Expert findings based on LGBM

SiQi Lu
201934729@sdu.edu.cn
School of Computer Science and Technology, Shandong University
Qingdao, Shandong

ABSTRACT

Accurately finding expert users who know a high level of professionalism and influence in various fields will help the system to accurately recommend questions to them, so that questions can be answered accurately and professionally in a timely manner [2]. This will help improve the knowledge level of knowing the knowledge base, increase user participation and community activity, and also provide high-quality information resources for external search engines.

At present, the mainstream online Q & A community experts have found that research mainly uses methods based on topic models and link analysis [3]. The method based on link analysis mainly uses the perspective of user-generated content to characterize the user's interest or professional distribution. The method based on link analysis mainly uses the network of link relationships between users to calculate the user influence through the iterative spread of the network [1].

This article uses a topic model-based method. First, the data extraction in the provided data set is analyzed and cleaned. Then the discrete features are digitally encoded by LabelEncoder. Finally, the processed data set is used to train the model and classified by LGBM For classification and prediction.

KEYWORDS

expert discovery, topic model, LGBM

1 INTRODUCTION

Online social network service websites are becoming more and more abundant, such as YouTube outside the palace, Twitter, and domestic Sina Weibo, etc. Users publish content and interactions on these social websites, and more and more users search and find information on social websites Interested users and content. "Information retrieval" was first proposed by Calvin Mooers. At first, information retrieval was mainly used to retrieve documents about specific content. With the popularization of the Internet, the meaning of information retrieval has now expanded to more fields. Among them, entity retrieval has become an important branch of information retrieval, and experts have found that it is an aspect that has been widely studied in entity retrieval.

Expert discovery is also called expert search, which mainly solves the problem of how to find experts in related knowledge fields and rank experts according to their professional level given the query conditions. In different types of social networks, it is very meaningful to study the problems discovered by experts. This article is based on watching the Mountain Cup competition, and mainly studies the problems found by experts who know about the community-based social network based on Q & A. Users can post questions, answer short questions, and share their understanding in the Zhihu website. At the same time, users can directly search for related keywords to find questions and answers previously raised.

The competition provided information about known questions, user portraits, user response records, and user acceptance invitation records, asking players to predict whether this user would accept invitations to a new question.

1. Problem information. Include <question id, time of question creation, topic of question, text of question, description of question>, etc.

2. User's response. Including <answer id, question id, author id, answer text, answer time, likes, favorites, thanks, comments, etc.

3. User portrait data. Including <user id, gender, frequency of active, topic of interest, long-term interest, salt value> and so on.

4. <topic, token (word), single word 64-dimensional embedding> data.

5. The invitation data of the last month includes <question id, user id, invitation time, whether to answer>.

2 RELATED WORK

When solving the classification problem, the traditional GBDT (Gradient Boosting Decision Tree) algorithm is generally used to solve it. First, the decision tree used by GBDT is the CART regression tree. Whether it is dealing with regression problems or binary classification and multi-classification, the decision tree used by GBDT All of them are CART regression trees. Why not use the CART classification tree? Because GBDT needs to fit gradient values, regression trees are used. The most important thing for a regression tree algorithm is to find the best dividing point, then the divisible points in the regression tree contain all the desirable values of all features. The criterion for the best division point in the classification

tree is entropy or Gini coefficient, both of which are measured by purity, but the sample labels in the regression tree are continuous values, so it is no longer appropriate to use indicators such as entropy. The square error is a good way to judge the degree of fit. However, the traditional GBDT algorithm has some problems, such as how to reduce training data, how to reduce features, and optimization of sparse data. The use of LGBM can better solve some problems.

For example, LGBM's Histogram algorithm has many advantages [4]. First of all, the most obvious is the reduction in memory consumption. The histogram algorithm not only does not need to additionally store the results of pre-sorting, but can only save the value after the feature is discretized, and this value is generally sufficient to store an 8-bit integer. Memory consumption Can be reduced to 1/8 of the original. Then the computational cost is also greatly reduced. The pre-sorting algorithm needs to calculate the gain of the split once for each eigenvalue traversal, while the histogram algorithm only needs to calculate k times (k can be considered a constant), and the time complexity is from O (# data # feature) is optimized to O (k # features).

On top of the Histogram algorithm, LightGBM is further optimized. First, it discards the level-wise decision tree growth strategy used by most GBDT tools and uses a leaf-wise algorithm with depth limitation. Level-wise data can split the leaves of the same layer at the same time. It is easy to perform multi-thread optimization. It also controls the model complexity and is not easy to overfit. But Level-wise is actually an inefficient algorithm, because it treats the leaves of the same layer indiscriminately, which brings a lot of unnecessary overhead. In fact, many leaves have lower split gains, so it is not necessary to search. And split. Leaf-wise is a more efficient strategy. Each time, from the current leaves, find the leaf with the largest split gain, and then split, and so on. Therefore, compared with Level-wise, Leaf-wise can reduce more errors and get better accuracy with the same number of splits. The disadvantage of Leaf-wise is that it may grow deeper decision trees and produce overfitting. Therefore, LightGBM adds a maximum depth limit on Leaf-wise to ensure high efficiency while preventing overfitting.

3 METHOD

First, the simplest description of the question is given: recommend a question Q to user U, and calculate the probability that user U will answer this question Q.

Specifically, the competition provided question information (*question_info_0926.txt*, which can create additional feature information for question Q), user portraits (which can create additional feature information for user U), and answer information (*answer_info_0926.txt*, Additional feature information can be created for both question Q and user U).

Data analysis

```
user_info.columns = ['用户id','性别','创作关键词','创作数量等级','创作热度等级','注册类型','注册平台','访问评率','用户二分类特征a','用户二分类特征b','用户二分类特征c','用户二分类特征d','用户二分类特征e','用户多分类特征a','用户多分类特征b','用户多分类特征c','用户多分类特征d','用户多分类特征e','盐值','关注话题','感兴趣话题']
```

Figure 1:user data

```
question_info.columns = ['问题id','问题创建时间','问题标题单字编码','问题标题切词编码','问题描述单字编码','问题描述切词编码','问题绑定话题']
```

Figure 2:question data

By analyzing the data in Figures 1 and 2, there are 21 features in the user data, of which 5 features (creation keywords, creation quantity level, creation popularity level, registration type, registration platform) are only available in the data set. Value, indicating that these 5 characteristics are completely useless and can be removed directly.

```
train_ans_feature = ans[(ans['a_day'] >= train_ans_feature_start) & (ans['a_day'] <= train_ans_feature_end)]

val_ans_feature = ans[(ans['a_day'] >= val_ans_feature_start) & (ans['a_day'] <= val_ans_feature_end)]

logging.info("train ans feature %s, start %s end %s", train_ans_feature.shape, train_ans_feature['a_day'].min(),
             train_ans_feature['a_day'].max())

logging.info("val ans feature %s, start %s end %s", val_ans_feature.shape, val_ans_feature['a_day'].min(),
             val_ans_feature['a_day'].max())
```

Figure 3:Processing time data

Secondly, the time data is processed, for example, the time for questioning and answering is limited, and a small part of the data outside the limit is deleted.

In order to analyze whether the features in the above two data sets have an impact on the prediction results (or whether they are strong and distinguishable features), we first merge the two data sets with the training set (*invite_info_0926.txt*), and Analyze some features by chart.

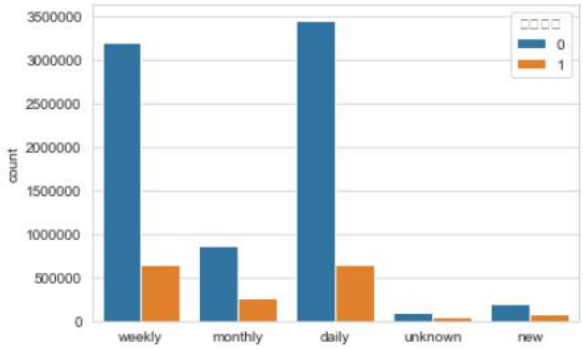

Figure 4:Visit frequency characteristics

For example, in the figure above, there are 5 categories in the feature, [weekly, monthly, daily, new, unknown]. As can be seen from the histogram below, different categories have completely different distributions. This feature is obviously a very different Very strong features.

Feature Processing

First, the feature is encoded. There are many encoding methods. OneHotEncoding, LableEncoding, etc. are commonly used. OneHotEncoding is applicable to the case where the

features lack an inherent order, and the number of features is less than 4, which is obviously not applicable to this dataset. LabelEncoding applies to the ordinal feature. So discrete features are digitally encoded by LabelEncoder. After subsequent analysis of the data, the features with good discrimination are counted for single features.

Model training and prediction

Use LightGBMClassifier for model training and final classification prediction. GridSearchCV was used to adjust the parameters of LightGBM during the adjustment of the model.

In the first step, we first set the learning rate to a higher value, here learning_rate = 0.1, then determine the type of estimator boosting, and the default is gbdt. The number of iterations. The parameter name is n_estimators. We can first set this parameter to a larger number, and then view the optimal number of iterations in the cv results. The second step is to improve the accuracy. Max_depth and num_leaves are the most important parameters to improve accuracy. max_depth is used to set the tree depth. The depth is expected to overfit. num_leaves: Because LightGBM uses a leaf-wise algorithm, num_leaves is used instead of max_depth when adjusting the complexity of the tree. The approximate distribution relationship: num_leaves = 2 m̂ax_depth, but its value should be set less than 2 m̂ax_depth, otherwise it may cause overfitting. The third step is to reduce overfitting, feature_fraction, min_data_in_leaf, and feature_fraction parameters to perform feature sub-assignment. This parameter can be used to prevent overfitting and improve training speed. min_data_in_leaf is a very important parameter, and its value depends on the sample trees and num_leaves of the training data. Setting this resistance can avoid generating an overly deep tree, but it may cause underfitting. The fourth step is to reduce the learning rate, learning_rate = 0.01, increase the number of iterations, and verify the model.

Results

As shown in the figure above,through the model after training, the accuracy of the test set reached 0.840358, and the accuracy of the final verification set reached 0.80951, which completed the task well.

Figure 5:Result

4 CONCLUSION

This paper introduces the selection of algorithms, data processing, parameter adjustment, etc., and finally obtains good accuracy. Experts find that the solution to the problem still needs to be improved and optimized. In summary, there are the following points:

(1) The online Q & A community serves as a platform for users to provide and gain knowledge. Its users will change over time, new users will join, and old users will be lost. Moreover, a user's level of expertise in different fields will change over time. Therefore, follow-up research can study the time factor as an important factor found by experts in the online Q & A community.

(2) With the rise and development of a new generation of online question-and-answer communities with social functions, when considering the influence of user networks, in addition to the traditional question-and-answer relationships, information social relationships such as mutual attention can also be considered for research .

(3) In terms of application, currently it mainly focuses on recommending new problems to domain expert users. It can also consider researching how to personally recommend users with higher levels of expertise to new users in related fields to increase the user viscosity of the Q & A community.

REFERENCES

[1] Youngdo Kim and Hawoong Jeong. 2011. Map equation for link communities. *Physical Review E* 84, 2 (2011), 026110.

[2] G Alan Wang, Jian Jiao, Alan S Abrahams, Weiguo Fan, and Zhongju Zhang. 2013. ExpertRank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems* 54, 3 (2013), 1442–1451.

[3] Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. 2011. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 ieee 11th international conference on data mining workshops*. IEEE, 344–349.

[4] Jin Zhang, Daniel Mucs, Ulf Norinder, and Fredrik Svensson. 2019. LightGBM: An Effective and Scalable Algorithm for Prediction of Chemical Toxicity–Application to the Tox21 and Mutagenicity Data Sets. *Journal of Chemical Information and Modeling* 59, 10 (2019), 4150–4158.