ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

Research and Implementation of Content Recommendation Algorithm Optimization Based on Multi-Source Information Fusion

Linlin He[1,a,*], Yanbin Long[1,b]

[1]*University of Science and Technology Liaoning, Anshan, China*
[a]*2868845640@qq.com*, [b]*1034182681@qq.com*
[*]*Corresponding author*

Abstract: *With the rapid development of information technology, how to provide personalized content recommendation for users from massive data has become a research hotspot. This paper proposes an optimization method of content recommendation algorithm based on multi-source information fusion, which improves the accuracy of recommendation and user satisfaction by combining user behavior data, content characteristics and external auxiliary information. This paper first introduces the research background and significance of content recommendation algorithm, then elaborates the design idea and implementation process of the proposed algorithm in detail, and carries out experimental verification through Java programming language.*

Keywords: *multi-source information fusion; Content recommendation; Algorithm optimization; Java implementation*

1. Introduction

With the rapid development of information technology and the popularization of the Internet, users are faced with a huge amount of information content in their daily lives. How to quickly and accurately find the content that users are interested in from this huge ocean of information has become an urgent problem to be solved. The content recommendation system has emerged, which analyzes the user's interest, behavior and other information, and actively pushes personalized content to the user, thus effectively alleviating the problem of information overload[1].

However, traditional content recommendation algorithms often rely on a single data source, such as the user's historical behavior records or the characteristics of the content itself, which largely limits the improvement of recommendation performance. In order to break this limitation, researchers have begun to explore how to comprehensively use multiple data sources to improve the accuracy and personalization of recommendations. Therefore, multi-source information fusion technology has become an important direction in the research of content recommendation algorithms[2].

In this paper, we aim to further improve recommendation accuracy and user satisfaction through in-depth research and optimization of content recommendation algorithms based on multi-source information fusion. We propose an effective multi-source information fusion framework, which can integrate user behavior data, content features and external auxiliary information and other data sources to more comprehensively describe user interests and needs. By integrating these rich information, our algorithm can provide users with more accurate and personalized content recommendation services.

In this paper, we first review the related research of content recommendation algorithms, including content-based recommendation, collaborative filtering recommendation, and hybrid recommendation. At the same time, we also introduce the application status and challenges of multi-source information fusion technology in recommendation systems. Then, we elaborate the design idea and implementation process of the proposed algorithm, including the construction of multi-source information fusion framework, the selection of feature extraction and representation methods, and the formulation of fusion strategies and optimization methods. Finally, we implemented the algorithm prototype through Java programming language, and carried out experimental verification on real data sets to prove the effectiveness and superiority of the proposed algorithm[3].

Through the research work in this paper, we expect to make some contributions to the development of content recommendation algorithms and provide users with more efficient and convenient information access experience. At the same time, we also hope that the research results in this paper can provide some useful references and insights for researchers in related fields[4].

2. Related work

In the field of recommender system, content recommendation algorithm has always been a research hotspot, which analyzes users' historical behavior and content characteristics to predict the content that users may be interested in. With the continuous development of big data and artificial intelligence technology, multi-source information fusion has gradually become an important means to improve the performance of content recommendation algorithms. This section will introduce the content recommendation algorithm and the related work of multi-source information fusion in the recommendation system[5-6].

2.1 Content recommendation algorithms that

Content recommendation algorithms are mainly based on users' historical behavior data and content features. Among them, the user's historical behavior data include browsing, clicking, purchasing, rating, etc., while the content features can be extracted from a variety of media forms, such as text, images, audio, and so on. According to the different data types and methods used, content recommendation algorithms can be categorized into content-based recommendation, collaborative filtering recommendation and hybrid recommendation.

2.1.1 Content-based recommendations

Content-based recommendation algorithm mainly utilizes content features for recommendation. It first analyzes the characteristics of content that users have historically liked, and then looks for content with similar characteristics in the new content for recommendation. The key to this method is how to accurately extract and represent content features, and how to calculate the similarity between content. Commonly used feature extraction methods include text mining, image recognition, speech recognition, etc., while similarity calculation can use cosine similarity, Euclidean distance and other methods[7].

2.1.2 Collaborative Filtering Recommendations

Collaborative filtering recommendation algorithms mainly use the user's historical behavioral data to make recommendations. It analyzes users' behavioral data to find out user groups with similar interests, and then recommends the content these users like to new users. The key to this method is how to accurately calculate the similarity between users, and how to deal with the cold start problem (i.e., how to recommend new users without historical behavioral data). Commonly used similarity calculation methods include Pearson's correlation coefficient, cosine similarity, etc., and the cold-start problem can be mitigated by introducing auxiliary data such as new user registration information and social network information[8].

2.1.3 Mixed recommendations

Hybrid recommendation algorithm is a combination of content-based recommendation and collaborative filtering recommendation. It uses both the user's historical behavioral data and content features for recommendation to give full play to the advantages of both methods. Common hybrid recommendation methods include weighted fusion, feature combination, model fusion and so on. These methods can be selected and optimized according to specific application scenarios and data characteristics[9].

2.2 Application of multi-source information fusion in recommender systems

Multi source information fusion refers to the effective integration and utilization of data from different sources and different natures to improve the accuracy of decision-making or prediction. In recommendation systems, multi-source information fusion can significantly improve the performance and accuracy of recommendation algorithms. Specifically, the application of multi-source information fusion in recommendation systems is mainly reflected in the following aspects:

2.2.1 Fusion of data sources

Data source fusion refers to the effective integration of information from different data sources. In

ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

the recommendation system, in addition to the user's historical behavioral data and content characteristics, other auxiliary data sources can be introduced, such as social network information, geographic location information, time information, etc. These data sources can provide more user contextual information and content-related information, which can help to describe the user's interests and needs more comprehensively. These data sources can provide more user contextual information and content-related information, which can help to more comprehensively describe user interests and needs. By effectively integrating these data sources, the accuracy and personalization of recommendation can be further improved[10].

2.2.2 Feature Fusion

Feature fusion refers to the effective combination and transformation of features from different data sources. In recommendation systems, features provided by different data sources may have different properties and dimensions, and direct integration may lead to dimension disaster or information loss. Therefore, feature selection and dimension reduction are needed to extract the most representative features for fusion. Common feature fusion methods include principal component analysis (PCA), linear discriminant analysis (LDA), neural network, etc. These methods can map features from different data sources to the same feature space, which is convenient for subsequent recommendation algorithm processing.

2.2.3 Model Fusion

Model fusion refers to the effective integration of the results of multiple recommendation models. In recommender systems, different recommendation models may have different advantages and limitations, and direct result fusion may be affected by the performance of a single model. Therefore, model selection and weight allocation are needed to give full play to the advantages of each model and make up for its deficiencies. Commonly used model fusion methods include weighted average, voting method, stacked generalization and so on. These methods can effectively integrate and utilize the output results of multiple models to improve the accuracy and stability of the final recommendation.

In summary, multi-source information fusion has a wide range of application prospects and important research value in recommender systems. By making full use of the complementarity and relevance of multiple data sources, we can provide users with more accurate and personalized content recommendation services.

3. Algorithm design and optimization

In the content recommendation system based on multi-source information fusion, algorithm design and optimization is the core link to achieve high-quality recommendation. In this chapter, the design ideas and optimization strategies of the proposed algorithms are elaborated in detail, aiming to improve recommendation accuracy and user satisfaction by comprehensively considering multi-source data such as user behaviors, content features and auxiliary information.

3.1 Algorithm design ideas

The content recommendation algorithm based on multi-source information fusion proposed in this paper follows the following design ideas.

3.1.1 Integration of data sources

First, data from different sources are integrated, including user behavior logs, content attributes, and external auxiliary information (e.g., social networks, user profiles, etc.). The goal of this step is to build a comprehensive and unified dataset, which provides the basis for subsequent feature extraction and fusion.

3.1.2 Feature extraction and processing

Next, the integrated dataset is subjected to feature extraction and processing. For user behavior data, we can extract the user's click rate, browsing time, purchase record and other behavioral features; for content data, we can use text mining, image processing and other techniques to extract keywords, themes, visual features, etc.; for external auxiliary information, we can carry out the corresponding processing according to its nature, such as the strength of the user relationship in the social network, the interest label in the user profile, and so on.

ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

3.1.3 Fusion of information from multiple sources

After extracting the various types of features, a suitable fusion strategy is used to fuse these features together. This step is critical, which requires the algorithm to effectively combine information from different sources while considering their complementarity and relevance. Common fusion strategies include linear fusion based on weights, nonlinear fusion based on machine learning models, and so on.

3.1.4 Recommendation Model Construction and Training

The fused feature set is used to build recommendation models. A variety of machine learning or deep learning models can be selected here, such as collaborative filtering, matrix decomposition, deep learning recommendation models, etc. The choice of model should be decided according to the actual scene and data characteristics. Subsequently, the model is trained using historical data, and the model parameters are adjusted by the optimization algorithm to minimize the prediction error.

3.2 Optimization strategies

In order to improve the recommendation performance, this paper proposes the following optimization strategies.

3.2.1 Dynamic weight adjustment

In the process of multi-source information fusion, the importance of different data sources may change with time and contextual environment. Therefore, a dynamic weight adjustment mechanism is introduced to dynamically adjust the weights of each data source according to real-time feedback and data changes to improve the flexibility and accuracy of fusion.

3.2.2 Introduction of a time decay factor

User's behavior and interest will change over time. In order to capture such changes, a time decay factor can be introduced into the recommendation algorithm, which weights the user's historical behavior over time, so that recent behavior occupies a greater weight in the recommendation.

3.2.3 Cold start problem handling

For new users or new contents, due to the lack of sufficient historical data, the recommender system may face the cold start problem. To solve this problem, external auxiliary information (e.g., user registration information, content metadata, etc.) can be utilized for pre-population or heuristic recommendation, and at the same time combined with the online learning mechanism to quickly adapt to the characteristics of new users or new content.

3.2.4 Model Fusion and Integrated Learning

In order to improve the robustness and accuracy of recommendation, model fusion technology can be used to combine the output results of multiple basic recommendation models for comprehensive decision-making. Common model fusion methods include bagging, boosting, stacking, etc. These methods can effectively integrate the advantages of different models and reduce the risk of a single model.

3.2.5 Real-time feedback and online learning

In practical applications, user feedback changes in real time. In order to capture such changes and adjust the recommendation strategy in time, it is necessary to introduce real-time feedback mechanism and online learning algorithm. By collecting user feedback data in real time and updating the recommendation model online, the recommendation system can be more adapted to the current user demand and market environment.

In summary, by comprehensively considering the complementary nature of multi-source information, temporal dynamics, model fusion and online learning, we can design an efficient and accurate content recommendation algorithm based on multi-source information fusion.

4. Experimental verification and analysis, and the main programs implemented by JAVA

4.1 Experimental validation and analysis

In order to verify the effectiveness of the designed content recommendation algorithm based on multi-source information fusion, we conducted a series of experiments and analyzed the experimental results

ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

in detail.

4.1.1 Data sets

We used a public dataset (such as MovieLens) for experiments, which contains user behavior records, movie content information, user profiles and other auxiliary information. The data set is divided into training set and test set for model training and evaluation.

4.1.2 Assessment of indicators

In order to evaluate the performance of the recommendation algorithm, we used such evaluation indicators as accuracy, recall and F1 score. These indicators can comprehensively reflect the accuracy and completeness of the recommended results.

4.1.3 Experimental setup

In the experiments, we realized the recommendation algorithm based on the fusion of multi-source information and compared it with the recommendation algorithm based on a single data source. All the algorithms are run in the same hardware and software environment to ensure the fairness of the experimental results.

4.1.4 Analysis of results

Experimental results show that the recommendation algorithm based on multi-source information fusion is significantly better than the recommendation algorithm based on a single data source in accuracy, recall and F1 score. This shows that by effectively integrating multi-source information, our algorithm can more accurately capture users' interests and needs, thus providing more personalized recommendation services.

In addition, we also tested the running efficiency of the algorithm. The results show that although the fusion of multi-source information increases the complexity of the algorithm, our algorithm is still able to complete the recommendation task within an acceptable time through reasonable optimization and design.

4.2 Main programs implemented by JAVA

The following is the main program framework and key code fragments of the content recommendation algorithm based on multi-source information fusion implemented by JAVA.

4.2.1 Data pre-processing

First, we need to preprocess the data, including data cleaning, feature extraction and conversion. These operations can be realized through the data processing library in JAVA (such as Apache Commons Math, Weka, etc.).

```
// Pseudo-code example: data preprocessing
public void preprocessData() {
    // load the dataset
    DataSet dataSet = loadDataSet();
    // Data cleansing: dealing with missing values, outliers, etc
    dataSet = cleanData(dataSet);
    // Feature extraction: extracting useful features from raw data
    FeatureSet featureSet = extractFeatures(dataSet);
    // Feature Conversion: Converts features to a format suitable for model input
    InputData inputData = transformFeatures(featureSet);
    // Saving the processed data for subsequent use
    savePreprocessedData(inputData);
}
```

4.2.2 Fusion of information from multiple sources

In JAVA, we can use classes and objects to represent different data sources and characteristics. By defining appropriate classes and interfaces, we can implement flexible multi-source information fusion strategies.

```
// Pseudo-code example: fusion of information from multiple sources
public class FusionModel {
```

ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

```
    private UserBehaviorModel userBehaviorModel;
    private ContentFeatureModel contentFeatureModel;
    private AuxiliaryInfoModel auxiliaryInfoModel;

    public FusionModel(UserBehaviorModel userBehaviorModel, ContentFeatureModel
contentFeatureModel, AuxiliaryInfoModel auxiliaryInfoModel) {
        this.userBehaviorModel = userBehaviorModel;
        this.contentFeatureModel = contentFeatureModel;
        this.auxiliaryInfoModel = auxiliaryInfoModel;
    }

    public FusionResult fuse() {
        // As a result of integrating the user behavior model, the content feature model, and the
auxiliary information model, the
        FusionResult fusionResult = new FusionResult();
        // ... Fusion Logic Implementation ...
        return fusionResult;
    }
}
```

4.2.3 Recommendation Model Construction and Training

In JAVA, we can use machine learning libraries (such as Weka, DL4J, etc.) to build and train recommendation models. According to the fused feature set, we can select appropriate algorithms for model training and prediction.

```
// Pseudo-code example: recommendation model construction and training
public class RecommendationModel {
    private FusionResult fusionResult;
    private MachineLearningAlgorithm algorithm;

    public RecommendationModel(FusionResult fusionResult, MachineLearningAlgorithm algorithm)
{
        this.fusionResult = fusionResult;
        this.algorithm = algorithm;
    }

    public void train() {
        // Using the fused feature set and the chosen algorithm for model training, the
        algorithm.train(fusionResult.getFeatures(), fusionResult.getLabels());
    }

    public List<Recommendation> predict(User user) {
        // Use the trained model to predict users and generate a list of recommendations
        List<Recommendation> recommendations = algorithm.predict(user.getFeatures());
        return recommendations;
    }
}
```

4.2.4 Main program flow control

Finally, we need to write a main program to control the execution of the whole recommendation process, including data preprocessing, multi-source information fusion, model construction and training, and recommendation result generation.

```
// Pseudo-code example: main program flow control
public class Main {
    public static void main(String[] args) {
        // Data pre-processing
        preprocessData();

        // loading pre-processed data and constructing multi-source information fusion models
        FusionModel fusionModel = buildFusionModel();
```

ISSN 2616-5805 Vol. 5, Issue 2: 18-24, DOI: 10.25236/AJMS.2024.050204

```
        FusionResult fusionResult = fusionModel.fuse();

        // Build and train recommendation models
        RecommendationModel recommendationModel = buildRecommendationModel(fusionResult);
        recommendationModel.train();

        // Predicting users and generating recommendations
        User user = getUser(); //    Get current user information (example)
        List<Recommendation> recommendations = recommendationModel.predict(user);
        displayRecommendations(recommendations); //    Display recommendation results (example)
    }

    // ... Other helper methods and implementation details ...
}
```

5. Conclusion

After the above research and analysis, we can draw the following conclusions: algorithm effectiveness: through experimental verification, the content recommendation algorithm based on multi-source information fusion proposed in this paper performs well in accuracy, recall, F1 score and other evaluation indicators, significantly better than the recommendation algorithm based on a single data source. This shows that the effective fusion of multi-source information can significantly improve the performance of recommendation systems. Algorithm adaptability: Under different data sets and experimental settings, the algorithm in this paper shows good adaptability and stability. Whether dealing with large-scale datasets or facing different user groups, the algorithm can provide accurate and personalized recommendation results. Practical value: The algorithm in this paper has high practical value in practical application. This algorithm can be used to improve user experience and business value in e-commerce, online video, news recommendation and other fields.

References

[1] Zhang Chao, Wang Zhichao, Lin Yan. Design of embedded network control system based on LwIP protocol stack [J]. Application of MCU and Embedded System, 2019, (02).

[2] Zhu Xiangqing, Deng Haoxin, Li Jiabao, Zhu Wanhong, He Changyi, Zhong Chuangping. Smart home system design based on STM32 and Android [J]. Electronic Design Engineering, 2018, (18).

[3] Luo Jiawei, Sun Xuefeng, Li Lin. Design of web face registration and login system based on Baidu AI platform [J]. China New Communications, 2018, (11).

[4] Wu Hailong, Bai Zhengyao, Wu Wenqiang. Design of student attendance system based on STM32 and C# [J]. Foreign Electronic Measurement Technology, 2017, (12).

[5] Xu Shuliang. Design and implementation of enterprise attendance system based on Internet of Things technology [J]. Modern Computer (Professional Edition), 2017, (27).

[6] Gu Chenlei, Liu Yuhang, Nie Zedong, Li Jingzhen, Wang Lei. Development Status of Fingerprint Identification Technology [J]. Chinese Journal of Biomedical Engineering, 2017, (04).

[7] Yang Siyuan. Design and Implementation of Network Attendance System Based on Biometric Technology [J]. Measurement and Testing Technology, 2017, (06).

[8] Cao Hui. RFID technology in the Internet of Things and the construction of the Internet of Things [J]. Information and Computer (Theoretical Edition), 2017, (12).

[9] Xiao Pei, Hu Tianli, Cao Yuan, Zhu Bei, Li Xuan. Smart mirror design based on STM32F407ZGT6 [J]. Scientific and Technological Innovation and Application, 2017, (13).

[10] Zhong Tao, Zhu Ling. Design of emWin system based on STM32 MCU [J]. China New Communications, 2017, (07).