GUID: A Knowledge Graph and Attention based User Interest Diffusion Process for Recommendation

Ding Tu,[1] Jiachen Liang,[2] Zheng Li[3]
CFETS Information Tecnology
tuding@chinamoney.com.cn,[1] liangjc16@tsinghua.org.cn,[2] lizheng@chinamoney.com.cn[3]

Abstract

In item recommendation domain, the explainability and diversity of the recommendation has been paid more and more attention by researchers, especially for some rewarding sensitive tasks, e.g., product recommendation in a financial area. To address this problems, traditional methods like collaborative filtering usually consider to utilize side information or external knowledge to improve the performance since the additional information captures some of the associations and implicit structures between items and user interests. Among them, graph neuron network based methods have been considered as a good alternative to utilize the high order relations in user-item interaction. In this paper, a graph attention and diffusion based recommendation process is proposed to combine item-user interaction and various side information in an unify framework. Compared to existing GNN based methods take embedding-based or path-based methods for recommendation, the proposed method can explicitly show an reasonable entity activation order in a direct way, which can be seen as an graph-based user interest propagation process or a candidate reasoning graph for the recommendation process. In the experiment part, we demonstrate that the graph diffusion process gains competitive results over several state-of-the-art baselines on several datasets, including books, music and financial products.

1 Introduction

Item recommendation has been a central issue in modern web business models, es-pecially in web service industry, e.g., E-commerce, social networks, and search en-gines. The task of item recommendation is to predict the future item that a web ap-plication user might be interested in with the user behavior data and other related data. Since it has has wide application space, the research community has made great efforts in this research area. The first widely used recommdation paradigm is collaborative filtering (CF). CF-based method utilizes user-item interaction history, e.g., rating, buying, browsing, and searching, to extract the potential relations between user and items and recommend similar items or items that a similar user has been concerned about. Due to its effectiveness and universality, CF-based methods has got great successful to solve this problem for several decades. However, traditional CF-based methods fail to use the rich side information in the recommendation con-text. The key difference between human decision process and a CF-based recom-mendation process is that the former usually considers their own situation and the related attributes of the item, which might exists be more clear in the side infor-mation, while the latter considers the similarities between user-item interaction history to fit user prereferences indirectly. This limits the performance and explainability of the results, while explainability and Interpretability is important for many high risk industry, e.g., financial industry or medical industry.

Knowledge base or knowledge graph is an important source of side information, e.g, there are rich structured and text data in financial industry. To leverage the power of these valuable information in recommendation, much work has been done to involve external knowledge in to the recommendation process. Existing KG-involved recommendation works can be divided into two types: the first type methods model-ing KG into low-dimensional vector-based embedings to represent the users and items, then use inner-product or a predict function to give the final result; the second type mehods uses paths that extracts from various relation patterns to model the correlations between different entity nodes.Among these methods, graph neron net-works (GNN) methods has attracted the attention of many researchers in recent years.

The advantage of GNN-based methods is that their modes can exploit the high or-der underlying relations in the data structure in an explicit way, therefore they can get better model explainability and a higher potential performance. The common way to generate recommendation explaination is to construct an explaination module, which might output several import factors or a chain of factors according to the model weights of the factors. While a real recommendation decision process made by a saler or a broker might be a slightly different. For example, they may observe the user behavior history and guess the factors why they behave like this, then they look for association factors based on the former ones and decide the candidates. The key difference is that in a real recommendation process, all relevant factors are involved in a contiuous, dynamic process and affects the process with different weights as a whole, which means they act like a dynamic graph.

According to the above considerations, we propose an activation Ggraph based User Interest Diffusion process (GUID) to explicitly exploit local structures of KG. The key idea behind this design is to simulate the dynamic evolving recommendation process upon a rich knowledge graph and consider all known factors in a single unified graph process. Compared to existing methods, the three advantage of our methods are: 1) It mainly considers local structures and relative closeness on KG, which can reduce the computation cost; 2) The graph activation state is an intermediate result in our process, and a sequence of activation graph rooted from a user node can be naturally regarded as a reasoning process of interests of the user, which means our method might have better explainability compared to path-based or key factor based methods; 3) It explores to treat The nodes form heterogeneous information network in an unified way in a single graph, which means the correlations from different relations can be considered at the same time.

In summary, our contributions in this paper are as follows:

- We propose a graph based user interest diffusion process utilizing the local structures of the entities, which models the user interests generation process as an unified dynamic activation graph through local and global attention mechanisms.
- We propose GUID, an end-to-end framework use the generated user interest graph to give appropriate item recommendations. Compared to other methods, the activation graph can be naturally regarded as an explanation process for the recommendations, and it exploits the user-item interactions and factors from heterogeneous knowledge graph in a unified way.
- We conduct experiments on three real-world recommendation scenarios, and the results prove the efficacy of GUID over several state-of-the-art baselines. And the proposed framework is applied in the recommendation scenarios of a financial exchange.

2 Related Works

To address the sparsity problem of collaborative filtering [He and Chua, 2017], researchers usually make use of side information, such as social networks or item attributes, to improve recommendation performance, Among various types of side information, knowledge graph (KG) usually contains much more fruitful facts and connections about items. It is an intuitive idea to use KG utilizing these side information to enhance the performance of recommender system [Li et al., 2019].

A KG is a type of directed heterogeneous graph in which nodes correspond to entities and edges correspond to relations, being a very general abstract descriptions of relation and interaction systems, are ubiquitous in different areas of science. Graph-based learning models have been successfully applied in social networks, link prediction [Cao et al, 2018], human-object interaction [Qi et al, 2018], particle physics [Choma et al, 2018] etc.

Inspired by the success of applying KG in a wide variety of tasks, researchers attempted to leverage KG to improve the performance of recommender systems. KG can help find the latent connections between entities and recommended items through semantic relatedness among items; the relations with various types on a KG is also helpful for extending a user's interests reasonably and increasing the diversity of recommended items [Ai et al., 2018].

Some researchers proposed path-based methods to utilize the KG for recommender systems, which utilize additional guidance for recommendations to explore the various patterns of connections among items in KG for recommendations [Wang et al, 2017]. For example, Personalized Entity Recommendation (PER) [Yu et al., 2014] and Meta-Graph Based Recommendation [Zhao et al, 2017] treat KG as a heterogeneous information network (HIN) and extract meta-path based features to represent the relations between users and items on the KG. But difficulty of reasoning over the hard-coded paths on heterogeneous knowledge graph prevents existing approaches from extract latent features on very different entities and relations [Wang et al, 2017]. Knowledge graph embedding-based is another method which pre-process a KG, entities and relations are learned as vector representations, and the connectivity between entities under a certain relation can be calculated in a soft manner based on their representations. For example, Deep Knowledge-aware Network (DKN) [Wang et al, 2018] make entity embeddings and word embeddings separately, then designs a CNN framework to combine them together for news recommendation. Collaborative Knowledge base Embedding (CKE) [Zhang et al, 2016] combines a CF module with knowledge embedding, text embedding, and image embedding of items in a unifed Bayesian framework. KG Embedding-based methods show high flexibility in utilizing KG to improve the performance of recommender systems.

3 Task Formulation

In this section, we first briefly give the definitions of basic concepts and terms, and then give the definitions of the problem. In the rest part of the paper, bold upper-case letters denote matrices and bold lower-case letters denote vectors. [·] denotes a sequence, { · } de notes a set, and <·> denotes a vector. | · | denotes the operation of vector concatenation.

Definition 1 (User-Item Interaction Graph): A user-item interaction graph is a bipartite graph $\boldsymbol{G_i} = (\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{R_i})$, where nodes in $\boldsymbol{G_i}$ denote the all relevant users $\boldsymbol{U}$

and relevant items $\boldsymbol{I}$. Each user and item are represented by a user feature vector and an item feature vector $\boldsymbol{u}, \boldsymbol{i} \in \mathbb{R}^d$, respectively. Edges(relations) in $\boldsymbol{G_i}$ denote the historical interactions $\boldsymbol{R_i} = \{ \boldsymbol{r_i} |\ \boldsymbol{r_i} = inter(\boldsymbol{u}, \boldsymbol{i}), \boldsymbol{u} \in \boldsymbol{U}\ and\ \boldsymbol{i} \in \boldsymbol{I}\}$ between the users and items in $\boldsymbol{G_i}$(e.g., purchasing or rating). Here $inter(\cdot\,,\cdot)$ indicates the interaction state of two nodes in the graph, where 1 means the two have interactions and vice versa.

Definition 2 (Knowledge Enhanced Interaction Graph): A knowledge enhanced interaction graph $\boldsymbol{G_{ki}} = \{\boldsymbol{E}, \boldsymbol{R}\ \}$ is built by fusing a user-item interaction graph $\boldsymbol{G_i}$ and related knowledge graph $\boldsymbol{G_k} = \{\boldsymbol{E_k}, \ \boldsymbol{R_k}\ \}$, where $\boldsymbol{E}$ denotes the entities and $\boldsymbol{R}$ denotes edges in graph. Each entity is represented by a numeric feature vector $\boldsymbol{e} \in \mathbb{R}^d$. The fusing process is accomplished by aligning nodes $\{\boldsymbol{u}, \boldsymbol{i}\ |\ \boldsymbol{u} \in \boldsymbol{U},\ \boldsymbol{i} \in \boldsymbol{I}\}$ in $\boldsymbol{G_i}$ and entities $\{\boldsymbol{e_k} \in \boldsymbol{E_k}\}$ in $\boldsymbol{G_k}$. The edges in $\boldsymbol{G_{ki}}$ is the union set $\{\ \boldsymbol{R_k} \cup\ \ \boldsymbol{R_i}\}$ of edges in $\boldsymbol{G_k}$ and $\boldsymbol{G_i}$.

Definition 3 (Interest Activation Graph): An interest activation graph $\boldsymbol{G_a}$is a subset $\{\boldsymbol{e_a}\ |\ \sigma(h(\boldsymbol{e_a})) > \tau,\ \boldsymbol{e_a}\ \in \boldsymbol{E}\}$ of $\boldsymbol{G_{ki}}$, where $\sigma(\cdot)$ is an activation function defined on $\boldsymbol{G_{ki}}$, $h(\,\cdot\,)$ gets the diffusion energy θ from the neighbors of an entity and τ is the threshold value to determine whether an entity is activated. All entities in an interest activation graph contains relevant entities for a specific user, which can be regarded as his activated potential interest. Each entity on $\boldsymbol{G_a}$ has a corresponding activation energy θ (or activation weight) ranging from 0 to1, with a larger θ means a more active interest.

Definition 4 (Interest Diffusion Process): The interest diffusion process is defined as a transition process between a sequence of interest activation graph $[\boldsymbol{G_a^0}, \boldsymbol{G_a^1}, \boldsymbol{G_a^2}, ..., \boldsymbol{G_a^{t-1}}, \boldsymbol{G_a^t}]$, where $\boldsymbol{G_a^t}$ is the interest activation graph at diffusion step *t*. Each interest activation graph is an intermediate transition state of an interest diffusion process, and the interest activation graph $\boldsymbol{G_a^t}$ at diffusion step *t* can be got by performing interest diffusion from $\boldsymbol{G_a^{t-1}}$ on $\boldsymbol{G_{ki}}$ with $\boldsymbol{G_a^t} = \{\boldsymbol{e_a^t}\ |\ \boldsymbol{e_a^t} \in diffu(\boldsymbol{e_a^{t-1}}, \boldsymbol{G_{ki}}),\ \}$. $diffu(\boldsymbol{e_a^{t-1}}, \boldsymbol{G_{ki}})$ returns a set of activated entities by diffusing activation energy to all neighbors of an entity in $\boldsymbol{G_a^{t-1}}$, where the neighbors of an entity is defined as $neg(\boldsymbol{e}) = \{\boldsymbol{e}\ |\ inter(\boldsymbol{e}, \boldsymbol{e_i}) > 0,\ \boldsymbol{e_i}\ \in \boldsymbol{E}\}$ on $\boldsymbol{G_{ki}}$. Fig.1 is an example of movie interest diffusion process, and each bar on a graph node is the activation energy.

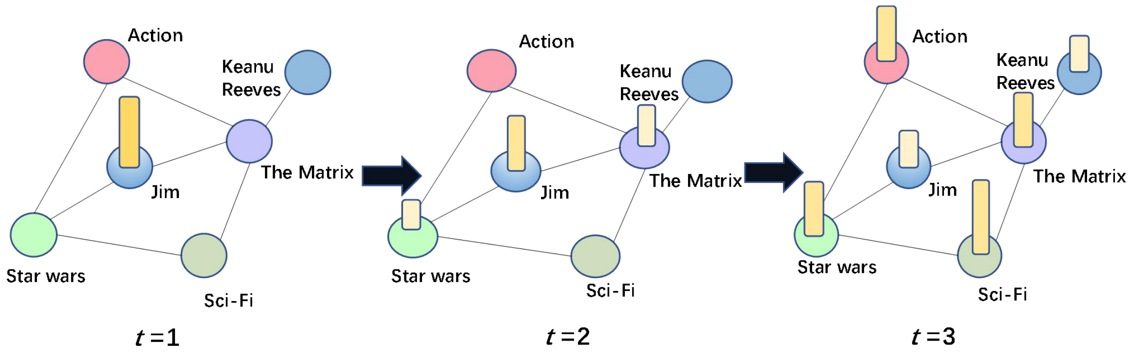

Figure 1: Interest Diffusion process

The task of a knowledge enhanced item recommendation can be formulated as: Given a knowledge enhanced interaction graph $\boldsymbol{G_{ki}}$ and a user *u*, to predict the potential interest of user *u* in an item *i* that has no interactions with *u* in the history . To achieve this goal, we explore to conduct an interest diffusion process to generate the interest activation graph of *u* on $\boldsymbol{G_{ki}}$.

4 Methodology

4.1 Framework

The overall structure of the proposed GUID network is shown in Fig.2, which takes user *u* and a knowledge enhanced interaction graph $\boldsymbol{G_{ki}}$ as input. The initial state is $\boldsymbol{G_a^0}$, which consist of only one activated entity corresponding to user *u*. Each entity in $\boldsymbol{G_a^t}$ is represented by a numeric feature vector $\boldsymbol{e_a}$ after processed by the embedding layers in Fig.2, and local and global attention mechanisms are exploited based on the embedding vectors to determine the strength of interest diffusion process. Then the most active neighbor entities at diffusion step *t* are added to $\boldsymbol{G_a^{t+1}}$ and activation energy $\boldsymbol{\theta}$ of all entities in $\boldsymbol{G_a^{t+1}}$ are updated. After several iterations the final recommendation items are selected from the interest activation graph according to their activation energy.

4.2 Entity Embedding

In most knowledge-graph based recommendation methods, the recommendation result is got by ordering items according to the inner-product of user embeddings $\boldsymbol{U}$ and item embeddings $\boldsymbol{I}$. To leverage the knowledge information in kg, the underlying embeddings usually needs to preserve the graph structure in the encoding implicitly.

However, the graph structure is explicitly exploited in GUID by diffusing along the geometric structures. Another fact is that entities related to same relations usually connected by same entities (attributes), which means they distribute on a local graph of $\boldsymbol{G_{ki}}$. If the related relations are important factors for the final recommendation, then the diffusion energy between entities in this local graph should be large, i.e., GUID learns embeddings with large correlations for the entities related to same relations. Thus, we do not explicitly consider to encode the knowledge graph structure and relations in the embedding process.

In GUID, all entities get corresponding embeddings $\boldsymbol{E}$ by ID. Since the diffusion process mainly consider local structures on a graph, GUID may encounter the problem of non-convergence or worse local minima. This may happen if too many irrelated entities are placed in a small area of the feature space when the average degree or the num of entities is too large. To avoid this problem, using an appropriate pretraining method to get a better initial state is a good way. The initial embeddings from a pretraining model can relatively distribute the entities uniformly in the vector space while preserves the relative closeness between the entities.

Here we can take similar pretraining model as in KGAT[Wang et al., 2019], which performs BPR[Rendle et al., 2009] on user-item interaction data to get the initial embeddings. Given the interaction history between users and items, BPR can generate user and item embeddings $\mathbf{E_U}$ and $\mathbf{E_I}$ that satisfy:

$$\mathbf{E_U}\,,\mathbf{E_I} = \max_{\mathbf{E_U}\,,\mathbf{E_I}} \sum_{e_u \in \mathbf{E_U}\,, \boldsymbol{e}_i, \boldsymbol{e}_j \in \mathbf{E_I}} \ln\sigma(\boldsymbol{e_u}\boldsymbol{e_i}^{\mathbf{T}} - \boldsymbol{e_u}\boldsymbol{e_j}^{\mathbf{T}}) - \lambda_{\boldsymbol{\Omega}} \parallel \boldsymbol{\Omega} \parallel^2$$

Here $\boldsymbol{\Omega}$ is the parameters in the model and $\lambda_{\boldsymbol{\Omega}}$ is a regularization parameter. $\boldsymbol{e_i}$ and $\boldsymbol{e_j}$ are representing vectors of item i and item j. For a large data set, parallel principal component analysis could also be used since we only need a similar dimension reduction method to anchor the initial nodes appropriately in the vector space.

4.3 Attentive Interest Diffusion

After the embeddings of the entities are generated, the activation energy θ_u of the entity corresponding to the input user $\boldsymbol{e_u}$ is set to 1 and it starts to diffusing its energy across $\mathbf{G}_{ki}$ with the interest diffusion process. Compared to the path-based methods, the advantage of GUID is that the candidate item is generated with a chain of interest activation graph rather than a sequence of graph entities, as shown in Fig. 1.

In a path-based method, an item recommendation might be generated along a sequence, e.g., $u_1 \rightarrow i_1 \rightarrow e_1 \rightarrow i_2$ or $u_1 \rightarrow i_1 \rightarrow e_1 \rightarrow i_3$, and the context information or high-order connectivity relations are encoded in the embeddings through aggregation operations. While in GUID, an item recommendation is generated among a sequence of graph, e.g., $\{u_1\} \rightarrow \{i_1, i_2, e_1\} \rightarrow \{e_2, i_3, u_2\} \rightarrow \{i_4, i_5\}$ or $\{u_1\} \rightarrow \{i_1, i_2, e_1\} \rightarrow \{i_3\}$, and the context entities can impact the diffusion process directly. The many-to-many activation way is much more flexible than the one-to-one propagation way using in the path-based methods, and another advantage is that it can utilize the interaction between all activated entities (more related entities) and potential neighbor entities, while other methods usually limits the high-order connectivity to a fixed number.

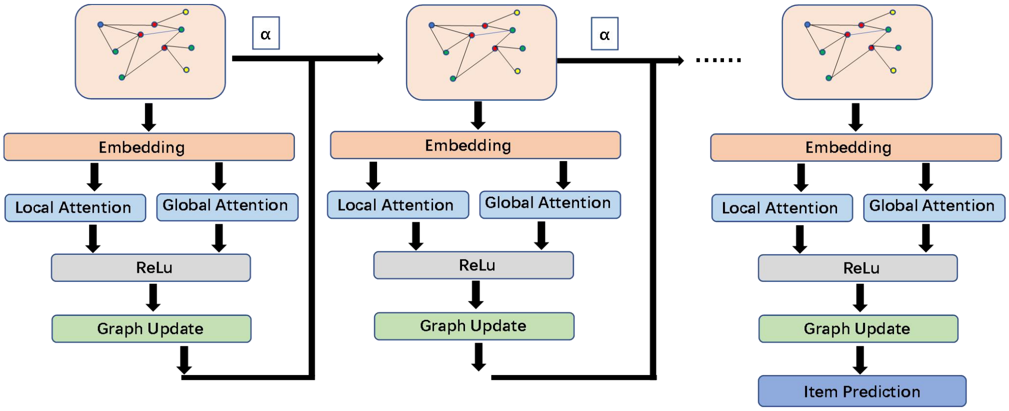

Figure 2: GUID Framework

Energy Diffusing. The interest diffusion process in GUID is accomplished by diffusing the activation energy of entity $\boldsymbol{e_a}$ in $\boldsymbol{G_a^t}$ to its neighbors neg$(\boldsymbol{e_a})$ on $\boldsymbol{G_{ki}}$. Then for an entity $\boldsymbol{e}$ in $\mathbf{E}_n = \{\text{neg}(\boldsymbol{e_a})\, |\boldsymbol{e_a} \in \mathbf{G}_a^t\}$, the energy it receives can be summed as:

$$\widehat{\theta_{\boldsymbol{e}}} = \sum_{\boldsymbol{e_a} \in \mathbf{G}_a^t \cap \text{neg}(\boldsymbol{e})} \mu(\boldsymbol{e_a}, \boldsymbol{e})\, \theta_{\boldsymbol{ea}}$$

Where $\mathbf{E}_{\boldsymbol{n}}$ is the union set of neighbors of all entities in $\boldsymbol{G_a^t}$, and $\theta_{\boldsymbol{ea}}$ is the activation energy of $\boldsymbol{e_a}$. $\mu(\boldsymbol{e_a}, \boldsymbol{e})$ is the diffusing strength between $\boldsymbol{e_a}$ and $\boldsymbol{e}$, which is generated through a local attention mechanism.

Local Graph Attention. The diffusing strength $\mu(\boldsymbol{e_a}, \boldsymbol{e})$ between an activated entity $\boldsymbol{e_a}$ and a neighbor entity $\boldsymbol{e}$ is defined as follows:

$$\mu_l(\boldsymbol{e_a}, \boldsymbol{e}) = \tanh\,(\mathbf{W}_{\boldsymbol{al}}\boldsymbol{e_a} + \boldsymbol{b_{al}})^{\mathrm{T}}\tanh(\mathbf{W}_{\boldsymbol{al}}\boldsymbol{e} + \boldsymbol{b_{al}})$$

Here $\boldsymbol{e_a}$ and $\boldsymbol{e}$ are first transformed by a linear mapping and then activated with a non-linear function tanh. If $\boldsymbol{e_a}$ and $\boldsymbol{e}$ have larger similarity in the transformed attention space, then more energy would diffusing from $\boldsymbol{e_a}$ to $\boldsymbol{e}$. To avoid unbalanced diffusing strength from different activated entities to $\boldsymbol{e}$, the diffusing strength of path start from $\boldsymbol{e_a}$ is normalized by:

$$\mu_l(\boldsymbol{e_a}, \boldsymbol{e}) = \frac{\exp\,(\mu_l(\boldsymbol{e_a}, \boldsymbol{e}))}{\sum_{\boldsymbol{e_t}\in \mathrm{neg}(\boldsymbol{e_a})} \exp\,(\mu_l(\boldsymbol{e_a}, \boldsymbol{e_t}))}$$

Unlike the embedding aggregation used in other graph neuron based methods, the local graph attention only considers the 1-hop structure of an activated entity, i.e., it focus on the local structure of $\boldsymbol{e_a}$. The message passing entities is only the activation energy rather than feature embeddings. Therefore, the local graph attention can be processed and optimized much faster than the embedding aggregation since it only concerns the relative order and closeness in a local region.

Global Graph Attention. The advantage of GUID is that it can exploit the context entities in $\boldsymbol{G_a^t}$ to influence the interest diffusion process explicitly, i.e., the long range connectivities between a history activated entity in $\boldsymbol{G_a^t}$ and a neighbor entity of $\boldsymbol{G_a^t}$ is considered. Through this, the input user entity or some highly related entities can exert influence on consequent diffusion processes continuously, and take more diversity into the generated recommendation. This is achieved by adopting a global graph attention mechanism. Unlike the micro structures used in local graph attention, the global graph attention mainly focus on the macro structures. First we give the definition of neighbor entities of $\boldsymbol{G_a^t}$ on $\boldsymbol{G_{ki}}$:

$$\mathrm{neg}(\boldsymbol{G_a^t}) = \{\mathrm{neg}(\boldsymbol{e_a}) \mid \boldsymbol{e_a} \in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}\}$$

Given a activated graph $\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}$ and a neighbor entity $\boldsymbol{e}$ of $\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}$, the global attention of $\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}$ on $\boldsymbol{e}$ is:

$$\mu_g(\boldsymbol{e_a}, \boldsymbol{e}) = \tanh\,(\mathbf{W}_{\boldsymbol{ag}}\boldsymbol{e_a} + \boldsymbol{b_{ag}})^{\mathrm{T}}\tanh(\mathbf{W}_{\boldsymbol{ag}}\boldsymbol{e} + \boldsymbol{b_{ag}})$$

$$\boldsymbol{e_a} \in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}\ \boldsymbol{e} \in \mathrm{neg}(\boldsymbol{G_a^t})$$

$$\mu_g(\boldsymbol{e_a}, \boldsymbol{e}) = \frac{\exp\,(\mu_g(\boldsymbol{e_a}, \boldsymbol{e}))}{\sum_{\boldsymbol{e_t}\in \mathrm{neg}(\boldsymbol{G_a^t})} \sum_{\boldsymbol{e_{at}}\in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}} \exp\,(\mu_g(\boldsymbol{e_{at}}, \boldsymbol{e_t}))}$$

The computation cost of global graph attention is almost proportional to the square of the entity size of $\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}$. To speed up the global graph process in a long diffusion sequence or a dense connected $\boldsymbol{G_{ki}}$, it can be simplified as:

$$g(\boldsymbol{G_a^t}) = \frac{\sum_{\boldsymbol{e_{at}}\in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}} \boldsymbol{e_{at}}\theta_{\boldsymbol{ea}}}{\sum_{\boldsymbol{e_{at}}\in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}} \theta_{\boldsymbol{ea}}}$$

$$\mu_g(\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}, \boldsymbol{e}) = \tanh\,(\mathbf{W}_{\boldsymbol{ag}} g(\boldsymbol{G_a^t}) + \boldsymbol{b_{ag}})^{\mathrm{T}}\tanh(\mathbf{W}_{\boldsymbol{ag}}\boldsymbol{e} + \boldsymbol{b_{ag}})$$

$$\mu_g(\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}, \boldsymbol{e}) = \frac{\exp\,(\mu_g(\mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}, \boldsymbol{e}))}{\sum_{\boldsymbol{e_t}\in \mathrm{neg}(\boldsymbol{G_a^t})} \exp\,(\mu_g(\boldsymbol{e_{at}}, \boldsymbol{e_t}))}$$

Here $g(\boldsymbol{G_a^t})$ is a weighted context vector of $\boldsymbol{G_a^t}$ using activation energy θ as weight.

Interest Graph Update. After the energy diffusion between entities, all states of the entities are updated. For a certain entity $\boldsymbol{e}$, its receiving energy is as follows:

$$\widehat{\theta_{\boldsymbol{e}}} = \begin{cases} \lambda \sum\limits_{\boldsymbol{e_a}\in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}} \cap \mathrm{neg}(\boldsymbol{e})} \mu_l(\boldsymbol{e_a}, \boldsymbol{e})\, \theta_{\boldsymbol{ea}} + (1-\lambda) \sum\limits_{\boldsymbol{e_a}\in \mathbf{G}_{\boldsymbol{a}}^{\boldsymbol{t}}} \mu_g(\boldsymbol{e_a}, \boldsymbol{e})\theta_{\boldsymbol{ea}} & \boldsymbol{e} \in \mathrm{neg}(\boldsymbol{G_a^t}) \\ 0 & \boldsymbol{e} \notin \mathrm{neg}(\boldsymbol{G_a^t}) \end{cases}$$

Here parameter λ controls the energy ratio of local graph attention and global graph attention. Since GUID is designed to exploit local structure of $\boldsymbol{G_{ki}}$, λ is usually set as a number closer to 1. The energy of $\boldsymbol{e}$ is updated by:

$$h(e, t) = \widehat{\theta_{\boldsymbol{e}}} + \alpha\theta_e^t$$

α is the decay factor to reduce the energy of existing activated entities, which can avoid that entities in the earlier activation graph take too large weight in $\boldsymbol{G_a^t}$**.** Then the energy of all entities is normalized as:

$$\theta_e^{t+1} = RELU\left(\frac{\exp\big(h(e,t)\big)}{\sum_{\boldsymbol{e_t}\in\, \boldsymbol{G_{ki}}} \exp\big(h(e,t)\big)} - \tau\right)$$

All entities with energy larger than a threshold τ are added into $\boldsymbol{G_a^{t+1}}$ and a new interest graph is generated**.** τ is a small value to control the number of activated entities in $\boldsymbol{G_a^{t+1}}$, as a user usually cannot handle too many interests at the same time. After this, a new interest diffusion process starts.

4.4 Optimization Algorithm

The out recommended item is selected from the activated entities of the latest $\boldsymbol{G_a}$ and ordered by corresponding activation energy**.** To improve the diversity of the recommendation result, we use a pair-wise rank loss similar to Ranknet[Burges et al, 2005]. Given an interest activation graph $\boldsymbol{G_a}$ of and historical items $\boldsymbol{I_u}$ of user u, then the optimization loss function of u is defined as follows:

$$\mathrm{P}(\boldsymbol{e_i}, \boldsymbol{e_j}) = \frac{\exp\,(\theta_{ei} - \theta_{ej})}{1 + \exp\,(\theta_{ei} - \theta_{ej})}$$

$$\mathcal{L}(\boldsymbol{e_u}) = - w_{ij}\ln\Big(\mathrm{P}(\boldsymbol{e_i}, \boldsymbol{e_j})\Big) - (1 - w_{ij})\ln\Big(1 - \mathrm{P}(\boldsymbol{e_i}, \boldsymbol{e_j})\Big) \quad (\boldsymbol{e_i}, \boldsymbol{e_j}) \in D_s$$

where

$$w_{ij} = \begin{cases} 1 & \boldsymbol{y}(\boldsymbol{e_i}) > \boldsymbol{y}(\boldsymbol{e_j}) \\ 0 & \boldsymbol{y}(\boldsymbol{e_i}) < \boldsymbol{y}(\boldsymbol{e_j}) \\ 0.5 & \boldsymbol{y}(\boldsymbol{e_i}) = \boldsymbol{y}(\boldsymbol{e_j}) \end{cases}$$

Here $\boldsymbol{y}(\boldsymbol{e_i})$ is the ground truth preference score of user u on item i, e.g., the ratings or purchase times. D_s is a sampled sub set of all items. In each training iteration, we randomly sample a minibatch of positive/negative interactions from ground truth and corresponding prediction from $\boldsymbol{G_a}$**.** The final loss is calculated by adding optimization loss of all users:

$$\mathcal{L} = \sum_{e_u \in \mathbf{E_U}} \mathcal{L}(\boldsymbol{e_u}) + \lambda_{\boldsymbol{\Omega}} \parallel \boldsymbol{\Omega} \parallel^2$$

Here **Ω** is the model parameters. The optimization problem is solved by employing a stochastic gradient descent (SGD) algorithm.

		Amazon-book	**Last-FM**	**Yelp2018**	**X-bond**
User-Item Interaction	#Users	70,679	23,566	45,919	1,583
	#Items	24,915	48,123	45,538	8,422
	#Interactions	847,733	3,034,796	1,185,068	1,823,462
Knowledge Graph	#Entities	88,572	58,226	90,961	2,352
	#Relations	39	9	42	10
	#Triplets	2,557,446	464,567	1,853,704	1,843,6273

Table 1: Data set information

5 Experiments

5.1 Experimental Setup

Dataset: To evaluate the effectiveness of KGAT, we utilize three public benchmark datasets: Amazon-book[1], Last-FM[2], and Yelp2018[3], which are publicly accessible and vary in terms of domain, size, and sparsity, the detail information is presented in Table 1.

For each dataset, we randomly select 80% of interaction history of each user to constitute the training set, and treat the remaining as the test set. From the training set, we randomly select 10% of interactions as validation set to tune hyper-parameters. For each observed user-item interaction, we treat it as a positive instance, and then conduct the negative sampling strategy to pair it with one negative item that the user did not consume before.

Evaluation Metrics:For each user in the test set, we treat all the items that the user has not interacted with as the negative items. Then each method outputs the user's preference scores over all the items, except the positive ones in the training set. To evaluate the effectiveness of top-K recommendation and preference ranking, we adopt two widely-used evaluation protocols [He and Chua, 2017]: recall@K and precision@K. By default, we set K = 20. We report the average metrics for all users in the test set.

Hyper Parameters: In our experiment, we set the learning rate to 0.001, the batch size to 1000, and the training epoch to 100 empirically. The datasets are split in chronological order with the first 70% for training, the last 20% for test, and the other 10% for validation.

In addition, the parameters of GUID is set as λ =0.8 τ=0.000001 , α=0.5, $\lambda_{\mathbf{\Omega}}$=0.05.

5.2 Parameter Evaluation

To study the model parameters that affects the recommendation performance, we choose two important factors: parameter λ that controls the energy ratio of local graph attention and global graph attention, and global interest diffusion number *t*.

Local Attention Ratio λ**:** To check the effect of different λ, we conduct experiment on Amazon data set with λ=0.5,0.7, 0.8,0.9. The result is shown in Table 2:

λ	**Recall@5**	**NDCG@5**	**Recall@20**	**NDCG@20**
0.5	0.0853	0.1031	0.1742	0.1445
0.7	0.0881	0.1069	0.1767	0.1475
0.8	**0.0888**	**0.1078**	**0.1771**	**0.1479**
0.9	0.0884	0.1078	0.1758	0.1477

Table 2: GUID performance with different λ on Amazon

The result shows that set λ=0.8 get the best result in these settings. As λ becomes closer to 1, the result of GUID appears better until λ=0.8. But the performance gain increases much slower as the increase of λ. If global attention occupies a large ratio in the interest diffusion process, the effect of local attention might be worse due to the impact of global attention.

Interest diffusion number *t*: In our experiment, *t* is defined as the interest diffusion number through global graph attention. The interest diffusion number through local graph attention is at least 3 (the least number to generate item recommendation in GUID). For interest diffusion process without global graph attention, the diffusing energy is updated only with local graph attention. *t* is set as 1,2,3,4 here. The result of this evaluation on Amazon is:

t	**Recall@5**	**NDCG@5**	**Recall@20**	**NDCG@20**
1	0.0843	0.0998	0.1706	0.1402
2	0.0861	0.1048	0.1739	0.1448
3	0.0888	0.1078	**0.1771**	0.1479
4	**0.0889**	**0.1089**	0.1762	**0.1485**

Table 3: GUID performance with different *t* on Amazon

The result shows that set *t* =3,4 get the best result in these settings. As *t* becomes closer to 3, the performance of GUID increases quickly. While t increase from 3 to 4, the performance gain little. The reason might be that when *t* is

[1] http://jmcauley.ucsd.edu/data/amazon/

[2] https://grouplens.org/datasets/hetrec-2011/

[3] https://www.yelp.com/dataset/challenge

small, the global attention can have good effect. While when t is large, the activated nodes in interest activation graph becomes more and this reduce the diversity of global attention, since the embeddings is mainly updated according to local structure and a larger t means the global attention should handle interactions across several local graph.

5.3 Comparison with Baseline on Public Dataset

To evaluate the effectiveness of the proposed method, we compare GUID with the baseline methods described above (NFM in [He and Chua, 2017]). All parameter settings of GUID is set as default, and Table 4 shows the result of all methods

	Amazon			**Last-FM**			**Yelp**		
method	Recall@20	Precision@20	NDCG@20	Recall@20	Precision@20	NDCG@20	Recall@20	Precision@20	NDCG@20
BPR	0.1305	0.0139	0.0694	0.0723	0.0303	0.0617	0.0678	0.0165	0.0435
NFM	0.1383	0.0138	0.0662	0.0742	0.0279	0.0591	0.0594	0.0143	0.0355
CKE	0.1382	0.0145	0.0671	0.0735	0.0309	0.0602	0.0657	0.0149	0.0372
KGAT	0.1411	0.0142	0.0751	0.0742	0.0313	0.0697	0.0678	0.0152	0.0423
GUID	**0.1771**	**0.0195**	**0.1479**	**0.1252**	**0.0528**	**0.1047**	**0.0691**	**0.0169**	**0.0481**

Table 4: Performance evaluation on public dataset

The result of GUID outperforms all baseline methods in this experiment section. On Amazon and Last-FM dataset, it outperforms the competitors with at least 20% improvement, which shows the great potential of GUID. All methods performs worse on Yelp dataset, but GUID still get the best performance on this data set, and the precision still has big improvement. The KG based methods does not outperform BPR on this data set, which might be because the KG on Yelp has less useful information or the interactions between entities are more complicated. Therefore, the improvement on Yelp is much less than on other data sets.

5.4 Comparison with Baseline on Financial Dataset

GUID has been applied in a real financial exchange scenarios to recommend financial products to potential users. In this section, we evaluate the performance of GUID on the sampled financial dataset- X-bond, and the result is presented in Table 5.

method	**Recall@5**	**Precision@5**	**NDCG@5**
BPR	0.2925	0.0722	0.0842
NFM	0.2916	0.0832	0.0735
CKE	0.2985	0.0817	0.0785
KGAT	0.3044	0.0812	0.0864
GUID	**0.4412**	**0.1312**	**0.1423**

Table 5: Performance evaluation on financial scenario

On X-bond the GUID gets the best performance, which is the only method that get recall value larger than 40%, and the improvement is nearly 50%. The user-item interactions in X-bond is much sparse than in the three public data set and the number of entities is relatively less than the others. Therefore, the impact of local graph will be large and the method relies on local structure might be a better choice.

6 Conclusions And Future Work

In this paper, we investigate the problem of item recommendation with knowledge graph and propose a activation graph based interest diffusion process, and presents a recommendation framework GUID based on the interest diffusion process, which takes the local structure of the entity interaction graph into consideration and models the user interest as a sequence of activation graph. The advantage of our method is that it mainly exploits local structures and the activation-graph based diffusion process is much more efficient than path-based methods, the effect of which has been proved in our experiment.

Our work could be extended in the following directions. First, the local structure used in this paper is relatively simple, which can be improved to exploit more complicated features. Second, there are also many methods that utilize local structures, the connection between them still need to be investigated. And lastly, the embedding method used in this paper is uniform for all entities, while the effect of local encoding methods should be explored in the future.

Acknowledgments

Thanks are due to Mr. Bao Xiaojing for valuable discussion. Thanks to the PBOC youth reasearch grant.

References

[Ai et al., 2018] Ai Qingyao, Vahid Azizi, Xu Chen and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms, 11*(9).

[Burges et al, 2005] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning (pp. 89-96),* August, 2005.

[Choma et al, 2018] Nicholas Choma, Federico Monti, Lisa Gerhardt, Tomasz Palczewski, Zahra Ronaghi, Prabhat, Wahid Bhimji, Michael M. Bronstein, Spencer R. Klein, and Joan Bruna. Graph neural networks for icecube signal classification. In *Proc. ICMLA,* 2018.

[Cao et al, 2018] Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. Neural Collective Entity Linking. In *COLING. 675–686.* 2018

[Hasson et al, 2020] Uri Hasson, Samuel A.Nastase, and Ariel Goldstein. Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks. *Neuron 105.3(2020):416-434*, 2020

[He and Chua, 2017] Xiangnan He and Tat-Seng Chua. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR. 355–364,* 2017

[Li et al., 2019] Qianyu Li, Xiaoli Tang, Tengyun Wang, Haizhi Yang and Hengjie Song. Unifying task-oriented knowledge graph learning and recommendation. *IEEE Access, PP*(99), 1-1.

[Qi et al, 2018] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV, pp. 401–417*, 2018.

[Rendle et al., 2009] Rendle, S. , Freudenthaler, C. , Gantner, Z. , and Schmidt-Thieme, L. . BPR: Bayesian personalized ranking from implicit feedback. *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. AUAI Press.mann.

[Rendle et al, 2011] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. *In SIGIR. 635–644,* 2011

[Vaswani et al, 2018] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, *6000–6010*, 2018.

[Wang et al, 2017] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item Silk Road: Recommending Items from Information Domains to Social Users. In *SIGIR. 185–194,* 2017

[Wang et al, 2018] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. DKN: Deep-Knowledge-Aware Network for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 1835–1844*, 2018

[Wang et al., 2018]Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 2018.

[Wang et al., 2019] Xiang Wang, Xiangnan He, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. KGAT: Knowledge Graph Attention Network for Recommendation. *the 25th ACM SIGKDD International Conference.* ACM, 2019

[Yu et al., 2014] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. *In Proceedings of the 7th ACM International Conference on Web Search and Data Mining. 283–292*, 2014.

[Zhang et al, 2016] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 353–362,* 2016.

[Zhang and Chen, 2018] Zhang Muhan, and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[Zhao et al, 2017] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2017.