Journal of Computing and Electronic Information Management

ISSN: 2413-1660

A weighted knowledge graph recommendation algorithm

Bo He [a], Ruimeng Cao [b]

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

[a]29659807@qq.com, [b]52980551@qq.com

Abstract: This paper proposed a weighted knowledge graph recommendation algorithm. Firstly, the similarity set of item entities is obtained by collaborative filtering recommendation algorithm, then we can obtain the knowledge graph with weight according to fusing the weight into it by calculating the weight between item entities and attribute entities. Finally, the entities in the weight knowledge map are embedded into a low dimensional vector by node2vec algorithm.

Keywords: Collaborative filtering recommendation algorithm, knowledge graph with weight, node2vec algorithm

1. Introduction

Until now the research and application of recommendation system have made great progress, and its types are also diverse, among their collaboration filtering recommendation algorithm as the most widely used method, which calculating the similarity of user or item entities, it has a good applicability and feasibility. However, it also faces many problems like data sparsity, cold boot in front of gigantic amount data. Therefore, some researchers considered using knowledge graph as auxiliary information to improve the recommendation results by replenishing the feature description of user and item entities [1-3]. Knowledge graph algorithm contains the background information of item entities and their relationships, which effectively solved the problems of data sparsity and cold boost, and improved the accuracy, diversity and interpretability of recommendation results. Reference fused the semantic information of knowledge graph and the collaborative filtering information to enhance the efficiency of recommendation. Reference regarded user entities as entities of knowledge graph, and the feedback of user entities on the item entities was regarded as a relationship of knowledge graph, it fused the rating of item entities into knowledge graph naturally, the similarity between user entities and item entities was calculated by knowledge graph embedding algorithm for realizing recommendation.

Consequently, this paper proposed a weighted knowledge graph recommendation algorithm. It fused the collaborative filtering recommendation information into knowledge graph, and built the knowledge graph with weight, which reduced the irrelevant attributes' influence of knowledge graph on recommendation results, and improved the efficiency of recommendation.

2. Related theories

2.1 Item entity-based collaborative filtering recommendation algorithm

The item entity-based collaborative filtering recommendation algorithm[4] is a nearest neighbor algorithm. The basic idea is that the recommended item entity is the counterpart of what user likes. The similarity can be calculated according to the exchanged information between user entity and item entity, then the item entity counterpart would be recommended to the user. The algorithm to calculate the similarity between items is as follows:

1. Jaccard similarity coefficient

$$sim(i,j)=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)|*|N(j)|}} \tag{1}$$

Where $N(i)$ is the number of users who like i, $N(j)$ is the number of users who like item j, and the numerator is the number of users who like two items at the same time.

2. Cosine similarity

$$sim(i,j)=\cos\theta=\frac{v_i * v_j}{\|v_i\| * \|v_j\|} \tag{2}$$

The cosine similarity can introduce user ratings into the similarity calculation, and solve the situation that the user may have purchased but disliked in Method (1).

The collaborative filtering recommendation algorithm is interpretable to some extent, but due to the lack of the context information, it is difficult to intuitively express its interpretable recommendation results.

2.2 Recommendation based on Knowledge Graph

As a kind of directed information heterogeneous network, the knowledge graph contains the relationship between entities, so it contains a large amount of background information of item entities and the relationship between them, besides, it can be combined with user-item network formed by user-behavior data to extend the hidden relatively relationship between user entities and item entities and replenish the interactive data between user entities and item entities. The efficiency of recommendation can be improved according to the above manipulate. Deep learning was firstly introduced into graph network as learning field by Deepwalk algorithm

[5,6]. Random walk algorithm samples the entity nodes in the graph and generates the node sequence , and then uses deep learning technology to map the node sequence to the low-dimensional space vector. The entities' nodes in graph are sampled by random walk algorithm, then the node sequence was created, then then the sequence would be mapped into low-dimensional space by deep learning technology. The more close and relative nodes, the more similarity of its space vector. Node2Vec proposed a random walk algorithm with bias by altering the sequence of random walk. Given the current vertex v, the probability of the next vertex x, accessed by random walk was:

$$P(c_i = x \mid c_{i\text{-}1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, if\,(v,x) \in E \\ 0, otherwise \end{cases} \tag{3}$$

Where the unnormalized probability of π_{vx} and Z is the normalized constant. Assuming that the current random walk passes through the vertex t to the vertex v, the relationship between the unnormalized probability and the weight is:

$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx} \tag{4}$$

Where w_{vx} is the weight of edge between nodes v and x , and the coefficient $\alpha_{pq}(t,x)$ is as follows:

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & if\ d_{tx} = 0. \\ 1 & if\ d_{tx} = 1. \\ \frac{1}{q} & if\ d_{tx} = 2. \end{cases} \tag{5}$$

Where d_{tx} represents the shortest distance between the previous node t and the next node x . Through the parameters p and q, both Depth-First-Search (DFS) and Breadth-First-Search (BFS) can be taken into account. BFS tends to walk around the initial node, which can reflect the microscopic characteristics of a node's neighboring. While DFS tends to move further away from the initial node, which can reflect the macroscopic characteristics of a node.

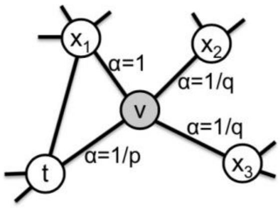

Figure 1 Node2vec algorithm

In general, we do not know the value of w_{vx}, Node2Vec algorithm initial node v and x weight to 1, which caused all the attributes have the same weight, the recommendation result is also susceptible by unrelated properties.

3. Knowledge Graph with weight recommendation algorithm based on collaborative filtering

3.1 Method

The proposed is based on the information of user-item rating matrix information fused into the knowledge graph to form a knowledge graph with weight to realize personal recommendation. The proposed utilizes the strength of knowledge graph which fused the features of semantics, context and heterogeneous information. Obtaining the similarity between item entities according to collaborative filtering recommendation algorithm. They would be more easily to find each other during random walking while the two item entities were alike, which means the edge connecting the two item entities owned bigger weight.

According to the above, the proposed built a knowledge graph with weight based on the original. Knowledge graph with weight was embedded into a low-dimensional space according to Node2Vec algorithm, then recommending by similarity between item entities. The proposed has the following advantages: a good explanation of knowledge graph after fusing the collaborative filtering recommendation algorithm, a higher accuracy and covering of recommendation results, a less influence of irrelevant attributes on recommendation results due to the low weight of irrelevant attributes which are hard to access during random walking.

We supposed there are m as number of user entities $U = (U_1, U_2, \cdots, U_m)$, n as number of item entities $I = (I_1, I_2, \cdots, I_n)$, user-item rating matrix is R_{m*n} , where R_{i*j} is the rating of user U_i on item I_j. According to the above formula (2) calculating the cosine similarity of the item entity vector. For item entity i , select the top k item entities with highest similarity into a set $S_i = \{S_i^1, S_i^2, \cdots, S_i^k\}$, where S_i^j is the j th counterpart of item i, the T attributes entities in the knowledge graph, which corresponding to the item entity i, is $P_i = \{P_i^1, P_i^2, \cdots, P_i^T\}$.

According to TF-IDF , we fused the calculation of similarity into the 'word frequency', then we proposed STF-IDF method to calculate the weight between attributes and item entities. The word frequency weight of specific attributes P of item entity i is as follow:

$$N_{i,p} = \sum_{j=1, p \in P_i}^{k} \frac{e^{sim(i,j)}}{n_{i,j}} \tag{6}$$

Where $sim(i, j)$ is the similarity between item i and item j , $n_{i,j}$ is the number of items i and item j which have the same attributes. The larger $sim(i, j)$, the higher similarity and the greater the weight of the same attribute large. The larger $n_{i,j}$, the smaller weight, When the similarity of two items is 0 and the same attribute is n , the weight of each attribute is $\frac{1}{n}$. The similarity word frequency STF after normalizing is as follow:

$$STF_{i,p} = \frac{N_{i,p}}{\sum_{t \in P_i} N_{i,t}} \tag{7}$$

The reverse set frequency is:

$$IDF_{i,p} = \log \frac{k}{f_{i,p} + 1} \tag{8}$$

Where $f_{i,p}$ is the probability that the item entity k appearing in the k similar item entities of attributes p, k is the number of similar sets, the larger $f_{i,p}$, the smaller $IDF_{i,p}$

The final weight of the attribute is:

$$W_{i,p} = STF_{i,p} * IDF_{i,p} \tag{9}$$

The larger $W_{i,p}$, the greater weight of attribute p in the item entity i, which means it can be a representative of this entity and vice versa. Incorporating the weight of item attributes into formula (4) makes the Node2Vec algorithm more towards nodes with heavier weights during random walking, thereby reducing the influence of irrelevant weights and improving the recommendation effect.

3.2 Algorithm design

Input: rating matrix data set D, knowledge graph (KG) output: top-N recommendation list 1 Using the score matrix data set D calculates the similarity between item entities. 2 Selecting its Top-k similar item entities for each item entity. 3 Using the STF-IDF method to calculate the weight between item entities and attributes to form a knowledge graph with weight. 4 Embedding the entities of knowledge graph with weight into low-dimensional vectors through Node2Vec algorithm. 5 Generating top-N recommendation list according to vector similarity.

4. Summary

This paper proposed a weighted knowledge graph recommendation algorithm for solving the problem that the traditional knowledge graph algorithm as heterogeneous information network for recommendation was easily influenced by irrelevant attributes. By integrating collaborative filtering information into the knowledge graph in a weighted manner, the recommendation effect is improved. Since the algorithm in this paper can only calculate the weight of an attribute for an item, and the knowledge graph includes not only the relationship between the item and the attribute, but also the relationship between the attribute and the attribute, the next step will try to improve the algorithm to calculate the weight between the attribute and the attribute.

However, the proposed just fit for calculating the weight of attribute entities to item entities, the relationship between attribute entities will be our further work.

Acknowledgments

This research is supported by the research fund for humanities and social sciences of the ministry of education under grant No.19XJA910001.

References

[1] Wang H,Zhang F,Xie X,et al.DKN:deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, 2018.1835–1844.
[2] Zhang F, Yuan N J, Lian D, et al. Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 353–362
[3] Wang H, Zhang F, Wang J, et al. RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management,2018. 417–426
[4] Dalal N , Triggs B . Histograms of Oriented Gradients for Human Detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE, 2005.
[5] Palumbo E , Rizzo G , Troncy R , et al. Knowledge Graph Embeddings with node2vec for Item Recommendation: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers[M]// The Semantic Web: ESWC 2018 Satellite Events. 2018.
[6] Perozzi B , Al-Rfou R , Skiena S. DeepWalk: online learning of social representations[C]. ACM SIGKDD International Conference on Know ledge Discovery and Data M ining,ACM,2014: 701-710.