

RESEARCH

Open Access



Recommendation method for fusion of knowledge graph convolutional network

Xiaolin Jiang¹, Yu Fu² and Changchun Dong^{1*} 

*Correspondence:

57102526@qq.com

¹ Jinhua Advanced Research
Institute, Jinhua 321013,
Zhejiang, China

Full list of author information
is available at the end of the
article

Abstract

In the application of internet of vehicles system, it is particularly important to obtain real-time and effective vehicle information and provide personalized functional services for vehicle operation. This algorithm combines knowledge graph technology with convolutional network and presents a new algorithm model, that is, when calculating the representation of a given entity in the knowledge graph, the information of the neighboring entity is combined with the deviation. Through the integration of neighbor entity information, the local neighborhood structure can be better captured and stored in each entity, and the weight of different neighbor entities depends on the relationship between them and the specific user, which can better reflect the user's personalized interests, in order to fully demonstrate the characteristics of the entity. Compared with the traditional coordinated filtering technology SVD model, this model has improved accuracy and F1 value.

Keywords: Knowledge graph, Recommended technology, Convolutional network

1 Introduction

The Internet of vehicles (IOV) is an integral part in the context of the Internet of things [1]. The equipment on the vehicle can obtain the vehicle information from the information network platform by wireless communication technology, making relevant recommendation technology, and viewing the vehicle in real time. The recommendation algorithm can provide personalized functional services for vehicle operation [2, 3]. The important idea of the KGCN model is summarized in that when calculating its feature representation for a given entity in the knowledge graph, it first performs a width search method to obtain its multi-hop associated entity in the knowledge graph [4], and then combines the relevant information of its neighbor nodes. The weights are aggregated and the features of the entity are merged. There are two main meanings of adopting this method [5, 6]: first, the calculation of the feature vector of the entity contained in the knowledge graph is weighted and aggregated with the information of the neighboring entity within a certain range of the entity, second, the information of the neighbor node is calculated for the node. The degree of selection is jointly influenced by the established entity and neighboring entities. This method not only effectively combines the semantic information between entities [7], but also characterizes the user's own interests. There

may be many in extreme conditions since the neighbor nodes of each entity are largely different. For this reason, drawing lessons from the idea of graph convolutional network, a concept of receptive field is defined in knowledge graph convolutional network [7]. The receptive field refers to a set of neighborhoods of a fixed size. In this way, the calculation amount of the knowledge graph convolutional network has a certain controllability, and this greatly improves the scalability of the algorithm. So as to achieve effective access to vehicle information, real-time information of the vehicle.

2 Methods

2.1 KGCN layer

The predictive equation for judging whether the users are interested in vehicle information in the IOV system.

$$\hat{y}_{uv} = F(u, v | \theta, Y, G) \quad (1)$$

where G is the knowledge graph composed of entity-relation-entity triples, Y represents the sparse matrix of a given user's vehicle, u represents user, v represents Vehicles, θ represents the set of learning parameters of the whole model.

2.1.1 Measure users' preference for different vehicle information relationships

$$\pi_r^u = g(r, u) \quad (2)$$

where g is aggregator, r represents relationship, π represents weight. Define function g : a certain function that can make $R^d \times R^d \rightarrow R$, here is an aggregator, used to calculate the score between the user and a certain. Where d is the dimension of the feature representation of the knowledge graph.

The real meaning here can be understood as (calculating the user's preference for different relationships). For example, if a user pays much attention to the vehicle service information of the corresponding online ride-hailing, he will pay more attention to the number of vehicle service times, user evaluation and other information, and will be more willing to use high-quality vehicles based on these judgments. Here, g function is used to measure the user's preference for different vehicle information relationships.

2.1.2 Use the linear combination of neighbor information to describe the neighborhood information of a node

$$v_{N(v)}^u = \sum_{e \in N(v)} \pi_{r_{v,e}}^u e \quad (3)$$

where $N(v)$ is the set of all entities directly connected to entity v (that is, within one hop), Note that the weight π here is not only related to the relationship between the two nodes v and e , but also related to the characteristics of the corresponding user u at this time. The weight here is the g function scores of all entities e and relations r in $N(v)$ corresponding to v and then normalized to formula 4.

$$\pi_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in N(v)} \exp(\pi_{r_{v,e}}^u)} \quad (4)$$

where $N(v)$ represents the set of all entities directly connected to entity v (that is, within one hop), where e is the characteristics of entity.

When calculating the representation of a given entity neighborhood, the vector scores of users and relationships are similar to the role of personalized filters. This processing is because the neighbor feature vectors are focused on the basis of these user-specific vectors.

2.1.3 Control the number of neighbor users

$$S(v) \rightarrow \{e | e \in N(v)\} \quad (5)$$

$$|S(v)| = K \quad (6)$$

where $N(v)$ represents the set of all entities directly connected to entity v (that is, within one hop), $S(v)$ is entity, (single-layer) accepting field, e is the characteristics of entity, K represents Hyperparameter, select K neighbor users.

In a real knowledge graph [7], there will be too many neighbor users in a certain node v , which will bring huge pressure to the calculation of the overall model. It does not use all its neighbor users, but randomly and uniformly sample a fixed-size set from its neighbor users. Instead, a hyperparameter K is defined. For each node v , only K neighbor users are selected for calculation. In other words, the neighborhood representation of v at this time is written as. In KGCN, because the calculation of the final feature of v is sensitive to these regions $v_{S(v)}^u$.

2.1.4 Define aggregator

The key point in the KGCN model is the fusion of the feature v of the entity and the neighbor feature $v_{S(v)}^u$. Three types of aggregators are defined in KGCN. In this experiment, an additive aggregator is used for comparison experiments.

Sum Aggregator. The sum aggregator is to perform arithmetic addition of two eigenvectors, and then add a nonlinear transformation.

$$\text{agg}_{\text{sum}} = \sigma(W \cdot (v + v_{S(v)}^u) + b) \quad (7)$$

In the formula, w is conversion weight, b represents bias, $S(v)$ represents physical (single-layer) acceptance field.

Splicing aggregator The splicing aggregator performs splicing operations on two feature vectors, and then adds a nonlinear transformation.

$$\text{agg}_{\text{concat}} = \sigma(W \cdot \text{concat}(v, v_{S(v)}^u) + b) \quad (8)$$

Neighbor aggregator The neighbor aggregator only uses the neighbor features of the entity v , and directly uses the neighborhood representation to replace the representation of the v node.

$$\text{agg}_{\text{neighbor}} = \sigma(W \cdot v_{S(v)}^u + b) \quad (9)$$

3 Algorithm

For a given entity, its final vector feature indicates that it must have a significant relationship with its neighbor users directly connected, which is called a first-order entity feature.

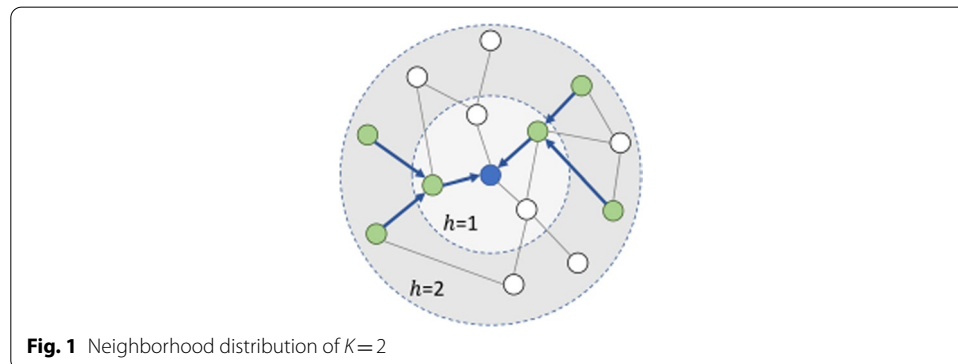
In order to further study the relationship between the two entities in the knowledge graph and explore other potential interests of users, the user's neighborhood set is expanded from one layer to multiple layers, that is, the selected single entity feature is transferred to a layer of adjacent neighborhoods, Then the first-order feature vector can be obtained, as shown in Fig. 1. In other words, the h -level features of a single entity emphatically aggregate the neighbor feature representations within h hops near itself. For this design, different aggregators are used to collect neighborhood information.

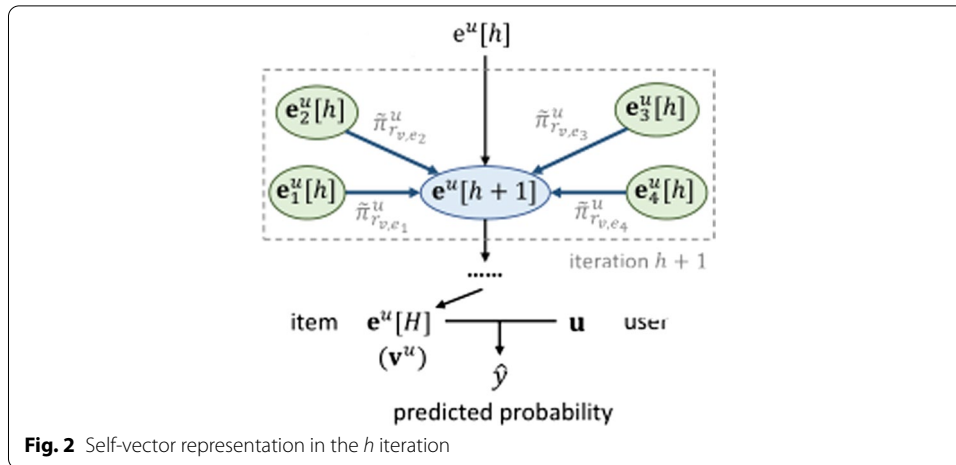
H represents the maximum number of receptive fields. For the obtained entity pairs of users and Vehicles, first obtain the receptive field size of the item through iteration, and then repeat the aggregation for H times. In the h time, calculate the neighborhood $e \in M[h]$ features of each entity. Then aggregate it with its own features $e^u[h-1]$ to obtain the feature representation of v in the next iteration, as shown in Fig. 2. V^u and user feature u are passed into the function together: $R^d \times R^d \rightarrow R^d$ to get the final predicted probability.

$$\hat{y}_{uv} = f(u, v^u) \quad (10)$$

where V^u represents H -order entity feature, Since the algorithm traverses all possible user-item pairs, in order to make the algorithm more efficient, the negative sampling technique is selected in the training of this article. Therefore, the specific loss function of the KGCN model is as follows:

$$\Gamma = - \sum_{u \in U} \left(\sum_{v: y_{uv}=1} \ell(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T^u} E_{v_i \sim P(v_i)} \ell(y_{uv}, \hat{y}_{uv}) \right) + \lambda \|F\|_2^2 \quad (11)$$





In the formul, ℓ represents a-cross-direction loss function, P represents the distribution of negative samples, T^u is the number of negative samples of user u , p represents uniform distribution.

$$T^u = |\{v : y_{uv} = 1\}| \quad (12)$$

The code of the KGCN algorithm is shown in Table 1.

4 Results and discussion

4.1 Results

This experiment uses a medical data database as a training experiment. The specific data is shown in Table 2.

The experimental data and various parameters are shown in Table 3.

In this experiment, a classic hidden factor model based on collaborative filtering, singular value decomposition (SVD) and the fusion knowledge graph convolutional network KGCN [8, 9], are used in the recommended technology to compare experiments, and the results can be compared to Table 4.

The $F1$ and accurate value line graphs of the training set, validation set, and test set of the data are shown in Figs. 3 and 4.

4.2 Discussion

The traditional collaborative filtering algorithm SVD does not have the auxiliary effect of the knowledge map, and the experimental effect is worse than that of KGCN. In the real relationship, using the attributes of users and vehicles, the various attributes do not exist alone, but are related to each other, forming a knowledge graph. Therefore, KGCN can better represent the relationship and perform better than SVD [9]. Because the multi-hop neighborhood structure is used, it shows that the knowledge graph captures neighborhood information is helpful for recommendation.

First, the influence of the number of samples in the neighborhood, the performance of the algorithm is further analyzed by changing the number of samples, as shown in Table 5. It can be concluded from the table that the best performance is achieved when the sampling

Table 1 KGCN algorithm code

Algorithm 1:KGCN algorithm
Input: Interaction matrix Y ; knowledge graph $G(\mathcal{E}, \mathcal{R})$; neighborhood sampling mapping $S: \mathcal{E} \rightarrow 2^{\mathcal{E}}$; trainable parameters: $\{u\}_{u \in \mathcal{U}}, \{e\}_{e \in \mathcal{E}}, \{r\}_{r \in \mathcal{R}}, \{W_i, b_i\}_{i=1}^H$; hyper-parameters: $H, d, g(\cdot), f(\cdot), \sigma(\cdot), \text{agg}(\cdot)$ Output: prediction function $F(u, v \Theta, Y, G)$ While KGCN not converge do For (u, v) in Y do $\{M[i]\}_{i=0}^H \leftarrow \text{Get-Receptive-Field}(v)$; $e^u[0] \leftarrow e, \forall e \in M[0]$; for $h=1, \dots, H$ do for $e \in M[h]$ do $e_{S(e)}^u[h-1] \leftarrow \sum_{e' \in S(e)} \pi_{e, e'}^u e'^u[h-1]$; $e^u[h] \leftarrow \text{agg}(e_{S(e)}^u[h-1], e^u[h-1])$; $v^u \leftarrow e^u[H]$; Calculate predicted probability $\hat{y}_{uv} = f(u, v^u)$; Update parameters by gradient descent; Return F ; Function Get-Reception-Field(v) $M[H] \leftarrow v$; for $h = H-1, \dots, 0$ do $M[h] \leftarrow M[h+1]$; for $e \in M[h+1]$ do $M[h] \leftarrow M[h] \cup S(e)$; Return $\{M[i]\}_{i=0}^H$;

Table 2 Training data

User	Items	Interactions	Entities	Relation	KG triples
8109	9424	401342	88649	18	94474

Table 3 Experimental parameters

K	d	H	λ	η	Batch size
8	64	1	2×10^{-5}	2×10^{-4}	256

Table 4 Comparison of data and experimental results

Model	AUC	F1
SVD	0.963	0.919
KGCN-sum	0.978	0.932

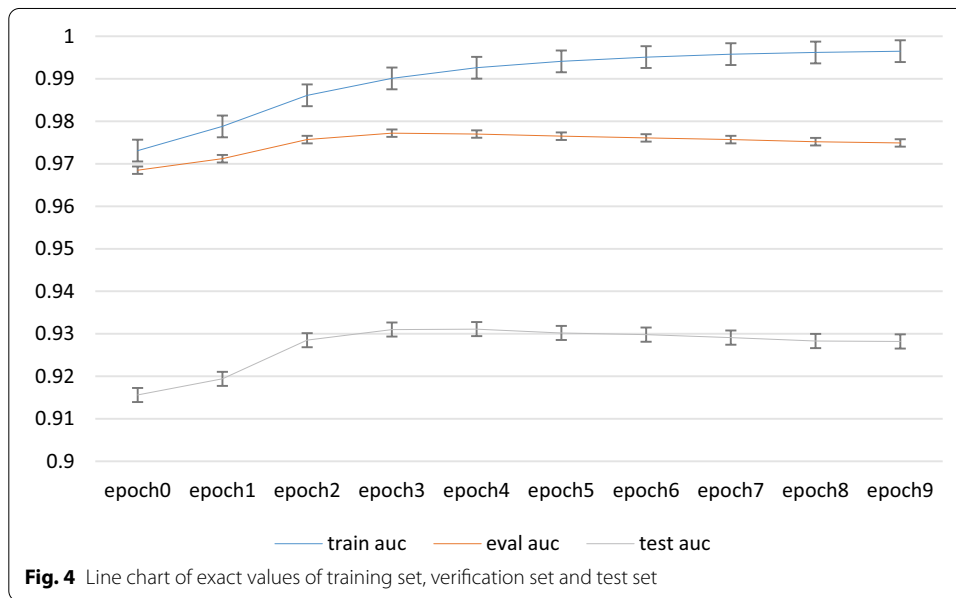
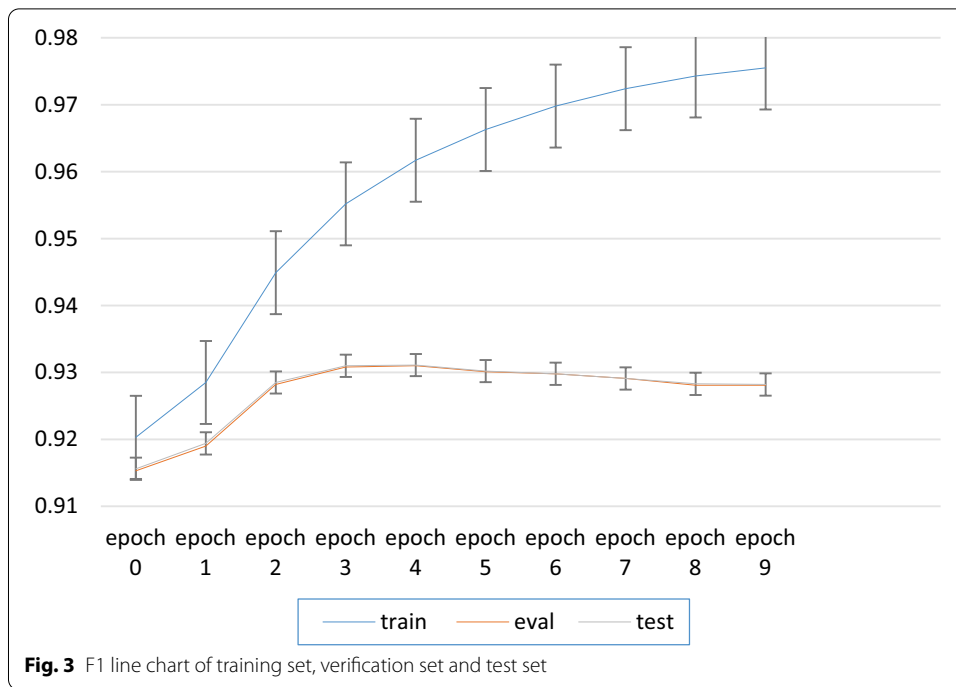


Table 5 The influence of increasing K value on accuracy

K	2	4	8	16	32	64
AUC	0.69	0.726	0.1738	0.726	0.710	0.729

Table 6 Influence of increasing H value on accuracy

H	1	2	3	4
AUC	0.739	0.731	0.635	0.503

format is 8. This is because a too small K does not have enough capacity to contain neighbor information, and a large K is easily affected by noise [10].

Second, adjust the influence of the number of receiving field layers on the experimental results. After experimenting with the number of layers H of the receiving field from 1 to 4, it is found that it has an effect on the performance of KGCN. The results are shown in Table 6. By comparing the data, it can be concluded that H is more sensitive to the experimental effect of the model than the number of samples K in the field. When $H=4$, the accuracy rate drops significantly. This is because too many layers of the receiving field have a negative impact on the experimental effect. This result is consistent with common sense judgment, because if the connected relationship in the knowledge graph is longer, these influencing factors at the end of the relationship chain are meaningless when recommending the similarity of Vehicles. In other words, H being 1 or 2 is sufficient for real life scenarios.

In the IOV system, for physical vehicle information extraction, the data required for training data is too high, and high-quality data has a large labor cost. Therefore, unsupervised learning is considered in subsequent studies to reduce the manual pressure as much as possible. For recommendation technology, and the algorithm is based on knowledge graph construction, the workload is relatively large.

5 Conclusions

The main content of this chapter is to in-depth study of the recommendation technology based on the knowledge graph-KGCN, this method is to aggregate the feature vector representation of the adjacent entities of a given entity [11, 12], and calculate the neighborhood features of the entity by weighting, and by setting a fixed-size neighborhood collection as the acceptance field, the calculation amount of the algorithm is controlled. By comparing with the traditional collaborative filtering algorithm, the result proves the superiority of the algorithm, and it is more suitable for the recommendation of knowledge graph. In the application of IOV system. The new algorithm is proposed to better obtain real-time and effective vehicle information, and provide personalized functions for vehicle operation to provide better services.

Abbreviations

IOV: Internet of vehicles; KGCN: Knowledge graph convolutional network; SVD: Singular value decomposition.

Acknowledgements

The authors acknowledged the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Authors' contributions

Xiaolin Jiang proposes the innovation ideas and theoretical analysis. Other authors also have contributed jointly to the manuscript. All of the authors have read and approved the final manuscript.

Funding

This work was supported in part by the Heilongjiang Province Natural Science Fund Project (No. F2020011) and Heilongjiang Provincial Postdoctoral Science Foundation (No. LBH-Z16054).

Availability of data and materials

The datasets used and analysed during the current study are available from the corresponding author on reasonable request.

Declarations**Competing interests**

The authors declare that they have no competing interests.

Author details

¹Jinhua Advanced Research Institute, Jinhua 321013, Zhejiang, China. ²Heilongjiang University of Science and Technology, Harbin 150000, Heilongjiang, China.

Received: 15 December 2021 Accepted: 24 February 2022

Published online: 28 March 2022

References

1. P. Zeng, Z. Chen, Y. Ma, G. Zhao, Design of IOV privacy protection authentication scheme based on blockchain. *Comput. Appl. Res.* **38**(10), 2919–2925 (2021)
2. X. Liu, X. Zhang, M. Jia et al., 5G-based green broadband communication system design with simultaneous wireless information and power transfer. *Phys. Commun.* **28**, 130–137 (2018)
3. Y. Xue, J. Jin, A. Song, Y. Zhang, Y. Liu, K. Wang, Relation-based multi-type aware knowledge graph embedding. *Neurocomputing* **456**, 11–22 (2021)
4. T. Phan, P. Do, Building a Vietnamese question answering system based on knowledge graph and distributed CNN. *Neural Comput. Appl.* **33**, 14887–14907 (2021)
5. L. Sang, M. Xu, S. Qian, X. Wu, Knowledge graph enhanced neural collaborative filtering with residual recurrent network. *Neurocomputing* **454**, 417–429 (2021)
6. H. Werneck, N. Silva, M. Viana, A.C. Pereira, F. Mourão, L. Rocha, Points of interest recommendations: methods, evaluation, and future directions. *Inform. Syst.* **101**, 101789 (2021)
7. R. Kojima, S. Ishida, M. Ohta, H. Iwata, T. Honma, Y. Okuno, kGCN: a graph-based deep learning framework for chemical structures. *J. Cheminform.* **12**(1), 1–10 (2020)
8. X. Wang, X. Liu, J. Liu, X. Chen, H. Wu, A novel knowledge graph embedding based API recommendation method for Mashup development. *World Wide Web* **24**(3), 869–894 (2021)
9. Information Technology-Cloud Computing; findings from nanjing university of science and technology reveals new findings on cloud computing (Kg2rec: Lsh-cf recommendation method based on knowledge graph for cloud services). *Comput. Technol. J.* (2020)
10. X. Liu, X. Zhang, Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer. *IEEE Internet Things J.* **6**(4), 5971–5980 (2019)
11. J. Ren, J. Long, Z. Xu, Financial news recommendation based on graph embeddings. *Decis. Support Syst.* **125**, 113115 (2019)
12. C. Ma, B. Zhang, A new query recommendation method supporting exploratory search based on search goal shift graphs. *IEEE Trans. Knowl. Data Eng.* **30**(11), 2024–2036 (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
