Proceedings of CONF-SEML 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA32620

A Fast and Accurate Recommendation System Based on a Simplified GCN Model

Jingzhi Lin

City University of Macau, Macau, China
15920895936@139.com

Abstract. The fast growth of digital content in streaming systems has made the problem of too much information more serious. It also brings big challenges to traditional recommendation methods that use experience-based similarity or simple neural network models. We put forward an improved graph convolution framework for personalized movie recommendations to solve three key problems: poor expandability, sparse data, and difficulties in modeling high-level interactions. First, we build a bipartite graph of users and items based on a big dataset of movie ratings. Then we use a simple multi-layer graph convolution method to get high-level collaborative information through standardized neighborhood spread. Different from standard LightGCN models that use inner-product calculation for scoring, our method combines an MLP prediction module with Batch Normalization, non-linear activation functions and Dropout regularization. This design lets us model the interactions between users and items more clearly and keeps the system structure efficient at the same time. The test results from big interaction data sets show the model has steady convergence and good generalization ability. It also gets competitive results in top-K recommendation tests, and there is no obvious overfitting during the training process. We find that mixing simple graph spread with non-linear prediction can improve both the ability to show data features and recommendation precision in big and sparse data environments. This research provides a framework that can expand well for recommendation systems with better structure. It also lays a good base for the future combination of graph learning and semantic model building.

Keywords: Movie Recommendation, Graph Convolutional Networks, Personalized Recommendation, Bipartite Graph, Deep Learning

1. Introduction

Against the backdrop of the deep integration of big data and the digital economy, digital information is growing at an exponential rate. When faced with massive volumes of content, users often struggle to efficiently obtain information that truly aligns with their personal interests. Personalized recommendation systems have thus become a critical infrastructure for alleviating the problem of information overload [1]. Not only do recommendation systems undertake the technical functions of information screening and precise distribution, but they also serve as a key driver for platforms to improve user retention, enhance user stickiness and achieve commercial monetization. In the

DOI: 10.54254/2755-2721/2026.BA32620

cultural and entertainment sector, movie recommendation systems, as an important application of personalized recommendation technology, have gradually become the core technical support connecting users and the content ecosystem with the digital transformation of the global film industry and the rapid development of streaming platforms. The continuous release of massive film and television resources and the increasing segmentation of user interests have made traditional distribution methods based on popular rankings or coarse-grained classification unable to meet the demands of refined recommendation. The film industry is in urgent need of building more efficient and intelligent recommendation models to achieve precise matching between content supply and individual needs.

With the continuous expansion of content scale on global streaming platforms and the diversified development of movie viewing scenarios, the recommendation demands of the film industry have also been upgraded [2]. On the one hand, the mode of content production and dissemination has shifted from a theater-centric single channel to an online platform-centric multi-supply structure, with film resources characterized by high-frequency updates, diverse genres and cross-media integration. On the other hand, users' movie viewing behaviors have gradually shown a trend of contextualization and personalization, expanding from single genre preferences to multi-dimensional demands such as emotion-oriented choices, social attributes and cross-domain interest linkage. Movie recommendation systems need not only solve the fundamental problem of "difficulty in finding movies", but also support the extension of viewing experience, the excavation of deep-seated preferences and cross-scenario content linkage, driving the transformation of the film industry from "content-driven" to "user-driven". In this context, the goal of recommendation systems is no longer limited to simple matching, but to achieve a higher level of individual customization and value mining.

However, traditional recommendation methods have gradually exposed obvious limitations in the big data environment. Collaborative filtering methods, which focus on user or item similarity, essentially rely on the statistical average of group preferences, making it difficult to depict individual differences of users in different scenarios. Meanwhile, their performance degrades significantly in scenarios with sparse data and cold start. Content-based recommendation methods can alleviate the cold start problem to a certain extent, but they over-rely on explicit feature tags, which makes it hard to explore the potential interest structure of users and easily leads to homogeneous recommendation results. As the scale of users and interaction data reaches tens of millions or even billions, traditional similarity calculation is faced with the problems of surging time complexity and declining response efficiency, which restricts both recommendation accuracy and system scalability.

Even if some studies have introduced deep models such as multi-layer perceptron, their structures often ignore the high-order relationships and complex topological information in user-item interaction data, making it difficult to perform effective modeling in complex network environments. Therefore, under the multiple challenges of expanding data scale, complicated structural relationships and deepened personalized demands, traditional methods can no longer meet the comprehensive requirements of modern movie recommendation systems for accuracy, efficiency and scalability.

Against the development of modern technology, Graph Neural Networks have grown fast, and Graph Convolutional Networks (GCN) [3,4] in particular have offered new and useful theoretical methods for the structured way to build recommendation systems. Movie recommendation can be seen as a user-item interaction graph in its basic form, and the nodes in this graph hold a lot of high-order connections and topological structure information. GCN can collect neighbor information and

DOI: 10.54254/2755-2721/2026.BA32620

spread feature data on graph structures, and it can catch the relational features of local and global parts well by doing so. In this way, it can realize the in-depth learning of feature representation for those complex interactive networks in recommendation systems. This way of building models based on graph structures has two key advantages: it can ease the problem of sparse data in recommendation systems, and it also improves how well the model can show high-order connections between different elements. What's more, it can make the final recommendation results much more accurate and stable in practical use. So, to build a recommendation system based on a simplified GCN model is not only fit with the developing direction of structured learning under the present big data condition, but also offers a practical way for making personalized movie recommendations that are efficient and stable. At the same time, it builds a firm technical base for the digital update and smart change of the whole movie industry.

2. Related works

Early movie recommendation systems relied primarily on users' historical interaction behaviors with items, among which collaborative filtering is one of the most widely adopted core methods. Such methods are independent of the explicit attributes of items or user profile information; instead, they generate recommendation results by mining the similarities between users or items based on the user-item interaction matrix. According to the different objects for similarity calculation, collaborative filtering methods are mainly divided into user-based collaborative filtering and item-based collaborative filtering.

User-based collaborative filtering calculates the interest similarity between the target user and other users, selects a group of similar users as neighbors, and recommends items favored by these neighbors but not yet accessed by the target user to them. This method can well capture the common preferences of user groups, and its recommendation results have a certain degree of interpretability, making it suitable for application scenarios where user interests are relatively stable. However, when the user scale is large or the interaction data is highly sparse, the time complexity of user similarity calculation increases significantly, and the dynamic changes of user interests over time will exert a negative impact on recommendation performance. In contrast, item-based collaborative filtering analyzes the similarity of items that are interacted with by the same group of users, and recommends other items highly similar to those the user has historically preferred to them. Since item similarity is relatively stable, this method features better scalability and lower computational overhead in scenarios with the rapid growth of user scale, and also has strong interpretability. Nevertheless, its performance is limited in scenarios with large differences in item attributes or cold start of new items, as new items cannot be integrated into the recommendation system in a timely manner due to the lack of sufficient interaction data. In addition to collaborative filtering ways, content-based recommendation methods work out recommendation results by analyzing the matching relation between the content features of items (such as text, tags and attribute information) and users' historical preference habits. These methods can ease the cold start problem to some degree and realize personalized recommendation with a high degree of matching for users. But the effect of these methods relies heavily on the accuracy and completeness of item attribute descriptions, and they often generate homogeneous recommendation results. This situation restricts the development of more diverse interests for users. As deep learning keeps developing, some researchers have introduced simple neural network models like multi-layer perceptrons (MLPs). These models conduct nonlinear modeling on the latent vectors of users and items, and this way can improve the capability of feature expression. Though these methods have made the recommendation performance better to a certain extent, they usually ignore the high-order structural information

DOI: 10.54254/2755-2721/2026.BA32620

contained in the interaction data between users and items, and their ability to build models for complex relationships is still quite limited.

To describe the complex relationships between users and items in a more effective way, researchers have started to treat recommendation problems as graph structure learning tasks and introduce representation learning methods based on graph structures. DeepWalk [5] and Word2Vec [6] are the typical representative models of such methods. DeepWalk carries out random walks on graph structures, takes node sequences as "word sequences", and uses the Skip-Gram model to learn the low-dimensional vector representations of various nodes. This method can capture the high-order neighbor relationships of nodes without relying on node attributes, and this makes it suitable for graph data that has rich structural information but lacks complete attribute information. But its random walk strategy is more inclined to build models for local structures, and its ability to depict global topological features is limited. Besides, there is still much room for improving its training efficiency when it is applied to large-scale graph data scenarios. Word2Vec was first used in the field of natural language processing, and it learns semantic vector representations through the co-occurrence relationships between words. It has obvious advantages such as high training efficiency and strong transferability. In recommendation systems, some researchers have tried to map user behavior sequences or item sequences into "text", and they do this to learn the potential semantic relationships behind the sequences. But this method only builds models based on linear sequences and does not have the ability to explicitly model the complex relationships in graph structures, so it is hard for it to fully adapt to the structural characteristics of the interaction networks between users and items.

In recent years, graph neural networks, especially graph convolutional networks, have received extensive attention in the field of recommendation systems. By conducting neighbor information aggregation on graph structures, GCNs realize end-to-end learning of node representations, which can capture both local and global structural features simultaneously and exhibit remarkable advantages in modeling the high-order correlation relationships between users and items. For this reason, this study adopts an enhanced graph convolution recommendation model based on LightGCN [7]. First, the model constructs a bipartite graph through the embedding representations of users and items, performs K-layer graph convolution propagation using a symmetric normalized adjacency matrix, and averages the embeddings of each layer to fuse multi-order neighborhood information. On this basis, this study introduces a multi-layer perceptron structure in the prediction layer, which consists of three fully connected layers and combines Batch Normalization, LeakyReLU/ReLU activation functions and Dropout regularization. This design is intended to enhance the nonlinear expressive ability and generalization performance of the model, thereby predicting users' ratings for items more accurately.

3. Dataset and preprocessing

This study uses the MovieLens dataset released by the GroupLens Research Group as the experimental basis for the research [8]. MovieLens is one of the most commonly used public datasets in the field of recommendation systems, and it is collected and sorted out by the GroupLens Research Laboratory of the University of Minnesota. The dataset includes real rating records that users gave to different movies, and the rating scores range from 0.5 to 5.0 points with an interval of 0.5 points each time. It has the clear characteristics of high sparsity and real authenticity, which makes it very fit for testing how well collaborative filtering and graph neural network recommendation algorithms work in practical use. This experiment takes the MovieLens-32M edition of the dataset for the related research work, and this edition has about 32 million rating

DOI: 10.54254/2755-2721/2026.BA32620

records in total. It also includes evaluation data from around 200,000 users on nearly 85,000 different movies (Figure 1). This specific dataset has a bigger overall data size and a more true-to-life sparse data distribution. Because of these advantages, it can better check the model's ability to expand and apply in different situations in the large-scale recommendation application scenarios.

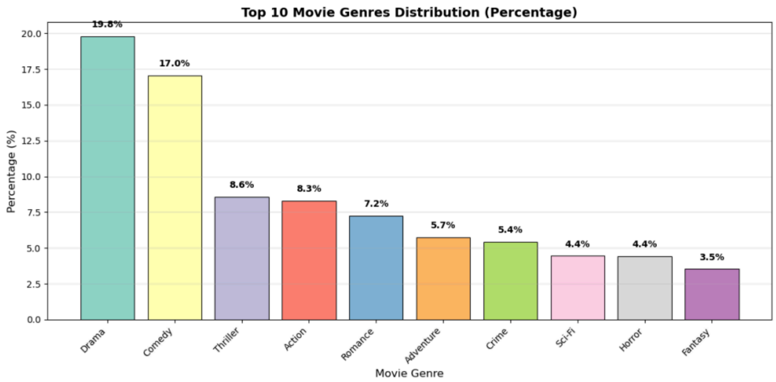

Figure 1. Top 10 movie genres distribution (percentage)

When we build a recommendation model with graph convolutional networks, we do systematic preprocess work on the original rating data of the model. First, we read the rating records of users and movies directly from the relevant dataset. To deal with the problem that user IDs and movie IDs in the original data have discontinuous and scattered distribution, we reassign numbers for them. We map all the users and movies to continuous whole number indexes that start from 0 through this way, and it thus forms a unified and tight index space for the model. After we finish the re-encoding of these indexes, we build the bipartite graph structure of users and movies on the basis of the original rating records we have read before. Each rating record is regarded as a weighted edge between a user node and a movie node, with the edge weight being the corresponding rating value.

By extracting user indices, movie indices and rating information, edge index representation and edge weight representation of the graph are constructed to depict the connection relationships and interaction intensity between nodes. Subsequently, the user-movie interaction matrix is converted into an adjacency matrix form suitable for graph convolutional networks.

The overall symmetric adjacency matrix containing user nodes and movie nodes is constructed by block concatenation of the user-movie interaction matrix and its transpose matrix, thus forming a unified undirected graph structure representation. On this basis, the adjacency matrix is transformed into a sparse representation form to adapt to the storage and computing requirements of large-scale sparse graph data.

Finally, the constructed edge data is randomly divided into a training set, a validation set and a test set in the ratio of 80%, 10% and 10% respectively. A two-stage random segmentation method is adopted in the division process to ensure the mutual independence of each data subset, which is used for model training, parameter tuning and performance evaluation.

4. Model

Based on the classic Light GCN model, this paper proposes an enhanced graph convolution recommendation model with two key improvements. Firstly, the model constructs trainable embedding matrices for user and item nodes respectively and generates initial representations through random initialization. Then, user embedding and item embedding are concatenated to form

DOI: 10.54254/2755-2721/2026.BA32620

a unified node representation matrix; on the basis of the user-item bipartite graph structure, adjacency relationships are normalized to enable multi-layer graph convolution propagation.

The model performs K layers of message passing, where each layer achieves weighted aggregation of neighbor node features through the normalized adjacency matrix to fuse high-order neighborhood information layer by layer. Unlike the standard GCN, no additional feature transformation matrix or nonlinear transformation is introduced in the propagation process, with only the linear propagation mechanism retained. Finally, the propagation results of each layer are concatenated and averaged to obtain the final node embedding representation that integrates multi-order neighborhood information.

The first core improvement is that, unlike the original Light GCN, which uses a simple inner product as the prediction function, this model introduces a MLP structure to perform nonlinear modeling on the concatenated interaction features. The prediction network consists of three fully connected layers and one output layer, with BatchNorm and activation functions applied between layers. The second core improvement is that, to enhance generalization ability, the Dropout mechanism is introduced in each hidden layer to alleviate overfitting; meanwhile, a weighted message passing method based on normalization coefficients is adopted in the graph convolution propagation stage to ensure numerical stability.

In summary, while retaining the lightweight propagation structure of Light GCN, the enhanced model significantly improves the expression ability and prediction flexibility of user-item interaction modeling by introducing two core innovations: a MLP and effective regularization strategies.

5. Results

This study conducted model training and evaluation on the MovieLens dataset, with a total of 2500 training iterations performed. The model's performance on the validation set was evaluated every 200 iterations. In the training process, Mean Squared Error (MSE) was adopted as the loss function, and Recall@20 and Precision@20 were used as the evaluation metrics for recommendation quality.

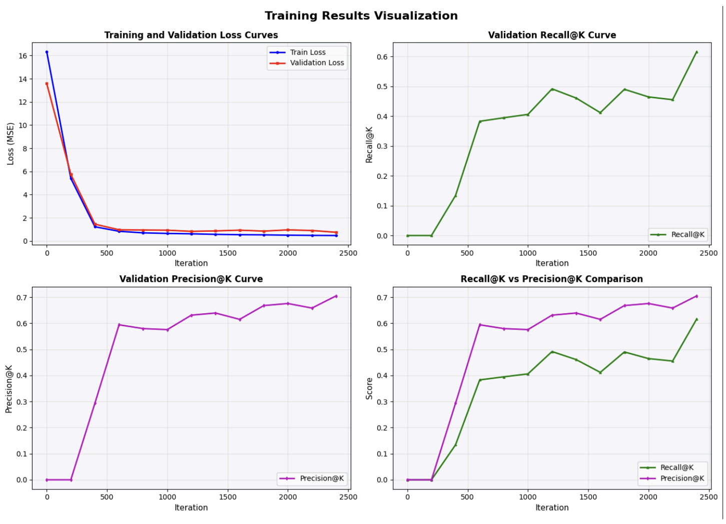

Figure 2. Model training process

We can see the training and validation loss curves in Figure 2, and it shows that as the number of training iterations goes up, the training loss (the blue curve) drops quickly at first and then stays stable, and the model converges little by little. The validation loss (the red curve) also has a fast

DOI: 10.54254/2755-2721/2026.BA32620

falling trend in the early part of the training process, and it becomes stable after about 500 training iterations.

The difference between the training loss and the validation loss is small, and these two curves have a similar changing trend. This shows that there is no clear overfitting happening when the model is being trained. This result comes from the regularization methods that are used in this research, such as weight decay (weight decay=0.01) and exponential learning rate decay (gamma=0.95), and these methods can stop the model from overfitting in an effective way.

We can also find the validation Recall@K curve (the green one) and the validation Precision@K curve (the purple one) in Figure 2. In the first 500 iterations, the Recall@K and Precision@K values on the validation set all stay at a quite low level. After 500 iterations, both of the two metrics start to rise quickly, and they keep getting better with small fluctuations all the time. This means that the model's ability to give recommendations becomes stronger and stronger as the training process moves forward.

These two metrics have a very consistent changing trend during the whole training process. This means that the model not only makes its recommendation accuracy better in the training, but also keeps a good recall ability at the same time.

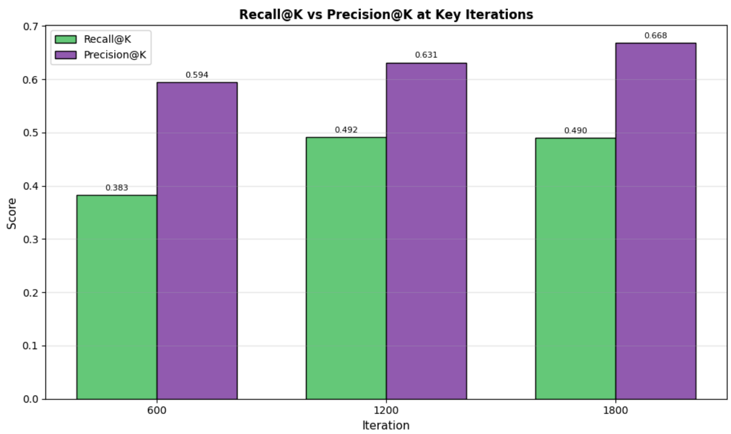

Figure 3. Model training with 600, 1200, 1800 iteration

As can be seen from Figure 3, at the iteration counts of 600, 1200, and 1800, the model's Recall@K and Precision@K remain relatively stable overall and show a positive trend of change.

Specifically, at 600 iterations, Recall@K is 0.383; it rises to 0.492 at 1200 iterations, and slightly drops to 0.490 at 1800 iterations, with minimal overall fluctuation. This indicates that the model's ability to recall relevant items remains stable across different training stages. Meanwhile, Precision@K gradually increases from 0.594 at 600 iterations to 0.631 at 1200 iterations and 0.668 at 1800 iterations, showing a continuous upward trend. This means the model does not experience significant fluctuations or performance degradation while improving recommendation accuracy.

From the perspective of the trend of change during the training process, neither of the two metrics exhibits the phenomenon of "rising first and then dropping significantly", nor do they show large fluctuations or abnormal peaks. Such a performance of stable change and gradual convergence usually indicates that the model is continuously learning effective features without overfitting to the training data.

DOI: 10.54254/2755-2721/2026.BA32620

6. Conclusion

This paper conducts research on recommendation models based on graph convolutional networks, and constructs a user-movie bipartite graph representation framework. On the basis of the Light GCN structure, a multi-layer perceptron and a nonlinear transformation module are introduced to form an enhanced recommendation model. The model realizes the aggregation of high-order interaction information through a multi-layer neighborhood propagation mechanism, and performs nonlinear modeling on the interaction features of users and items by means of a deep fully connected network in the prediction stage, thereby improving the model's expressiveness and prediction flexibility. Compared with traditional collaborative filtering methods, the method proposed in this paper can capture complex correlation relationships at the graph structure level and enhance the feature mapping capability at the prediction level simultaneously. It achieves the combination of "lightweight propagation + deep prediction" in structural design, and carries out structural expansion and practical exploration for the application of graph neural networks in recommendation systems.

But this research has some clear limitations in its design and implementation. First of all, in terms of data use, we only use user rating records to build the interaction graph in this study, and we do not take user attribute information and item semantic features into consideration here. This makes the model mainly depend on collaborative feature signals for the learning of feature representations, and it is hard for the model to find the potential semantic connections and users' real preferences at the content level. In addition, the dataset used in the research has the problem of unbalanced sample distribution: for example, popular movies get much more interaction from users than the long-tail movies that are less popular, and there is a big difference in the number between highly active users and low-active users. Such an unbalanced sample distribution will bring bias towards popular content and exposure-related bias to the model, and this makes the model tend to focus more on the popular content that already exists in the dataset. What's more, the improved model adds a multi-layer nonlinear structure in the prediction step; this structure does help the model have a better ability to express different features, but it also makes the number of parameters and the complexity of calculation go up at the same time. This may bring big challenges to the training cost and the efficiency of inference when the model is used in large-scale application situations. At the same time, the collaborative recommendation models based on graph structure still meet big difficulties in cold start situations. This problem is especially obvious when new users or new items have no interaction records at all, and it is really hard for the model to generate stable feature representations for them in such cases.

The future research work of this topic can focus on the combination of different data types and the use of Large Language Models (LLMs) [9]. First of all, we can use LLMs to do semantic processing on the text information of movies, including the brief introduction of movie plots, the review texts written by users and the tag descriptions of movie features. Then we can combine these high-dimensional semantic feature representations with the feature information from the graph structure, and build a graph recommendation model with enhanced content features [10]. This research method can make up for the shortcomings of using only the collaborative feature signals in the model, and it can also provide extra semantic support for the model when it faces cold start situations. In addition, LLMs can also be used to analyze the content made by users, such as the comments users write and the text records of their historical behavior on the platform. By deeply understanding the language used by users and the semantic meaning behind it, we can make a more detailed description of what users are interested in, and on this basis, we can realize personalized recommendations based on semantic meaning for users. What's more, researchers can also do further

DOI: 10.54254/2755-2721/2026.BA32620

exploration on the combination of parameters or feature representations between graph-based neural networks and LLMs [11]: for example, we can use the constraints from the graph structure to optimize the feature embeddings of users or items that are generated by language models, or we can turn the recommendation task into a natural language inference problem by using a special learning method with prompt words, and thus build a recommendation system with generative features. Additionally, in the actual application of recommendation models, we can combine the recommendation systems with LLMs for conversation to build interactive recommendation systems. These systems can let the model catch the changes of users' preferences in a dynamic way through multi-round conversations with users, and then provide more flexible personalized service for different users in practical use.

From the perspective of the overall development trend of deep learning, recommendation systems are gradually evolving from single-structure models to graph models, generative models and multi-modal fusion models. Graph neural networks excel at depicting structured relationships, while LLMs possess prominent advantages in semantic understanding and generation capabilities; the combination of the two is likely to become a key research direction for recommendation systems in the next stage. At the same time, future research should also focus on the interpretability, fairness and ethical issues of models, and avoid exacerbating algorithmic bias and the filter bubble effect while improving model performance. In general, the work of this paper provides an experimental foundation for the structural enhancement of graph convolutional recommendation models, and also offers theoretical and practical references for subsequent research combining LLMs with graph representation learning.

References

[1] C. N. Sunilkumar, "A review of movie recommendation system: Limitations, survey and challenges, " ELCVIA Electron. Lett. Comput. Vis. Image Anal., vol. 19, no. 3, pp. 18–37, Sep. 2020, doi: 10.5565/rev/elcvia.1232.

[2] R. Lavanya, U. Singh, and V. Tyagi, "A comprehensive survey on movie recommendation systems, " in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Mar. 2021, pp. 532–536. doi: 10.1109/ICAIS50930.2021.9395759.

[3] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review, " Comput. Soc. Netw., vol. 6, no. 1, p. 11, Nov. 2019, doi: 10.1186/s40649-019-0069-y.

[4] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: A systematic review and new directions, " Artif. Intell. Rev., vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.

[5] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations, " in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, in KDD '14. New York, NY, USA: Association for Computing Machinery, Aug. 2014, pp. 701–710. doi: 10.1145/2623330.2623732.

[6] K. W. Church, "Word2Vec, " Nat. Lang. Eng., vol. 23, no. 1, pp. 155–162, Jan. 2017, doi: 10.1017/S1351324916000334.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation, " in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, in SIGIR '20. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 639–648. doi: 10.1145/3397271.3401063.

[8] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context, " ACM Trans Interact Intell Syst, vol. 5, no. 4, p. 19: 1-19: 19, Dec. 2015, doi: 10.1145/2827872.

[9] Z. Zhao et al., "Recommender systems in the era of large language models (LLMs), " IEEE Trans. Knowl. Data Eng., vol. 36, no. 11, pp. 6889–6907, Nov. 2024, doi: 10.1109/TKDE.2024.3392335.

[10] J. Chen, X. Wang, S. Zhao, and Y. Zhang, "Content-enhanced network embedding for academic collaborator recommendation, " Complexity, vol. 2021, no. 1, p. 7035467, Jan. 2021, doi: 10.1155/2021/7035467.

[11] W. Li, Y. Peng, M. Zhang, L. Ding, H. Hu, and L. Shen, "Deep model fusion: A survey, " IEEE Trans. Neural Netw. Learn. Syst., pp. 1–17, 2025, doi: 10.1109/TNNLS.2025.3628666.