H&M Personalized Fashion Product Recommendation Using LightgbmRanker

Dan Xian[1], Shaozan Cui[2], Bo Wang[3], and Lishuai Cui[4](✉)

[1] Northeastern University, San Jose, USA
xian.d@northeastern.edu
[2] Nanjing University of Finance & Economics, Shanghai, China
[3] College of Information and Electrical Engineering, China Agricultural University, Beijing, China
[4] King's College London, London, UK
lishuai.cui@kcl.ac.uk

Abstract. The product recommendation system is a system that analyzes user preferences, searches for a large number of product information in the e-mall, and recommends products that may be of interest to users, providing an intelligent shopping experience for online shopping users. It can help users more accurately and quickly discover interesting and high-quality product information, enhance the value of information, and improve the user's online shopping experience. In this article, we use data provided by H&M to construct a recommendation system using LightgbmRanker. To evaluate our experiment's performance, we do compared competitions. We use map as the evaluation metric. The result shows that our LightgbmRanker owns the highest 0.0282 among these models, which is 0.063, 0.056 higher than SVD, TF recommender respectively.

Keywords: recommendation system · LightGBM Ranker · MAP · feature engineering

1 Introduction

With the advent of the Internet era, the characteristics of mobile Internet sharing at will have catered to the convenient and diversified needs of modern consumers, online shopping has become a very common but important shopping method in life, and the e-commerce market has been stimulated with great vitality. The core competition point of major e-commerce companies has also shifted from basic services to providing personalized product recommendation services to users, and the product recommendation system has become one of the core functions of e-commerce platforms. As the category of goods grows, users will spend more money to proactively obtain the items they want through simple retrieval, and the product recommendation system is the most promising way to solve this problem. The product recommendation system is a system that analyzes user preferences, searches for a large number of product information in the e-mall, and recommends products that may be of interest to users, providing an intelligent

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 201–208, 2023.
https://doi.org/10.2991/978-94-6463-198-2_23

shopping experience for online shopping users. It can help users more accurately and quickly discover interesting and high-quality product information, enhance the value of information, and improve the user's online shopping experience.

In this article, we use data provided by H&M to construct a recommendation system using LightgbmRanker. Related work is described in sect. 2, and we introduce our methodology and experiment in sects. 3 and 4.

2 Related Work

In the context of the rapid development of e-commerce, personalized recommendation algorithms that tap user interest in massive items have become the core of the recommendation system. Today's mainstream recommendation technologies are CB (Content-Based), CF (Collaborative Filtering), deep learning-based recommendation, and hybrid recommendation.

1) Content-Based recommendation

CB is the earliest technology used in the recommendation system. The basic idea is: according to the user's historical behavior, to obtain the user's interest preferences, to recommend users to similar to their interest preferences. For feature extraction for targets, [1] introduced a vector space model technique, [2] comprehensively considered the distribution of word frequency and document frequency in the target and the target, and used the word frequency-inverse document frequency technology TF-IDF (Term Frequency- Inverse Document Frequency, TF-IDF) for text preprocessing, and targeted recommendation [3, 4] for users. [5] used the idea of relevance feedback to explore the implicit relationship between user behavior and preferences to improve the effect of target recommendation5.

The advantages are that the principle is easy to understand, the implementation is simple, and the recommendation results can be clearly interpreted: but the disadvantage is: it is impossible to actively discover the potential interest that the user has not shown: it can only process text, and it is impossible to process video, audio, and other targets that are difficult to directly summarize the features.

2) Collaborative filtering recommendation

CF is a very classic technique in recommender systems. The basic idea is that users with similar interests may be interested in the same things. The basic operation of the collaborative filtering algorithm is to analyze user preferences, find similar user groups, and then determine the preference degree of a single user to the target according to the preference program of the user group to the target.

Lu Z and others have proposed a recommendation algorithm based on Naive Bayes to classify target types [7] proposed a collaborative filtering algorithm with uncertain neighbors, which effectively alleviated the problem of data sparsity by calculating the similarity between users and products and using similar objects of products as the recommendation results.

Recommendation based on collaborative filtering can make more accurate target recommendation in the case of learning a large amount of data. But the disadvantages are also obvious: a large amount of data is needed to generate a more accurate user portrait; there is a cold start problem for new users or new targets: the algorithm has

a performance bottleneck, and when there is too much data, the time complexity of calculating the user group feature portrait will be greatly increase.

3) Deep learning recommendation

In addition to the above two traditional recommendation techniques, the application of deep learning technology to the recommendation system has also become a research hotspot. Deep learning is a branch of machine learning and an extension of traditional artificial neural networks, which achieve complex function approximations by building nonlinear mapping functions in multiple hidden layers, and build recommended models that are more in line with user interests. [6] proposed an algorithm that combines collaborative filtering with restricted Boltzmann machine RBM (Restricted Boltzmann Rachine, RBM), which is the earliest recommended technique based on deep learning, using a two-layer RBM model to extract data features, training the user's preference data as input, calculating the hidden factor vector of the hidden layer through conditional probability, and using the hidden factor vector of the hidden layer to reverse the user's liking degree when predicting.

The advantage of the recommended technology of deep learning is that it is possible to learn more abstract essential characteristics from the complex multidimensional data of the user and the target: layer-by-layer training in multiple hidden layers effectively reduces the training difficulty; Features can be learned and extracted automatically without the need for manual operation. However, the recommendation technology based on deep learning is extremely dependent on machine computing power, and there is no effective interpretation of the recommendation results.

4) hybrid recommendation

Hybrid recommendation is to combine the advantages of various recommendation techniques to solve the problem that the accuracy of single-model recommendation is not high. As a semantic web information tool, the knowledge graph contains semantic information and mesh path structure information to provide a new way of thinking for solving the target recommendation problem. This paper mainly examines recommended techniques based on knowledge graphs.

- Our Contribution

  – We utilize LightgbmRanker to create product recommendations.
  – We introduce our dataset and do feature engineering.
  – In the experiments process, we do the comparing experiments and the result shows that our model performed better than the other models.

3 Methodology

The traditional GBDT and XGBoost algorithms have quite good efficiency, but in the case of large amount of data or too high feature dimension, the efficiency and scalability of these algorithms cannot meet the actual needs. Timeliness is important for real-time online consumer behavior analysis systems. The LightGBM algorithm can solve these

problems very well, so LightGBM is applied to the online consumer behavior prediction system.

- Principles of LightGBM

LightGBM (Lightweight Gradient Decision Tree) is a gradient boosting framework based on the decision tree algorithm proposed by Microsoft to support efficient well line training. The algorithm has not made many changes in principle, and the principle of the algorithm is the same as that of XGBoost, which uses the Taylor expansion of the loss function as an approximation of the current error to obtain a new decision tree. The innovation of this algorithm lies in the application of GOSS algorithm and EFB algorithm to speed up operation. When achieving the same accuracy as GBDT, LightGBM is about 10 times faster. The following mainly introduces the GOSS algorithm and the EFB algorithm.

1) GOSS algorithm

The GOSS (Gradient based One-Side Samplin, gradient-based, one-sided sampling) algorithm optimizes the algorithm speed from a sample perspective. Sample importance is measured differently in different algorithms. In the SVM, the support vector closest to the separated hyperplane is the most important: in the AdaBoost algorithm, the sample weight is a measure of importance, and increasing the weight of the mispartitioned sample makes the next iteration focus on training the mispartitioned sample. In GBDT, the gradient of the sample can be used as a basis for evaluating the importance, if the gradient of the sample is small, it means that the sample has been trained and the training residual is small. Similarly, LightGBM can use the gradient of a sample as a measure of its importance.

However, if all the small gradient samples are deleted, it is easy to lack the data set information and cause a loss of accuracy. The GOSS algorithm can solve this problem by retaining large gradient samples, randomly selecting a part of the small gradient samples and multiplying the weights, so that without changing the original data set information, more attention can be paid to those undertrained samples.

The specific step of the GOSS algorithm is to first sort the samples according to the absolute value of the gradient, retain the large gradient sample of the first $a \times 100\%$, then randomly select $b \times 100\%$ of the small gradient sample in the remaining dataset, and finally multiply the small gradient sample by the constant $(1 - a)/b$. Therefore, the LightGBM algorithm uses a gradient based on each sample to optimize the training samples.

2) EFB algorithm

The EFB (Exclusive Feature Bunding) algorithm optimizes the algorithm speed from a feature.

perspective. In high-dimensional sparse spaces, most features take non-zero values almost at different times, meaning they are mutually exclusive. The EFB algorithm takes advantage of the mutexity of high-dimensional sparse features to bundle mutex features together to form a feature and build it into a histogram, which can reduce the training complexity and further speed up the operation speed.

There are two difficulties in the implementation process: how to determine which features are mutually exclusive, and how to bundle these mutually exclusive features. For the first problem, it can be solved by a greedy algorithm: for the second problem,

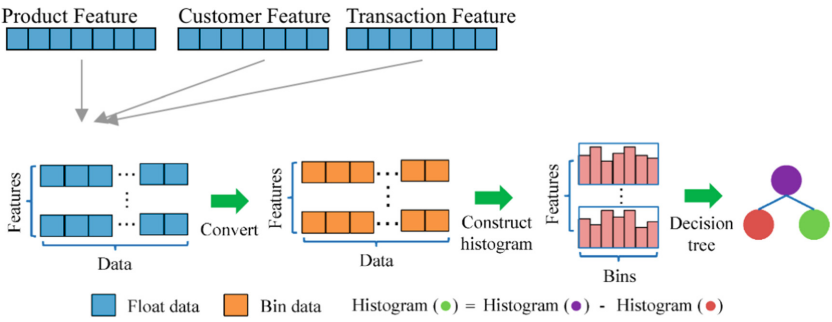

Fig. 1. LightGBMRANK mode

the goal of placing the mutex features into buckets of different histograms is achieved by increasing the offset of the mutex features. The EFB algorithm bundles and merges sparse mutex features, reduces the feature set, and can further improve the efficiency of the algorithm.

- Pros and cons of LightGBM

LightGBM adopts optimization strategies such as histogram-based sorting algorithm and tree generation based on leaf nodes. The advantages of the algorithm are that using histogram has lower training execution time and higher operating efficiency, and storing discrete values instead of continuous values makes memory Reduced consumption, supports category features and parallel processing, and is often used in industrial big data.

- LightGBMRANK model

Our LightGBMRANK. Model is shown in Fig. 1. There are three kinds of data: product features, customer feature and transaction features. And these float data are converted into bin data. We construct histogram and get the decision tree.

4 Experiments

- Experiments data

In our paper, we use the data from previous transactions and customer and product meta data provided by H&M. The available meta data spans from simple data, such as garment type and customer age, to text data from product descriptions, to image data from garment images. We need to predict what articles each customer will purchase in the 7-day period immediately after the training data ends. The metadata for each article_id available for purchase are 25 kinds: article_id, product_code, prod_name, product_type_name, product_group_name, graphical_appearance_no, graphical_appearance_name, colour_group_code, etc., which are detailed describe in articles.csv. “Images” is a folder of images corresponding to each article_id. The metadata for each customer_id in dataset are 7 kinds: customer_id,

FN, Active, club_member_status, fashion_news_frequency, age, postal_code which are presented in customers.csv.

We do feature engineering, and the number of product types per each product group is shown in Fig. 2, which shows accessories owns the most product types. And the top 5 product types are accessories, shoes, garment upper body, underwear and swimwear respectively. Furthermore, we draw product name wordcloud in Fig. 3, which shows the most frequent product name in the data.

- Training parameters

The LightGBM's parameters according to empirical methods and grid search are shown in Table 1.

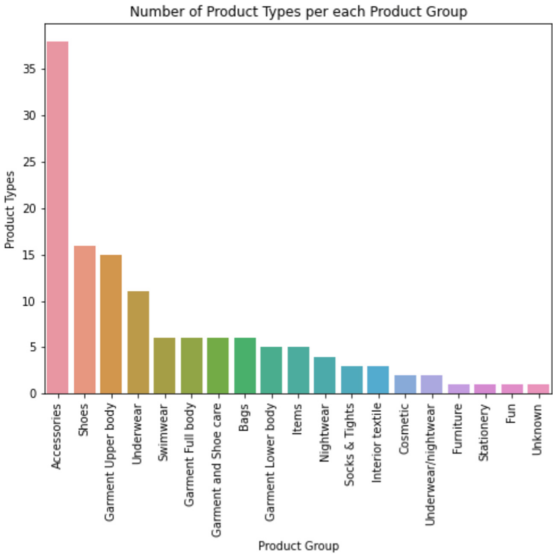

Fig. 2. Number of product types per each product group.

Fig. 3. Product name Wordcloud

Table 1. Training parameters

n_estimators	3000
learning_rate	0.001
num_leaves	20

Table 2. Experiment result

Models	MAP@12
SVD	0.0219
TF recommender	0.0226
LightgbmRanker	0.0282

- Evaluation metrics

The evaluation metric is mean Average Precision @ 12 (MAP@12):

$$MAP@12 = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{min(m,\,12)}\sum_{k=1}^{min(n,12)} P(k) \times \mathrm{rel}(k)$$

where U is the number of customers, $P(k)$ is the precision at cutoff k, n is the number predictions per customer, m is the number of ground truth values per customer, and $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant (correct) label, zero otherwise.

- Experiment result

To evaluate our experiment's performance, we do compared competitions. The higher metric is, the better our model will be. The experiment result is shown in Table 2. Our LightgbmRanker owns the highest 0.0282 among these models, which is 0.063, 0.056 higher than SVD, TF recommender respectively.

5 Conclusion

In our paper, we proposed a product Recommendation system using the LightgbmRanker model. The feature engineering is conducted for three kind of features: Product features, Customer features and Transaction features. These features will be as an input for the ranker algorithm using Goss and EFB algorithm to calculate the histogram and build the decision trees. Compared to other models or methods like SVD, TF recommender, the proposed model owns the highest recall score which is 0.0282. In the future, we will dig into more features especially the text features to build a better recommendation system.

References

1. Salton G, Wong A, Yang C S. A vector space model for automatic indexing. Communications of the ACM, 1975, 18(11): 613-620
2. Agnihotri D, Verma K, Tripathi P. Variable global feature selection scheme for automatic classification of text documents. Expert Systems with Applications, 2017, 81: 268-281
3. Tang L, Long B, Chen B C, et al. An empirical study on recommendation with multiple types of feedback. in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 283–292
4. Xia Z, XuS, Liu N, et al. Hot News Recommendation System from Heterogeneous Websites Based on Bayesian Model. The Scientific World Journal, 2014, (2014–6–26), 2014, 2014:7
5. Wei J, He J, Chen K, et al. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems with Applications, 2017, 69: 29-39
6. Liang P, Lan Y, Guo J, et al. Text Matching as Image Recognition. in: Proceedings of the AAAI Conference on Artificial Intelligence. 20 l 6,30 (1):2793–2799