Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/69/20241529

Analysis of recommendation systems based on knowledge graphs

Ziwei Zhou

School of Computing and Data Science, Xiamen University Malaysia, Sepang, Selangor Darul Ehsan, Malaysia, 43900

swe2109551@xmu.edu.my

Abstract. Due to the rapid development of internet technology, recommendation systems have played a crucial role in improving user experience and enhancing user retention. Knowledge graphs (KG), a technique capable of capturing complex semantic relationships and contextual information, are gradually included in recommendation systems to augment their accuracy and intelligence. This paper reviews the application of knowledge graphs in recommendation systems, analyzing their unique advantages in handling user-item relationships. This paper comprehensively analyzes embedding methods based on tensor decomposition and translation, which primarily designed for static knowledge graphs. This study thoroughly explains the advantages and disadvantages of these methods in practical application. Moreover, this paper discusses the challenges faced by dynamic knowledge graphs, such as temporal data processing, real-time updates, and inference, proposing potential solutions and future research areas. Integrating knowledge graphs with machine learning techniques enhances the ability of social media recommendation systems to understand user preferences and deliver highly personalized recommendations effectively. This study provides theoretical support and practical guidance for the implementing of knowledge graphs in recommendation systems, holding significant academic value and practical significance.

Keywords: knowledge graphs, machine learning, recommendation systems

1. Introduction

With the advancement of internet technology, this world has witnessed an era of information explosion. The abundance of online content, such as products, movies, and news, not only provides users with a wide range of options but also leads to the problem of information overload. This is evident when users attempting to quickly and effectively sift through a substantial volume of information to find what they requires. The proliferation of information in the internet era has become an acute issue requiring immediate attention.

The emergence of recommendation systems helps enhance users experience by providing more accurate information. Recommendation systems can automatically assist users in discovering preferred information from a large volume of data, offering more personalized data services. Traditional methods for developing recommendation systems include content-based recommendation systems and collaborative filtering-based (CF-based) recommendation systems. However, both options have significant disadvantages [1].

DOI: 10.54254/2755-2721/69/20241529

To address the limitations of traditional recommendation systems, some scholars have introduced knowledge graphs (KG) in recommendation systems. A knowledge graph is a database that represents the relationships between entities using a graph structure [2]. It facilitates researchers better understand and organize information and knowledge. In recommendation systems, the interconnections among various types in the knowledge graph help improve the accuracy of suggestions and increase the diversity of recommended items. What's more, KG enhances the transparency of recommendation algorithms.

This paper aims to review the current status and advancements in the application of KG in recommendation systems. This research will introduce the basic principles and methods of knowledge graph technology. Furthermore, it will discuss the current challenges and obstacles, and offers both theoretical foundations and practical insights on the utilization of knowledge graphs in recommendation systems.

2. Research Background

2.1. Classification of Recommendation Systems

The concept of recommendation systems was introduced by Resnick in 1994, since then, it has been widely used as a subclass of information filtering systems [3]. The main purpose of recommendation systems is to suggest items that users might be interested in. These items or content can include movies, music, books, news articles, products, etc.

Current recommendation algorithms fall into three categories: content-based, collaborative filtering-based (CF-based), and hybrid.

Content-based algorithms analyze item characteristics to recommend similar items to users. While effective, they often lack diversity and fail to meet varied user needs.

CF-based algorithms use user-item interactions, split into user-based, item-based, and model-based recommendations [4]. User-based methods find similar users to predict preferences but struggle with large datasets and cold starts. Item-based methods calculate item similarities, offering low complexity but less personalization. Model-based methods, using techniques like Probabilistic Matrix Factorization (PMF) and Singular Value Decomposition (SVD), predict user behavior effectively but require substantial data and resources [5-6].

Hybrid algorithms combine different techniques to leverage their strengths and mitigate weaknesses. Pre-fusion methods merge independent recommendation lists, post-fusion presents lists separately, and inter-fusion integrates results using advanced algorithms. While enhancing accuracy and coverage, hybrids face challenges in determining optimal algorithm weights and ensuring data compatibility.

To address the limitations of traditional systems, researchers are increasingly incorporating knowledge graph technology to enhance recommendation performance.

2.2. Knowledge Graphs and Their Advantages

Knowledge graphs are a structured method of representing knowledge, used for organizing and managing vast amounts of information and knowledge. They represent relationships between entities in a graphical structure, enabling computers to better understand and infer knowledge. In 2012, Google introduced the concept of knowledge graphs to identify and disambiguate entities in text, enriching search results with semantic summaries, and providing links to related entities in exploratory searches. The basic unit of a KG is the "entity-relation-entity" triplet, defined as $G = (E, R, S)$, in which E represents entities in the knowledge base that correspond to real-world objects or concepts, such as people or products. Each entity has a unique identifier and may possess multiple attributes to describe its characteristics, such as text, numbers, or dates. R represents relationships, which denote connections or associations between entities. Relationships describe certain connections or interactions between entities, such as "owns," "located_in," or "is." S represents the set of triplets in the knowledge base. Each triplet consists of a head entity, a relationship, and a tail entity.

DOI: 10.54254/2755-2721/69/20241529

Knowledge graphs provide rich semantic information that helps recommendation systems better understand the relationships between users and items. This allows for more accurate capturing of user interests and needs, enhancing the accuracy and personalization of recommendations. Additionally, knowledge graphs assist in addressing the cold start problem by using related information to provide personalized recommendations for new users, mitigating the cold start impact. Knowledge graphs also offer better explanation and transparency for recommendation systems.

3. Methods for Knowledge Graph-Based Recommendation Systems

The embedding approach is one of the most representative methods for constructing knowledge graph-based recommendation systems. Embedding-based methods preprocess knowledge graphs using Knowledge Graph Embedding (KGE) algorithms, mapping entities and relationships from the knowledge graph to continuous vectors in a low-dimensional vector space (i.e., embeddings) to capture semantic associations between entities. Machine-learning models can effectively learn from these feature vectors. There are two widely used triplet fact embedding methods created by researchers.

3.1. Tensor Factorization Based Methods

Tensor is a general term for high-dimensional arrays, and tensor factorization is the process of decomposing a high-dimensional array into multiple lower-dimensional matrices [7]. This method represents triplets (h, r, t) in a knowledge graph as a three-dimensional tensor. By decomposing this tensor, high-dimensional data is mapped to a lower-dimensional space, capturing complex interactions between entities and relationships. Specifically, each triplet (h, r, t) in the knowledge graph corresponds to an element in the tensor, representing the weight of the connection between head entity h and tail entity t through relationship r [8]. This section will discuss several common methods.

RESCAL is a representative method of this category. It first constructs a three-dimensional tensor similar to a graph adjacency matrix, and then decomposes the tensor to obtain an asymmetric relationship matrix [9]. Its scoring function can be expressed as:

$$X_k \approx AR_kA^T$$

where X_k represents the three-dimensional tensor, A is an $n \times r$ matrix representing the latent components of entities, r represents the features (or dimensions) of each entity [10], and R is an $r \times r$ asymmetric matrix specifying the interactions between the latent components of entity objects. This matrix can model the interactions between different entities based on specific relationships, revealing the latent semantic features of entities.

RESCAL was one of the earliest tensor decomposition-based methods proposed. It has a simple structure and strong expressive capability, making it well-suited for the high-dimensional and sparse nature of knowledge graph relationships. However, it has high computational costs and limited scalability.

DistMult is a simplified version of the RESCAL model. In DistMult, relationships are represented as diagonal matrices, meaning each relationship only requires a vector representation instead of a full matrix. Because relationships are represented by diagonal matrices, DistMult can only capture symmetric relationships [11]. Its scoring function can be expressed as:

$$f_r(h, t) = h^TM_rt$$

where the diagonal matrix M_r is composed of the vector of relationship r. This scoring function essentially is the sum of the element-wise product of three vectors. Since each relationship only requires a vector representation, the number of parameters in DistMult is less than models that need a full matrix representation for relationships. Consequently, DistMult has lower computational complexity and higher computational efficiency compared to other more complex models. Therefore, it is rather easy to handle large-scale knowledge graphs. However, its most obvious drawback is the inability to model asymmetric relationships.

DOI: 10.54254/2755-2721/69/20241529

HolE (Holographic Embeddings) is another commonly used model that captures complex interactions between entities and relationships by combining the advantages of tensor decomposition and holographic techniques [10]. Specifically, HolE uses circular correlation to model the interactions between entities and relationships. Circular correlation is a method that combines two vectors, h and t, to effectively capturing their higher-order interactions. The calculation of circular correlation is defined as follows:

$$(\mathrm{h} \star \mathrm{t})_{\mathrm{k}} = \sum_{\mathrm{i}} \mathrm{h}_{\mathrm{i}} \mathrm{t}_{(\mathrm{i}+\mathrm{k}) \bmod \mathrm{n}}$$

where n is the dimension of the vectors. Therefore, HolE's scoring function can be defined as:

$$\mathrm{f}_{\mathrm{r}}(\mathrm{h}, \mathrm{t}) = \mathrm{r}^{\mathrm{T}}(\mathrm{h} \star \mathrm{t})$$

HolE accelerates the computation of circular correlation, making it more computationally efficient than traditional tensor decomposition methods. What’s more, since circular correlation is asymmetric, the HolE model can capture asymmetric relationships, representing the complex interactions between entities and relationships. At the same time, the number of parameters in the HolE model is relatively small, and close to DistMult, but it can represent more complex relationships. However, the main drawback of HolE is the implementation of circular correlation being relatively complex, making the model more challenging to implement.

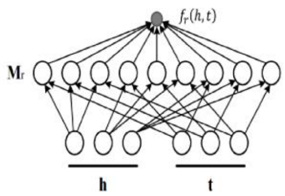

Figure 1. RESCAL Model Diagram [9]

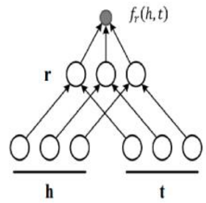

Figure 2. DistMult Model Diagram [11]

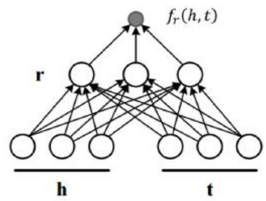

Figure 3. HolE Model Diagram [12]

3.2. Translation-Based Methods

Translation-based methods treat relationships in knowledge graphs as "translation" operations from the head entity to the tail entity. The optimization goal is to make the vector of the head entity plus the relationship vector close to the vector of the tail entity. This approach uses geometric relationships in vector space to represent the associations between entities and relationships. The most representative methods in this category are the Trans series of graph embedding methods, including TransE, TransH, TransR, TransD, and others [13-16].

The basic idea of TransE is to make the vector of the head entity h plus the relationship vector r as close as possible to the vector of the tail entity t (h + r $\approx$ t), using either the l1 or l2 norm to measure "closeness". TransE defines a scoring function to measure this association, defined as:

$$\mathrm{f}_{\mathrm{r}}(\mathrm{h}, \mathrm{t}) = ||\mathrm{h} + \mathrm{r} - \mathrm{t}||_{\mathrm{l}_1/\mathrm{l}_2}$$

DOI: 10.54254/2755-2721/69/20241529

TransE is a simple, intuitive method with high computational efficiency and strong interpretability. However, its drawback is that it is difficult to capture complex many-to-many relationships (e.g., one entity can be associated with multiple entities).

TransH is used to handle one-to-many or many-to-one relationships. It maps the head entity h and tail entity t onto a specific hyperplane w_r, resulting in vectors $h_\perp$ and $t_\perp$. And the relationship vector d_r is considered a transformation operation of the projected entity vectors on the hyperplane w_r. The projected entity vectors are:

$$h_\perp = h - w_r^T h w_r,\ \ t_\perp = t - w_r^T t w_r$$

and the scoring function is:

$$f_r(h,t) = ||h_\perp + d_r - t_\perp||_{l_1/l_2}$$

The advantage of TransH is that it represents relationships on a relation-specific hyperplane, which allows better handling of many-to-many relationships. It projects entities onto different hyperplanes for different relationships, providing more flexibility so that the same entity can have different representations in different contexts. However, TransH also has its drawbacks. Introducing relation-specific hyperplanes increases the computational complexity of the model, making training TransH potentially more challenging.

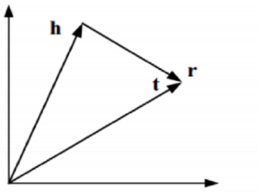

Figure 4. Trans E model diagram [13]

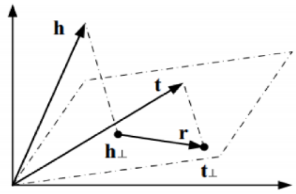

Figure 5. TransH model diagram [14]

TransR considers that entities have different aspects, and different relationships focusing on different aspects, requiring representation in multiple semantic spaces. Therefore, a relation matrix M_r is first established to map entities into the relation-specific space of r, and then optimize the objective $h_r + r \approx t_r$. The projected entity vectors are:

$$h_r = hM_r,\ \ t_r = tM_r$$

And its scoring function is similar to TransE, which is expressed as:

$$f_r(h,t) = ||h_r + r - t_r||_{l_1/l_2}$$

Since entities and relationships are projected and operated in different spaces, TransR can handle many-to-many relationships more effectively, avoiding the oversimplification of entity embeddings. However, TransR also has significant drawbacks. The projection matrix M_r is determined solely by the relationship, ignoring the issue that head entities and tail entities may have different attributes. This single relationship projection may not fully capture the complex relationships between entities.

Based on TransR, TransD further posits that the mapping relationship should be determined by both entities and relations. TransD embeds entities and relations in the knowledge graph through dynamic mapping matrices, with the improved projection matrix determined jointly by entities and relations. All entity objects and relation objects are composed of two vectors: one to represent the semantic information of the object itself and another to dynamically construct the mapping matrix. The projection matrices for head entities and tail entities are represented as:

$$M_{rh} = r_p h_p^T + I,\ \ M_{rt} = r_p t_p^T + I$$

DOI: 10.54254/2755-2721/69/20241529

where I represents the identity matrix, and r_p, h_p, t_p are the mapping vectors used to construct the mapping matrix. The projected entity vectors are represented as:

$$h_\perp = M_{rh}h, \quad t_\perp = M_{rt}t$$

and the scoring function is expressed as:

$$f_r(h,t) = ||h_\perp + d_r - t_\perp||_{l_1/l_2}$$

TransD's use of dual projection matrices allows the model to represent entities and relations at a finer granularity, addressing the issue of head entities and tail entities possibly having different attributes. This makes the model more flexible in capturing the diversity and complexity of entities and relations.

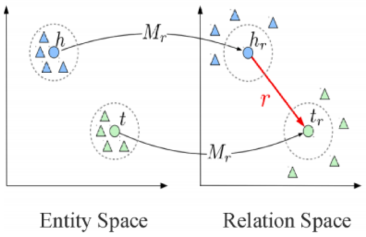

Figure 6. TransR model diagram [15]

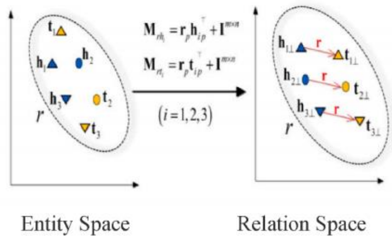

Figure 7. TransD model diagram [16]

3.3. Existing Challenges and Future Research Directions

Despite their achievements, KG-based recommendation systems face challenges. Current methods targeting static knowledge graphs assume a fixed structure during training. Each update, like adding new entities or relationships, requires extensive recalculations, consuming significant computational resources. Constructing KGs often leads to fact omission, especially in rare relationships or long-tail entities, affecting the recommendation system's effectiveness. Future research can explore knowledge graph completion techniques, such as probabilistic learning-based path ranking algorithms (PRA) and representation learning-based methods, to address these gaps.

4. Conclusion

This paper explores recommendation systems based on knowledge graphs, analyzing their advantages and challenges. This article offers a systematic and thorough introduction to tensor decomposition-based and translation-based approaches for embedding knowledge graphs, and explores their suitability and constraints in the context of static knowledge graphs. In addition, this paper addresses the challenges faced by these systems, highlighting critical issues in dynamic recommendation systems, fact omission in knowledge graphs, and the construction and completion of knowledge graphs. Therefore, this paper significantly contributes to the promotion of sustainable development and advancement in this particular domain.

There are many unresolved issues in the research of knowledge graphs in recommendation systems that persist in the future. Important areas for enhancing recommendation system effectiveness involve the development and upkeep of superior dynamic knowledge graphs, the effective completion of knowledge graphs, and the integration of multimodal input into knowledge graphs. Moreover, the collaborative construction and completion of cross-domain knowledge graphs will help provide more diverse and personalized recommendation services.

In general, the potential for using knowledge graphs in recommendation systems is extensive. In the future, recommendation systems will improve their understanding of user needs, offer more accurate and personalized recommendations, and boost user experience and happiness through ongoing technological innovation and research advancements.

References

[1] KWON, Y., LEE, Y., RHU, M., et al. (2021). Tensor casting: co-designing algorithm-architecture for personalized recommendation training. In 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Seoul: IEEE, 235–244.

[2] Zhao, Y., Liu, L., Wang, H., Han, H., & Pei, D. (2023). Survey of Knowledge Graph Recommendation System Research. Journal of Frontiers of Computer Science and Technology (04), 771-791.

[3] Resnick, P., Iacovou, N., Suchak, M., et al. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (pp. 175–186).

[4] Huang, L., Jiang, B., Lv, S., Liu, Y., & Li, D. (2018). Survey on Deep Learning Based Recommender Systems. Chinese Journal of Computer, (07), 1619-1647.

[5] Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. In Proceedings of the 20th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc. (pp. 1257-1264).

[6] Yin, F., Song, Y., & Li, A. (2019). Collaborative filtering algorithm based on singular value decomposition of local optimization and K-means clustering. Journal of Nanjing University of Science and Technology, 43(06), 720-726. DOI: 10.14177/j.cnki.32-1397n.2019.43.06.008.

[7] Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM review, 51(3), 455-500

[8] Liu Wei, Xie Lujun, Zhang Zhihui, & Chen Yafan. (2024). Reasoning of temporal knowledge graphs based on tenser decomposition embedding. Journal of Beijing Information Science and Technology University (Science and Technology Edition), (01), 49-54. doi:10.16508/j.cnki.11-5866/n.2024.01.008.

[9] Nickel, M., Tresp, V., & Kriegel, H. P. (2011, June). A three-way model for collective learning on multi-relational data. In Icml (Vol. 11, No. 10.5555, pp. 3104482-3104584).

[10] Wu, Y., Zhu, D., Liao, X., Zhang, D., & Lin, K. (2017). Knowledge graph reasoning based on paths of tensor factorization. Pattern Recognition and Artificial Intelligence, 30(5), 473-480.

[11] Yang, B., Yih, W. T., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

[12] Nickel, M., Rosasco, L., & Poggio, T. (2016, March). Holographic embeddings of knowledge graphs. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).

[13] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26.

[14] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI conference on artificial intelligence (Vol. 28, No. 1).

[15] Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. 1).

[16] Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 687-696).