Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 1, No. 2, 2022

Design and Implementation of Insurance Product Recommendation System

JiaPeng Zhang*

Qingdao University, Shandong Province, China.
* **Corresponding author**: Email: porter_zhang@163.com.

Abstract: In this paper, a hybrid collaborative filtering recommendation algorithm with Latent Dirichlet Allocation (LDA) and Alternating Least Squares (ALS) is proposed, which can effectively alleviate the cold start problem in traditional recommendation system and improve the accuracy of recommendation. At the same time, real-time recommendation according to the real-time behavior of users is realized in the real-time recommendation system, which effectively improves effectiveness of the recommendation results. In the system construction, user historical data is collected and analyzed by big data tools. Based on the Spark framework, the offline and real-time recommendation of insurance products are effectively combined and realized, with personalized recommendation as an important function. The designed system can improve the purchase rate and experience of users.

Keywords: Big Data Analysis; Insurance Product Recommendation; Hybrid recommendation algorithms; Spark Framework.

1. Introduction

With the improvement of people's living standards, more and more people began to consider the rational allocation of assets, and purchasing insurance became one of the important choices for people. At present, there are nearly a thousand insurance sales companies in China, and there are as many as tens of thousands of insurance products, and policyholders face a situation in which they have no way to start when choosing insurance products on their own. In the traditional insurance sales industry, there are often problems such as customers not being able to accurately find insurance products that meet their own needs, and insurance salespeople recommending insurance products for users with strong subjective factors. With the advent of the Internet era, the traditional insurance sales industry is also relying more and more on the application of Internet technology, and a large amount of user behavior data is generated in the process of online product sales. It is of great significance to design and implement a set of personalized recommendation system for insurance products based on big data technology, which not only improves the product recommendation efficiency of insurance companies, but also provides users with insurance products that are more in line with their own needs and improves the user experience.

At present, in the traditional recommendation system, usually due to the small number of behavior data of new users, the sparseness of the prediction score matrix is large, resulting in poor recommendation effect, which usually becomes a cold start problem of the recommendation system; In addition, with the increase of user behavior data and the increase of the number of recommended items, the calculation complexity of the recommendation algorithm increases, resulting in poor timeliness of the recommendation results.

In view of this, this paper designs and implements an insurance product recommendation system that integrates offline recommendation and real-time recommendation. In the offline recommendation module, a collaborative filtering hybrid recommendation algorithm of Latent Dirichlet Allocation (LDA) and alternating Least Squares (ALS) is used, which combines the advantages of document topic algorithm and alternating least squares algorithm to alleviate the cold start problem caused by user information loss. In the real-time recommendation module, the real-time stream processing operation combined with the spark streaming framework of the offline module is applied to achieve the effect of real-time recommendation according to the real-time behavior of the user, thus solving the problem of poor timeliness of the traditional recommendation system.

2. Research Situation

The recommendation system has a long history, recommendation technology is the core of the recommendation system, and the use of appropriate recommendation technology can effectively improve the user experience. Huang Liwei et al. summarized the deep learning technology of the current mainstream recommendation system, and proposed that the main task of the recommendation system is to combine massive data with the user's preference needs for model construction. Zhang L et al. based on the traditional matrix decomposition model, combined the advantages of the asymmetric factor model by adding user-evaluated item information to the loss function, and improved the model prediction accuracy. Chai Z Y et al. modeled the recommendation system as a multi-objective optimization problem, first using singular value decomposition to obtain the recommendation list, and then using the multi-objective immunization algorithm to optimize the recommendation list.

At present, the recommended system is used in a wide range of business scenarios. In the field of news recommendation, Wu Yanwen et al. people based on user collaborative filtering technology, by obtaining mobile terminal data to continuously correct user preference values, while punishing users with similarity values through the impact of news popularity, and finally refer to the nearest neighbor set to top-N sort the unread news of users to obtain recommendation results. Li Bo fully considers the user habits, combines popularity and characteristics into a variety of recommendation algorithms, and finally obtains the recommendation results through the method of sorting learning. When recommending books, Wang Shunzhen and others started from the diversification of readers' needs and divided the recommendation system into three parts: data collection subsystem, data processing subsystem and data reporting subsystem, analyzing and summarizing the needs of different users, and achieving the purpose of book recommendation through the aggregation strategy of social networks. He Sheng et al.When designing the recommendation system, they first analyzed the requirements of literature recommendation from the aspects of literature search, browsing, and analysis, and then realized the association of various literatures to achieve the purpose of optimizing literature search through "string matching" and "similarity measurement".

3. System Design

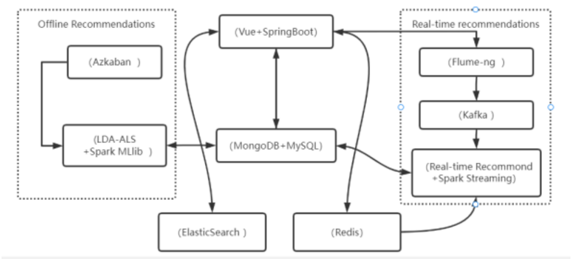

Figure 1. System Design

Based on the real user record data set of an insurance technology company and based on its real business data architecture, this system constructs a personalized recommendation system for this insurance product. The system architecture diagram is shown in Figure 1, which includes three modules: data processing, offline recommendation and real-time recommendation, and comprehensively uses the LDA-ALS recommendation algorithm to provide hybrid recommendation, and completes the closed-loop business from front-end application, background service, algorithm design and implementation, and platform deployment.

3.1. Data Processing

Based on the real user record data set of an insurance technology company and based on its real business data architecture, this system constructs a personalized recommendation system for this insurance product. The system architecture.

The data storage section is subdivided into four modules: comprehensive business services, business databases, search services, and cache services. The integrated service uses Vue to display the user's front-end screen, and the back-end adopts the SpringBoot framework for business logic processing. The business database adopts the document database MongoDB and the relational database MySQL as the database, which mainly saves the business logic data; The search service uses ElasticSearch to implement a content-based search service; The caching service implements the high-speed requirements for the real-time recommendation system part of the data by using Redis as the caching database.

3.2. Offline Recommendations

The offline recommendation part is divided into work scheduling service and offline recommendation service, of which the work scheduling service acts on the offline recommendation part, and Azkaban is used to schedule the algorithm at fixed intervals. The offline recommendation service uses Spark MLlib for service construction, data calculation based on LDA-ALS algorithm, obtains the user-product prediction scoring matrix, and stores the results into the service database to realize the offline recommendation of the product.

3.3. Real-time Recommendations

The real-time recommendation section is divided into log collection service, message buffering service, and real-time recommendation service. In particular, the log collection service uses Flume-ng to capture the one-time scoring behavior of users in the service platform for products and transmits them to the Kafka cluster in real time. The message buffering service uses Kafka to cache the data, receives the data collection request from Flume, and then passes the data to the real-time recommendation system part of the project; The real-time recommendation service uses the Spark Streaming framework to process the data of real-time recommendations by receiving data cached in Kafka, a designed recommendation algorithm (Real-time Recommendation), and the results are merged and updated to the MongoDB database.

4. System Implementations

4.1. Offline Recommended Module Design

The offline recommendation service is to calculate and save periodic results through the offline recommendation algorithm by using all the historical data owned by the user, and the frequency of the recommendation result change depends only on the frequency of offline algorithm scheduling. The offline recommendation service mainly obtains some indicators that can be counted and calculated in advance (such as the popularity of insurance products, the viewing rate of insurance types, product search rankings, etc.), and provides data support for real-time recommendation modules and front-end services.

The data processing flow of the offline recommendation module is shown in Figure 2, the user in the process of system use through the front-end interface to achieve the call to the background service, the back-end service records the user's use logs, through the big data tool for log collection and log cleaning and storage to the system's data warehouse, based on LDA-ALS collaborative filtering recommendation algorithm for data calculation, get the user's product recommendation results and store the calculation results to the database, and finally recommend the processed results to the user through the front-end. This makes it possible to recommend personalized insurance products for them.

4.2. Offline Recommendation Module Algorithm Core

The LDA (Latent Dirichlet Allocation) topic model [21], also known as the three-tier Bayesian probability model, consists of a three-tiered structure of words, topics, and documents. LDA uses the method of bag of words, the algorithm can give the topic of each document in the document set in the form of a probability distribution, so that after analyzing some documents to extract their topic distribution, you can perform topic clustering or text classification according to the topic distribution. Suppose that the document collection W has m documents with n words per document, and the collection W contains k topics, using μ and η to represent the probability distribution of documents—topics and topics—words, respectively.

In the recommendation system, the user's scoring matrix for the product (hereinafter referred to as the user scoring matrix) can be built according to the user's score on the product (the user's degree of interest in the product) and other data such as the user's past operation log. The user rating matrix is a sparse matrix, and the elements in the matrix that are not empty represent the user's rating of a product in the past time, and the empty elements indicate that the user has not interacted with the product. Through the known user scoring matrix can be calculated by the ALS (Alternating Least Squares) algorithm to build a prediction scoring matrix, the matrix is a full matrix, the prediction scoring matrix will be the user scoring matrix empty elements filled in the prediction score, after calculating the prediction scoring matrix can be based on the user's prediction score for the project to personalize the user to recommend products.

The ALS algorithm splits the user scoring matrix into two matrices, that is, the user scoring matrix is broken down into the user's preference matrix P for the implied features of the item and the implied feature matrix Q contained in the project. Suppose the user product scoring matrix is R, and now there are i user j products, if you want to find l hidden features, the task now is to find the matrix P and Q so that the product of P and Q is approximately equal to R, that is, the user's product scoring matrix R is decomposed into two low-dimensional matrices multiplied, such as Equation (1):

$$\widehat{\mathrm{R}} = \mathrm{P} \times \mathrm{Q}^{\mathrm{T}}$$

Cold start usually refers to the situation when there is little or no new user data in the system, and the single ALS algorithm has a poor recommendation effect in the case of cold start [23], so the combination of LDA and ALS algorithm improves the performance of the algorithm. As shown in Table 1, the system divides the user's interest in the product "degree" into different behavioral association scores (-1 to 5 points), the higher the score, the higher the user's interest in the product, through the extraction of the user's behavior log to obtain the user's behavior score of the product, based on the recorded log data through the LDA-ALS algorithm model data calculation for personalized product recommendations.

The system uses the user's message and comment data as the data source of the LDA algorithm. The log file of an insurance technology company used by this system contains the user's article comments, dynamic messages, etc., and the system uses it as the basic data source for users when entering the product purchase module. The data processing process of the LDA-ALS model is shown in Figure 4, first extract the user's comment text, obtain the text topic through the LDA algorithm, that is, get the type of product that the user may be interested in, you can initially construct the user-product scoring matrix to alleviate the cold start problem, and then apply the ALS algorithm to calculate the user's prediction scoring matrix, and finally recommend the product with the higher prediction score for the user according to the derived prediction scoring matrix.

5. Experimentation and Analysis

5.1. Experimental Environment and Data

The dataset used in this paper is from an insurance technology company, including 90 days of APP logs totaling about 55 GB, including about 8,000 registered users, 300 insurance products and corresponding attributes, about 200,000 records. The first 70 days of logs are used as the training set, and the last 20 days of logs are used as test sets to test the user's effect under the use of the recommendation system, and the indicator is the proportion of users who click on the recommended results.

5.2. Evaluation Criteria for Experiments

For the evaluation of the recommendation effect of different recommendation algorithms experiments using root mean square error (RMSE) as the evaluation criterion for model evaluation, RMSE is to calculate the deviation between the user's prediction score and the user's actual score to measure the accuracy of the prediction, as shown in the formula (2), where N represents the number of products, observed d represents the actual score, predicted as the prediction score, the smaller the overall RESM of the recommendation algorithm, the better the recommendation effect.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(observed_t - predicted_t)^2}$$

5.3. Recommended Performance Evaluation

This experiment evaluates the RMSE value of ALS and LDA-ALS two different recommendation algorithms under different iterations, the abscissa represents the number of iterations, the ordinate represents the RMSE value of each iteration of different algorithms, as shown in the figure, with the increase of the number of iterations, the RMSE value of each algorithm is decreasing, and the LDA-ALS algorithm used in this study is 14% lower than the RMSE after 10 iterations of a single ALS. Therefore, the LDA-ALS algorithm used in this experiment is significantly improved compared with the traditional ALS algorithm to recommend accuracy. Moreover, in the first five iterations, the RMSE value of the LAD-ALS algorithm is reduced by 38% compared with that of a single ALS algorithm, which can be obtained that the LDA-ALS algorithm proposed by this research can effectively alleviate the cold-start problem of a single ALS algorithm.

To verify the effectiveness of the LDA-ALS algorithm, this experiment evaluates five different recommendation algorithms, ALS, bias SVD [24], SVD++ [25], AutoSVD++ [26], LF-WLC [27], and their RMSE values are checked as shown in Table 2. The experiments all use the same data set, and the RMSE averaged by each recommendation algorithm in the experiment is used as the comparison result, which can be obtained that the LDA-ALS algorithm proposed in this paper has a good recommended result.

Table 1. Algorithms Comparing

algorithms	RMSE
ALS	1.029
SVD	0.911
SVD++	0.870
AutoSVD++	0.842
LF-WLC	0.835
LDA-ALS	0.785

In order to verify the recommendation effect of offline recommendation and real-time recommendation model, this experiment uses the click rate of recommendation results comparing different recommendation modules as the evaluation index. From Figure 8, it can be clearly seen that the click rates of products recommended for users after the application of the insurance recommendation model has been significantly improved, of which the click rate of products recommended for users after the real-time recommendation model and the offline recommendation model is 16% and 23%, respectively.

References

[1] Qazi M, Fung G M, Meissner K J, et al. An insurance recommendation system using Bayesian networks.Proceedings of the Eleventh ACM Conference on Recommender Systems[J]. Como, Italy. 2017. 274-278.

[2] REINSEL D,GANTZ J,RYDNING J.The digitization of the world:from edge to core[J].IDC White Paper, ,2018,41(07):1619-1647.

[3] Impact of data characteristics on recommender systems performance[J].Gediminas Adomavicius,Jingjing Zhang. ACM Transactions on Management Information Systems（TMIS）. 2012 (1):79-89

[4] Zhang L, Liu X, Cao Y, et al. O-Recommend: An Optimized User-Based Collaborative Filtering Recommendation System[C]//2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2018: 212-219.

[5] Chai Z Y, Li Y L, Han Y M, et al. Recommendation system based on singular value decomposition and multi-objective immune optimization[J]. IEEE Access, 2018, 7: 6060-6071.

[6] Yang Q. A novel recommendation system based on semantics and context awareness[J]. Computing, 2018, 100(8): 809-823.

[7] ÖZCAN İ, ÇELİK M. Developing Recommendation System Using Genetic Algorithm Based Alternative Least Squares [C]//2018 International Conference on Artificial Intelligence and Data Processing (IDAP). IEEE, 2018: 1-5.

[8] Shaikh S, Rathi S, Janrao P. Recommendation system in E-commerce websites: A Graph Based Approached[C]//2017 IEEE 7th International Advance Computing Conference (IACC). IEEE, 2017: 931-934.

[9] Nilashi M, Ibrahim O B, Ithnin N, et al. A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA-ANFIS[J].Electronic Commerce Research & Applications, 2015.

[10] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C] //Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2020: 426-434.

[11] Zhang S, Yao L, Xu X. AutoSVD++ An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders [C]//Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. 2021: 957-960.