirically, for fair comparison with NGCF, without specification, we use 3 GNN layers with embedding size of {100, 100, 100} for the Graph-based encoder. The dimension of the embeddings for the elite opinion elicitation process is also set to be 100. The node dropout rate is set to be 0.1. While calculating the reconstruction loss in Equation 6, the negative sampling rate is set to be 10. That is for each item with positive feedback, we randomly select 10 items that the user has left no feedback on as negative samples. We train the model for 100 epochs or stop until it achieves the best performance on the validation set.

4.3 Baseline Comparison

As an overall comparison, we summarize the results of Top-K recommendation at K=5 and K=10 in Table 2. And Δ represents the improvement of GoRec over the best baseline methods. GoRec achieves the best performance under different K for both communities on all the metrics (*Pre*cision, *Re*call, F1 and NDCG).

Starting from ItemPop of recommending the most popular items, with matrix factorization, BPRMF can improve ItemPop by 56.61%. Then, NGCF extends BPRMF by concatenating the embedding generated from multiple GNN layers and achieves 8.13% and 4.82% improvement on Goodreads and Epinions, which shows the significance of paying attention to the high-order connectivity between users and items in recommendation.

Comparing GoRec with the baseline models which are designed to enhance the recommendation utilizing the semantic information from knowledge graphs (MKR, CKE and KTUP), we can see GoRec outperforms all of them, which indicates the proposed model is a good fit for eliciting the elite opinions and incorporating them to infer user preferences. Among those models, we find that MKR performs worst because it is designed to optimize the AUC and thus may not be a good fit for the Top-K recommendation task. KTUP improves BPRMF by transferring embeddings learned from the relational structured data. And we can see it outperforms BPRMF by 2.14% and 5.59% in recommendation for Goodreads and Epinions, which shows the effectiveness of treating the opinion triplets similar as the fact triplets in structured knowledge graph. In CKE, it encodes the structured data with TransR [20], which can provide more flexibility and is more powerful in handling the *many-to-many* relationships between KOLs and items.

4.4 Ablation Analysis of GoRec

GoRec provides improved recommendations by modeling the influence of KOLs in the community. In this section, we want to evaluate how each component of GoRec impacts the recommendation quality by comparing it with its variants (in Table 3). Note that *GARec* represents the Neural Graph-Autoencoder model, which can be treated as a variant of the proposed *GoRec* without considering the influence of KOLs.

Diffusion via High-order Connectivity. Compared to the basic autoencoder for recommendation (Autorec) [26], GARec comprises multiple GNN layers to encode also the high-order connectivity of users and items. The collaborative denoising Autoencoder (CDAE) [42] is an advanced version of Autorec with an additional embedding vector to characterize user preferences. In GARec, we use GNN layers to capture multi-order connectivity in the user-item interaction graph and a fully-connected layer to reconstruct the feedback vector. We find that by taking the high-order connectivity into consideration, GARec can outperform CDAE by 4.46% and

	Goodreads			Epinions		
	F1	NDCG	Δ(%)	F1	NDCG	Δ(%)
CDAE	9.82	30.04	-	4.60	6.57	-
GARec	9.99	32.20	4.46	4.70	6.62	1.47
GARec+*User*	10.75	32.72	9.20	4.87	7.00	6.21
GARec+*Item*	10.69	32.80	9.02	4.90	6.91	5.84
GoRec	11.02	33.61	12.05	4.97	7.16	8.51

Table 3: Ablation Analysis of GoRec (K=10). ***All the results are in percentage.***

1.47% in Goodreads and Epinions correspondingly. Thus we can conclude that it is necessary to model the high-order proximity between users and items in a sparse scenario. The proposed structure is capable in capturing the opinions and signals diffusion in the community, leading to improved item recommendation.

Fusing Elite Opinions into Diffusion Process. KOLs can directly influence their followers, starting from which their opinions diffuse in the whole community. While modeling user preferences, GARec+*User* extends GARec by fusing the initial user embedding with the attentive aggregation of all the KOLs the user is following, and thus outperforms GARec in both Goodreads and Epinions. This confirms the importance of capturing and modeling the influence of KOLs to their followers in such a direct process. Additionally, KOLs can change how the community views an item by publishing reviews, ratings and tags. In this way, they add community-specific/KOL-defined features to items. In GoRec, we model this process with a fusing layer to enrich the initial embeddings of items. Thus extending from GARec, we add in the item embeddings generated from elite opinion elicitation, and then end up with GARec+*Item*. We find that in both Goodreads and Epinions, by modeling the influence of KOLs on how users view the items, we can further improve the quality of our recommendations. Also we find that the improvement from GARec+*User* is larger than that of GARec+*Item*, which indicates that the influence from whom the user is following is more direct and more significant. And the combination of both diffusion processes leads to even better improvement in both communities.

4.5 Visualization of Elite Opinions

Finally, we want to gain insight into the elite opinions we obtain with the translation-based embedding model in Section 3.2. Based on the opinions embedding $\mathbf{o}_r^e$ from Equation 2, how do the KOLs in Goodreads view or define items in several ambiguous book genres? A tag left by the KOL indicates the genre to which the KOL believes the book should belong. Thus in Figure 5, after deducing their dimensions with t-SNE [21], we plot the embeddings for 5 ambiguous tags and the neighboring elite opinion words. This figure indicates how the Goodreads community defines these genres. We find that in Goodreads, "Sherlock", "detective" or "hacker" related contents are likely to be categorized as Mystery. Books talking about "diet", "biography" or life of "youtubers" are nonfiction. KOLs describe horror contents and paranormal contents closely which cover topics like "zombie", "ghost", "werewolf" and "reaper". And the Goodreads community pays attention to "historic" books for various countries like Scotland and Germany.

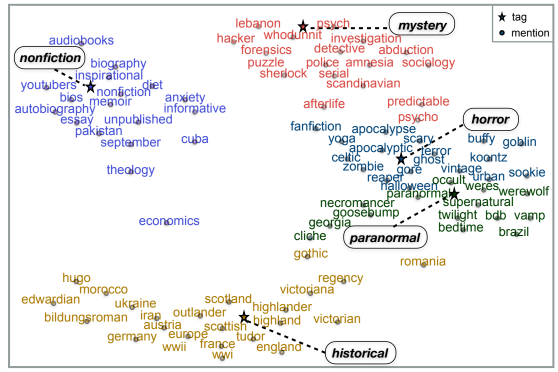

Figure 5: Visualization of Elite Opinions in Goodreads. We show 5 ambiguous tags ("Tag: nonfiction", "Tag: mystery", "Tag: horror", "Tag: paranormal", "Tag: historical") and their neighboring elite opinions (words in black).

5 RELATED WORK

Graph-based Recommendation. There are previous works on exploiting graph structure in recommendation [11, 12, 28, 38, 40, 43, 44]. PsiRec [12] and HOP-Rec [38] employ random walk to generate pseudo linkage between users and items to alleviate the sparsity in feedback data. These models are sensitive to a separate random walk process and thus produce unstable results.

Building on GNN [17], PinSage [46] is proposed to generate embeddings of items (pins in Pinterest), considering both the graph structure (item-item graph) and item features (text and image features), which can be applied for downstream recommendation tasks (e.g. recommend items whose embeddings are the nearest neighbors of the query item's embedding). In this work, we focus on utilizing user-item interactions to model users and items simultaneously for direct recommendation. GCMC [1] is proposed for rating prediction, which handles user-item interaction on different rating levels with separate GNN layers. NGCF [37] improves neural CF [11] utilizing high-order connectivity in the structure of user-item bipartite graph. Different from the Graph-Autoencoder component (GARec) in our GoRec, NGCF concatenates the embedding of multiple graph layers to generate the final latent representations and trains with pair-wise loss. All these methods ignore the impact of KOLs in the communities.

Knowledge and Content Enhanced Recommendation. There is a growing interest in incorporating various auxiliary information into recommendation to deal with the challenge of sparse interactions between users and items. One direction is to extract content information of items to enrich the latent representations, like visual features from images [10, 23], topical information from text descriptions [16, 47] or audio features from multimedia [34].

On the other hand, there are also works on transferring semantic knowledge learned from the relational graph of many knowledge bases (e.g., DBpedia, YAGO, Freebase) for item recommendation [3, 36, 47]. KTUP [3] and MKR [36] combine the tasks of item recommendation and Knowledge Graph (KG) completion, and thus transfer information of entities to items. However, there are challenges in utilizing the KG in recommendation systems. Update and maintenance of the semantic databases highly depends on the contributors. It takes effort to find the correct mapping between items and entities in the KG. An incorrect mapping will introduce noise and hurt the recommendation. Thus, in our work, instead of relying on externally managed KGs, we focus on eliciting elite opinions from KOLs in the community itself, to characterize items and users.

6 CONCLUSION

We propose a novel recommendation system to provide improved item recommendation by taking the influence of key opinion leaders into consideration while exploiting user preferences. It is able to elicit the elite opinions from key opinion leaders with a translation-based embedding method. Meanwhile, building upon multiple GNN layers, the proposed framework can efficiently model the opinion diffusion process. Through experiments on Goodreads and Epinions, the proposed model outperforms state-of-the-art approaches in Top-K recommendation. In the future, we are interested in further exploring how the influence of KOLs can be transferred cross-platform. We also want to develop a flexible model to support some newly-emerging types of opinions (like video blogs).

ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-1841138.

REFERENCES

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

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Chua Tat-seng. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preference. In *WWW*.

[4] Luis V Casaló, Carlos Flavián, and Sergio Ibáñez-Sánchez. 2018. Influencers on Instagram: Antecedents and consequences of opinion leadership. *Journal of Business Research* (2018).

[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*.

[6] Marijke De Veirman, Veroline Cauberghe, and Liselot Hudders. 2017. Marketing through Instagram influencers: the impact of number of followers and product divergence on brand attitude. *International Journal of Advertising* (2017).

[7] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep Anomaly Detection on Attributed Networks. In *SDM*.

[8] Kaize Ding, Yichuan Li, Jundong Li, Chenghao Liu, and Huan Liu. 2019. Graph Neural Networks with High-order Feature Interactions. *arXiv preprint arXiv:1908.07110* (2019).

[9] Leisa Reinecke Flynn, Ronald E Goldsmith, and Jacqueline K Eastman. 1996. Opinion leaders and opinion seekers: Two new measurement scales. *Journal of the academy of marketing science* (1996).

[10] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.

[12] Yun He, Haochen Chen, Ziwei Zhu, and James Caverlee. 2018. Pseudo-Implicit Feedback for Alleviating Data Sparsity in Top-K Recommendation. In *ICDM*.

[13] Wen-Tai Hsieh, Tsun Ku, Chen-Ming Wu, and Seng-cho T Chou. 2012. Social event radar: a bilingual context mining and sentiment analysis summarization system. In *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 163–168.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets.. In *ICDM*.

[15] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*.

[16] Giannis Karamanolakis, Kevin Raji Cherian, Ananth Ravi Narayan, Jie Yuan, Da Tang, and Tony Jebara. 2018. Item Recommendation with Variational Autoencoders and Heterogeneous Priors. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. 10–14.

[17] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[18] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).

[19] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *CIKM*.

[20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

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

[22] Rui Neves-Silva, Marta Gamito, Paulo Pina, and Ana Rita Campos. 2016. Modelling influence and reach in sentiment analysis. *Procedia CIRP* 47 (2016), 48–53.

[23] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural personalized ranking for image recommendation. In *WSDM*.

[24] Douglas W Oard, Jinmook Kim, et al. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.

[26] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*.

[27] Aviv Shoham and Ayalla Ruvio. 2008. Opinion leaders and followers: A replication and extension. *Psychology & Marketing* 25, 3 (2008), 280–297.

[28] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based Social Recommendation via Dynamic Graph Attention Networks. In *WSDM*.

[29] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2019. Node Injection Attacks on Graphs via Reinforcement Learning. In *arXiv preprint arXiv:1909.06543*.

[30] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *WSDM*.

[31] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2019. Robust graph neural network against poisoning attacks via transfer learning. In *arXiv preprint arXiv:1908.07558*.

[32] Mike Thelwall and Kayvan Kousha. 2017. Goodreads: A social network site for book readers. *JASIST* 4 (2017), 972–983.

[33] Tiana Tucker. 2011. Online word of mouth: characteristics of Yelp. com reviews. *Elon Journal of Undergraduate Research in Communications* 2, 1 (2011), 37–42.

[34] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *NeurIPS*.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

[36] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *WWW*.

[37] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[38] Ziqi Wang, Yuwei Tan, and Ming Zhang. 2010. Graph-based recommendation on social networks. In *APWeb*.

[39] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

[40] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In *SIGIR*.

[41] Shaomei Wu, Jake M Hofman, Winter A Mason, and Duncan J Watts. 2011. Who says what to whom on twitter. In *WWW*.

[42] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*.

[43] Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *ACL*.

[44] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *RecSys*.

[45] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. 2015. RAIN: Social Role-Aware Information Diffusion.. In *AAAI*.

[46] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*.

[47] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*.

[48] Yin Zhang and James Caverlee. 2019. Instagrammers, Fashionistas, and Me: Recurrent Fashion Recommendation with Implicit Visual Influence. In *CIKM*.

[49] Ling Zhu, Guopeng Yin, and Wei He. 2014. Is this opinion leader's review useful? Peripheral cues for online review helpfulness. *Journal of Electronic Commerce Research* 15, 4 (2014), 267.

[50] Ziwei Zhu, Jianling Wang, and James Caverlee. 2019. Improving Top-K Recommendation via Joint Collaborative Autoencoders. In *WWW*.