ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

Enhancing Book Recommendation Systems through the Application of Sparse Features in Wide and Deep Learning Models

Guo Jiajie[1,a,*]

[1]*Software Engineering Institute of Guangzhou, Guangzhou, China*
[a]*229559413@qq.com*
[*]*Corresponding author*

Abstract: *In processing large-scale user-item data, recommendation systems often face the challenge of information overload, making it critical to improve recommendation accuracy and efficiency. In recent years, deep learning-based recommendation algorithms have gradually become mainstream. However, due to the high sparsity of user-item interaction data, fully leveraging these sparse features to enhance model performance remains a pressing challenge in the field of recommendation systems. In this context, this paper investigates the application of sparse features in the Wide and Deep Learning model, based on the publicly available Book-Crossing dataset. To evaluate model performance, this study compares Random Forest, Gradient Boosting Decision Trees (GBDT), Multilayer Perceptron (MLP), and Wide and Deep Learning models using three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC). The experimental results demonstrate that the Wide and Deep Learning model outperforms the other models, particularly in terms of AUC and RMSE, confirming its advantages in handling sparse data and improving the performance of recommendation systems.*

Keywords: *Wide and Deep Learning Model, Book Recommender System, Feature Engineering, Sparse Features*

1. Introduction

As In the era of big data, the problem of information overload has become increasingly prominent with the rapid growth of internet users and the surge of online information. Recommendation systems, as an effective filtering tool, help users find relevant information and products by mining user behavioral data and item characteristics. However, as data scale expands and user behavior becomes more complex, recommendation systems face challenges when processing high-dimensional sparse data, especially in the case of book recommendations, where rating data is often sparse, making it difficult to provide accurate recommendations.

The study of personalized recommendation algorithms has always been a central topic in the field of recommender systems. Xu Enyuan proposed a personalized book recommendation system that improves recommendation accuracy and user satisfaction by integrating user behavior data, which has been widely applied in university libraries[1]. Tu Tie et al. employed a collaborative filtering algorithm combined with clustering and prediction imputation methods to address the performance degradation caused by data sparsity in traditional collaborative filtering[2]. Zhao Jie improved the accuracy of recommendations by enhancing the user interest model[3].

In the context of university digital libraries, Ling Yaoyin further explored the application of personalized recommendation technologies and analyzed the importance of user behavior data for system optimizatio[4]. Building on this, Yu Chaoyu constructed a Bayesian network and proposed a differentiated intelligent recommendation algorithm, effectively balancing recommendation accuracy and diversity[5]. Chen Changhua, utilizing the ISODATA clustering algorithm, investigated how to enhance recommendation system performance in large-scale datasets[6].

The continued development of collaborative filtering techniques has driven the emergence of improved algorithms. Wang Qianli optimized the ALS collaborative filtering algorithm, proposing a more efficient recommendation method[7], while Shi Fengyuan designed and implemented a recommendation analysis system based on big data technology, effectively handling massive data to enhance the responsiveness and accuracy of recommendation algorithms[8]. Yan Fan's research focused

ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

on book recommendation using semantic information, highlighting the role of natural language processing in improving the semantic understanding capabilities of recommendation systems[9].

Wang Yuqin proposed a hybrid algorithm that combines the advantages of collaborative filtering and content-based filtering, effectively addressing issues of diversity and cold start in recommendation systems[10]. Wu Simin analyzed the integration of user information needs and personalized recommendation systems, further enhancing the level of personalization in recommendations[11]. Additionally, Zhang Huaigao proposed a solution for privacy protection in personalized recommendations, particularly suitable for e-book store scenarios[12].

Wang Dafu, by constructing user profiles, proposed a personalized recommendation model based on user behavior data, which has been applied in university library personalized recommendations[13]. Bao Yan designed a recommendation system based on long- and short-term preferences, solving the problem of balancing users' long-term interests and short-term needs[14]. Chen Linghong, utilizing knowledge graphs and reader profiles, proposed an intelligent book recommendation system, significantly improving the accuracy and interpretability of recommendation algorithms[15].

To enhance the diversity of recommendation results, Zhong Zufeng proposed a re-ranking-based recommendation algorithm, which avoids the common "long tail" effect in traditional recommendation algorithms[16]. Gao Xia studied the application of full-text retrieval technology in book search, improving search efficiency and user experience by integrating recommendation algorithms[17]. Meanwhile, Zheng Xin proposed an improved recommendation system model based on collaborative filtering, considering time factors and user characteristics to enhance recommendation accuracy[18].

Song Chuping, focusing on large-scale data scenarios in university libraries, proposed an improved collaborative filtering method that optimizes recommendation performance[19]. Finally, Fu Yongping proposed a personalized collaborative filtering recommendation method based on Bayesian networks, effectively addressing the problem of sparse data and significantly improving recommendation accuracy[20]. Heng-Tze Cheng et al. proposed the "Wide & Deep Learning" framework, which combines wide linear models and deep neural networks to leverage both for solving memorization and generalization issues in recommendation systems[21].

In conclusion, existing research has improved recommendation systems' performance in handling data sparsity, recommendation accuracy, and efficiency through various methods, providing important theoretical and practical support for this study on the application of the Wide and Deep Learning model in processing sparse features.

2. Principles of the Models

2.1. Random Forest

Random Forest is an ensemble learning method based on decision trees. Its core idea is to improve the accuracy of classification or regression tasks by constructing multiple decision trees. Each decision tree is trained on a randomly sampled subset of the original training data, and the node splits in each tree are determined by randomly selecting a portion of the features. The final prediction is obtained through majority voting (for classification tasks) or averaging (for regression tasks) of the predictions from all trees. By introducing randomness, Random Forest reduces the risk of overfitting that can occur in individual decision trees and is capable of handling large-scale data and high-dimensional features. It also demonstrates strong robustness against noise. Moreover, Random Forest provides feature importance measures, which help in understanding which features contribute significantly to the model’s predictions.

2.2. Gradient Boosting Decision Tree

Gradient Boosting Decision Trees (GBDT) is an iterative ensemble learning algorithm that constructs multiple weak learners, typically decision trees, to gradually reduce the residuals in each iteration. Each new tree is trained to correct the errors (residuals) made by the previous tree, allowing the new tree to complement the shortcomings of its predecessors. Through this process, multiple weak learners are progressively combined to form a strong learner. GBDT uses the negative gradient of the loss function to guide the generation of new trees, thereby continuously optimizing the overall model performance. GBDT has strong fitting capabilities and can handle complex nonlinear relationships, making it widely applicable to tasks such as classification, regression, and ranking. Its key advantage lies in its flexibility,

ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

allowing the use of different loss functions to suit various tasks.

2.3. Multilayer Perceptron

The Multilayer Perceptron (MLP) is a classical neural network architecture consisting of an input layer, hidden layers, and an output layer. Each neuron in one layer is fully connected to every neuron in the next layer, with information propagating forward from the input layer to the output layer. This network introduces nonlinearity through activation functions, enabling the model to learn complex nonlinear relationships. During training, the backpropagation algorithm is used to compute the gradient of the loss function with respect to each weight, and the weights are updated using gradient descent to progressively minimize the loss. MLP are widely used in tasks such as classification and regression and are particularly well-suited for processing structured data and time series. One of the key advantages of MLP is their flexibility; by increasing the number of hidden layers, the model's capacity for learning complex patterns can be significantly enhanced.

2.4. Wide and Deep Learning model

The Wide and Deep Learning model combines the strengths of both generalized linear models and deep neural networks. The generalized linear model (Wide component) is used for memorizing features, particularly for handling sparse features such as direct combinations of categorical features. The deep neural network (Deep component) learns complex feature interactions through multiple hidden layers, capturing the nonlinear relationships between features. The Wide component excels at processing explicit features, while the Deep component uncovers implicit feature combinations. This design enables the model to balance both memorization and generalization. Consequently, the Wide and Deep Learning model is widely applied in recommendation systems, especially in scenarios requiring the processing of large-scale sparse data. By combining the efficiency of linear models with the generalization capabilities of deep models, this approach enhances predictive accuracy.

3. Experimental Analysis

3.1. Data and Preprocessing

This study utilizes the Book-Crossing dataset, which is a publicly available dataset widely used in book recommendation system research, created by Cai-Nicolas Ziegler in 2004. The dataset contains global users' book ratings and related information, providing a rich source of data for the development and research of personalized recommendation systems. It is a large-scale dataset, comprising 278,858 users, 271,379 books, and 1,149,780 rating records. Due to the high sparsity of user, book, and rating data, the Book-Crossing dataset presents challenges for recommendation system development, particularly in effectively addressing data sparsity.

The dataset consists of three main parts. First, the user data contains detailed information such as user ID, age, and address. Some users may not have provided age information or may have supplied invalid data. Second, the book data includes detailed information about books, such as ISBN (International Standard Book Number), title, author, publication year, and publisher. Although some books may lack complete metadata (e.g., missing titles or authors), each book's ISBN serves as a unique identifier, ensuring the uniqueness of records in the dataset. Lastly, the rating data captures users' ratings of books on a scale from 1 to 10, with a rating of 0 indicating that the user has not rated the book.

In the data preprocessing phase, missing values in the user ID, address, ISBN, author, publication year, and publisher fields were filled with default values, with no additional preprocessing required. These fields were encoded during the feature engineering phase. For the age field, we excluded outliers (ages below 3 or above 100) and discretized the remaining values to better represent the distribution of user ages. Specifically, K-Means clustering was applied to divide the ages into six groups: under 18, 18-25, 26-31, 32-40, 51-60, and 60 and above. Each group was encoded as an integer from 1 to 6. This binning method helps to enhance the aggregation of data within each age group, thereby improving the model's ability to handle age-related features.

3.2. Feature Engineering

In this study, the CRC32 hash function was applied to encode user IDs, addresses, ISBNs, authors,

ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

publication years, publishers, and the binned values of user ages. To prevent hash collisions, a "salting" mechanism was introduced by using the field name as the salt, ensuring the uniqueness of hash values across different fields.

To extract key information from book titles, the TF-IDF algorithm was applied to each word in the titles, and the top 5000 words with the highest weights were selected as tags for the books. These tags were then encoded using the hash function, providing richer feature representations for the model.

Additionally, this study constructed comprehensive user and book profiles to enhance the model's ability to capture underlying patterns in the data. User profile features include the total number of book ratings, the user's ratings for specific book tags, and the number of ratings for each book tag. Furthermore, features such as the difference between the highest and lowest book ratings given by the user were included to reveal the user's rating distribution. The standard deviation of the user's ratings across different book tags was also calculated to reflect the consistency or diversity of the user's preferences.

On the book profile side, features such as the total number of ratings for each book and the average rating were constructed. Additional features include the variance of book ratings, which measures the consistency of user ratings for a given book, and the distribution of book ratings (e.g., median, quartiles) to identify extreme cases in book evaluations. The previously extracted book tag list was also incorporated as part of the book profile features.

3.3. Model Selection and Construction

In the model construction process, different feature processing methods were chosen to meet the specific needs of each model. For the Random Forest (RF) and Gradient Boosting Decision Tree (GBDT) models, the raw features were directly used as input.

The Multilayer Perceptron (MLP) model built in this study consists of two hidden layers, each containing 64 neurons, with the Rectified Linear Unit (ReLU) as the activation function. The output layer is a single neuron, using the Sigmoid activation function to map the output to the [0, 1] range. Since the rating scale is [1, 10], the output is multiplied by 10. The model's input features are one-hot encoded using hashed values. The loss function employed is the cross-entropy loss function.

The Wide and Deep model constructed in this study comprises a logistic regression model as the Wide component, which primarily takes hashed one-hot encoded inputs, such as user ID, ISBN, and binned age values. The Deep component includes three layers: the first two layers consist of 64 neurons each, while the third layer contains 32 neurons, all utilizing the ReLU activation function. The Wide and Deep components are concatenated, and their outputs are combined using weighted summation. The final output uses the Sigmoid activation function to map the result to the [0, 1] range, and since the rating scale is [1, 10], the output is multiplied by 10. The input features are also one-hot encoded using hashed values, and the cross-entropy loss function is employed.

3.4. Evaluation and Comparison of Prediction Results

This study evaluates the four models on different datasets using three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC).

Mean Absolute Error (MAE) measures the average absolute difference between the predicted and actual values. It is calculated by taking the absolute differences between predicted and actual values, and then averaging them, reflecting the magnitude of prediction bias. A lower MAE indicates a smaller prediction error, meaning the model's predictions are closer to the actual values, thus demonstrating better predictive performance. MAE assigns equal weight to all errors, making it suitable for scenarios where sensitivity to large and small errors is comparable.

Root Mean Squared Error (RMSE) is another metric for evaluating model prediction errors. It is calculated by taking the square root of the average squared differences between predicted and actual values. Compared to MAE, RMSE is more sensitive to larger errors because the error squaring process amplifies larger deviations. Therefore, RMSE is better suited for assessing a model's performance in handling extreme prediction errors. A lower RMSE indicates better overall predictive performance, especially in cases where larger errors are of greater concern.

Area Under the Curve (AUC) evaluates the performance of binary classification models based on the Receiver Operating Characteristic (ROC) curve. The AUC value ranges between 0 and 1, indicating the model's ability to distinguish between positive and negative classes. AUC values closer to 1 indicate

ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

stronger classification ability, whereas values closer to 0.5 suggest that the model's classification performance is akin to random guessing. A key advantage of AUC is its robustness to imbalanced class distributions, making it well-suited for classification tasks where class imbalance is present.

These three metrics offer different perspectives in evaluating model performance.

In terms of feature construction and model training, this study utilized Spark to construct a pipeline for data preprocessing and feature engineering. Scikit-learn was used to train and evaluate the Random Forest and Gradient Boosting Decision Tree (GBDT) models, while TensorFlow was employed to train and evaluate the Multilayer Perceptron (MLP) and Wide and Deep models. Table 1 provides a comparative analysis of the evaluation metrics for the four models across various datasets.

Table 1: Comparison Table of Four Model

Model	Dataset	MAE	RMSE	AUC
Random Forest	Training	0.845	1.085	0.728
	Validation	0.91	1.15	0.7
	Test	0.935	1.18	0.693
GBDT	Training	0.795	1.04	0.752
	Validation	0.85	1.11	0.73
	Test	0.87	1.135	0.721
Multilayer Perceptron	Training	0.765	1.02	0.78
	Validation	0.82	1.085	0.769
	Test	0.835	1.1	0.76
Wide and Deep model	Training	0.745	0.995	0.804
	Validation	0.79	1.05	0.788
	Test	0.805	1.065	0.78

Based on the prediction results from the test set, it can be observed that the Wide and Deep model, which integrates sparse features such as user ID and book ISBN, outperforms the other models in terms of MAE, RMSE, and AUC. Additionally, the Wide and Deep model demonstrates better performance on the test set compared to the MLP model, indicating that the Wide and Deep model exhibits stronger generalization ability.

4. Conclusions

This study, based on the Book-Crossing dataset, investigates the application of sparse features in the Wide and Deep Learning model, aiming to address the prevalent issue of user-item interaction data sparsity in recommender systems. By comparing the performance of Random Forest, Gradient Boosting Decision Trees (GBDT), Multilayer Perceptron (MLP), and Wide and Deep Learning models, the evaluation was conducted using three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC). The experimental results demonstrate that the Wide and Deep Learning model excels at handling sparse data, significantly outperforming other models, especially in terms of AUC and RMSE, showcasing superior generalization ability and predictive performance. Additionally, this study optimized feature engineering through various approaches, such as using TF-IDF to extract book tag features and applying hash encoding to handle the sparse features of users and books. A comprehensive analysis reveals that the Wide and Deep Learning model has significant advantages in addressing data sparsity and improving the performance of recommender systems, confirming its value in the application of personalized recommendation systems.

References

[1] Xu Enyuan. Design of personalized book recommendation system for university libraries[J]. Science & Technology Information, 2023(13): 207-210.
[2] Tu Tie, Liu Bin. Research on collaborative filtering book recommendation algorithm based on clustering and prediction filling[J]. Journal of Guizhou Institute of Technology and Applications, 2024(3): 100-105.
[3] Zhao Jie. Research on personalized book recommendation algorithm based on improved user interest model[J]. Machine Tool & Hydraulics, 2018(6): 193-198.
[4] Ling Yaoyin. Research on the application of personalized recommendation technology in university digital libraries[J]. Information Recording Materials, 2023(10): 79-97.

ISSN 2706-655X Vol.6, Issue 5: 106-111, DOI: 10.25236/IJFET.2024.060515

[5] Yu Chaoyu. Research on differentiated intelligent book recommendation algorithm based on Bayesian network[D]. Master's Thesis, Beijing Jiaotong University, 2022.
[6] Chen Changhua. Research on personalized book recommendation method based on ISODATA clustering algorithm[J]. Software Engineering and Applications, 2022(4): 35-40.
[7] Wang Qianli. Research on book recommendation algorithm based on improved ALS collaborative filtering[J]. Library Science Research, 2021(7): 58-65.
[8] Shi Fengyuan. Design and implementation of book recommendation analysis system based on big data[D]. Master's Thesis, Nanjing University of Science and Technology, 2023.
[9] Yan Fan. Research on book recommendation method based on book semantic information[D]. Master's Thesis, Dalian University of Technology, 2023.
[10] Wang Yuqin. Research on personalized library recommendation system based on hybrid algorithm[J]. Science and Publishing, 2022(10): 23-29.
[11] Wu Simin. Integration of user information needs analysis and personalized book recommendation system[J]. Literature and History Expo, 2023(10): 67-72.
[12] Zhang Huaigao. Research on personalized recommendation scheme for e-book stores based on privacy protection[D]. Master's Thesis, East China Normal University, 2021.
[13] Wang Dafu. Research on personalized book recommendation for university libraries based on user profiling[J]. Journal of Henan Normal University, 2022(3): 95-103.
[14] Bao Yan. Design of personalized book recommendation system based on long- and short-term preferences[J]. Inside and Outside the Lantai, 2024(19): 70-72.
[15] Chen Linghong. Research on book recommendation based on knowledge graphs and reader profiling[D]. Master's Thesis, Huazhong University of Science and Technology, 2023.
[16] Zhong Zufeng. Research on book recommendation algorithm to enhance diversity based on re-ranking[J]. Computer Science, 2023(9): 100-108.
[17] Gao Xia. Application of full-text retrieval technology in book retrieval[J]. Journal of Zhongyuan Institute of Technology, 2024(2): 85-89.
[18] Zheng Xin. Application of collaborative filtering algorithm in book recommendation systems[J]. Journal of Fuzhou Business College, 2023(12): 60-62.
[19] Song Chuping. Application of an improved collaborative filtering method in university library book recommendation[J]. Library and Information Work, 2016(24): 86-91.
[20] Fu Yongping. Research on personalized collaborative filtering recommendation method based on Bayesian network[J]. Journal of Intelligence, 2023(3): 45-53.
[21] Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems[J]. arXiv preprint arXiv:1606.07792, 2016.