An Optimized linear Scheme with Liblinear
CS420 Coursework: Text Classification

Author

Jinning Li, 515030910592

Shanghai Jiao Tong University

Jinning Li@Computer Science

April 23, 2017

Abstract

This Classification model can well solve the given binary classification problem. The accuracy is great.

In the beginning, my model use both the bigram parsing method and the Jieba Parsing method to get the terms. Then, the model use the scalable term selection method to filter the meaning less dimension. The feature map is created. After that, This model use Linear Regularized Logistic Regression Method to operate the training. And The method of bisection is employed to find the optimized parameters. Last, standardize the test data. apply the trained model, the probability predicted is achieved.

The light spot of this model is the term Parsing and feature selection. And some Limitation also exists. Parameter selection should be optimized in the future work.This classification model can also be applied to advertisement classification, criminal distinguishment.

1 Introduction

1.1 Background

As aggregators, online news portals face great challenges in continuously selecting a pool of candidate articles to be shown to their users.

Typically, those candidate articles are recommended manually by platform editors from a much larger pool of articles aggregated from multiple sources. Such a hand-pick process is labor intensive and time-consuming.

In this task, we study the editor article selection behavior and propose a learning by demonstration system to automatically select a subset of articles from the large pool.

2 Methodology

The method how I use the liblinear package to do the classification modeling and get the results.

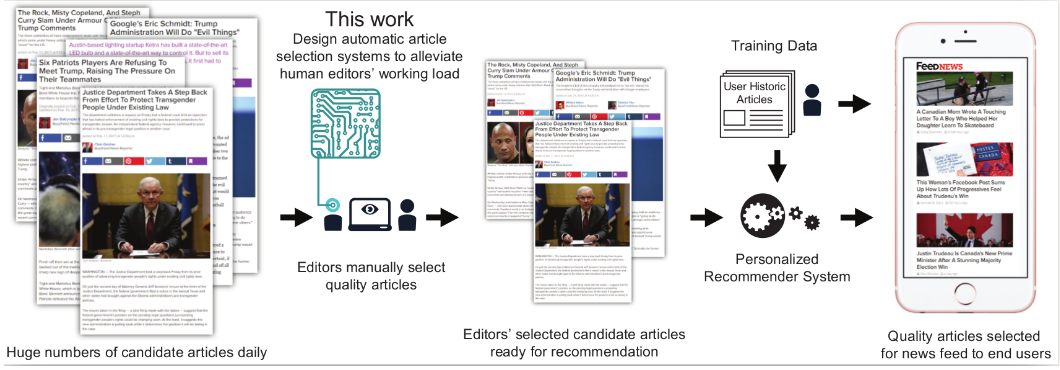
This work
Design automatic article selection systems to alleviate human editors' working load
Editors manually select quality articles
Huge numbers of candidate articles daily
Editors' selected candidate articles ready for recommendation
Training Data
User Historic Articles
Personalized Recommender System
FeedNEWS
Quality articles selected for news feed to end users

Figure 1: Background of text Classification

Table 1: Notations and descriptions

Notation	Description
$\boldsymbol{x}$	The pre-defined value of positive user response.
y	The true label of user response.
p	The predicted probability $Pr(\hat{y} = 1 \vert \boldsymbol{x})$.
df	Document Frequency.
PR	Relative probability ratio

2.1 Prepare Works

The works before the modeling are illustrated.

2.1.1 Text Standardlizing

To solve the text classification problem, the first subproblem is that how to standardize the text data. The source text files are of a kind of strange Unico code type.

And in the page there are lots of useless words, for example, symbols, spaces, URL links, picture links, etc. I abondon of optimize these words.

Abandon the symbols and space. Replace URL with Chinese Word ”Link” to represent here the author use a webpage link. Replace Picture links with ”Picture” to represent here the author use a picture.

And I think the title of the text well include the content of the text. So I write the title for five times to the standardized text files to rise the weight of words in the title.

2.1.2 Word Parsing

For the Parsing works, the most efficient method is the **bigram method**. The bigram method is to parse the text two by two. For example, ”ABCD” will be parsed to ”AB”, ”BC”, and ”CD”. this method will produce lots of meaningless phrases, but those will affect the result little, except that those enlarge the calculation process because this method produce a charactoristic space of higher dimension.

And another weakness of the bigram method is it can only extract the phrases of size two. So to optimize this, I use the JieBa package to extract the word equal or more than two. And I combine this two method to get a more charactoristic space.

There may be a question that JieBa can also extract phrases of size two, can we abondon the bigram method? the answer is not because of my experiments. So I think the seemly meaningless phrases can also represent the features of text files to some extent.

By using the combined method, the performance of my model is improved.

2.2 Training

In the subsection 2.1, we have converted the initial text files into phrases. By count the frequencies of these phrases and formulated these feature and its weight, we get the feature map.

As for the feature select method, I refer **the Scalable Term Selection** method[1]. This method separately measure the discriminability and the coverage of a term, The basic guideline is that the phrases should not be highly correlated. As noted in the notation 1, assume the *df* as the document frequency, the PR:

$$PR(t_i, c) = log(\frac{P(t_i|c_+)}{P(t_i|c_-)}) = log(\frac{df(t_i, c_+)/df(c_+)}{df(t_i, c_-)/df(c_-)}) \tag{1}$$

And the selection criterion is given by:

$$\varsigma(t_i; \lambda) = (\frac{\lambda}{PR(t_i)} + \frac{1-\lambda}{log(df(t_i))})^{-1} \tag{2}$$

The value λ is selected based on the test data.

And after that, I use the liblinear package to train my model. By experiment, I choose the best regression method **regularized logistic regression method** to train my model. Then the optimization problem is created 3:

$$min_w \ \sum |w_j| + C \sum log(1 + exp(-y_i w^T x_i)) \tag{3}$$

This method need us to optimize two parameters:

- C, the cost parameter to limit the violation.
- P, the sensitivity of supportvector regression loss.
- $Emps$, the precision of the iteration.

To select the best parameters, I divide the test data into two parts. The first part includes 80% of the initial data and the second part includes 20% of the initial data. The first part is used to train the model and the second part is used to evaluate the model locally.

And I set the $Emps$ as 0.001, and the P as 0.1, and for the C, I use method of bisection to select the value of C.

And the value of c in the best performance is about 2.15.

After select the best parameter, we have get the optimized traning model.

2.3 Predict

For the Predict Process, we first extract the feature vector of each text, using the same method illustrated in the section of preparation 2.1. Then, input the feature vector to operate the liblinear prediction, the last result is achieved.

3 Discussion and Future works

3.1 Discussion

- This Text Classification Model can Predict the binary classification problem well.
- The parts of term Parsing and feature selection performs very well.
- But the training part performs not so good, I think there is some problem in the parameter selecting.
- The Final score is 0.86957, which is very close to the baseline.
- If time permits, I believe a higher score will be achieved.

3.2 Future works

The training process should be optimized. Loss Function, parameters, etc. The neural network model can be employed to improve the performance. And we can also try to apply this model into the muti-classification problem. This text classification model can also be used to solve the advertisement classification and many other problems.

References

[1] J. Li and M. Sun. Scalable term selection for text categorization. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 774–782, 2007.