Information Sciences 584 (2022) 170–183

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Food recommendation with graph convolutional network

Xiaoyan Gao [a,b], Fuli Feng [c], Heyan Huang [a,b,*], Xian-Ling Mao [a,b], Tian Lan [a,b], Zewen Chi [a,b]

[a] School of Computer, Beijing Institute of Technology, Beijing 100081, China
[b] Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, Beijing 100081, China
[c] School of Computing, National University of Singapore, 117417, Singapore

A R T I C L E I N F O

Article history:
Received 2 March 2021
Received in revised form 9 October 2021
Accepted 13 October 2021
Available online 19 October 2021

Keywords:
Food recommendation
Food-related relations
Graph convolutional network
Information propagation
High-order connectivity

A B S T R A C T

Food recommendation has attracted increasing attentions to various food-related applications and services. The food recommender models aim to match users' preferences with recipes, where the key lies in the representation learning of users and recipes. However, ranging from early content-based filtering and collaborative filtering methods to recent hybrid methods, the existing work overlooks the various food-related relations, especially the ingredient-ingredient relations, leading to incomprehensive representations. To bridge this gap, we propose a novel model *Food recommendation with Graph Convolutional Network* (FGCN), which exploits ingredient-ingredient, ingredient-recipe, and recipe-user relations deeply. FGCN employs the information propagation mechanism and adopts multiple embedding propagation layers to model high-order connectivity across different food-related relations and enhance the representations. Specifically, we develop three types of information propagation: (1) ingredient-ingredient information propagation, (2) ingredient-recipe information propagation, and (3) recipe-user information propagation. To validate the effectiveness and rationality of FGCN, we conduct extensive experiments on a real-world dataset. The results show that the proposed FGCN outperforms the state-of-the-art baselines. Further in-depth analyses reveal that FGCN could alleviate the sparsity issue in food recommendation.

1. Introduction

Food recommender systems play important roles in a wide spectral of lifestyle applications such as Koubei[1] and NYT Cooking[2]. This is because food choices play an important role in users' daily life, *e.g.,* affecting users' healthy [6,9]. The target of a food recommender system is to predict users' preference over recipes, which can be served by restaurants or cooked by the user depending on the business of the application. The task of building a food recommender system is typically formulated as a machine learning task [28,32], *i.e.,* learning users' preferences from historical interactions on the recipes. Aiming to properly match the user with a recipe, the key lies in the representation learning of user and recipe. In this work, we explore the central theme of learning comprehensive user and recipe representations to enhance food recommendation.

* Corresponding author at: School of Computer, Beijing Institute of Technology, Beijing 100081, China.
E-mail address: hhy63@bit.edu.cn (H. Huang).

[1] https://www.koubei.com/.
[2] https://cooking.nytimes.com/.

https://doi.org/10.1016/j.ins.2021.10.040

Existing food recommender models are mainly in three categories that emphasize different types of information for representation learning:

- *Collaborative filtering.* These methods [28,32] learn user and recipe representations from historical interactions through the general collaborative filtering models. These methods are easy to implement, but ignore the domain knowledge in food recommendation. They thus lead to coarse-grained recipe representation, degrading food recommendation performance.
- *Content-based filtering.* These methods explore food contents (*e.g.,* ingredients and photos) to construct recipe representation [23,27,33]. They largely ignore user interests, failing to please the user.
- *Hybrid methods.* Recent work [14] incorporates food contents into the collaborative filtering model, justifying the benefit of considering richer food information.

In this light, we further consider the various food-related relations, especially the ingredient-ingredient relations. Fig. 1 provides an intuitive example where the user, recipe, and ingredient are represented as an orange circle, gray rectangle, and green triangle, respectively. The dotted line represents the relation connecting two ingredients, *e.g.,* having the same cooking method. Due to such connections, recipes i_2 and i_4 are indeed similar to each other, while their food content features lack overlap. As such, ignoring the ingredient-ingredient relation will fail to recommend i_4 when serving the user u_2, pushing food recommendation to account for such relations. However, the target is non-trivial to achieve since the complex interactions of ingredient-ingredient, ingredient-recipe, and recipe-user require a careful model design.

Towards this end, we propose *Food Graph Convolutional Network* (FGCN), which represents the rich interactions as a graph and distill useful signals from the local structure among nodes (*i.e.,* users, recipes, and ingredients). To enhance the representation learning, FGCN performs information propagation over the graph with multiple graph convolution layers, which model the complex and high-order connectivity. In this way, FGCN is able to exploit ingredient-ingredient, ingredient-recipe, recipe-user semantics deeply. More specially, we design three kinds of information propagation: (1) ingredient-ingredient information propagation, (2) ingredient-recipe information propagation, and (3) recipe-user information propagation. Extensive experiments on the real-world dataset validate the effectiveness of the proposed FGCN, which outperforms the state-of-the-art method [35] by 5.4%. In summary, this work makes the following main contributions:

- To the best of our knowledge, this is the first work to consider various food-related relations in food recommendation.
- We develop a novel model FGCN, which learns comprehensive user and recipe representations by exploiting the graph of ingredient-ingredient, ingredient-recipe, and recipe-user relations.
- We conduct extensive experiments on the real-world food dataset, where the proposed FGCN achieves state-of-the-art performance.

In the following, Section 2 describes the task formulation, Section 3 introduces the proposed framework, then Section 4 presents experiment settings and analyzes the results. Finally, Section 5 reviews related work, followed by the conclusions in Section 6.

2. Task Formulation

Let $\mathcal{G}$ denote a heterogeneous graph with three types of nodes to represent users, recipes, and ingredients. The connections within $\mathcal{G}$ can be seen as three subgraphs: (1) the user-recipe bipartite graph, which encodes the user-recipe interactions; (2) recipe-ingredient bipartite graph, which represents the relations between recipes and ingredients; and (3) ingredient graph, which encodes the relations among ingredients. In particular,

- **User-Recipe Bipartite Graph:** It includes the historical user-recipe interactions, which is formulated as:
$$\mathcal{G}_1 = \{(u, y_{ui}, i) | u \in \mathcal{U}, i \in \mathcal{I}\}, \tag{1}$$
where $\mathcal{U}$ and $\mathcal{I}$ denote the user and item set, respectively. $y_{ui} = 1$ denotes whether a user u has interacted (*e.g.,* consumed or rated) a recipe $i, y_{ui} = 1$; otherwise, $y_{ui} = 0$.
- **Recipe-Ingredient Bipartite Graph:** The containment between ingredients and recipes is represented as:
$$\mathcal{G}_2 = \{(i, g_i^k, k) | i \in \mathcal{I}, k \in \mathcal{K}\}, \tag{2}$$
where k denotes the ingredient ID within the ingredient set $\mathcal{K}$. $\mathbf{g}_i \in \mathbb{R}^{|\mathcal{K}|}$ is a multi-hot encoding for recipe i where $g_i^k = 1$ means recipe i contains the ingredient k; otherwise, $g_i^k = 0$.
- **Ingredient Graph:** Ingredients sharing the same attributes[3] are also connected. Formally,
$$\mathcal{G}_3 = \{(k, a, k\prime) | k \in \mathcal{K}, a \in \mathcal{A}, k\prime \in \mathcal{K}\}, \tag{3}$$

[3] In this work, we consider three kinds of attributes: ingredient color, ingredient shape, and cooking methods.

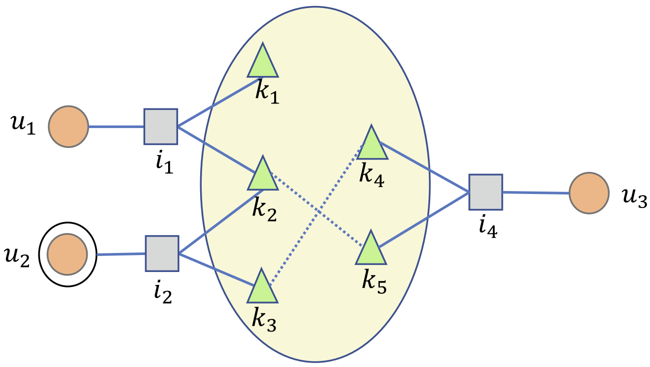


Fig. 1. An illustration of food-related relations, where node u_2 represents the target user to serve.

where a is an attribute from the attribute set $\mathcal{A}$. Details of the above attributes are introduced in Section 3.2.1.

The objective of food recommendation is to learn an interaction function from the heterogeneous graph: $\hat{y}_{ui} = s(\mathcal{G}, \Theta)$, which predicts the probability that user u would interact with recipe i. Θ represents the parameters of $s(\cdot)$. Specially, we formulate the food recommendation task as:

Input: A heterogeneous graph $\mathcal{G}$ which consists of the user-recipe bipartite graph $\mathcal{G}_1$, recipe-ingredient bipartite graph $\mathcal{G}_2$, and ingredient graph $\mathcal{G}_3$.

Output: An interaction function $\hat{y}_{ui} = s(\mathcal{G}, \Theta)$, which outputs the probability that user u will interact recipe i.

3. Methodology

As shown in Fig. 2, the framework consists of three key components: (1) embedding layer, which includes user embeddings, recipe embeddings, and ingredient embeddings; (2) information propagation, which purifies the representations by exploiting ingredient-ingredient, ingredient-recipe, and user-recipe relations; (3) prediction layer, which aggregates the purified representations and predicts the score for each user-recipe pair. In the following, we elaborate on the components one by one.

3.1. Embedding Layer

User and Recipe Embedding. As embedding-based models have made great success in exploiting collaborative signals between user and item [26,36,44,45], we parameterize each user or recipe with an embedding vector, which is formulated as:

$$\boldsymbol{p}_u = \boldsymbol{P}\boldsymbol{e}_u, \quad \boldsymbol{q}_i = \boldsymbol{Q}\boldsymbol{e}_i, \tag{4}$$

where $\boldsymbol{p}_u$ is the embedding of user u, and $\boldsymbol{q}_i$ is the embedding of recipe i. $\boldsymbol{P} \in \mathbb{R}^{D\times|\mathcal{U}|}$ and $\boldsymbol{Q} \in \mathbb{R}^{D\times|\mathcal{I}|}$ are the matrix format, where D is the embedding size, $|\mathcal{U}|$ and $|\mathcal{I}|$ represent the number of users and recipes, respectively. Note that $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ are one-hot vectors.

Ingredient Embedding. Ingredients are the essential components of each recipe, which largely affect user preference [12]. We thus project ingredients into the same embedding space of users and recipes, *i.e.,* each ingredient k corresponds to an embedding vector $\boldsymbol{x}_k$. In this way, we represent the ingredient list of recipe i (*i.e.,* $\mathbf{g}_i$) as a set of embedding vectors $\mathcal{X}_i$. Formally,

$$\mathcal{X}_i = \{\boldsymbol{x}_k | g_i^k = 1\}. \tag{5}$$

3.2. Information Propagation

There are three kinds of information propagation: (1) ingredient-ingredient information propagation, which purifies ingredient representations by ingredient graph; (2) ingredient-recipe information propagation, which aggregates ingredient semantics into the recipe representations; (3) recipe-user information propagation, which refines the users' representations through historical recipes. In the following, we introduce these three components step by step.

3.2.1. Ingredient-Ingredient Information Propagation

Considering that the users will consume ingredients with similar attributes [5], we summarize such ingredient-ingredient relations into a graph. Specifically, we consider three types of relations between ingredients: ingredient color, ingredient shape, and cooking method.

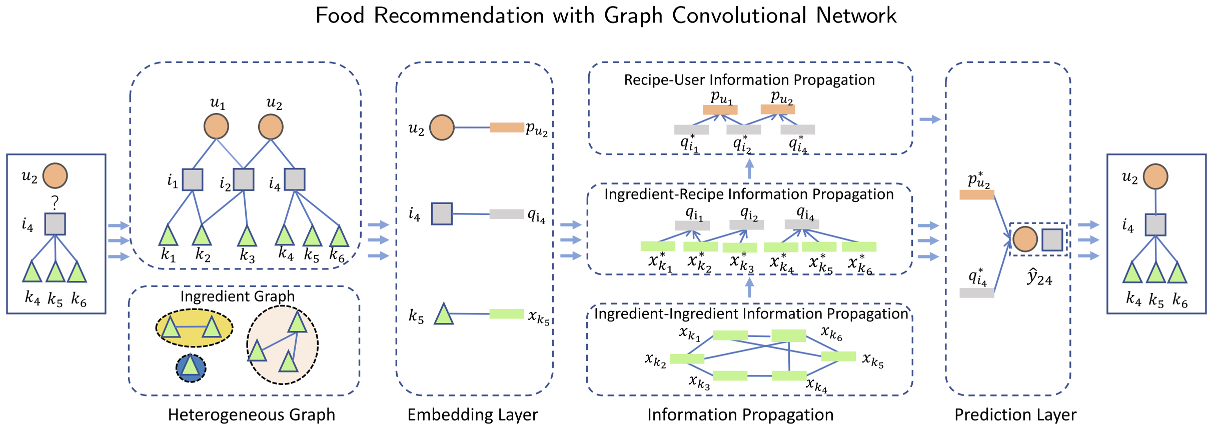


Fig. 2. Illustration of the proposed FGCN.

- **Ingredient Color.** We link the ingredients that share the same color such as white, black, red, green, and yellow. This is because color affects the appearance of recipes and reveals nutritional information. For example, green ingredients are beneficial for users' stomachs as they can maintain acid-base balance. The users would like to choose some recipes since they seem tasty or are helpful for health.
- **Ingredient Shape.** The shape of the ingredients often affect the taste and absorption of the ingredient, especially for users with oral and gastrointestinal diseases. We thus construct the link between ingredients with the same shape. In particular, we consider six shape attributes: slice, dice, minced, powder, roll, and shred.
- **Cooking Method.** Moreover, we take six cooking methods into account, which are deep-fry, dry, fry, steam, boil, and pickle. The motivation is that different cooking methods would have a great impact on user preferences over food. For instance, elderly people would like to eat stem and boil chickens, youngsters would choose fry or even deep-fry chickens instead.

Our key consideration is that the connectivity of sharing the same attribute should be encoded into ingredient representations. Moreover, ingredients sharing different attributes can contribute different information. In this light, the ingredient-ingredient propagation handles each attribute in $\mathcal{A}$ separately. Formally,

$$\boldsymbol{m}^{a}_{k\leftarrow k'} = \frac{1}{|\mathcal{N}^{a}_{k}|}\sum_{k'\in\mathcal{N}^{a}_{k}} \boldsymbol{W}_1 \boldsymbol{x}^{l-1}_{k'}, \tag{6}$$

where $\boldsymbol{m}^{a}_{k\leftarrow k'}$ denotes the message passed to ingredient node k from its neighbor ingredients connected with attribute a; $\mathcal{N}^{a}_{k}$ represents neighbor ingredients in attribute a for the ingredient node k; and $\boldsymbol{W}_1$ is the weight matrix. As there are three kinds of connections in the ingredient graph $\mathcal{G}_3$, we consider two ways to aggregate information across attributes, which are formulated as:

$$\text{sum}: \boldsymbol{x}^{l}_{k} = \sum_{a\in\mathcal{A}} \boldsymbol{m}^{a}_{k\leftarrow k'}; \text{concate}: \boldsymbol{x}^{l}_{k} = \text{CON}(\{\boldsymbol{m}^{a}_{k\leftarrow k'} | a\in\mathcal{A}\}), \tag{7}$$

where l denote the iteration of the propagation. Note that one iteration corresponds to a graph convolutional layer. In order to model high-order connectivity information, we employ L layers, and aggregate their outputs to get the final ingredient representation:

$$\boldsymbol{x}^{*}_{k} = CON\left(\{\boldsymbol{x}^{l}_{k} | k\in[0, L]\}\right). \tag{8}$$

3.2.2. Ingredient-Recipe Information Propagation

A recipe contains multiple ingredients. For example, recipe *steam egg custard* consists of three ingredients: egg, minced pork, and chopped green onion. Considering that the ingredient list plays a key role in describing recipe [4], we exploit ingredient information to enhance recipe representation.

Ingredient-Wise Information Construction. For a recipe, we refine its representation with the linked ingredients in $\mathcal{G}_2$. Formally, the recipe representation $\boldsymbol{q}_i$ receive information from its ingredient neighbors $\mathcal{N}_q$,

$$\boldsymbol{m}_{x\rightarrow q} = \frac{1}{|\mathcal{N}_q|}\sum_{x\in\mathcal{N}_q} \boldsymbol{x}^{*}, \tag{9}$$

where $\boldsymbol{m}_{x\rightarrow q}$ is the message propagation from neighbors; $\frac{1}{|\mathcal{N}_q|}$ is the normalization term to deal with different neighbor numbers and accelerate the training.

Recipe-Wise Information Aggregation. To obtain the final recipe representation, we aggregate the information propagated from neighbor ingredients and ego representation. Specifically, we formulate the aggregation function as,

$$\boldsymbol{q}_i^* = F(\boldsymbol{q}_i, \boldsymbol{m}_{x\rightarrow q}), \tag{10}$$

where $\boldsymbol{q}_i^*$ denotes the refined recipe representation, which is affected by recipe ID embedding $\boldsymbol{q}_i$ and information propagated from neighbor ingredients. Intuitively, we can adopt the *sum* and *concate* function as the aggregation function $F(.)$. Considering that $\boldsymbol{q}_i$ and $\boldsymbol{m}_{x\rightarrow q}$ are heterogeneous, we adopt more advanced implementations:

- *GCN Aggregation* [18] adopts a nonlinear function after adding these two representations directly. The aggregation is formulated as,

$$F_{gcn}(\boldsymbol{q}_i, \boldsymbol{m}_{x\rightarrow q}) = f(\boldsymbol{W}_2(\boldsymbol{q}_i + \boldsymbol{m}_{x\rightarrow q})), \tag{11}$$

  where $\boldsymbol{W}_2$ is the paramters to be learned, $f(.)$ is a non-linear activation function such as LeaklyReLU.
- *Bi-Interaction Aggregation* [8,35] employs a nonlinear function after two operation: sum and element-wise between recipe ID embedding $\boldsymbol{q}_i$ and neighbor ingredients information $\boldsymbol{m}_{x\rightarrow q}$. The aggregation is formulated as,

$$F_{bi}(\boldsymbol{q}_i, \boldsymbol{m}_{x\rightarrow q}) = f(\boldsymbol{W}_3(\boldsymbol{q}_i + \boldsymbol{m}_{x\rightarrow q}) + f(\boldsymbol{W}_4(\boldsymbol{q}_i \odot \boldsymbol{m}_{x\rightarrow q})), \tag{12}$$

  where $\boldsymbol{W}_3, \boldsymbol{W}_4$ are the parameters.

3.2.3. Recipe-User Information Propagation

Recent studies have shown that user-recipe collaborative signals could help infer user's preference over recipes [14,32]. As such, we enrich user representations via integrating historical recipes according to the structure of $\mathcal{G}_1$. To purify the target user representation, we obtain information passing from historical recipes. Formally, the user u receive information propagation from recipe neighbors $\mathcal{N}_p$ in the heterogeneous graph,

$$\boldsymbol{m}_{q\rightarrow p} = \frac{1}{|\mathcal{N}_p|}\sum_{q\in\mathcal{N}_p}\boldsymbol{q}_i^*, \tag{13}$$

where $\boldsymbol{m}_{q\rightarrow p}$ is the message propagation from neighbors; $\frac{1}{|\mathcal{N}_p|}$ is a normalization term.

Similar to recipe-wise information aggregation, we update user representation through aggregating neighbor recipes and ego representation. The aggregation function is formulated as,

$$\boldsymbol{p}_u^* = F(\boldsymbol{p}_u, \boldsymbol{m}_{q\rightarrow p}), \tag{14}$$

where $\boldsymbol{p}_u^*$ denotes the refined target user representation, which is impacted by user ID embedding $\boldsymbol{p}_u$ and information propagated from neighbor recipes; $F(.)$ denotes the aggregation function, similar with recipe-wise information aggregation, which could be implemented as:

$$F_{gcn}(\boldsymbol{p}_u, \boldsymbol{m}_{q\rightarrow p}) = f(\boldsymbol{W}_5(\boldsymbol{p}_u + \boldsymbol{m}_{q\rightarrow p})), \tag{15}$$

$$F_{bi}(\boldsymbol{p}_u, \boldsymbol{m}_{q\rightarrow p}) = f(\boldsymbol{W}_6(\boldsymbol{p}_u + \boldsymbol{m}_{q\rightarrow p})) + f(\boldsymbol{W}_7(\boldsymbol{p}_u \odot \boldsymbol{m}_{q\rightarrow p})), \tag{16}$$

where $\boldsymbol{W}_5, \boldsymbol{W}_6$, and $\boldsymbol{W}_7$ are the parameters.

3.3. Prediction Layer and Objective Function

Prediction Layer. After the above steps, we obtain the refined user representation $\boldsymbol{p}_u^*$ and recipe representation $\boldsymbol{q}_i^*$. To predict how likely user u would consume recipe i, we adopt the inner product on their representations as:

$$\hat{y}_{ui} = \boldsymbol{p}_u^{*T}\boldsymbol{q}_i^*. \tag{17}$$

Objective Function. In order to optimize parameters in FGCN, we employ a pairwise learning approach [41]. From the pairwise learning perspective, the score of an observed interaction should be higher than unobserved ones. Formally, we assign 1 to the observed interaction, and we assign 0 to the unobserved one, which is formulated as:

$$\mathcal{D}_{\mathcal{S}} = \{(u,i,k)|y_{ui} = 1 \wedge y_{uk} = 0\}, \tag{18}$$

where $\mathcal{D}_{\mathcal{S}}$ denotes the training set, (u,i,k) is the triples, in which a user u consumed the recipe i, but has never consumed the recipe k before.

We adopt the widely used Bayesian Personalized Ranking (BPR) loss [19] to optimize model parameters Θ. Formally,

$$\mathcal{L} = \sum_{(u,i,k)\in\mathcal{D}_S} - ln\sigma(\hat{y}_{ui} - \hat{y}_{uk}) + \lambda||\Theta||^2. \tag{19}$$

where σ denotes the sigmoid function, λ represents the regularization term.

4. Experiments

We perform extensive experiments on a large-scale dataset to answer the following research questions:

- **RQ1:** How does FGCN perform as compared with the state-of-the-art food recommendation methods?
- **RQ2:** How do different components (*i.e.,* model depth and aggregator selection) influence the effectiveness of FGCN?
- **RQ3:** What is the key hyper-parameter of FGCN and how does it impact FGCN's performance?
- **RQ4:** How does FGCN perform with respect to different sparsity levels in user groups?

In the following, we first describe the experimental settings, then answer the above research questions.

4.1. Experimental Settings

Dataset Description. We conduct experiments on a large-scale dataset since Allrecipes [14] is the only available public dataset in the food recommendation task. The dataset was collected from Allrecipes.com, which contains a user-recipe bipartite graph and a recipe-ingredient bipartite graph. To evaluate the effectiveness of FGCN, we build the test set by the latest 20% historical interactions, and construct the training set by the remainder data. To tune hyper-parameters, we randomly select 10% of interactions from the training set as validation set. We then obtain 57,828 users, 43,272 recipes with 38,130 ingredients and 681,656 interactions. Note that the users both in the training set and the testing set will be retained.

Besides the user-recipe bipartite graph and recipe-ingredient bipartite graph, we need to generate the ingredient graph. We build links among ingredients that share the same attributes. For example, since attribute "black" occurs both in ingredient "black soya bean" and "black fungus", there will be an edge linking them. Then we could quickly generate the triplet (black soya bean, black, black fungus).

Evaluation Protocol. For the test set, given a user and her/his consumed recipes, we regard the recipes as positive samples. Moreover, we sample some negative instances and pair totally 500 recipes for each user. Note that the negative instances are recipes the user did not consume before. Then the models predict the ranking score for each instance and output the recommendation list. To evaluate the effectiveness of food recommendation, we adopt two widely-used evaluation metrics for recommendation: *Recall* and *NDCG*. For both metrics, we average the values over all users. Note that the range of values is [0,1]. A higher value indicates better recommendation performance.

Baselines. To verify the effectiveness of FGCN, we select the baselines in the following:

- MF [30]: Since the model is popular in collaborative filtering methods, we choose it as a representative method for collaborative filtering. To optimize the model, MF employs the popular BPR as loss function. Note that this model only considers ID embeddings as representations of users and recipes.
- FM [29]: The method adopts second-order factorized interactions between input features for recommendation. Especially, we consider user ID, recipe ID, and ingredients as the input features in this work.
- KGAT [35]: The method adopts graph neural network framework to explicitly model high-order relations. In particular, it constructs a collaborative knowledge graph and employs attention mechanism to discriminate the importance of the neighbors.
- HAFR-non-v [14]: This is the state-of-the-art food recommendation approach, which exploits hierarchical attention mechanism to incorporate user-recipe interactions and recipe ingredients into deep neural network.

Parameter Settings. We implement our FGCN based on TensorFlow. To compare with the baselines fairly, we set the embedding dimension of the models as 64. Besides, we perform a grid search to tune the hyper-parameters: the learning rate is searched in [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05], the coefficient of L_2 is tuned among $[1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}]$. When the coefficient L_2 equals to $1e^{-4}$ and learning rate equals to 0.0001, FGCN performs best. Moreover, we set model depth as three to model the third-order connectivity. In particular, the hidden dimensions are [64, 32, 16].

4.2. Model Comparison (RQ1)

To investigate the effectiveness of the proposed method (FGCN), we study how FGCN performs as compared with the above-mentioned baselines. Table 1 shows the performance comparison. Analyzing the results, we have the following observations:

Table 1
Performance of compared methods.

Methods	Recall@10	NDCG@10
MF	0.0526	0.0311
FM	0.0537	0.0315
HAFR-non-v	0.0565	0.0330
KGAT	0.0580	0.0472
FGCN	0.0611*	0.0498*

- Our proposed FGCN performs the best on both metrics. The result verifies the capability of FGCN to model high-order connectivity and exploiting ingredient-ingredient, ingredient-recipe, recipe-user relations deeply, which benefit from information propagation mechanism and multiple embedding propagation layers.
- Compared with KGAT - the representative graph-based recommendation, FGCN exhibits an average improvement of 5.4%, which justifies the importance of exploiting relations among ingredients. Note that the input setting is the same for KGAT and FGCN except for ingredient relations.
- Specifically, compared with HAFR-non-v, the state-of-the-art food recommendation method, FGCN exhibits an average improvement of 29%. We attribute the remarkable improvement to the expressive power of graph convolutional network in modeling food-related relations.
- KGAT achieves the best performance among the baselines, which demonstrates that graph structure is beneficial for enhancing the representation learning in food recommendation. HAFR-non-v performs weaker since it builds a hierarchical architecture merely. Lastly, FM outperforms MF since FM benefits from seconder-order factorized interactions between features.

To evaluate the ranking performance of FGCN, we show the performance of Top-N food recommendation in Fig. 3, in which the ranking position varies from 5 to 20. Clearly, FGCN consistently outperforms other baselines on both Recall and NDCG metrics. This verifies the effectiveness and robustness of FGCN to enhance the representation learning.

4.3. Study of FGCN(RQ2)

To get deep insights into the architecture of FGCN, we analyze its impact from different perspectives. Firstly, we investigate how different layer numbers affect the model performance. Secondly, we examine how aggregators in ingredient graph influence performance. Lastly, we study how performances are affected by different aggregators in ego representations and neighbor representations.

4.3.1. Effect of Model Depth

We then examine whether FGCN could benefit from multiple embedding propagation layers. Specially, we vary the embedding propagation layer number L in [1, 2, 3, 4, 5]. Table 2 shows the performance of FGCN in different layer numbers. For example, FGCN-1 indicates FGCN with only one embedding propagation layer. From Table 2, we observe that:

- FGCN performance could be enhanced substantially by enlarging the model depth. Specially, both FGCN-2 and FGCN-3 consistently outperform FGCN-1 on both metrics. Benefit from second-order and third-order connectivities, FGCN is capable of modeling high-order food-related relations among nodes (*i.e.,* ingredients, recipes, users), thereby achieve better performance.
- Further stacking too many layers in FGCN, we find that the performance drops quickly, like FGCN-4 and FGCN-5. The results indicate that the representation learning maybe hurt by noises from the too deep model, which easily causes overfitting and degrades the performance.

4.3.2. Effect of Aggregators in Ingredient Graph

To study the effect of aggregators in ingredient graph, we conduct experiments in different variants of FGCN. Table 3 shows the FGCN performance with different aggregators in ingredient graph. Specially, sum (*cf.* Section 3.2.1) and and concat (*cf.* Section 3.2.1) are termed FGCN_{sum} and FGCN_{con}, respectively. From Table 3, we observe that:

- FGCN_{sum} consistently achieves better performance than FGCN_{con}. We attribute the improvement to better incorporating the characteristics of individual connection with sum operation.
- Jointly analyzing Table 1 and Table 3, we observe that, even with concate aggregator, FGCN_{con} is superior to the baselines on both metrics. The result verifies the robustness and effectiveness of the ingredient graph in modeling connections among ingredients.

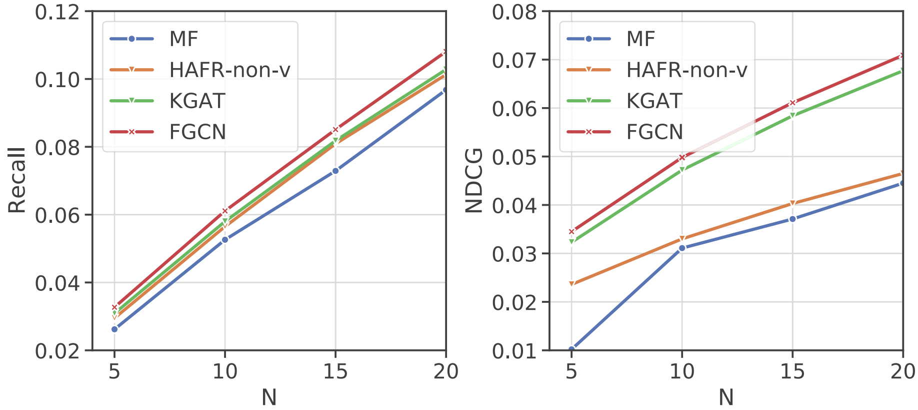


Fig. 3. Performance of Top-N food recommendation.

Table 2
Effect of embedding propagation layer numbers.

	Recall@10	NDCG@10
FGCN-1	0.0556	0.0445
FGCN-2	0.0607	0.0500*
FGCN-3	0.0611*	0.0498
FGCN-4	0.0588	0.0478
FGCN-5	0.0498	0.0403

Table 3
Effect of aggregators in ingredient graph.

Aggregator	Recall@10	NDCG@10
$FGCN_{con}$	0.0600	0.0491
$FGCN_{sum}$	0.0611*	0.0498*

4.3.3. Effect of Aggregators for Ego Representations and Neighbor Representations

To investigate how different aggregators for ego representations and neighbor representations would affect the performance of FGCN, we perform experiments for two variants of FGCN: $FGCN_{GCN}$ and $FGCN_{Bi}$. Table 4 summarizes performances for these two variants of FGCN. Note that $FGCN_{GCN}$ (*cf.* Section 3.2.2) and and $FGCN_{Bi}$ (*cf.* Section 3.2.2) are two variants of FGCN with GCN and Bi-Interaction, respectively.

From Table 4, we observe that $FGCN_{Bi}$ beats $FGCN_{GCN}$ on both metrics clearly. We attribute the improvement to feature interactions, which models affinities between ego representation and neighbor representations. Besides, the result justifies the effectiveness and rationality of the Bi-interaction aggregator to capture heterogeneity of these two representations.

4.4. Hyper-parameter Studies (RQ3)

Hyper-parameters play a vital role in affecting the recommendation [20,21]. Thereafter, we investigate how the performance of FGCN and HAFR-non-v are impacted by hyper-parameters. In particular, we examine the influence of the regularization term λ. Fig. 4 shows the performance of FGCN as adjusting the strength of regularization, from which, we observe that:

- Compared with HAFR-non-v, FGCN is consistently superior to it in various values of λ. HAFR-non-v leverage hierarchical structure for learning nodes (*i.e.,* ingredient, recipes, and users) representations. In contrast, FGCN beats it by graph structure, which benefits from the expressive power of graph convolutional network again.
- When the value of λ is larger than $1e^{-3}$, the performance of FGCN and HAFR-non-v drops quickly. The result illustrates that setting proper regularization is important for neural network-based recommendation, which is consistent with prior efforts like [14].

Table 4
Effect of aggregators for ego representations and neighbor representations.

Aggregator	Recall@10	NDCG@10
$FGCN_{GCN}$	0.0578	0.0465
$FGCN_{Bi}$	0.0611*	0.0498*

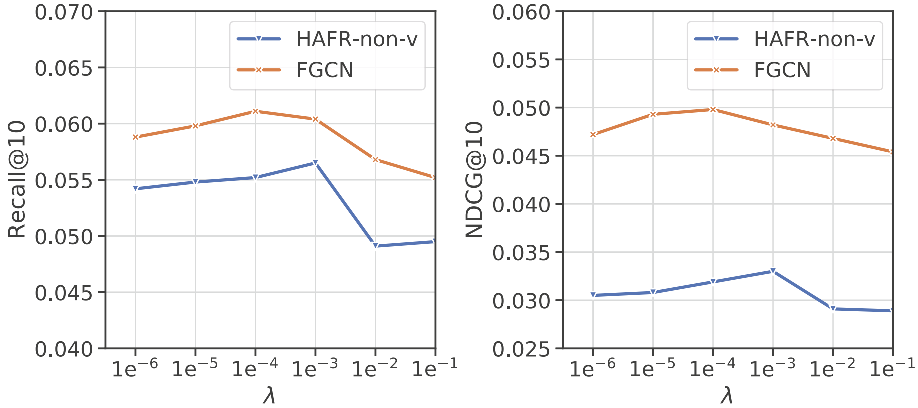


Fig. 4. Performance of FGCN regarding Recall@10, NDCG@10 as adjusting the strength of regularization.

4.5. Performance over Different Sparsity Levels of User Group(RQ4)

To investigate whether FGCN could alleviate the sparsity issue in food recommendation, we compare FGCN with KGAT over different sparsity levels of user groups. Specially, we split the test set into four groups with respect to the interaction number in each user. Note that we keep the equal interaction size in these four groups. Fig. 5 displays the model comparison. From Fig. 5, we can observe that:

- The proposed FGCN outperforms KGAT consistently in all the user groups. This justifies the capability of FGCN to alleviate the sparsity issue since FGCN could leverage information across different kinds of connections with the ingredient graph.
- Clearly, with the interaction number per user becomes large, the performance of both models increases. This illustrates that there exists richer information in denser user-recipe interactions. As a result, the model could easily capture users' preferences.

5. Related Work

We firstly review graph-based recommendation, then introduce food recommendation methods in this section.

5.1. Graph-based Recommendation

Graph-based recommendation aims to infer user preference by exploiting user-item interaction graph. ItemRank [15] and BiRank [17] are early proposed in the research line. These two models employ label propagation mechanism to fed user preferences into graphs. In particular, they define items the users interacted with as labels, then propagate the labels. As such, the model could predict item scores for the target users. However, these methods simply use interactions directly without model parameters, which is detrimental for optimization and easily hurts the recommendation performance.

Recently, with rapid development of Graph neural networks (GNNs) and their applications on relational data [1,13,22,37,38,40,42,47], graph-based recommendation becomes popular. GNNs construct a set of elements and their relations in form of graph. Due to its expressive capability of representations, great efforts have been paid by both academia and industry.

Prior studies on GNNs-based recommendation like GC-MC [2], which develop a graph-based auto-encoder framework to construct the user-item graph. Specifically, by employing message passing, the model adopts auto-encoder to generate representations for a user and an item. Then the generated user and item representations are utilized to reconstruct rating links via a bilinear decoder. However, the method could not be scalable to web-scale scenarios. Towards this end, PinSage [43] is proposed, which relies on efficient random walks to re-organize graph convolutions and produces node representations. To encode collaborative signals in the embedding process, Wang et al. propose NGCF [36], which stacks multiple layers to perform embedding propagation, which exploits the collaborative signals under graph neural network. However, NGCF directly applies graph neural network to perform embedding propagation, which leads to complex implementions and time-

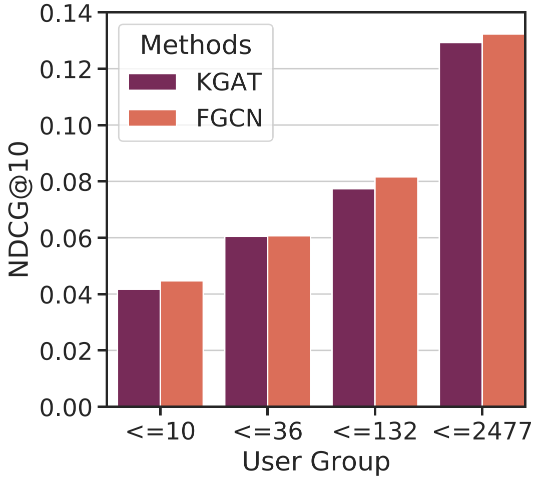


Fig. 5. Performance over the sparsity distribution of user groups.

consuming training. In order to make a simpler but efficient model, He et al. develops LightGCN [16], which removes two designs from NGCF: nonlinear activation and feature transformation. Although LightGCN is efficient, the approach merely adopts user-item interactions, which overlooks the attributes among items. Recently, Wang et al. [35] builds KGAT, which leverages graph convolutions via modeling item-attribute and user-item interactions. Despite the success of KGAT, the model ignores exploiting relations among attributes, which leads to suboptimal item representation.

In our work, we build a heterogeneous graph, which consists of three types of nodes: ingredients, recipes, and users. The proposed model benefit from information propagation over the built graph since graph convolutional network is capable of learning ingredient, recipe, and user representations through exploiting food-related relations.

5.2. Food Recommendation

Since food recommendation is a key component in facilitating everyday life, it receives lots of attentions recently [24]. One paradigm of food recommendation is collaborative filtering. Collaborative filtering methods aim to learn user and recipe representations from historical interactions with the general collaborative filtering models. Trattner et al. [32] tries popular collaborative filtering models. After testing, LDA, which beats all other ones, considers recipes as words and users as documents. Pecunne et al. [28] also conducts experiments in three collaborative algorithms: ALS, BPR, and LMF. Clearly, these methods belong to general collaborative filtering models, which are easy to implement. Nevertheless, they overlook domain knowledge in food recommendation. Due to the lack of rich yet complicated food domain knowledge, it easily leads to coarse-grained recipe representation hence degrades food recommendation performance.

Another important research line of food recommendation is content-based filtering. Content-based filtering mines food contents merely to construct recipe representation. Content-based filtering can be divided into three methods: ingredient-based, photo-based, and profile-based. (1) The first approaches infer user preferences from recipe ingredients to make recommendations [12,27,31]. Freyne et al. [12] explores recipes-ingredient relationships by two phases: break down and reconstruction. Break down phase transfers ratings from recipe to ingredients, instead, reconstruction transfers ratings from ingredients to recipes. By simply summing ratings of the associated ingredients in the recipe, the method recommends recipes to the target user. To predict scores for the target user, Teng et al. [31] feed ingredient features in machine learning classifier. (2) The second approaches demonstrate that recipe photos reveal important information of the recipes [9,23,39], which would be beneficial for recommendation. 3) Except for the above food contents, some efforts focus on exploring user profiles for food recommendation [33]. They largely ignore user interests, failing to please the user.

Recently, researchers realize that food contents should not be ignored in collaborative filtering based framework. Towards this end, Gao et al. [14] develops hierarchical attention mechanism to simultaneously incorporates collaborative signals and food contents for food recommendation. However, the method simply adopts attention mechanism to incorporate ingredients as recipe representation, which overlooks the various food-related relations. Lastly, Meng et al. [23] proposes PiNet for food recommendation and ingredient prediction simultaneously. Even though PiNet could work well via the multi-task learning perspective, it still ignores the relations among ingredients. Moreover, the method could not handle the complex interactions of ingredient-ingredient, ingredient-recipe, and recipe-user. Therefore, this hurts food recommendation performance easily.

Compared with the above efforts, the proposed FGCN has the following advantages: (1) compared with collaborative filtering methods, FGCN employs domain knowledge (*i.e.*, food-related relations) to learn fine-grained recipe representation, thus enhances the recommendation; (2) compared with content-based approaches, FGCN focuses on personalized food recommendation task, hence could capture users' preferences easily; (3) compared with hybrid methods, FGCN considers various food-related relations (*i.e.,* ingredient-ingredient, ingredient-recipe, and recipe-user) then elaborates them in a heterogeneous graph to enhance the representation learning.

6. Conclusion

This work argues that there exist complex relationships among ingredient-ingredient, ingredient-recipe, and recipe-user interactions, which are important for food recommendation. In this light, we propose Food recommendation with Graph Convolutional Network (FGCN), which employs the information propagation mechanism and adopts multiple embedding propagation layers to model high-order connectivity and enhance the representation learning. Extensive experiments conducted on a large-scale dataset demonstrate the rationality and effectiveness of FGCN to model food-related relations among ingredients, recipes, and users in a heterogeneous graph. Besides, the in-depth analyses of FGCN demonstrate the usefulness and necessity of model components.

This work focuses on the accuracy of recommendation, ignores the other important evaluation for recommendation such as diversity [3,7], which might be suboptimal *w.r.t.* long-term user experience. Moreover, the work only explores internal knowledge among ingredients, recipes, and users, but ignores external knowledge such as the ingredient-disease relation, which might cause unhealthy recommendation.

In future, we will explore the following directions: (1) the diversity for food recommendation; (2) more advanced information propagation mechanisms [10,46]; and (3) the causation beyond the simple connection [11,34]. Besides, to construct more comprehensive food knowledge graph, we will explore the extraction of food-oriented relations from other sources such as relevant documents [25].

CRediT authorship contribution statement

Xiaoyan Gao: Conceptualization, Methodology, Software, Writing - original draft. **Fuli Feng:** Methodology, Supervision, Writing - review & editing. **Heyan Huang:** Supervision, Writing - review & editing. **Xian-Ling Mao:** Supervision, Writing - review & editing. **Tian Lan:** Software. **Zewen Chi:** Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

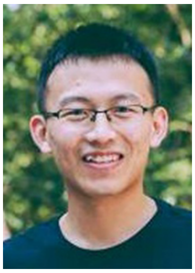
This work is supported by the National Natural Science Foundation of China [Grant Nos. No. U19B2020, 61772076, 61751201].

References

[1] Bachmann, G., Bécigneul, G., Ganea, O., 2020. Constant curvature graph convolutional networks, in: International Conference on Machine Learning, PMLR. pp. 486–496.
[2] Berg, R.v.d., Kipf, T.N., Welling, M., 2018. Graph convolutional matrix completion, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1–8.
[3] X. Cai, Z. Hu, J. Chen, A many-objective optimization recommendation algorithm based on knowledge mining, Information Sciences 537 (2020) 148–161.
[4] D. Cao, N. Han, H. Chen, X. Wei, X. He, Video-based recipe retrieval, Information Sciences 514 (2020) 302–318.
[5] J. Chen, L. Pan, Z. Wei, X. Wang, C.W. Ngo, T.S. Chua, Zero-shot ingredient recognition by multi-relational graph convolutional network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 10542–10550.
[6] M. Chen, X. Jia, E. Gorbonos, C.T. Hoang, X. Yu, Y. Liu, Eating healthier: Exploring nutrition information for healthier recipe recommendation, Information Processing & Management 57 (2020) 102051.
[7] W. Chen, P. Ren, F. Cai, F. Sun, M. De Rijke, Multi-interest diversification for end-to-end sequential recommendation, ACM Transactions on Information Systems (TOIS) 40 (2021) 1–30.
[8] Y. Chen, Y. Wang, P. Ren, M. Wang, M. de Rijke, Bayesian feature interaction selection for factorization machines, Artificial Intelligence 103589 (2021).
[9] D. Elsweiler, C. Trattner, M. Harvey, Exploiting food choice biases for healthier recipe recommendation, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 575–584.
[10] F. Feng, X. He, H. Zhang, T.S. Chua, Cross-gcn: Enhancing graph convolutional network with k-order feature interactions, IEEE Transactions on Knowledge and Data Engineering. (2021).
[11] F. Feng, W. Huang, X. He, X. Xin, Q. Wang, T.S. Chua, Should graph convolution trust neighbors? a simple causal inference method, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1208–1218.
[12] Freyne, J., Berkovsky, S., 2010. Intelligent food planning: personalized recipe recommendation, in: Proceedings of the 15th international conference on Intelligent user interfaces, pp. 321–324.
[13] S. Fu, W. Liu, D. Tao, Y. Zhou, L. Nie, Hesgcn: Hessian graph convolutional networks for semi-supervised classification, Information Sciences 514 (2020) 484–498.
[14] X. Gao, F. Feng, X. He, H. Huang, X. Guan, C. Feng, Z. Ming, T. Chua, Hierarchical attention network for visually-aware food recommendation, IEEE Trans. Multimedia 22 (2020) 1647–1659.
[15] Gori, M., Pucci, A., Roma, V., Siena, I., 2007. Itemrank: A random-walk based scoring algorithm for recommender engines., in: IJCAI, pp. 2766–2771.

[16] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.
[17] X. He, M. Gao, M.Y. Kan, D. Wang, Birank: Towards ranking on bipartite graphs, IEEE Transactions on Knowledge and Data Engineering 29 (2016) 57–71.
[18] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: ICLR.
[19] Y.C. Lee, T. Kim, J. Choi, X. He, S.W. Kim, M-bpr: A novel approach to improving bpr for recommendation with multi-type pair-wise preferences, Information Sciences 547 (2021) 255–270.
[20] Li, X., Jiang, W., Chen, W., Wu, J., Wang, G., Li, K., 2020. Directional and explainable serendipity recommendation, in: Proceedings of The Web Conference 2020, pp. 122–132.
[21] Lin, Y., Moosaei, M., Yang, H., 2020. Outfitnet: Fashion outfit recommendation with attention-based multiple instance learning, in: Proceedings of The Web Conference 2020, pp. 77–87.
[22] Z. Liu, P. Qian, X. Wang, Y. Zhuang, L. Qiu, X. Wang, Combining graph neural networks with expert knowledge for smart contract vulnerability detection, IEEE Transactions on Knowledge and Data Engineering (TKDE) (2021) 1, https://doi.org/10.1109/TKDE.2021.3095196.
[23] L. Meng, F. Feng, X. He, X. Gao, T.S. Chua, Heterogeneous fusion of semantic and collaborative information for visually-aware food recommendation, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 3460–3468.
[24] W. Min, S. Jiang, R. Jain, Food recommendation: Framework, existing solutions, and challenges, IEEE Transactions on Multimedia 22 (2019) 2659–2671.
[25] G. Nan, Z. Guo, I. Sekulić, W. Lu, Reasoning with latent structure refinement for document-level relation extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1546–1557.
[26] J. Ni, Z. Huang, J. Cheng, S. Gao, An effective recommendation model based on deep representation learning, Information Sciences 542 (2021) 324–342.
[27] N. Nilesh, M. Kumari, P. Hazarika, V. Raman, Recommendation of indian cuisine recipes based on ingredients, in: 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), 2019, pp. 96–99.
[28] Pecune, F., Callebert, L., Marsella, S., 2020. A recommender system for healthy and personalized recipes recommendations., in: HealthRecSys@ RecSys, pp. 15–20.
[29] S. Rendle, Factorization machines with libfm, ACM Transactions on Intelligent Systems and Technology (TIST) 3 (2012) 57.
[30] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L., 2009. Bpr: Bayesian personalized ranking from implicit feedback, in: UAI, pp. 452–461.
[31] Teng, C.Y., Lin, Y.R., Adamic, L.A., 2012. Recipe recommendation using ingredient networks, in: WebSci, pp. 298–307.
[32] Trattner, C., Elsweiler, D., 2017. Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems, in: WWW, pp. 489–498.
[33] W. Wang, L.Y. Duan, H. Jiang, P. Jing, X. Song, L. Nie, Market2dish: Health-aware food recommendation, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 17 (2021) 1–19.
[34] W. Wang, F. Feng, X. He, H. Zhang, T.S. Chua, click is not equal to like: Counterfactual recommendation for mitigating clickbait issue, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1288–1297.
[35] X. Wang, X. He, Y. Cao, M. Liu, T.S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.
[36] X. Wang, X. He, M. Wang, F. Feng, T.S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019, pp. 165–174.
[37] L. Wu, L. Chen, P. Shao, R. Hong, X. Wang, M. Wang, Learning fair representations for recommendation: A graph-based perspective, in: Proceedings of the Web Conference 2021, 2021, pp. 2198–2208.
[38] Y. Xu, C. Han, J. Qin, X. Xu, G. Han, S. He, Transductive zero-shot action recognition via visually connected graph convolutional networks, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 3761–3769.
[39] L. Yang, C.K. Hsieh, H. Yang, J.P. Pollak, N. Dell, S. Belongie, C. Cole, D. Estrin, Yum-me: a personalized nutrient-based meal recommender system, ACM Transactions on Information Systems (TOIS) 36 (2017) 7:1–7:31.
[40] X. Yang, X. Du, M. Wang, Learning to match on graph for fashion compatibility modeling, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 287–294.
[41] X. Yang, P. Zhou, M. Wang, Person reidentification via structural deep metric learning, IEEE transactions on neural networks and learning systems 30 (2018) 2987–2998.
[42] Y. Yang, L. Wu, R. Hong, K. Zhang, M. Wang, Enhanced graph learning for collaborative filtering via mutual information maximization, in, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 71–80.
[43] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.
[44] W. Yuan, H. Wang, X. Yu, N. Liu, Z. Li, Attention-based context-aware sequential recommendation model, Information Sciences 510 (2020) 122–134.
[45] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, F. Klanner, Modeling hierarchical category transition for next poi recommendation with uncertain check-ins, Information Sciences 515 (2020) 169–190.
[46] Zhu, H., Feng, F., He, X., Wang, X., Li, Y., Zheng, K., Zhang, Y., 2020. Bilinear graph neural network with neighbor interactions, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 1452–1458.
[47] Zhuang, Y., Liu, Z., Qian, P., Liu, Q., Wang, X., He, Q., 2020. Smart contract vulnerability detection using graph neural network, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 3283–3290.

Xiaoyan Gao is currently pursuing the Ph.D. degree in the School of Computer Science, Beijing Institute of Technology. She received the B.E. degree in software engineering from Fuzhou University, Fujian, in 2012. Her research interests include recommendation and natural language processing. She has published several works in MM, TMM, and TOIS.

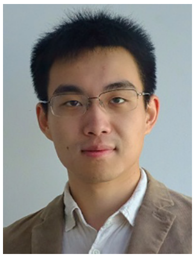


Fuli Feng is a research fellow in National University of Singapore (NUS). He received his Ph.D. in Computer Science from NUS in 2019. He has over 40 publications appeared in several top conferences such as SIGIR, WWW, and ACMMM, and journals including TKDE and TOIS. His work on recommender system has received the Best Poster Award in WWW 2018 and the Honorable Mention of Best Paper Award in SIGIR 2021. Moreover, he has served as the PC member for top-tier conferences including SIGIR, WWW, SIGKDD, NeurIPS, and ACMMM, and the invited reviewer for prestigious journals such as TPAMI, TNNLS, TOIS, and TKDE.

Heyan Huang received the BE degree in computer science from Wuhan University, in 1983, the ME degree in computer science and technology from the National University of Defense Technology, in 1986, and the PhD degree from the China Academy of Sciences, Institute of Computer Technology, in 1989. Now, she is a professor, doctoral tutor, and president in School of Computer, Beijing Institute of Technology, and the director of the Research Center of High Volume Language Information Processing and Cloud Computing. Her current research interests mainly focus on natural language processing.

Xian-Ling Mao received the PhD degree from Peking University, in 2013. Now, he is an assistant professor in the School of Computer Science, Beijing Institute of Technology. His current research interests include Topic Modeling, Learning to Hashing, and Question Answering. Various parts of his work have been published in top conferences and journals including AAAI, TKDE, IJCAI, EMNLP, COLING and CIKM.

Tian Lan is currently pursuing the Ph.D. degree in the School of Computer Science, Beijing Institute of Technology. He received the bachelor's degree from the Beijing institute of technology in 2019. His research interests include natural language processing, automatic evaluation and dialogue systems.



Zewen Chi is currently pursuing the Ph.D. degree in the School of Computer Science, Beijing Institute of Technology. He received the B.E. degree in Computer Science from Beijing Institute of Technology in 2018. His research interests include natural language processing and information extraction.