Interpretable Recommendation System Based on Knowledge Map Features

Hongyu Chen¹, Jingyang Liu² and Mao Yang³

¹Zhongbei University, Taiyuan, Shanxi, 030051, China
²Harbin University of Technology, Harbin, Heilongjiang, 150080, China
³Southwest University of Science and Technology, Mianyang, Sichuan, 610000, China

Abstract. At present, it can be explained that the recommendation system is mostly designed for a specific recommendation model, and its scalability is weak. It is not enough for the emerging recommendation models, such as complex and mixed models with deep neural networks. Knowledge maps as a highly readable external knowledge carrier provide great possibilities for improving the ability of algorithms to interpret. In essence, the knowledge map is intended to describe the various entities or concepts and their relationships that exist in the real world, which constitute a huge semantic network diagram, nodes represent entities or concepts, and edges are composed of attributes or relationships. Based on the current research status, this paper analyzes and studies the concept of knowledge map, the combination of knowledge map and recommendation system, the object of interpretability and the application of knowledge map in interpretability model, combined with relevant recommendation models. The knowledge map, open up the relationship between the media, flexibly choose the most suitable medium according to the specific situation to recommend and explain the user.

1. Introduction

The essence of the recommendation question is to replace the user's evaluation of items that they have never touched. The recommendation system is generally divided into two types of rating predictions and click rate estimates. The score prediction refers to predicting the user's evaluation of the item. For example, in the movie recommendation, the system needs to predict the user's rating of the movie, and based on this, push the movie that the user may like. In this scenario, the data we often use is the user's rating data on movies that have been viewed in history. This information can express the user's preference for the movie, so it is also called explicit feedback. The click rate estimation refers to predicting whether the user clicks on the item. For example, in the news recommendation, the system needs to predict the probability that the user clicks on a certain news to optimize the recommendation scheme. This information can only express the user's behavior characteristics, but does not reflect the user's preferences. Degree, so this information is also called implicit feedback.

Traditional recommendation systems use explicit/implicit information as input for prediction, and there are two main problems. The first is the sparsity problem. In actual scenarios, the interaction information between users and items is often very sparse. In movie recommendations, there are often tens of thousands of movies, but there are often only dozens of movies that users have overplayed. Using so few observations to predict a large amount of unknown information can greatly increase the risk of overfitting. The second is the cold start problem. For newly added users or items, there is no
corresponding historical information, so it is difficult to accurately model and recommend. A common idea to solve the sparsity and cold start problems is to introduce some additional information as input in the recommendation algorithm. Auxiliary information can enrich the description of users and items, enhance the mining ability of recommendation algorithms, and effectively compensate for the sparse or missing interaction information.

2. Knowledge Map

In recent years, as a new auxiliary information, knowledge map has gradually attracted the attention of scholars. The Knowledge Map was officially proposed by Google on May 17, 2012. The original intention was to improve the search engine's ability to improve the user's search quality and search experience. With the development and application of artificial intelligence technology, knowledge map has gradually become one of the key technologies, and has been widely used in the fields of intelligent search, intelligent question and answer, personalized recommendation, content distribution and so on.

The knowledge map contains rich semantic associations between entities, providing a potential source of auxiliary information for the recommendation system. Introducing the knowledge map into the recommendation system can bring the following characteristics to the recommendation system:

- **Accuracy:** Knowledge maps introduce more semantic relationships to items, allowing deeper discovery of user interests.
- **Diversity:** Linking types through different relationships in the knowledge map is conducive to the divergence of recommendation results.
- **Interpretable:** The knowledge map can connect the user's history and recommendation results, thereby improving the user's satisfaction and acceptance of the recommendation results, and enhancing the user's trust in the recommendation system.

The knowledge map contains various nodes such as entities, semantic classes, attributes, and relationships. An entity refers to something that is distinguishable and independent, such as an individual, a city, a certain plant, etc., a commodity, etc. The entity is the most basic element in the knowledge map, different there are different relationships between entities.

Semantic classes are collections of entities with the same characteristics, such as countries, nations, books, computers, etc. Concepts mainly refer to collections, categories, object types, types of things, such as characters, geography, and so on. An attribute refers to an attribute value that points to it from an entity. Different attribute types correspond to edges of different types of attributes. The attribute value mainly refers to the value of the specified attribute of the object. A relationship is formalized as a function that maps k points to a Boolean value. On the knowledge map, the relationship is a function that maps k graph nodes (entities, semantic classes, attribute values) to Boolean values. A triple is a general representation of a knowledge map. Its basic forms mainly include (entity1-relationship-entity2) and (entity-attribute-value).

3. Knowledge Map and Recommendation System Combination

3.1. Feature-based recommendation method

The feature-based recommendation method mainly extracts the attributes of users and items from the knowledge map as features, and puts them into traditional models, such as FM model, LR model, and so on. This is not specifically for knowledge map design, and it is not possible to introduce relationship features.

3.2. Path-based recommendation method

A path-based recommendation method that treats a knowledge map as a heterogeneous information network and then constructs meta-path or meta-graph-based features between items. Simply put, meta-path is a specific path connecting two entities, such as "actor->movie->director->movie->actor". This meta-path can connect two actors, so it can be regarded as a way to explore the potential relationships between actors. The advantage of this type of method is that it fully and intuitively utilizes the network
structure of the knowledge map. The disadvantage is that you need to manually design meta-path or meta-graph, which is difficult to achieve optimal in practice. At the same time, this method cannot be in the entity. Applies to scenes in the same domain (such as news recommendations) because we can't predefine meta-path or meta-graph for such scenes.

3.3. Knowledge map feature learning

The knowledge map feature learning learns a low-dimensional vector for each entity and relationship in the knowledge map, while maintaining the original structure or semantic information in the graph. In general, there are two types of model classification for knowledge of feature map learning: distance-based translation models and semantic-based matching models.

3.3.1. Distance-based translation model

This type of model uses a distance-based scoring function to evaluate the probability of a triple, and the tail node as a result of a head node and relationship translation. Representatives of such methods are TransE, TransH, TransR, and the like.

![Distance-based translation model](image1)

The basic ideas of the above three methods are the same. This article uses TransE as an example to introduce the core ideas of these methods. In space, the head node h, the relationship r, and the tail node t of the triple have corresponding vectors. If the result of h + r is closer to t, then we think that these vectors can represent the entities in the knowledge map well. And relationship.

3.3.2. Semantic-based matching model

The semantic-based matching model uses the similarity-based scoring function to evaluate the probability of the triples, and maps the entities and relationships into the implicit semantic space for similarity metrics. Representatives of such methods are SME, NTN, MLP, NAM, and the like.

![Semantic-based matching model](image2)

The core of the above method is to construct a two-class model, and input h, r and t into the network. If (h, r, t) exists in the knowledge map, the probability of approaching 1 should be obtained. If it does not exist, it should be Get a probability of approaching zero.
4. Interpretable Goals

The most common view about interpreting machine learning goals is to increase the transparency of machine learning methods. Taking the deep learning method as an example, many people regard it as a black box when debugging deep neural networks. We can only understand the input and output of the black box, it is difficult to understand the working principle in the black box. This causes problems such as unpredictable and difficult to debug machine learning model results, which ultimately affects the in-depth understanding of the machine learning model and the further improvement of the results. Another goal is to increase people's trust in machine learning algorithms. In key areas such as medical, financial, military, and political, the results of machine learning have a great impact, while explaining machine learning can help decision makers decide whether to trust the results of machine learning.

When we look at the user, we find that the explanation not only helps us improve the understanding of the model, but also its readability is critical. If the explanation is too complicated, it takes time and effort to understand and requires a strong knowledge of machine learning, which runs counter to the original intention of enhancing understanding. In some areas, the results need to be presented to the average user. For example, recommend a book to the user. If you can explain to the user why you can recommend this book with easy-to-understand explanation, you can greatly improve the effectiveness of the recommendation and even the persuasiveness of the recommendation. This is true for both users and advertisers. Important application significance, but the current academic community has less discussion on this aspect. This paper argues that the inclusion of users in more conceivable machine learning considerations will greatly enhance the application and research value of interpretable methods.

The goals of readability, effectiveness, and persuasiveness are collectively referred to as quality of interpretation. The goals that can explain machine learning can also be grouped into two broad categories: model interpretability (model-oriented) and interpretation quality (user-oriented), which complement each other and constrain each other. Studies have shown that increasing transparency helps to increase the persuasiveness of interpretation; transparency and readability need to be balanced, and effectiveness is designed to help users make the decisions that are best for them, which is also contradictory to persuading users to accept certain decisions. At the office. How to balance the different goals depends on the specific application scenarios.

5. Interpretable Recommended Model

5.1. Interpretable recommended process

The process that can be explained is mainly related to several key elements in the recommendation: user set \( U \), item set \( V \), interpreted recommendation system \( f(u, v) \), recommended item set \( V' \) of recommendation system, interpretation module and its output Interpretation \( z \). The most common one is post-processing. The post-processing method is explained after the recommendation result has been given. The content of the explanation is not affected by the recommendation system \( f(u, v) \). Even if a recommendation system is changed, as long as the same user and item are given, the explanation is the same. This method is mainly optimized for interpretation quality, but the model is poorly interpreted and relatively easy to implement. The main application scenarios include advertising e-commerce platform, news, music, movie recommendation and so on.

The second process that can explain the recommendation is embedded. The embedded approach incorporates the interpretation module into the construction of the recommendation system. The interpretation module often selects the characteristics of the item and selects the item that has the greatest impact on the accuracy of the recommendation as an explanation. The features of the item used for explanation here are often phrases, sentences or pictures. Compared with post-processing, the embedded method has high model interpretability, but it is difficult to ensure the quality of interpretation. For example, it is difficult to ensure the consistency and consistency between interpretations, so it is more suitable for researchers and algorithm developers.

Most of the interpretation modules of the embedded method are shallow, such as the topic model used in RecSy's paper "Hidden factors and hidden topics: Understanding rating dimensions with review
text", SIGIR's paper "Explicit factor models for explainable recommendation based" The matrix decomposition used in on phrase-level sentiment analysis, and the single-layer attention network used in the WWW paper "Neural attentional rating regression with review-level explanations".

5.2. Depth interpretable recommended model
In the paper "Explainable Recommendation through Attentive Multi-View Learning", the initial structure of the deep interpretable network was constructed by using Microsoft Concept Graph, and the parameters of each layer in the depth interpretable network were optimized by Attentive Multi-View Learning. Not only can the recommendation accuracy and usefulness be improved, but also the user's hierarchical interest can be automatically modeled in an unsupervised manner. For example, we can know whether users are only interested in sushi (low-level features) or in overall material (high-level features). The model framework is shown below.

![Figure 3. Depth explains the recommended model](image)

5.3. Encapsulated interpretable method
Post-processing and embedded processes focus on quality and one on model interpretation. The encapsulated approach does not require changes to the existing recommendation system, but places the interpretation module and the recommendation system in a relatively equal position, and the interpretation module generates model-related interpretations by interacting with the recommendation system. This can not only freely control the quality of interpretation, but also ensure the interpretability of the model. At the same time, it does not need to design different interpretation methods for different recommendation models like the embedded method. It is a good way to coordinate model interpretability and model quality. Method.

![Figure 4. Illustrates the recommended packaged process](image)
The encapsulated interpretable method is implemented using reinforcement learning. Specifically, it is to use the enhanced learning framework shown in the figure below to explain any recommended model, which can ensure the interpretability of the model and the quality of interpretation. In this framework, the recommended recommendation model is part of the environment. There are two agents in the framework, in which the agent 1 is responsible for generating an explanation, and the agent 2 is responsible for predicting the output of the interpreted recommendation model by using the interpretation (the user scores the item). These two agents treat the model that needs to be interpreted as a black box, and the rewards obtained by interacting with the environment determine the direction of optimization.

![Figure 5. Illustrates the recommended enhanced learning framework](image)

Here, the reward given by the environment is determined by two aspects. If the agent can use the interpretation to accurately predict (impersonate) the results of the interpreted recommendation model, it is rewarded, which reflects the ability to interpret the recommended model. If the explanation given by the agent is refined, coherent, and readable, it will be rewarded, in order to improve the quality of the interpretation itself. Such a framework is applicable to any recommended model, while the explanatory power and interpretation quality are high.

6. Conclusion

As a recommended area to be explored less, it can be explained that many aspects of the recommendation are worth studying and exploring. Knowledge maps as a highly readable external knowledge carrier provide great possibilities for improving the ability of algorithms to interpret. For example, knowledge maps can be used to open up associations between different media. The recommended explanations generated by the existing interpretable recommendations are often limited to one of the articles as the medium, the user as the medium, or the feature as the medium. The association between the three types of media is not enough. If you can use the knowledge map, open up the relationship between these three types of media, and flexibly choose the most suitable medium to recommend and explain the user according to the specific situation. In an era that can explain the increasing importance of artificial intelligence, combining knowledge maps with deep learning can be a very promising direction.

References

[1] Facing the cold start problem in recommender systems [J]. Blerina Lika, Kostas Kolomvatsos, Stathes Hadjieffthymiades. Expert Systems With Applications. 2014 (4)

[2] Comparing context-aware recommender systems in terms of accuracy and diversity [J]. Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione. User Modeling and User-Adapted Interaction. 2014 (1-2)

[3] Hybreek: A software framework for developing context-aware hybrid recommender systems[J]. Tim Hussein, Timm Linder, Werner Gaulke, Jürgen Ziegler. User Modeling and User-Adapted Interaction. 2014 (1-2)
[4] Summary of Recommendation System for Knowledge Mapping [J]. Chang Liang, Zhang Weitao, Gu Tianlong, Sun Wenping, Bin Chenzhong. Journal of Intelligent Systems. 2019 (02)

[5] Knowledge map to realize the intelligentization of big data applications [J]. Liu Ruibao. Software and Integrated Circuits. 2018 (08)

[6] International Data Mining Research Hotspots and Frontier Visualization Analysis [J]. Zhao Rongying, Yu Bo. Modern Intelligence. 2018 (06)

[7] Interpretation of the secret knowledge map behind the map [J]. Technical House. Computer enthusiasts. 2017 (15)

[8] Progress in knowledge mapping learning and reasoning [J]. Wu Yunbing, Yang Fan, Lai Guohua, Lin Kaibiao. Small microcomputer system. 2016 (09)

[9] Research on Personalized Learning Resources Recommendation Based on Knowledge Mapping [J]. Huang Huasheng. Software Engineering. 2018 (10)

[10] Research on User Interest Discovery and Personalized Recommendation Based on Social Labeling [J]. Wang Xiaotong, Zhao Jing, Xu Zuoning. Modern Intelligence. 2018 (07)

[11] Interpretable study of personalized recommendations [D]. Zhang Yongfeng. Tsinghua University, 2016