

Original Research Article

A simple advertisement image recommendation system based on user profiles

Enhui Yu

Liaoning University of Science and Technology, Anshan, Liaoning, 114051, China

Abstract: The purpose of this paper is to design and implement a simple advertisement image recommendation system based on user profiles. The system constructs a user profile by analyzing the user's historical behavior, interests and preferences, and makes use of advanced machine learning models for personalized recommendation of advertising images. This paper firstly introduces the theoretical and technical background of the recommendation system, and then elaborates the requirements, outline design and detailed design of the system. In order to improve the recommendation effect, this paper also explores the click rate prediction method based on Z-NFFM (a kind of improved neural network factorization machine) model and the advertisement cold start strategy based on LightGBM-LR (Light Gradient Boosting Machine combined with logistic regression) model. The experimental results show that the system can effectively improve the click-through rate of advertisements, optimize the user experience, and bring higher conversion efficiency for advertisers.

Keywords: User profiling; Z-NFFM model; LightGBM-LR model; Click-through rate prediction; Advertisement cold start

1. Preamble

With the rapid development of the Internet, digital advertising has become an important means of enterprise marketing. In the field of advertising, the traditional creative design and delivery of advertising images rely on manual experience, it is difficult to meet the user's personalized needs and preferences, which can easily lead to user resistance and low conversion rate of advertising. In today's digital era, personalized content recommendation has become an important means to improve user experience and business marketing results. This project aims to build an intelligent advertisement image recommendation system, which can not only accurately push personalized advertisement images to users according to their interests, behavioral patterns and contextual environments, but also ensure that the relevance and attractiveness of the advertisements reach the optimal level through a series of complex data analysis and technological reasoning. By improving the relevance and attractiveness of advertisements, not only the user's satisfaction can be enhanced, but also the effect of advertisement delivery can be significantly improved, thus creating a win-win situation for both users and advertisers^[1].

2. Theories and techniques related to recommendation systems

2.1. Overview of recommender systems

Recommender systems, as a form of information filtering, aim to recommend content or products that may be of interest to a user from a large number of available options based on the user's interests, historical behavior, or other relevant information. The core goals of recommender systems are to improve user satisfaction, increase user engagement, and generate higher revenues for content or service providers^[2].

Basic concept: recommender systems generate personalized recommendation lists by analyzing the

interaction history between users and items (e.g., merchandise, movies, music, etc.) and predicting the degree of user preference for untouched items. Classification: Collaborative Filtering: Recommendation based on the similarity of users or items. User-user collaborative filtering recommends items based on similar user preferences; item-item collaborative filtering recommends items based on similarities between items. Content Recommendation: By analyzing the content characteristics of the items (such as text description, tags, metadata, etc.) and the user's historical preferences, recommending items that match the user's interests. Hybrid recommendation: combining the advantages of collaborative filtering and content recommendation, using multiple recommendation strategies to generate comprehensive recommendation results to improve the accuracy and diversity of recommendations. Recommendation based on association rules: analyze the association rules in the transaction data to find out the association relationship between the items, so as to make recommendations. Recommendation based on deep learning: using deep learning models (e.g. neural networks) to automatically learn complex feature representations of users and items to improve the accuracy of recommendations^[3].

2.2. User profile construction

User profile is a comprehensive and detailed description of user characteristics, which is constructed based on the user's historical behavior, preferences, attributes and other information, and is the basis for recommender systems to achieve personalized recommendations. Concept: User profile is a multi-dimensional collection of user information, which integrates the user's static attributes (e.g., age, gender, geographic location) and dynamic behaviors (e.g., browsing records, purchase history, comments, etc.) to form a comprehensive understanding of the user. Construction process: Data collection: Collect user data from multiple sources (e.g. user registration information, transaction records, social media activities, etc.). Data pre-processing: Clean the data, deal with missing values, remove noise, and ensure data quality. Feature engineering: Extract and select features that are useful for user profile construction, which may include statistical features, text features, image features, etc.^[4-5].

3. A simple advertisement image recommendation system based on user profiles

3.1. System requirements

In order to realize an efficient, secure and user-friendly advertisement image recommendation system, we need to define the functional requirements, performance requirements, and security and privacy requirements of the system. These requirements will guide the design and implementation of the system to ensure that the system meets the expectations of users and advertisers.

functional requirements

User Profile Construction.

Data collection: The system should be able to automatically collect user data from multiple sources (e.g., user registration information, browsing history, transaction records, social media activity, etc.)^[6-7].

Feature extraction: based on the collected data, the system should be able to extract features that are useful for the construction of user profiles, including the user's basic information, interests and preferences, and behavioral patterns.

Model training: Using machine learning or deep learning algorithms to model user data to form an accurate user profile.

3.2. System outline design

The overall architecture of a simple advertisement image recommendation system based on user profiles can be divided into four main layers: data layer, processing layer, recommendation layer and display layer. The following is a brief description of the system architecture diagram and its layers.

Data Layer.

Data sources: including user registration information, historical behavior data, advertising image information, social media data, etc.

Data storage: using relational databases (e.g., MySQL) or non-relational databases (e.g., MongoDB) to store structured and unstructured data. For large-scale data, may also need to use distributed storage systems (such as Hadoop HDFS).

Processing layer.

Data cleaning and pre-processing: removing duplicate data, dealing with missing values, standardizing data formats, etc.

Feature engineering: extracting useful features from raw data and constructing feature vectors for user profiles.

Data analysis: In-depth analysis of user behavior to identify user interests and preferences.

In response to the above description of the recommendation algorithm module, we can provide Python-based pseudo-code examples to demonstrate the core implementation logic. Please note that since the implementation details of specific recommendation algorithms (e.g., content-based recommendation, collaborative filtering recommendation, or hybrid recommendation) may be more complex and dependent on specific libraries and data structures, the following code will focus on showing the overall process and key steps rather than the complete algorithm implementation.

```
# Data preparation
def load_data():
    # Assuming that user behavior data and ad image data have been stored in files in some way
    user_behavior_df = pd.read_csv('user_behavior.csv')
    ad_image_df = pd.read_csv('ad_image.csv')
```

```
# Data preprocessing and feature extraction (specific implementation omitted here)
# ...
```

```
return user_behavior_df, ad_image_df
```

Algorithm selection (Random Forest is used here as an example, other models may be chosen in practice)

```
def select_algorithm(data_characteristics, business_requirements):
    # Selection of appropriate algorithms based on business needs and data characteristics
    # It is assumed here that a random forest classifier was chosen as an example
    return RandomForestClassifier()
```

```

# Model training and parameter tuning
def train_and_tune_model(algorithm, X_train, y_train):
    # Parameter definition (this is an example, in practice it should be defined according to the needs of the
    algorithm)
    param_grid = {
        'n_estimators': [10, 50, 100],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10]
    }

    # Selection of tuning method (grid search is used here as an example)
    tuner = GridSearchCV(estimator=algorithm, param_grid=param_grid, cv=5, scoring='accuracy')

    # Model training and tuning
    tuner.fit(X_train, y_train)

    # Output optimal parameters and models
    print(f"Best parameters found: {tuner.best_params_}")
    best_model = tuner.best_estimator_

    return best_model

# Recommendation generation
def generate_recommendations(best_model, new_data):
    # Use the trained model to make predictions on new data (this is an example, in practice it may involve
    more complex recommendation logic)
    predictions = best_model.predict(new_data)
    # Convert predictions into a list of recommendations (omit specific implementation here)
    # ...
    return recommendations_list

# Effectiveness evaluation
def evaluate_model(best_model, X_test, y_test, evaluation_metrics):
    # Model evaluation using test data (accuracy and AUC are examples here)
    predictions = best_model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
    auc = roc_auc_score(y_test, predictions) # Note: for categorization problem, AUC is a common
    evaluation metric, but click-through rate and conversion rate may need custom function to calculate

```

```

# Output the results of the assessment
print(f'Model Accuracy: {accuracy}')
print(f'Model AUC: {auc}')

# Model optimization based on business requirements (implementation omitted here)
# ...

# The main function
def main():
# Load data
user_behavior_df, ad_image_df = load_data()

# Data preprocessing and feature extraction (specific implementation is omitted here, but should include
conversion of raw data into a format acceptable to the model)
# ...
# Suppose X is the feature matrix,y is the target variable,X_train, X_test, y_train, y_test are the training
set and test set respectively

```

Data preparation: In practical applications, the data preparation stage may involve complex processes such as data cleaning, feature extraction and feature selection. These processes are crucial for improving the performance of recommendation algorithms.

Algorithm selection: choose a suitable recommendation algorithm according to business needs and data characteristics. For example, for image-rich advertisement recommendation scenarios, content-based recommendation algorithms can be considered; for user behavior data-rich scenarios, collaborative filtering recommendation algorithms can be considered; in order to synthesize the advantages of multiple algorithms, hybrid recommendation algorithms can be considered.

Model training and parameter tuning: Model training and parameter tuning are key steps to improve the performance of recommendation algorithms. In practical applications, it may be necessary to try multiple tuning methods and parameter combinations to find the optimal model configuration.

4. Click-through rate prediction based on the Z-NFFM model

4.1. Introduction to the Z-NFFM model

Fundamentals

Z-NFFM (Zoom-in Neural Field-aware Factorization Machine) model is a click-through rate prediction model that combines the advantages of neural network and field-aware factorization machine (FFM). Based on the traditional factorization machine (FFM), it introduces the domain-aware features and the nonlinear expressive ability of neural networks, which can better capture the interactions between features and performs well especially on highly sparse datasets.

structural characteristics

The structure of Z-NFFM model mainly includes input layer, Embedding layer, Bi-Interaction layer, hidden

layer and output layer. Among them, the Embedding layer is used to convert sparse features into dense vectors with low dimensions; the Bi-Interaction layer realizes feature crossover and captures the second-order feature interactions through the element product operation; the Hidden layer learns the higher-order nonlinear feature interactions by using the neural network; and the Output layer outputs the estimation results of the click-through rate.

4.2. Model training and evaluation

Data preparation

Before model training, data preparation is required. This includes the collection of user behavior data, advertising information data, etc., and data cleaning, preprocessing and feature extraction. Data cleaning mainly removes duplicate data, deals with missing values and outliers, etc.; data preprocessing includes data standardization, normalization, etc.; feature extraction extracts useful features from the original data for model training.

feature engineering

Feature engineering is one of the key steps in model training. In the Z-NFFM model, it is necessary to select appropriate features and perform feature combination. The selection of features should be based on business requirements and data analysis results, and features with strong correlation with the target variable (i.e., click rate) should be selected. At the same time, feature crossover is also needed to capture the interaction between features. Feature crossover can be obtained by manual combination or automatic learning using deep learning models.

5. Concluding remarks

We have deeply explored the problem of advertisement cold launch based on LightGBM-LR model, and elaborated the design and implementation of cold launch strategy from the advantages of the combination of LightGBM and logistic regression, and verified the validity and practicability of the model through experiments. The following is a summary and review of the content of this paper, as well as an outlook on the future research direction.

In conclusion, this paper has achieved some results on the problem of advertisement cold start, but there are still many issues that deserve in-depth research. In the future, we will continue to explore more efficient and accurate recommendation algorithms and strategies to cope with the increasingly complex recommendation scenarios and user needs. Meanwhile, we also look forward to communicating, learning and advancing with our peers to promote the development of recommendation system.

(Liaoning University of Science and Technology Innovation Training Program for Undergraduates 2025)

References

- [1] Li Xianwei . Research on Spark-based recommender system [D]. Hangzhou : Zhejiang University of Technology ,2017:14.
- [2] Jiao Jian . Research on collaborative filtering recommendation algorithm based on Spark [J]. Computer Programming Skills and Maintenance ,2020(3):40-41.
- [3] Liu Shanshan . Dynamic user personalized recommendation based on hybrid collaborative filtering in big

- data [J]. *Software Engineering* ,2019,22(3):16-19.
- [4] Shi Aiwu , Li Denggui . Design and Implementation of Movie Recommendation System Based on Spark and Microservice Architecture [J]. *Computer Knowledge and Technology* ,2021(5):78-80.
- [5] Shin Jin-Xiang , Bao Mei-Ying . Optimized recommendation algorithm gate based on user clustering and item segmentation [J]. *Computer System Applications* ,2019,28(6):159-164.
- [6] Q. Tong , Q. Liu , S. Xu , et al. Research on Intelligent Recommendation System for E-commerce Based on Related Items [J]. *Enterprise Technology and Development* ,2019(12):79-80.
- [7] Wang Tengyu . Analysis and design of e-commerce personalized recommendation system in the era of big data [J]. *Think Tank Times* ,2020(8):132-133.