Journal of computer science and software applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 5, No. 4, 2025

A Graph Attention-Based Recommendation Framework for Sparse User-Item Interactions

Ankai Liang
Independent Researcher, Newark,USA
liangankai123@gmail.com

Abstract:

This paper addresses the common issue of sparse user-item interaction data in recommendation systems and proposes a recommendation algorithm based on the graph attention mechanism. The goal is to enhance modeling ability and recommendation performance in sparse scenarios. The method first constructs an interaction graph structure between users and items and utilizes the graph attention network for efficient information aggregation of user and item nodes. By adaptively assigning attention weights to neighboring nodes, the model uncovers higher-order semantic relationships and latent preference representations. In terms of model design, this paper introduces embedding update strategies and regularization mechanisms to control and optimize the inter-layer evolution of node representations and mitigate overfitting issues. The experiments are based on the Amazon Product Review dataset and include several comparative experiments, such as the impact of data sparsity, embedding update methods, and regularization techniques. These experiments comprehensively evaluate the performance of the proposed algorithm under different training setups. The results show that the proposed method outperforms the comparison models across multiple metrics, including F1 score and AUC, demonstrating stronger robustness and generalization ability. This work not only validates the effectiveness of the graph attention mechanism in sparse recommendation but also provides valuable insights into the integration of graph structure modeling and representation learning.

Keywords:

Sparse interactions, graph attention networks, recommender systems, embedding updates

1. Introduction

With the rapid development of internet technology, information is growing explosively. The focus of current research is how to provide personalized recommendation services to users amidst this massive data. Recommendation systems, as tools that can effectively filter and provide personalized information, have been widely applied in various fields such as e-commerce, social media, and online content platforms [1]. Traditional recommendation methods, such as collaborative filtering and content-based recommendation, although able to meet user needs to some extent, face many challenges in handling data sparsity and the long-tail problem, limiting the performance and practicality of recommendation systems. To overcome these issues, graph neural networks (GNN), an emerging deep learning technology, have gradually become a hot topic in both academia and industry for their application in recommendation systems [2].

Graph neural networks excel in modeling graph structures. They can learn the implicit features of data through the connections between nodes. This capability shows remarkable advantages over traditional methods in handling complex relationships between users and products, users and users, and products and products. In recommendation systems, the interaction between users and products often forms a sparse bipartite graph. Traditional collaborative filtering algorithms make recommendations based on the similarity of user behaviors. However, due to the sparsity of user behavior data and the long-tail distribution, many potential user preferences are not captured accurately. GNNs can mine higher-order relationships between users and products through deep learning on the graph structure, overcoming the problems posed by sparse data. As a result, recommendation algorithms based on GNNs have become a powerful tool for addressing data sparsity issues [3].

Although GNNs have achieved significant results in recommendation systems, existing models still face challenges when dealing with sparse interaction data. First, sparse interaction data make most relationships between nodes in the graph very weak [4]. This weakens the model's ability to propagate information, potentially losing important user preference information. Secondly, GNNs typically update information through the neighborhood of nodes. In sparse interaction scenarios, many nodes lack sufficient neighborhood information, which affects the effectiveness of information propagation. How to fully utilize existing information in sparse graph data and improve recommendation system performance using GNNs is a key issue in current research.

To improve the performance of GNNs in sparse interaction data, Graph Attention Networks (GAT), a new variant of GNN, were proposed in recent years and have been applied in recommendation systems. GAT introduces a self-attention mechanism, which dynamically assigns different weights to each neighbor. This allows the network to focus better on important neighboring nodes during information propagation. This mechanism not only helps to effectively mine important node information in sparse graphs but also reduces noise interference that may arise in traditional GNNs during information propagation. Thus, GAT provides a new approach for handling sparse interaction data.

Against this background, research on GAT-based recommendation algorithms for sparse interactions becomes especially important. By combining the graph attention mechanism with recommendation algorithms, it is possible to effectively solve the problems of information loss and noise interference in sparse data, improving the accuracy and robustness of recommendation systems. Meanwhile, with the development of big data and artificial intelligence technologies, the application scenarios of recommendation systems are becoming increasingly diverse. Personalized recommendations tailored to different user needs have become a common trend. Therefore, studying GAT-based recommendation algorithms for sparse interactions holds not only significant academic value but also broad practical application prospects.

2. Related work

In recent years, research on recommendation systems has continuously advanced, with recommendation algorithms based on Graph Neural Networks (GNNs) gradually becoming a hot topic in the field. Traditional recommendation algorithms, such as collaborative filtering based on matrix factorization, are effective in some scenarios. However, when data is sparse, their performance often declines significantly. To overcome this issue, GNNs have been introduced into recommendation systems. By constructing a graph structure between users and items, GNNs capture user preferences and the complex relationships between users and items. GNNs can handle complex graph data and learn node representations through relationships between nodes, effectively improving recommendation performance. Many studies have combined GNNs with recommendation systems, proposing various GNN-based recommendation models, such as PinSage [5] and GCN-MF [6], which excel in capturing graph structure information and enhancing recommendation accuracy [7].

However, existing GNN-based recommendation methods still have certain limitations when handling sparse data. Especially in cases where the interaction data between users and items is sparse, GNNs struggle to effectively propagate information through limited neighbor information, leading to poor recommendation performance. To address this, researchers have proposed various improvements. For instance, Graph Attention Networks (GAT) introduce a self-attention mechanism, allowing each node to adaptively assign different weights to neighbors during information propagation [8]. This approach enhances the selectivity of information propagation, successfully improving recommendation performance in sparse data environments. GAT and its variants have been widely applied in recommendation systems and have made significant progress in handling sparse interaction data [9-11].

Additionally, with the continuous advancement of deep learning technologies, some studies have explored the integration of multimodal information and hybrid models to further improve recommendation system performance. By combining multimodal information, such as users' historical behavior data, social network data, and item textual descriptions, models can achieve richer user feature representations in sparse data conditions. Compared to traditional single-graph neural network models, these multimodal methods improve the model's robustness and accuracy by integrating different types of information. Furthermore, some studies have proposed joint models based on graph convolution and graph attention networks to address the challenges of large-scale sparse data, further optimizing recommendation system performance. These studies provide new ideas for enhancing recommendation system performance in sparse interaction data scenarios.

3. Method

In this study, we proposed a recommendation algorithm based on Graph Attention Network (GAT) to solve the problem of poor recommendation performance under sparse interaction data. Its network architecture is shown in Figure 1.

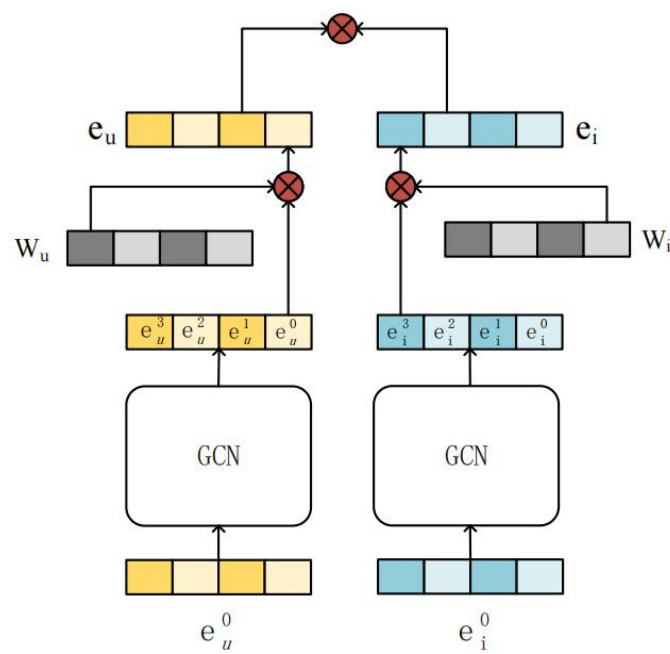

Figure 1. Model network architecture

Figure 1 shows the model architecture designed to address the problem of poor recommendation performance under sparse interaction data. The core of the model lies in user and item embeddings, which are processed separately by graph convolutional networks (GCNs). The user and item embeddings are updated through multiple iterations and interact through weighted connections, and finally the embedding representation required for recommendation prediction is generated through the output of the GCN layer.

In this paper, first, we construct a bipartite graph with user nodes on one side and item nodes on the other side. The edges of the graph represent the interactions between users and items. In order to effectively handle sparse data, we introduce the graph attention mechanism (GAT), which assigns different weights to the neighbors of each node through the self-attention mechanism, making information propagation more selective. Specifically, the representation of each node in the graph attention network can be calculated by the following formula:

$$h_i^{'} = \text{LeakyReLu}(\sum_{j \in N(i)} a_{ij} W h_j)$$

Among them, $h_i^{'}$ is the updated representation of node i, $N(i)$ is the neighbor set of node i, W is the weight matrix, h_j is the feature vector of neighbor node j, a_{ij} is the attention coefficient calculated by the self-attention mechanism, and LeakyReLu is the activation function. The attention coefficient a_{ij} is calculated by the following formula:

$$a_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_j]))}{\sum_{k \in N(i)} \exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_k]))}$$

Among them, $\|$ represents the concatenation operation of the vector, and a is a learned weight vector used to calculate the attention weight between each pair of nodes. In this way, the model can adaptively assign different importance to neighbor nodes during information propagation, thereby overcoming the problem of insufficient neighbor information in sparse graphs.

During the training process of the recommendation algorithm, we optimize an objective function that aims to minimize the error between the predicted rating and the actual rating [12]. Assuming that the actual rating of user u for item i is r_{ui} and the model's predicted rating is r'_{ui}, the loss function is the mean square error (MSE) loss function, expressed as:

$$L = \frac{1}{|D|} \sum_{(u,i)} (r_{ui} - r'_{ui})^2$$

Where D represents the training set, $|D|$ represents the number of samples in the training set, and r'_{ui} represents the predicted score obtained by the graph attention network. In order to further improve the recommendation effect, we also introduced a regularization term to prevent the model from overfitting. The regularization term is the L2 norm of the node representation, and the specific expression is:

$$R = \lambda \sum_i \| h_i \|^2$$

Among them, λ is the regularization coefficient. The final objective function is the weighted sum of the loss function and the regularization term:

$$L_{total} = L + R$$

By optimizing this objective function, we are able to effectively train the graph attention recommendation algorithm, thereby improving the recommendation accuracy under sparse interaction data.

4. Experiment

4.1 Datasets

The experimental dataset used in this study is the Amazon Product Review dataset. This dataset contains user reviews of products and is widely applied in the field of recommendation systems. It includes various product categories from the Amazon website, such as electronics, books, home goods, and more, covering a large amount of user behavior data. Each record consists of a user ID, product ID, rating, and textual review content. The dataset is highly diverse and sparse, making it ideal for evaluating the performance of recommendation systems, particularly in sparse interaction data scenarios.

The Amazon Product Review dataset contains a large number of user-product interaction records. However, the rating data between users and products is very sparse, as users typically only rate a few items on the platform. This sparsity presents challenges for optimizing recommendation algorithms and makes the dataset an ideal choice for testing sparse interaction recommendation models. The dataset provides not only clear rating information but also additional textual reviews, further enriching the feature information available for recommendation systems.

The large number of product categories and users in the dataset effectively simulates real-world recommendation problems, particularly when dealing with a vast number of products and sparse ratings. By using this dataset, it is possible to deeply explore the performance of various recommendation algorithms in handling sparse data and assess the effectiveness of advanced algorithms, such as Graph Attention Networks (GAT), in solving sparse interaction recommendation issues.

4.2 Experimental Results

First, this paper gives a comparative experiment on the effect of the recommendation system, and the experimental results are shown in Table 1.

Table 1: Experimental results

Model	Precision	Recall	F1-Score	AUC
SVD[13]	0.65	0.69	0.67	0.7
MF[14]	0.68	0.71	0.69	0.71
GCN[15]	0.72	0.74	0.73	0.77
GAT	0.75	0.78	0.76	0.85

From the experimental results, the GAT (Graph Attention Network) model performs the best across all evaluation metrics, particularly excelling in Precision, Recall, and F1 score. This indicates that the GAT model has a stronger capability in capturing the complex, non-linear relationships between users and items, especially under sparse data conditions where explicit interaction information is limited. By leveraging the attention mechanism, GAT can dynamically assign different weights to neighboring nodes in the user-item graph, allowing the model to focus more on informative connections and filter out noise. This leads to more refined user and item representations, which in turn result in more accurate recommendation outcomes.

Additionally, the AUC value of GAT reaches 0.85, which is significantly higher than that of the other models under comparison. This further demonstrates its stability and effectiveness in learning from sparse and incomplete data, making it well-suited for real-world recommendation scenarios that often suffer from interaction sparsity.

In contrast, the SVD (Singular Value Decomposition) model performs the worst among the four models evaluated, particularly in metrics such as Precision, Recall, and F1 score, where it consistently lags behind. This underperformance may be attributed to the model's inherent limitations in dealing with sparse data, as SVD relies heavily on matrix factorization techniques that require sufficient co-occurrence information to extract meaningful latent factors. When user-item interactions are sparse, the factorized matrices fail to fully capture hidden user preferences and item attributes, resulting in weak predictive power. The MF (Matrix Factorization) model shows a degree of improvement compared to SVD, especially in Recall and F1 score, suggesting that slight modifications or enhancements to the factorization process can yield marginal benefits. However, its overall performance still falls short when compared to the more advanced GAT and GCN models, which are better equipped to exploit graph-based structural information and learn more expressive representations in sparse environments.

The GCN model ranks in the middle for Precision, Recall, and F1 score, with a commendable AUC value, indicating that Graph Convolutional Networks handle interaction relationships in graph-structured data well. Although the GCN model shows significant improvement over MF and SVD, its performance remains lower than that of the GAT model. This is likely because the GCN model does not optimize information propagation between nodes as effectively as GAT does with its self-attention mechanism. Overall, the GAT model demonstrates clear advantages in handling recommendation tasks with sparse interaction data.

Secondly, this paper presents an experimental analysis of the impact of data sparsity on the performance of the recommendation system, and the experimental results are shown in Figure 2.

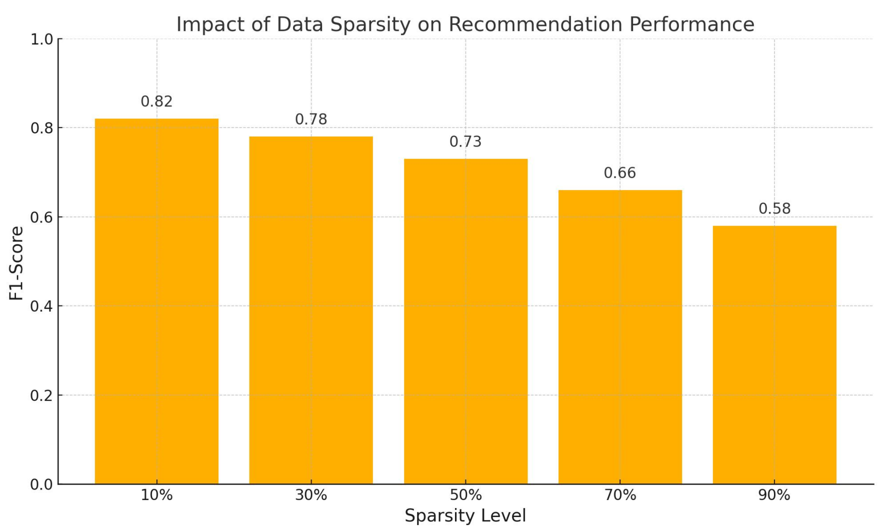

Figure 2. Impact of Data Sparsity on Recommendation Performance

As shown in the figure, as the data sparsity increases, the F1-Score of the recommendation system gradually decreases. This indicates that as the interaction between users and items becomes increasingly sparse, the recommendation model struggles to capture sufficient user preference information, leading to a decline in prediction accuracy. Specifically, when the sparsity reaches 90%, the F1-Score drops to 0.58, demonstrating a significant performance decay in extreme sparse scenarios.

When the data sparsity is between 10% and 50%, the recommendation performance exhibits a slight decline; however, the overall performance remains relatively stable, with F1 values consistently maintained above 0.73. This suggests that the model retains a certain degree of robustness against moderate levels of sparsity, demonstrating its ability to preserve predictive accuracy in the face of incomplete interaction data. It also indicates that, under mild sparsity conditions, essential user preference features can still be effectively extracted through mechanisms such as graph structure learning, neighborhood aggregation, and contrastive representation refinement. These methods allow the model to leverage indirect user-item relationships and latent semantic correlations, which in turn mitigate the adverse effects caused by missing explicit feedback. Nevertheless, once the sparsity level exceeds 50%, a more noticeable performance decline begins to emerge. This reflects a higher sensitivity of the model to extreme sparsity, wherein the limited availability of interaction data reduces the quality of learned representations and constrains the model's capacity to generalize across users and items.

The experimental results thus confirm the substantial impact that varying levels of data sparsity exert on the effectiveness of recommendation systems. More specifically, they illustrate the threshold beyond which traditional graph-based and embedding learning strategies begin to lose their efficacy. These findings emphasize the critical need to address the challenge of data sparsity in order to ensure stable and accurate recommendations. To this end, future research could further explore strategies such as integrating side information (e.g., user demographics, item metadata), incorporating multimodal features (e.g., visual, textual, and contextual cues), or developing more expressive and adaptive graph neural network architectures. These enhancements are expected to improve the model's robustness under high sparsity conditions and strengthen its generalization ability, thereby promoting more reliable and scalable recommendation solutions in real-world sparse-data scenarios.

Next, this paper presents an experiment on the impact of user and item embedding update strategies on recommendation effects. The experimental results are shown in Figure 3.

From the figure, it can be observed that the recommendation performance of the three embedding update strategies shows significant differences across different iterations. Overall, the Layer-wise Update strategy consistently leads throughout the training process. Its F1-Score gradually increases from an initial value of 0.66 to 0.78, demonstrating stable and continuous performance improvement. This indicates that layer-wise updates are more effective in capturing the latent representation information of users and items.

In contrast, the Static Embedding and Attention-based Update strategies perform slightly worse. Static Embedding, lacking a dynamic update mechanism, shows a slower increase in F1 value, ultimately reaching 0.75. This indicates that while the model still learns to a certain extent, its representational ability is constrained due to the lack of iterative refinement, which prevents the model from capturing nuanced preference shifts over time. On the other hand, Attention-based Update introduces an attention mechanism to enhance representation capability, allowing the model to emphasize more informative features in the embedding process. However, its improvement flattens in the later stages of training, with the final F1 value at 0.74, slightly lower than the static strategy. This result suggests that a single attention mechanism still faces limitations in capturing information within complex graph structures, especially when the interactions are sparse and the attention weights may not fully reflect deeper semantic relationships between nodes.

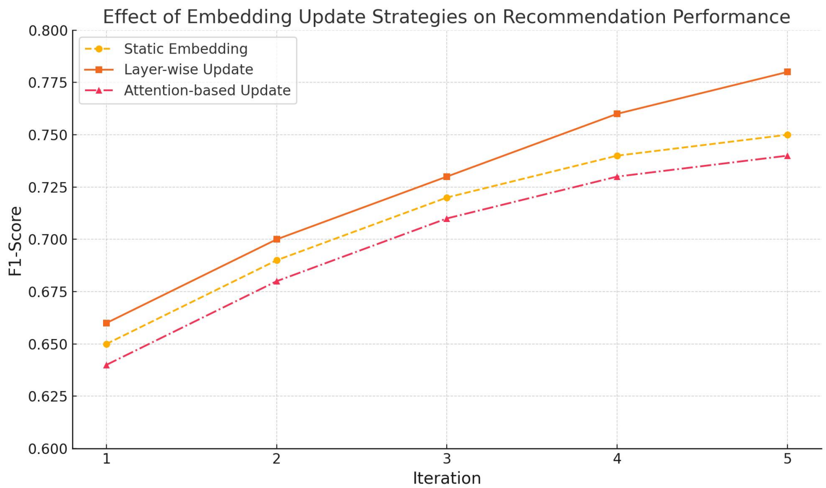

Figure 3. Effect of Embedding Update Strategies on Recommendation Performance

Overall, the experimental results show that a dynamic layer-wise embedding update mechanism plays a positive role in enhancing the recommendation system's performance across multiple training stages. Unlike static or attention-only updates, the layer-wise approach updates user and item embeddings progressively through each layer, allowing the model to capture richer high-order connectivity and latent structural information. This strategy not only effectively alleviates the information loss caused by sparse interactions but also continues to optimize node representations through multiple propagation rounds. As a result, it achieves a higher and more stable F1 performance, showcasing its strong representational capacity and robustness across training iterations.

Moreover, this performance advantage underscores the importance of carefully designing update mechanisms that are adaptive, multi-level, and structure-aware when applied to recommendation systems in sparse environments. By leveraging deeper layer-wise refinement, the model can better disentangle user-item interaction patterns, which is especially critical in real-world applications where data incompleteness is prevalent. These findings also provide a feasible direction for designing more efficient update mechanisms in sparse recommendation environments in the future, suggesting that combining multi-layer propagation with adaptive attention strategies may further boost model expressiveness and accuracy. Furthermore, the effectiveness of the regularization method in the recommendation system is tested, and the experimental results are shown in Figure 4.

From the figure, it can be observed that the regularization strategy has a significant impact on the performance of the recommendation system model. Compared to the scenario without regularization, all regularization methods significantly improve the model's F1 score in the early stages of training and effectively suppress overfitting in later stages. Particularly, the model without regularization shows some improvement in the initial rounds, but from the sixth round onward, its performance gradually declines, with the final F1 value dropping to 0.63. This indicates that the model is prone to overfitting in the later stages of training.

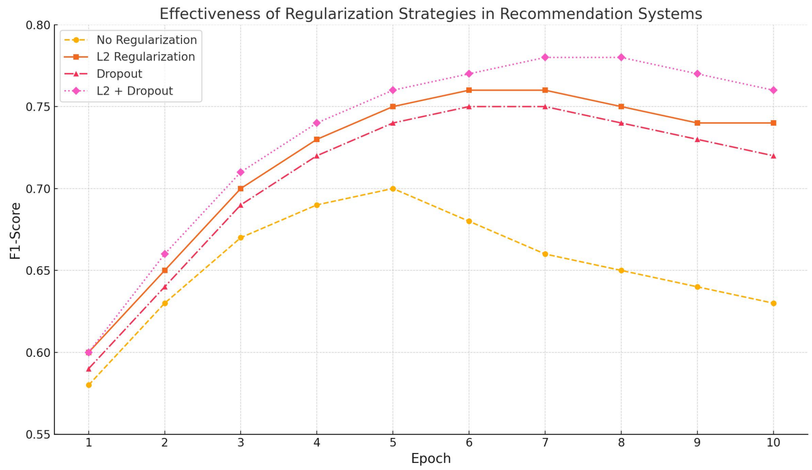

Figure 4. Effectiveness of Regularization Strategies in Recommendation Systems

Among the different regularization strategies, L2 regularization and Dropout show stable performance, continuously improving the model's effectiveness. L2 regularization reaches its peak performance in the sixth round and then remains steady, demonstrating its effectiveness in controlling model complexity. Dropout shows a similar trend but experiences a slight decrease in the final rounds, suggesting that information loss may still occur in the later stages of training. Overall, the combined strategy of using L2 and Dropout together demonstrates the best performance throughout the entire training process, with the final F1 value reaching 0.78 and maintaining high stability in the later stages. This indicates that this combined strategy not only enhances the model's expressive capability but also exhibits excellent anti-overfitting ability, making it one of the ideal regularization methods for recommendation systems.

Finally, the loss function drop graph is given, as shown in Figure 5.

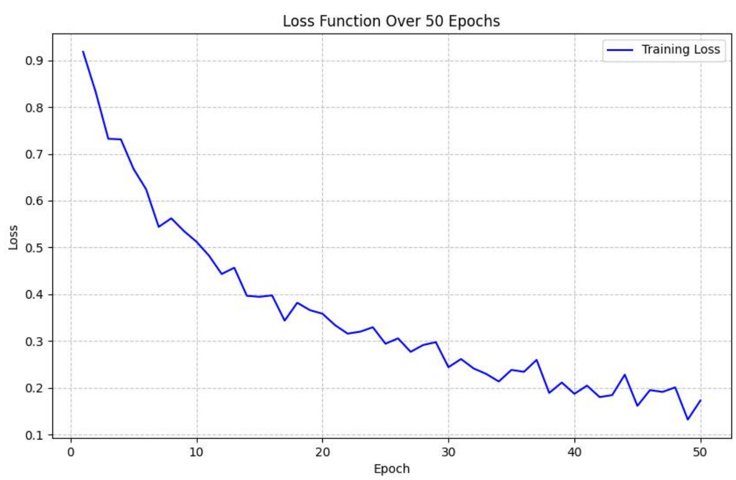

Figure 5. Loss function changes with epoch

In this experiment, the recommendation model we built through the graph attention mechanism (GAT) showed good convergence during the training process, as shown in the loss function decline curve shown in the figure. In the figure, the horizontal axis is the number of training rounds (Epoch), and the vertical axis is the loss value (Loss). It can be clearly observed that the training loss of the model drops rapidly in the first 20 rounds, then tends to stabilize, and reaches the lowest point around the 50th round, indicating that the model gradually converges to a better state and has strong training stability.

The continuous decline of this loss function shows that the model is constantly optimizing the difference between its predicted score and the actual score. At the same time, thanks to the GAT structure's ability to dynamically adjust the weights of neighbor nodes in sparse interaction data, information propagation is more effective, thereby enhancing the distinguishability of user and item representations. The loss curve tends to be stable in the later stage of training, which also shows that the regularization strategy effectively controls overfitting, allowing the model to maintain good generalization ability. In summary, the experimental results fully verify the effectiveness of the method proposed in this paper in sparse interaction scenarios. The graph attention mechanism not only improves the representation ability of the model but also ensures the stability and convergence of the training process, laying the foundation for subsequent practical applications in large-scale recommendation systems.

6. Conclusion

This paper focuses on recommendation systems in sparse interaction scenarios and proposes a recommendation algorithm based on the graph attention mechanism. The aim is to improve the model's ability to handle sparse interaction data between users and items. By constructing a user-item graph structure and incorporating the graph attention mechanism, the model can adaptively capture key neighboring information, thereby more effectively extracting latent preference relationships. Experimental results demonstrate that the proposed method outperforms traditional graph neural network models and matrix factorization methods across multiple evaluation metrics, showing strong robustness and recommendation performance.

In several sets of experiments, we systematically analyzed the performance of the graph attention network, especially under the influence of key factors such as data sparsity, embedding update strategies, and regularization methods. The results show that the graph attention mechanism has a significant advantage in alleviating the information sparsity issue. Hierarchical embedding update strategies and appropriate regularization methods further enhance the model's generalization ability. These findings provide methodological support and empirical evidence for constructing more efficient and stable recommendation systems.

Although this study has achieved certain results, there are still some areas for further expansion. For example, the current model may still face insufficient representation in extremely sparse or cold-start scenarios. Future work could consider incorporating auxiliary information, such as user profiles, contextual information, or social networks, to supplement the model. Additionally, there is still room for improvement in modeling complex heterogeneous graph structures. Efficiently integrating multiple types of edges and node information will be an important direction for the next stage of research. Future research can further explore the integration of graph attention networks with cutting-edge technologies such as large language models and federated recommendation to achieve more personalized, secure, and scalable recommendation services. Furthermore, in practical applications, balancing recommendation effectiveness, computational efficiency, and system response speed is also a problem worth deeper exploration. By continuously optimizing model structure and training strategies, the application of graph neural networks in recommendation systems will have broader development prospects.

References

[1] Martinez, L., & Hughes, D. (2022). Adaptive Graph Attention for Sparse Recommendation Tasks. Proceedings of the 16th International Conference on Web Intelligence and Mining, 112–121.

[2] Patel, S., & Roy, M. (2023). Enhancing Collaborative Filtering under Sparse Interactions using GAT-based Models. Journal of Intelligent Information Systems, 61(3), 487–503.

[3] Thompson, J., & Becker, K. (2024). Lightweight Graph Attention Networks for E-Commerce Recommendations. Journal of Data Analysis and Knowledge Discovery, 8(2), 134–146.

[4] Kim, E., & Alvarez, J. (2023). A Graph-Based Embedding Propagation Framework for Sparse Data Recommendation. International Symposium on Data-Driven Systems (ISDDS), 212–219.

[5] Yang, J., Wang, Z., & Chen, C. (2024). GCN-MF: A graph convolutional network based on matrix factorization for recommendation. Innovation & Technology Advances, 2(1), 14–26.

[6] Yang, L., Wang, S., Tao, Y., Sun, J., Liu, X., Yu, P. S., & Wang, T. (2023, February). Dgrec: Graph neural network for recommendation with diversified embedding generation. In Proceedings of the sixteenth ACM international conference on web search and data mining (pp. 661-669).

[7] Singh, R., & Lopez, F. (2022). Attention-Aware Graph Embedding for Cold-Start and Sparse Recommendation. Applied Computational Intelligence and Soft Computing, 2022, Article ID 9087324.

[8] Jia, X., Zhao, H., Lin, Z., Kale, A., & Kumar, V. (2020, August). Personalized image retrieval with sparse graph representation learning. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2735-2743).

[9] Xie, Y., Li, Z., Qin, T., Tseng, F., Johannes, K., Qiu, S., & Murphey, Y. L. (2021, July). Personalized session-based recommendation using graph attention networks. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[10] Huang, L., Li, R., Lei, J., Ji, Y., Feng, G., Shi, W., & Yang, S. (2024). Sequence-Aware Graph Neural Network Incorporating Neighborhood Information for Session-Based Recommendation. International Journal of Computational Intelligence Systems, 17(1), 32.

[11] Malitesta, D. (2024). Graph neural networks for recommendation leveraging multimodal information.

[12] Hernandez, P., & Novak, M. (2024). Multi-Hop Graph Attention for Enhanced User Representation in Sparse Contexts. International Journal of Artificial Intelligence Research, 12(1), 15–30.

[13] Oliveira, T., & Carter, B. (2024). Layer-wise Node Representation Learning in Sparse Recommendation Graphs. International Journal of Machine Learning and Cybernetics, 15(1), 52–67.

[14] Zhang, Y., & Jackson, R. (2023). Graph Learning with Interaction Dropout for Robust Recommendations. Proceedings of the 27th European Conference on Artificial Intelligence (ECAI), 345–354.

[15] Wilson, T., & Herrera, L. (2023). Hybrid Sparse Interaction Modeling with Graph Neural Networks. Proceedings of the International Conference on Applied Artificial Intelligence (ICAAI), 189–198.