Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

POI Recommendations with the Use of Knowledge Graph Convolutional Networks

Nabin Paudyal [a], Arun Kumar Timalsina [b]

[a, b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] 075mscsk010.nabin@pcampus.edu.np, [b] t.arun@ioe.edu.np

Abstract

The lack of check-in data, which is a form of implicit feedback data, is a major problem prevalent in almost all of the prevalent Point of Interest (POI) recommendation systems. The problem of cold start and sparsity actually is a commonly occurring theme in collaborative filtering based recommendation systems for all item types. On the other hand, availability of different kinds of contextual information for the POIs creates another unique challenge in regards to leveraging them in the most effective manner. So researchers tend to collect information about attributes of users and items and design algorithms that can make use of these side information along with the user-item interaction data to effectively understand the users' preferences towards items. Knowledge graph (KG) is one of such approaches which consists of tuples representing relationships existing between two entities. Graph Neural Networks (GNN) meanwhile are gaining popularity because of their power in modeling the dependencies between nodes in a graph. They are able to generate rich contextual embedding for entities without having to explicitly specify features and attributes for the nodes representing the entities. In this paper, we propose to combine the contextual information provided by KG with the power of GNN at modeling node dependencies in the resulting Knowledge Graph Convolution Network (KGCN) to generate POI recommendations for users. The experiments carried out on Foursquare dataset exhibited performance improvement of 24.19%, 13.20% and 16.27% respectively for top 5, 10 and 20 POI recommendations in terms of F1-score. Similarly, for Gowalla dataset, the performance improvement observed was 19.35%, 10.29% and 6.77% respectively for top 5, 10 and 20 POI recommendations in terms of F1-score.

Keywords

Graph Neural Networks, Graph Machine Learning, Knowledge Graphs, POI Recommendation System

1. Introduction

The end goal for any recommendation system is to be able to predict beforehand the user interest for a new item based upon user's preferences throughout history, his/her personalized needs, and also the specific properties and traits present in the item, in order to suggest the most appropriate item for the user, enhance the level of satisfaction for the user, and enable the user to make decisions in a much more efficient manner. In the present times of big data, usage of classical recommendation systems is very narrow to solve data mining related challenges [1]. The use of knowledge graphs offers a much more efficient way for designing recommender systems under the premise of big data.

A knowledge graph is basically a directed labeled graph where the labels have well-defined meanings. A directed labeled graph is made up of nodes, edges, and labels. Any entity can act as a node, such as people, company, department, computer, book, car and so on. Two nodes are linked via an edge and the relationship of interest between the nodes is captured by the edge. Some examples of such relationships are the friendship relationship between two people, network connection between two computers and employee-employer relationship between a company and an employee.

A Point of Interest (POI) is a specific location which might be of interest to a visitor. A POI can be anything from restaurant, theatre, stadium, park and grocery store. In strictest terms, a POI is a place on earth associated with specific latitude and longitude. In the context of this paper, we refer to potential places of visit for travelers as POIs. The goal of a POI recommender system is to suggest to the travelers the most relevant places for visit catering to their specific interests.

Making effective recommendations for POIs has historically been challenged with two major problems. First of all, the check-in data for users is very low as compared to other information such as user clicks for items on e-commerce recommendation systems, and thus the recommendation methods face the problem of data sparsity. Researchers observed that density of check-ins for user-POI interaction matrix is just about 0.05% [2], which is considerably small when compared with 1.2% [3] for Netflix data. Furthermore, the feedback that can be inferred from check-in is of implicit type which further makes the problem of POI recommendations challenging.

Knowledge graphs (KG) have emerged as one of the best ways to use contextual information [4] for POI recommendations. The major concept behind using KG to address recommendation problems is to identify the features present in the KG in an effective manner. Lately, network representation learning is identified as a very popular research direction in the field of machine learning. The field of network representation learning [5] strives to learn, for each network node, a low-dimensional representation also at the same time maintaining the original structural information. This technique has emerged as a highly effective mechanism to learn the features in the graph.

In this paper, we explore the problem of KG based POI recommendations with the use of GNNs. The major objective is to represent high-order structure along with logical information in the knowledge graph. This is inspired by the work done by other researchers [6, 7, 8] in combining graph convolutional networks (GCN) along with KG to better understand user preferences towards items of different types. We attempt to explore the potential of this approach in the context of POI recommendations for users. The major underlying principle behind KGCN is to combine information from neighbours along with the entity's bias present in the KG. This approach brings with it two major advantages, one of successfully capturing local proximity structure and the other of adjusting weights for neighbours based on the users' personal preferences for relations of particular types.

Our contribution via this paper can be summarized as follows:

- We propose knowledge graph convolutional networks (KGCN) as an end-to-end framework to understand the preferences of user on travel knowledge graphs for the purpose of generating effective POI recommendations for the user.
- We carry out experiments on the datasets compiled from the checkins on two popular location based social networks in Foursquare and Gowalla to test the effectiveness of our approach. The results validate that our approach is able to outperform existing baselines in the domain of POI recommendations.

2. Literature Review

After the introduction of knowledge graphs by Google in 2012 [9], they were applied by scholars in the domain of recommendation systems in several fields and exciting results were achieved in different applications. The data available in DBpedia, Geonames, and Wikidata was used by Lu [10] to build a knowledge graph consisting of world tourist attractions, in order to recommend tourist attractions. Recommendation for books was also explored by Noia et al. [11] with the use of a knowledge graph.

Both types of vector representations - handcrafted feature vectors along with learned representations, for graphs and relational structures, help in the application of techniques for standard data analysis along with machine learning techniques for the structures. Different such methods for generating embeddings have been explored in the domain of machine learning and knowledge representation [12]. But vector embeddings have attracted very little traction in terms of theoretical viewpoint. Vector embeddings have the ability to bridge the shortcoming between the "discrete" relational data world and the "differentiable" world of machine learning and for this exact cause have a great database research potential. However, very little work has been carried out on relational data embeddings apart from the knowledge graphs' binary relations.

The focus in existing bodies of work is mostly related to the recommendation of point of interests (POIs) with the use of social networks based on location. Extensive work has been done in this area [13]. In general, existing works have used various classes of data: contextual data, social information, categories and tags. Using social information basically comprises the use of information related to the places that a user's friends have visited. It has been demonstrated that there is a rather low overlap of destinations among friends and using this social information is of little worth to improve the performance of recommendations.

Contextual data is helpful as the geographical information is very helpful as users have a tendency to visit the places located nearer to each other. The temporal information can be used since many users visit multiple places at varying times and often visit the identical places within an identical time frame. The use of different context-based data such as weather has also been widely discussed [14].

Memory-based collaborative filtering (CF) based approaches, the likes of user-based and item-based CF, have been widely used for POI recommendations in the past. Ye et al. [15] combined both geographical and social preferences into user-based CF model to recommend POIs. The experiments mostly point that user-based in general produce better results than item-based CF in the context of POI recommendations. Levandoski et al. [16] further enhanced the item-based CF approach by taking into longer travel distances as less favorable for travelers. Various model-based CF techniques as well have been tried in the domain of POI recommendation. Noulas et al. [17] observed that user-based matrix factorization (MF) yields worse results in comparison to that of item- based approaches for POI recommendation. Traditional MF techniques best suited for explicit feedback data was used to generate POI recommendation in their work, resulting in subpar performance.

Li, Xutao et. al [18] proposed a geographical factorization method based on ranking, which they named as Rank-GeoFM, for the purpose of recommending POIs, which deals with the two major problems of data requiring implicit feedback and making optimal use of context information for POI recommendations. In the model put forward by the authors, they took into consideration that the frequency of check-in directly correlates with users' preferences and through the ranking of the POIs, learned the factorization.

The method proposed in our work heavily draws upon GCN, in particular the non-spectral method, which directly computes over the original graph and defines convolution operations over a group of nodes. The work in particular focuses on a specific type of graph i.e. knowledge graph. To incorporate the neighbourhoods of varying sizes and maintain the property of sharing weights which is inherent in CNNs, researchers propose to use techniques such as learning a particular weight matrix for each and every degree [19], sampling regions of local proximity in graphs and only including fixed size of neighbours.

3. Methodology

3.1 Machine Learning on Graphs

3.1.1 Graph Neural Networks

A Graph Neural Network is a form of Neural Network that works with the graph structure directly. Node classification is a common use of GNN. In general, every node in the graph has a label, and we want to predict the labels of the nodes without using ground truth.

Each node v is defined by its feature x_v and associated with a ground-truth label t_v in the node classification problem setting. The goal is to use the labels of the labeled nodes in a partially labeled graph G to predict the labels of the unlabeled nodes. It learns to represent each node as a d dimensional vector (state) h_v containing information about its surroundings.Specifically,

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}) \tag{1}$$

where $x_co[v]$ denotes the features of the edges connecting with v, $h_ne[v]$ denotes the embedding of the neighboring nodes of v, and $x_ne[v]$ denotes the features of the neighboring nodes of v. The transition function f is used to project these inputs onto a d-dimensional space. We can use the Banach fixed point theorem to rephrase the following equation as an iterative updating process because we're looking for a unique solution for h_v.

$$H^{t+1} = F(H^t, X) \tag{2}$$

H and X denote the concatenation of all the h and x, respectively.

The output of the GNN is computed by passing the state h_v as well as the feature x_v to an output function g.

$$o_v = g(h_v, x_v) \tag{3}$$

3.1.2 Graph Convolutional Networks

Because the filter parameters are often shared across all sites in the graph, Graph Convolutional Networks

(GCN) are perhaps the most prevalent graph neural network architecture in use today. The purpose of these models is to train a function of signals/features on a graph G = (V,E) that accepts the following as input:

- A feature description x_i for every node i; summarized in a N×D feature matrix X(N: number of nodes, D: number of input features).
- A representative description of the graph structure in matrix form; typically in the form of an adjacency matrix A

and produces a node-level output Z (an N×F feature matrix, where F is the number of output features per node). Graph-level outputs can be modeled by introducing some form of pooling operation.

Every neural network layer can then be written as a nonlinear function.

$$H^{l+1} = f(H^l, A) \tag{4}$$

with H^0 = X and H^L = Z (or z for graph-level outputs), L being the number of layers. The specific models then differ only in how f is chosen and parameterized.

As an example, let's consider the following very simple form of a layer-wise propagation rule:

$$f(H^{l+1}, A) = \sigma(AH^l W^l) \tag{5}$$

where W^l is a weight matrix for the l-th neural network layer and σ is a non-linear activation function like the ReLU.

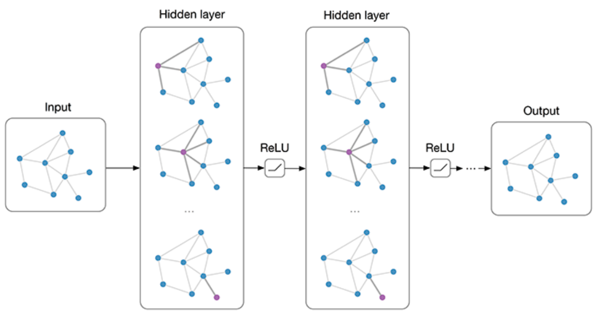

Figure 1: Architecture of GCN

3.1.3 Knowledge Graphs

A knowledge graph is made up of triples of entity-relation-entity (h,r,t). The head and tail are represented by h and t, respectively, while the relationship between the head and tail is represented by r. The link between the things represented in the network is described by a knowledge graph. A triple *(Big Ben, location.location.city, London)*, for example, shows the relationship that Big Ben is one of the places in London.

3.2 Problem Formulation

The problem we're dealing in this paper can be formally defined as a knowledge-graph-aware POI recommendation problem. In a typical POI recommendation scenario, we are given a set of M users represented by U = $\{u_1, u_2, ..., u_M\}$ and set of N POIs represented by V = $\{v_1, v_2, ..., v_N\}$. We are also provided with a user-POI interaction matrix Y $\in R^{MxN}$ defined by user's feedback, which is check-in information in the context of our work. Given a knowledge graph G = (h, r, t) where h $\in$ E, r $\in$ R and t $\in$ E, the goal is to learn $\hat{y}_{uv}$ = F(u, v | Y, G), where $\hat{y}_{uv}$ represents the probability that the user u will be interested in the POI v.

3.3 KGCN Layer

In this research, we propose KGCN as a way to capture high-order structural proximity among entities in a knowledge graph. This subsection begins by detailing a single KGCN layer. For pair of user u and item (entity) v, N(v) can be used for denoting the set of entities which are directly connected to v and r_{e_i,e_j} for representing the relation between entity e_i and e_j . A function g : $R^d \times R^d \to$ R (e.g., inner product) can also be used to compute the score between a user and a relation:

$$\pi_r^u = g(u, r) \tag{6}$$

where u $\in R^d$ and r $\in R^d$ are the representations respectively for user u and relation r , and d is the dimension of representations. In general, π_r^u represents the importance of relation r to user u. For example, a user may be more interested in the POIs of same "city" with his/her historically liked ones, while another user may have greater inclination towards the ones with the "category" of the POI.

To represent the importance of neighbours for the POI v, we combine the neighbourhood for v in a linear fashion.

$$v_{N(v)}^u = \sum_{e \in N(v)} \tilde{\pi}_{r_{v,e}}^u e \tag{7}$$

where the normalized relation between user and a relation is given by the following equation:

$$\tilde{\pi}^{u}_{r_{v,e}} = \frac{exp(\pi^{u}_{r_{v,e}})}{\sum_{e\in\ N(v)} exp(\pi^{u}_{r_{v,e}})} \tag{8}$$

and e represents the entity. The user-relational scores are a type of filter that adds personification to the neighbors, allowing them to be paired with a bias computed using user-specific relational scores.

We have used three different types of aggregators to combine the neighbours in a KGCN layer.

- Sum aggregator sums the two representing vectors and applies linear transformation to the result.

$$agg_{sum} = \sigma(W \cdot (v + v^{u}_{S(v)}) + b) \tag{9}$$

- Concat aggregator concatenates the two representing vectors and applies linear transformation to the result.

$$agg_{concat} = \sigma(W \cdot concat(v, v^{u}_{S(v)}) + b) \tag{10}$$

- Neighbour aggregator directly takes the representation for the neighbourhood of the entity.

$$agg_{neighbour} = \sigma(W \cdot v^{u}_{S(v)} + b) \tag{11}$$

3.4 Learning Algorithm

The steps involved in the learning algorithm for our proposed approach can be summarized in the following steps.

1. Get all of your neighbors' incoming messages.
2. By conducting an aggregation process, you can combine all of those messages into a single message.
3. With a learnable weight matrix, matrix multiplication of the neighborhood message.
4. With a learnable weight matrix, multiply the initial node message by a matrix.
5. Steps 3 and 4 should be combined.
6. Apply a ReLU activation function to the total.
7. Repeat the process for as many layers as you'd like. The output of the final layer is the outcome.

We use BCE loss as the loss function and Adam as the optimizer for our training model.

4. Experiments

4.1 Data Set

4.1.1 Knowledge Graph

The knowledge graph for POI is constructed using Google's Knowledge Graph Search API. We use the following relationships to represent a POI in the KG.

- containsPlace
- address
- aggregateRating
- type
- isAccessibleForFree

4.1.2 User-POI Interaction

The user-POI interaction matrix is constructed from the check-in information available on the datasets for the two highly popular location based social networks.

- **Foursquare**

  The dataset consists of global information about check-in of visitors on Foursquare during the time period between April 2012 and September 2013.

Entities	Count
Checkins	1,94,108
Users	2,321
POIs	5,596
Sparsity	99.18%

Table 1: Foursquare metrics

- **Gowalla**

  The dataset consists of global information about check-in of visitors on Gowalla during the time period between February 2009 and October 2010.

Entities	Count
Checkins	4,56,967
Users	10,162
POIs	24,237
Sparsity	99.88%

Table 2: Foursquare metrics

4.2 Parameters

There are different parameters involved in our method, choice of which impacts the performance of the overall end-to-end experiments.

- **Neighbour sampling size(K):**

  This refers to the number of neighbours used while aggregating neighbourhood information for a node.

- **Embedding dimension(d)**

  The dimension of embedding represents the length of the vector representing a node.

- **Depth of receptive field(H)**

  This refers to the ability of the model to capture long distance relationships.

- **Aggregation function(σ)**

  This refers to the multiple ways in which we can combine the information coming from the neighbouring nodes.

4.3 Baselines

We compare the performance of our approach with several existing baselines in the domain of POI recommendation. All of the compared methods are KG-free methods.

- PMF [20], which stands for Probabilistic Matrix Factorization, is a commonly used factorization method common for other user-item recommendation problems as well.

- GeoMF [21], which stands for Geographical Matrix Factorization, is a popular and established method used for POI recommendation.

- Rank-GeoFM, which stands for Ranking based Geographical Factorization Method, is the current state-of-art for POI recommendation.

4.4 Experimental Setup

We computed the AUC for different values of the parameters for both the datasets and used the best performing model to run the final set of experiments. The value of k = 2, d = 4, H = 1 and σ = sum were set initially and for each instance, different values were used for a specific parameter while setting all others to constant.

	Foursquare	Gowalla
2	0.791	0.782
3	0.794	0.786
4	0.795	0.788
5	0.798	0.792
6	0.794	0.789

Table 3: AUC Results for different K

	Foursquare	Gowalla
4	0.789	0.775
8	0.793	0.780
16	0.797	0.782
32	0.793	0.786
64	0.790	0.784
128	0.789	0.778

Table 4: AUC Results for different d

	Foursquare	Gowalla
1	0.724	0.682
2	0.746	0.713
3	0.738	0.724
4	0.724	0.712

Table 5: AUC Results for different H

	Foursquare	Gowalla
sum	0.794	0.672
concat	0.790	0.654
neighbour	0.682	0.642

Table 6: AUC Results for different σ

4.5 Evaluation

We evaluate the results obtained using our approach with the results obtained from other baseline methods and compare the results between them in terms of the following aspects. The values of N are varying and set to 5, 10 and 20.

- **Pre@N:**

  Precision compares the extent to which the visitors actually visit the POIs recommended by our model.

- **Rec@N:**

  Recall compares the extent to which the model actually recommends the POIs actually the visitors are intereste in to them.

- **F1@N:**

  This is simply the harmonic mean of the Precision and Recall scores.

5. Results and Discussion

5.1 Results on Foursquare Dataset

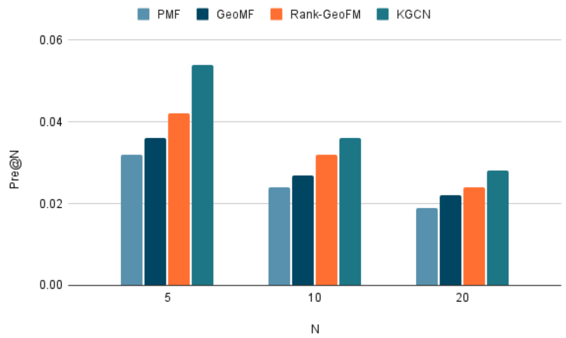

Figure 2: Pre@N-Foursquare

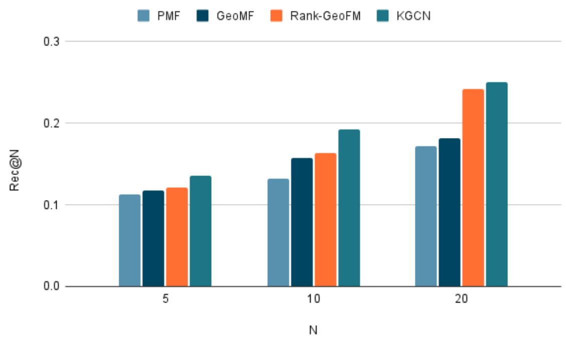

Figure 3: Rec@N-Foursquare

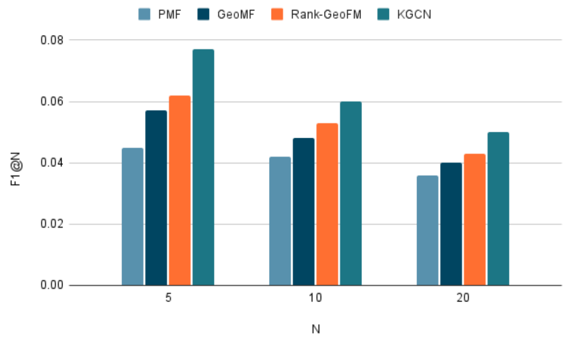

Figure 4: F1@N-Foursquare

5.2 Results on Gowalla Dataset

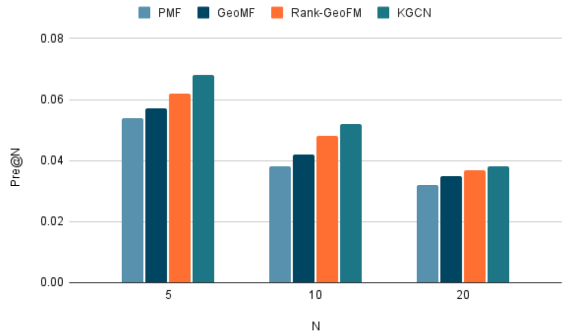

Figure 5: Pre@N-Gowalla

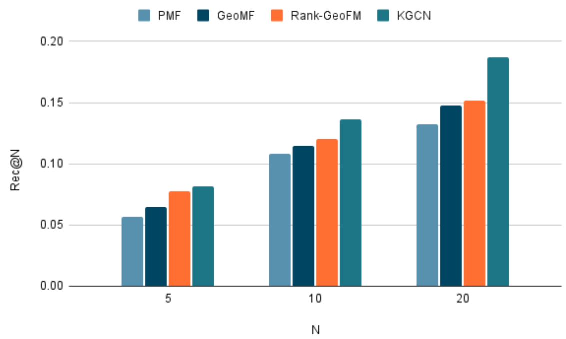

Figure 6: Rec@N-Gowalla

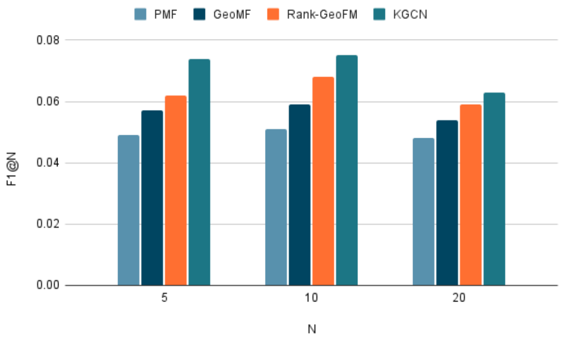

Figure 7: F1@N-Gowalla

5.3 Discussion

From the experiments, we observed that our approach is able to obtain better results compared to existing baselines in POI recommendation. We obtained optimal results in terms of AUC when the value of neighbouring sample size was taken as 5 for both datasets. The best value for the embedding dimension was 16 for Foursquare and 32 for Gowalla dataset. Fousquare performed best with the receptive field size of 2 but best performance on Gowalla was achieved when this value was set to 3. The optimal aggregation method for both the datasets was found to be summation.

6. Conclusion

With this paper, we propose to use knowledge graph convolutional networks for generating POI recommendations for users. KGCN is an extension over GCN which combines GNN with KG which combines information from neighbouring nodes in a careful yet biased manner. Because of this, the model is able to learn structural as well as logical information from the KG along with the user's specific preferences. We also applied this method taking a minibatch approach, so that it can be effectively applied on large datasets and knowledge graphs. By carrying out various experiments on the real world datasets constructed for Foursquare and Gowalla, the method is found to be performing better than current state-of-art methods in POI recommendation.

We have also been able to identify several avenues to continue work and build upon the findings of the experiments carried out so far.

1. Use different approaches to generate node embeddings apart from the GCN approach used in this paper.
2. This paper has mostly focused on leveraging KGs for POI. Constructing KGs for users and exploiting their relational information could be another potential area to obtain better recommendations.

References

[1] Will Serrano. Intelligent recommender system for big data applications based on the random neural network. *Big Data and Cognitive Computing*, 3:15, 02 2019.

[2] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. New York, NY, USA, 2015. Association for Computing Machinery.

[3] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. 9(2), 2007.

[4] Sihang Hu, Zhiying Tu, Zhongjie Wang, and Xiaofei Xu. A poi-sensitive knowledge graph based service recommendation method. In *2019 IEEE International Conference on Services Computing (SCC)*, pages 197–201, 2019.

[5] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*.

[6] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. New York, NY, USA, 2019. Association for Computing Machinery.

[7] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. New York, NY, USA, 2019. Association for Computing Machinery.

[8] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. New York, NY, USA, 2019. Association for Computing Machinery.

[9] Amit Singhal. Introducing the knowledge graph: things, not strings. 2012. https://blog.google/products/search/introducing-knowledge-graph-things-not/.

[10] Chun Lu, Philippe Laublet, and Milan Stankovic. Travel attractions recommendation with knowledge graphs. pages 416–431, 11 2016.

[11] Tommaso Di Noia, Iván Cantador, and Vito Ostuni. Linked open data-enabled recommender systems: Eswc 2014 challenge on book recommendation. volume 475, pages 129–143, 05 2014.

[12] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. New York, NY, USA, 2017. Association for Computing Machinery.

[13] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: A survey. *GeoInformatica*, 19, 07 2015.

[14] Context-aware points of interest suggestion with dynamic weather data management. page 12. Springer International Publishing, 2013.

[15] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. New York, NY, USA, 2011. Association for Computing Machinery.

[16] Justin J. Levandoski, Mohamed Sarwat, A. Eldawy, and M. Mokbel. Lars: A location-aware recommender system. *2012 IEEE 28th International Conference on Data Engineering*, pages 450–461, 2012.

[17] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 144–153, 2012.

[18] Xutao Li, Gao Cong, Xiaoli li, Tuan-Anh Pham, and Shonali Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. *SIGIR*, 09 2015.

[19] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2007.

[21] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. New York, NY, USA, 2014. Association for Computing Machinery.