The Frontiers of Society, Science and Technology
ISSN 2616-7433 Vol. 6, Issue 9: 35-40, DOI: 10.25236/FSST.2024.060906

Improving Movie Recommendations Using Multilayer Perceptron Model and Sparse Feature Integration

Guo Jiajie[1,a,*]

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

Abstract: *In handling large-scale user and item data, recommendation systems often face the challenge of information overload, making the improvement of recommendation accuracy and efficiency a critical research direction. In recent years, recommendation algorithms based on deep learning have increasingly dominated the field, with feedforward neural networks gaining widespread application due to their flexibility and scalability. However, the high sparsity of user-item data presents a key challenge in effectively leveraging these sparse features to enhance model performance. This study utilizes the Movielens movie dataset and generates sparse features through feature engineering, proposing a multilayer perceptron model based on sparse features. The model is compared with classical models such as logistic regression, random forest, and gradient boosting decision trees (GBDT). Experimental results demonstrate that the feedforward neural network based on sparse features exhibits significant advantages in performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC). This study provides both theoretical foundations and practical guidance for optimizing recommendation systems in sparse data environments, with important application value.*

Keywords: *Feedforward Neural Network, Feature Engineering, Sparse Features, Movie Recommendation System*

1. Introduction

As Recommendation systems have become essential tools in modern information processing and are widely applied across various domains such as e-commerce, online media, and social platforms. As the volume of internet data grows rapidly, providing users with personalized and accurate recommendations from vast amounts of data has emerged as a critical problem. Traditional recommendation methods, such as collaborative filtering and content-based recommendation, face limitations when dealing with data sparsity and the cold start problem. In recent years, with the rapid development of deep learning, recommendation systems based on neural networks have demonstrated strong feature learning capabilities and superior performance. Among these, feedforward neural networks have become a research hotspot in the field of recommendation systems due to their flexibility and ability to fit highly nonlinear data.

However, user-item interaction data in recommendation systems often exhibits high sparsity, making it challenging for traditional deep learning methods to fully capture the underlying information within this sparse data. To address this challenge, feature engineering techniques have been widely employed in handling sparse data, enabling the effective extraction and transformation of user and item features. This significantly improves the model's generalization ability and recommendation accuracy. Therefore, combining sparse features with deep learning models to construct efficient recommendation systems has become a critical research direction.

Early recommendation systems primarily relied on collaborative filtering (CF) and content-based recommendation methods. Wang Guang proposed a content-based weighted granular sequence recommendation algorithm, which enhances recommendation accuracy by combining user preferences and movie content[1]. Jin Xin's research focused on applying deep learning technology to movie recommendation systems, demonstrating that deep learning models significantly improve recommendation performance when handling large-scale data[2]. Additionally, improving traditional algorithms is a common approach. Wang Yan introduced an efficient SVD++ algorithm with improved learning rates to address the shortcomings of collaborative filtering in handling sparse data[3].

Building on this, more researchers have introduced complex models and new technologies to optimize recommendation systems. Zhao Guisheng designed a movie recommendation system based on the IRGAN model and Hadoop, showcasing the potential of generative adversarial networks (GANs) in recommendation systems[4]. Gu Yiran proposed a recommendation algorithm based on movie attributes and interaction information, which further enriched the data dimensions in movie recommendation and enhanced accuracy[5]. At the same time, Zhang Yanliang combined dynamic user profile tags with KNN classification to improve user preference modeling and effectively solve the cold start problem[6].

Multidimensional data fusion has also been applied to recommendation systems. Jin Shanshan and colleagues proposed a multidimensional data fusion-based movie recommendation system, which combined movie posters, user features, and other multidimensional features. This approach improved recommendation accuracy using deep learning models[7]. Additionally, Wang Meishen’s research explored long-term recommendation methods based on bipartite networks, aiming to capture users' long-term preference changes through network structures[8]. Hu Xuelin provided a comprehensive review of recommendation algorithms based on distributed representation techniques, emphasizing their ability to better capture implicit relationships between users and items[9].

In recent years, convolutional neural networks (CNNs) have increasingly demonstrated advantages in handling multimodal information. He Jie utilized CNNs to process multimodal information, proving their potential application in movie recommendation[10]. Liu Hualing’s review systematically summarized the latest advances in deep learning for content-based recommendation algorithms, highlighting the extensive application prospects of deep learning models in big data environments[11].

To address the issue of sparsity in recommendation systems, Chen Na and colleagues proposed a multi-criteria decision-making recommendation system based on neural networks. They applied fuzzy theory to mitigate uncertainties in user evaluations, thereby enhancing system robustness[12]. Liu Xiaowei’s research focused on the application of collaborative filtering in personalized movie recommendation, proposing a hybrid recommendation strategy that combines user ratings with content features[13]. Simultaneously, Li Kunlun introduced an attention mechanism and an improved TF-IDF algorithm to enhance recommendation accuracy and efficiency[14].

In terms of hybrid recommendation, Tan Taizhe proposed a hybrid recommendation system based on an attention model, integrating multiple recommendation algorithms and improving recommendation precision through the attention mechanism[15]. Zhong Zhifeng introduced a hybrid recommendation model based on the least squares method, optimizing recommendation results through weight distribution and least squares calculation. The experiments demonstrated that this model performed well across various datasets[16].

Moreover, He Ming proposed a collaborative filtering recommendation algorithm that integrates category information and user interest levels, further improving recommendation accuracy by combining user interests and item categories[17]. Li Menghao conducted a detailed analysis of the progress in recommendation algorithms, particularly in tackling challenges related to data sparsity and cold start issues[18]. Huang Bo’s review emphasized the diversity of recommendation system applications and explored how optimizing algorithms can enhance user experience[19]. Lastly, Zhang Fuguo introduced a recommendation algorithm based on criterion-level feature cross-fusion, which improved the flexibility and adaptability of recommendation systems through feature interaction[20].

In summary, current research in recommendation systems has gradually shifted from traditional collaborative filtering and content-based recommendations to emerging technologies such as deep learning, hybrid recommendation models, and multidimensional feature fusion. These studies provide essential theoretical foundations and technical support for improving recommendation system accuracy, efficiency, and their ability to handle complex data. This study, based on the Movielens movie dataset, uses feature engineering to generate sparse features and constructs a multilayer perceptron recommendation model. We compare it with classical recommendation models such as logistic regression, random forest, and gradient boosting decision trees (GBDT), focusing on their performance in handling sparse data. The experimental results show that the feedforward neural network based on sparse features demonstrates significant advantages in performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC). Through this study, we aim to provide new insights and solutions for optimizing recommendation systems in sparse data environments.

2. Principles of the Models

2.1. Logistic Regression Model

Logistic Regression (LR) is a widely used linear model for classification tasks, particularly suitable for binary classification problems. Its core concept involves using a linear regression model to compute a weighted sum of the input features, which is then mapped to the [0, 1] interval through the logistic (Sigmoid) function, thereby estimating the probability of an event belonging to a particular class. Logistic regression optimizes the model parameters through Maximum Likelihood Estimation (MLE), ensuring that the log-likelihood function reaches its maximum value. Due to its simplicity, efficiency, and interpretability, logistic regression is commonly employed in recommendation systems to address the cold start problem or serve as a baseline model for evaluating the performance of more complex models.

2.2. Multilayer Perceptron

The Multilayer Perceptron (MLP) is a typical artificial neural network consisting of an input layer, one or more hidden layers, and an output layer. The neurons in each layer are fully connected to those in the next layer, allowing information to flow from the input layer to the output layer without forming any loops, thereby ensuring unidirectional data flow. MLFN adjusts the weights through the backpropagation algorithm, gradually reducing output error. The nonlinear activation functions in the hidden layers provide the model with strong nonlinear mapping capabilities, enabling it to effectively capture complex input-output relationships. Due to its flexibility and powerful feature extraction capabilities, MLFN performs exceptionally well when handling high-dimensional sparse features, making it particularly suitable for user behavior prediction and classification tasks in recommendation systems.

2.3. Random Forest

Random Forest (RF) is an ensemble learning method based on decision trees. It enhances the model's generalization ability by constructing multiple independent decision tree models and introducing randomness. Specifically, Random Forest generates different training sets for each tree by randomly selecting samples and features, and then aggregates the output of all decision trees by voting or averaging to produce the final prediction. Random Forest is highly effective in handling high-dimensional data, avoiding overfitting, and demonstrating strong robustness against missing values and noise. Therefore, in recommendation systems, Random Forest is often used to capture the nonlinear relationships between users and items.

2.4. Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT) is an ensemble learning method that incrementally builds multiple weak classifiers, typically decision trees, and integrates them. GBDT utilizes an additive model and a forward stage-wise algorithm to gradually reduce prediction errors. Each decision tree generated in each step aims to fit the residuals from the previous stage, ensuring that the new tree corrects the errors made by the previous model. Through iterative updates, GBDT ultimately forms a powerful predictive model. Due to its ability to handle complex nonlinear relationships and its superior performance with both continuous and categorical features, GBDT is frequently used in recommendation systems to solve personalized recommendation problems, especially excelling in predicting user behavior preferences.

3. Experimental Analysis

3.1. Data and Preprocessing

This study utilizes the MovieLens 1M dataset, which contains 1 million rating records from 6,000 users on nearly 4,000 movies. The dataset is divided into three files: user data, movie data, and rating data. The user data file includes user ID, gender, age, occupation ID, and postal code fields; the movie data file contains movie ID, movie name, and movie genre fields; and the rating data file includes user ID, movie ID, rating, and timestamp fields.

During the data preprocessing phase, some fields required no further processing. For instance, user ID, occupation ID, and movie ID. The gender field, which takes values of F (female) and M (male), also required no additional handling. However, for the age field, discretization was needed to better represent the age distribution of users. This study applied the K-Means clustering algorithm to divide the age field into seven intervals: below 18, 18-25, 26-35, 36-45, 46-51, 51-56, and above 57. Each interval was encoded as an integer ranging from 1 to 7. This binning approach helps enhance the aggregation of data within each age group, improving the model's ability to handle age-related features.

The movie genre field is a list, as a single movie can belong to multiple genres. To handle this field, a dictionary was first constructed to map each movie genre to a numerical value, and the genres of each movie were then converted into a numerical list based on this dictionary. Similarly, for the movie name field, a dictionary mapping each word in the movie titles to a numerical value was created, and the movie titles were converted into corresponding numerical lists. This approach ensures the effective utilization of both movie names and genres in the model.

3.2. Feature Engineering

In this study, the CRC32 hash function was applied to user ID, occupation ID, movie ID, gender, age buckets, movie genre types, and words in movie titles to generate corresponding encodings. To prevent hash collisions, a "salting" mechanism was introduced during the hashing process, with field names used as salt values to ensure the uniqueness of the hash values for different fields.

Additionally, a rich set of user profile features and movie profile features were constructed to enhance the model’s ability to capture user behavior. The user profile features included the total number of movies rated, the total number of ratings for different movie genres, and the average rating for various movie genres. Furthermore, the following features were added: the user's viewing frequency, calculated based on rating timestamps to measure user activity; the total number of ratings within a specific time frame (e.g., the past month) to reflect the user's recent behavioral preferences; the difference between the highest and lowest rated movies to reveal the distribution of the user's ratings; and the standard deviation of the user's ratings across different genres, indicating the user's consistency or diversity of preferences for various movie genres.

On the movie profile side, in addition to constructing the total number of ratings and the average rating for each movie, the following features were also included: the rating variance, used to measure the consistency of user evaluations for a particular movie; the rating distribution trend (e.g., median, quartiles) to identify extreme cases in movie evaluations; the gap between the movie’s release date and the user’s rating time, reflecting the trend in the movie's popularity; and changes in the number of viewers over different time periods to capture the variation in the movie's popularity.

Moreover, to extract key information from movie titles, the TF-IDF algorithm was applied to each word in the title, and the top 1000 words with the highest weights were selected as additional labels for the movies. These labels were processed similarly to the movie genre field, with the hash function being used to encode them, thereby providing the model with richer movie features.

3.3. Model Selection and Construction

In the model construction process, different feature processing methods were selected to meet the needs of various models. For The Multilayer Perceptron (MLP) and Logistic Regression models, the engineered input features described earlier were used. In contrast, the Random Forest (RF) and Gradient Boosting Decision Tree (GBDT) models directly used the raw features as input.

The MLP model constructed in this study includes two hidden layers, each containing 128 neurons, with the activation function set to ReLU (Rectified Linear Unit). The output layer consists of a single neuron, using the Sigmoid activation function to map the output to the [0, 1] range. Since the ratings are on a [0, 5] scale, the output is multiplied by 5. Given that the input layer includes high-dimensional sparse features such as user ID and movie ID, the number of neurons in the input layer is accordingly large, and the input layer exhibits noticeable sparsity. The cross-entropy loss function is used as the loss function. The Logistic Regression model, as a baseline classifier, uses the same input features. For the Random Forest and GBDT models, in addition to using the raw features as input, the hashed features were also included.

3.4. Evaluation and Comparison of Prediction Results

This study uses three key metrics—Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Area Under the Curve (AUC)—to evaluate the performance of the four models across different datasets. These metrics not only provide a comprehensive reflection of the predictive accuracy of the models but also help us better understand the applicability of the models in recommendation systems.

Mean Absolute Error (MAE) measures the average difference between the predicted values and the actual values. It is calculated as the mean of the absolute errors of all samples. The advantage of MAE lies in its interpretability, as it directly reflects the deviation between the model's predictions and the actual values. A smaller MAE indicates more accurate predictions. In recommendation systems, MAE is commonly used to evaluate the precision of the model’s rating predictions, offering a clear representation of the overall error when predicting user preferences.

Root Mean Squared Error (RMSE) is another metric for measuring prediction error. It is calculated by taking the square root of the mean of the squared errors, making it more sensitive to larger errors. Compared to MAE, RMSE places greater emphasis on the model's performance in handling extreme predictions by amplifying the impact of larger deviations. In recommendation systems, RMSE is particularly useful for assessing whether a model exhibits significant error when predicting user ratings, helping identify areas where the model may struggle with complex user behavior.

Area Under the Curve (AUC) is a common metric used to evaluate the performance of binary classification models. It calculates the area under the Receiver Operating Characteristic (ROC) curve, reflecting the model's ability to distinguish between positive and negative samples. In recommendation systems, AUC effectively measures the ranking ability of the model, i.e., whether it can accurately prioritize content that users are interested in. A higher AUC, closer to 1, indicates stronger discriminatory power between positive and negative samples, demonstrating better ranking performance in recommendation systems and enhancing user satisfaction.

In terms of feature engineering and model training, this study built a complete data processing pipeline based on Spark to automate the processing of user and movie features. The models were then trained and evaluated using the Scikit-learn and TensorFlow platforms. Specifically, Scikit-learn was used to train the Logistic Regression, Random Forest, and Gradient Boosting Decision Tree (GBDT) models, while TensorFlow was employed for the construction and training of the Multilayer Perceptron. This multi-platform collaboration ensures the efficiency and accuracy of feature engineering and model training, providing a solid foundation for the final evaluation results. Table 1 compares the evaluation metrics of the four models across different datasets.

Table 1: Comparison Table of Four Model

Model	Dataset	MAE	RMSE	AUC
Multilayer Perceptron	Training	0.45	0.6	0.78
	Validation	0.5	0.65	0.77
	Test	0.52	0.67	0.76
Logistic Regression	Training	0.68	0.88	0.67
	Validation	0.72	0.91	0.64
	Test	0.75	0.94	0.63
Random Forest	Training	0.55	0.75	0.72
	Validation	0.6	0.8	0.71
	Test	0.62	0.82	0.69
GBDT	Training	0.54	0.74	0.73
	Validation	0.58	0.78	0.73
	Test	0.6	0.8	0.72

The prediction results on the test set reveal that the Multilayer Perceptron model, which incorporates sparse features such as user ID and movie ID, demonstrates superior performance in terms of MAE, RMSE, and AUC metrics. This result indicates that the inclusion of sparse features helps deep learning models more effectively capture complex patterns and relationships within the data. Furthermore, it confirms the critical role these features play in enhancing the model’s expressive power and improving predictive accuracy.

4. Conclusions

This study presents a Multilayer Perceptron model based on sparse features, aimed at optimizing movie recommendation systems through in-depth analysis of the MovieLens 1M dataset. During the experiments, sparse features such as user ID and movie ID were incorporated to construct more expressive user and movie profile features, which were then fed into various models for performance comparison. The results, compared against Logistic Regression, Random Forest, and Gradient Boosting Decision Tree (GBDT) models, indicate that the Multilayer Perceptron exhibits significant advantages when processing sparse feature data, achieving optimal results in key metrics such as MAE, RMSE, and AUC.

The study demonstrates that sparse features provide critical support in enabling deep learning models to capture the complexity of data, effectively improving recommendation accuracy and model generalization. By introducing feature engineering techniques and hash encoding, this study not only improves model performance in sparse data environments but also validates the feasibility and superiority of deep learning models in large-scale recommendation systems. This provides new insights and practical guidance for the development and application of future recommendation systems, offering significant theoretical value and practical prospects.

References

[1] Wang G, Zhao J, Li X, et al. Weighted Granular Sequence Recommendation Algorithm Based on Content[J]. Computer Science, 2020, 47(3): 72-78.

[2] Jin X, Zhang Y, Wang H, et al. Research on Movie Recommendation System Based on Deep Learning[J]. Computer Engineering and Applications, 2019, 55(18): 235-240.

[3] Wang Y, Liu M, Zhao L, et al. An Efficient SVD++ Algorithm with Improved Learning Rate[J]. Modern Computers, 2021, 47(3): 56-63.

[4] Zhao G S, Zhang H, Wang X, et al. Design of a Movie Recommendation System Based on IRGAN Model and Hadoop[J]. Computer Engineering, 2020, 46(9): 67-71.

[5] Gu Y R, Zhang J, Li H, et al. Movie Recommendation Algorithm Based on Movie Attributes and Interaction Information[J]. Computer Applications, 2021, 41(11): 302-308.

[6] Zhang Y L, Wang Z, Liu Y, et al. Research on KNN Classification Recommendation Algorithm Based on Dynamic User Portrait Tags[J]. Journal of Software, 2020, 31(9): 2762-2770.

[7] Jin S S, Chang H Z, et al. Design of a Film and Television Recommendation System Based on Multidimensional Data Fusion[J]. Modern Electronics Technology, 2023, 45(17): 123-128.

[8] Wang M S, Chen X, Zhao L, et al. Long-Term Recommendation Based on Bipartite Networks[J]. Software Guide, 2020, 19(7): 34-40.

[9] Hu X L, Li Z, Wang M, et al. A Review of Recommendation Algorithms Based on Distributed Representation Technology[J]. Application Research of Computers, 2018, 35(12): 3789-3793.

[10] He J, Wang Z, Liu Y, et al. Multimodal Information Processing Methods and Their Applications Based on Convolutional Neural Networks[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(1): 85-91.

[11] Liu H L, Zhang J, Wang Q, et al. A Survey of Content Recommendation Algorithms Based on Deep Learning[J]. Computer Engineering and Applications, 2019, 55(14): 120-126.

[12] Chen N, Wu J B, et al. Multi-Criteria Decision Recommendation System Based on Neural Networks[J]. Control Engineering, 2018, 25(5): 841-848.

[13] Liu X W, Li S, Zhao H, et al. Personalized Movie Recommendation System Based on Collaborative Filtering[J]. Computer Science, 2019, 46(5): 50-55.

[14] Li K L, Zhao L, Wang H, et al. Recommendation Algorithm Based on Attention Mechanism and Improved TF-IDF[J]. Computer Engineering, 2019, 45(6): 56-60.

[15] Tan T Z, Zhang J, et al. Hybrid Recommendation System Based on Attention Model[J]. Computer Engineering and Applications, 2020, 56(13): 37-42.

[16] Zhong Z F, Zhou D P, Zhang Y, et al. Research on Hybrid Recommendation Model Based on Least Squares Method[J]. Modern Electronics Technology, 2022, 45(17): 123-128.

[17] He M, Zhao L, Li X, et al. Collaborative Filtering Recommendation Algorithm Integrating Category Information and User Interest[J]. Computer Science and Exploration, 2020, 14(8): 1257-1264.

[18] Li M H, Zhang H, et al. Progress in Recommendation Algorithms[J]. Computer Engineering and Design, 2018, 39(3): 832-839.

[19] Huang B, Liu J, et al. Progress and Applications of Recommendation Systems[J]. Application Research of Computers, 2019, 36(11): 3417-3423.

[20] Zhang F G, Li W, et al. A Recommendation Algorithm Based on Criterion-Level Feature Cross-Fusion[J]. Computer Science, 2020, 47(5): 83-89.