Optimization and Machine Learning, Spring 2021 Course Project

Qiu Longtian
2018533107
qiult@

Fu Xiang
2018533104
fuxiang@

Shi Qianjing
2018533194
shiqj@

1 Introduction

1.1 Data Set

Since our project uses the NLP method, so it's better for the data set if it's binary like 0 and 1 or True and False. Thus we chose data from Kaggle, a website focusing on machine learning.

The labeled data set consists of 25,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of "Negative", and rating >=7 have a sentiment score of "Positive." No individual movie has more than 30 reviews.

File description:MovieReviewTrainingDatabase.csv is the labeled training set. The file is comma-delimited and has a header row followed by 25,000 rows containing the sentiment and the text for each review.

Data fields of the data set:sentiment - Sentiment of the review; "Positive" for positive reviews and "Negative" for negative reviews;

Review - Text of the review.

1.2 Practical Use

The data set is binary, containing films review and whether it's positive or negative, so by using various analysis methods, we divide the data set into a training set and testing set, thus analyzing a review of a specific film is positive or negative, which can help people judge the usefulness of a film review.

1.3 Class Related

We plan to use the neural network, linear classifier and principal component analysis learned in class to do this task.

2 Methodology

The methodology and experiment sections should describe clearly what you did (e.g., enough details for someone to re-implement your algorithms, if needed.), and provide some summary tables or figures that illustrate your experimental results, along with some descriptive interpretations.

2.1 Words Embedding

Word embedding is a popular topic in natural language processing which is designed to represent words and documents in a low dimensional vector space. So that the words with similar meanings will

Preprint. Under review.

have a similar representation. We have tried two kinds of methods when doing document embedding on the movie reviews. The first kind of method is based on **Bag of Words** to calculate and collect the frequency of every word in a document. Another kind of method is called **Topic Modeling** which focuses on extracting the topic of a document.The detailed introduction of methods are as following:

Doc2Vec is an algorithm based on distributed bag of words and distributed bag of words models. The neural network of Doc2Vec contains one extra input compare with Word2Vec which is the document id so the topic or the main idea of a document is stored. With the help of Doc2Vec, we train Doc2Vec model on movie reviews and derive the vector representation of movie reviews.(1)

Latent Dirichlet Allocation is used to infer the topic of a document from a training corpus. The LDA algorithm estimates the importance of a word by the frequency of the words in the current document and the whole corpus with a prior distribution of the word. We train the LDA model from IMDB movie review corpus and derive a vector representation of our movie reviews.

BERTopic leverages BERT embedding and c-TF-IDF to create a cluster so as to extract the topic of documents and preserve the words representing the topic in the meantime. We use our movie reviews to train BERTopic model and derive a vector representation.

Sentence-BERT is a modification of the pre-trained BERT network used to derive sentence embeddings where siamese and triplet network structure are the main modification. We use a pre-train model from SBERT official website and predict the vector representation of movie reviews.(2)

Top2Vec is designed to do topic modeling and semantic search which leverages word semantic embedding and joint document. We use our movie reviews and derive the vector representation of movie reviews. (3)

The above algorithms are the available and feasible ways to do word embedding we found. Since the Doc2Vec is more stable for document representation and the effect of topic modeling is agnostic, in the final feature vector, we take feature vector from Doc2Vec as majority and feature vector from topic modeling methods as supplementary.

2.2 Classifier

The classifier is to classify the inputs' labels(or classes) according to the input variables. There are a lot of methods to solve such classification problems. Here, three of them are chosen to solve the classification of movie reviews.

The first one is the neural network method. Here, the input variable is an n-dimension vector. (210<=n<=280, changing referring to the features) The design of the neural network is figure 1. The whole network is a ccn(completely connected network) with the Relu function as the activation function. The first layer contains n cells so that all variables of each input vector can be included. Then, all cells are connected to the next layer of 100 cells with a Relu function dealing with the outputs. This step is to conclude the information to the more general topics. After that, 100 cells are connected to the next layer of 50 cells with a Relu function again, and the following layer is of 20 cells, which is doing the same thing as above. Finally, we connect the 10 cells to 1 cell to get the probability of whether the movie review is positive.

The second one is SVM(support vector machine). The input vector can be represented as a point in the n-dimension space, and we can find the decision boundary through this method. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. We can use this method to classify the different kinds of movie reviews.

The third one is the Random Forest method. It is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

	Neural Networks	SVM	RandomForest	Adaboost
Combination.csv(270)	0.1095	0.117	0.152	0.1095
BertTopic+doc2vec.csv(210)	0.247	0.2475	0.2915	0.2475
LDA+doc2vec.csv(280)	0.234	0.2325	0.3	0.234
SBERT+doc2vec.csv(280)	0.1065	0.113	0.1315	0.1065
top2vec+doc2vec.csv(280)	0.193	0.1895	0.241	0.1925

Table 1: Error rate of methods

2.3 Ensemble

AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems, it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

3 Experiment

In the experiments part, we run an experiment on 12000 data. To better select the best feature vector and classification algorithm, we tried to use five combinations of features 1. combination of vector five methods 2. LDA + Doc2Vec 3.SBERT + Doc2Vec 4. Top2Vec + Doc2Vec 5. BERTTpoic + Doc2Vec. The feature vector generated by Doc2Vec is fixed at 200. In combination 1, the size of the feature vector is at most 20 except for Doc2Vec. For combinations 2,3,4 and 5, the feature vector is of size 80 except for Doc2Vec. To ensure the size limitation is satisfied, we use Principal Component Analysis to do dimension reduction. All the error rates using different features and methods are shown below.

4 Conclusion

In this project, we propose a state-of-art method to classify movie reviews into two categories, negative and positive. The most challenging part of our project is to convert words to vectors while preserve the information in the movie reviews and choosing the right classifier to predict the sentiment of the movie reviews.The method basically consists of two parts. The first part is word embedding where we convert words into a vector representation. In word embedding, we try a combination of two kinds of methods. The first kind is extracting the word representation based on BOW from a movie review. We apply doc2vec to derive the vector representation for a document. The second kind of method is topic modeling. We have tried LDA, SBERT, BERTTopic, and Top2Vec to generate topic vector representation for the movie review. After generating these two parts, we combine the vector of two parts together as our final feature vector.

Then, we start to use different methods to classify the movie reviews.

The first one is the neural network method. Here, the input variable is an n-dimension vector. (210<=n<=280, changing referring to the features) The design of the neural network is as following.

The whole network is a ccn(completely connected network) with the Relu function as the activation function. The first layer contains n cells so that all variables of each input vector can be included. Then, all cells are connected to the next layer of 100 cells with a Relu function dealing with the outputs. This step is to conclude the information to the more general topics. After that, 100 cells are connected to the next layer of 50 cells with a Relu function again, and the following layer is of 20 cells, which is doing the same thing as above. Finally, we connect the 10 cells to 1 cell to get the probability of whether the movie review is positive. The error rate of testing relating to epoch number is as following.

The other methods are as above: SVM and Random Forest algorithm. We use the sklearn library to train the models and predict our labels.

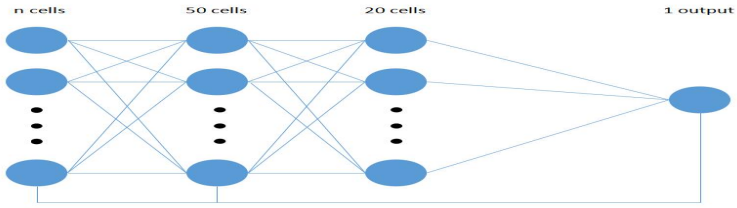

Figure 1: neural

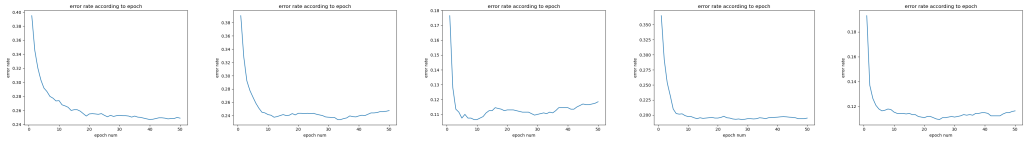

Figure 2: Neural Error Rate

Lastly, referring to AdaBoost, we create an ensemble model containing all methods above. We calculate the weights on each method to get the total model prediction. The weights calculation way is as following:

$$D_1(i) = \frac{1}{m}, \epsilon_t = \Sigma_{error} D_t(i)$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{-\alpha_t} if y_i = h_t(x_i)$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{\alpha_t} if y_i \neq h_t(x_i)$$

$$\alpha_t = \frac{1}{2} \lg(\frac{1-\epsilon_t}{\epsilon_t})$$

$$H_{final}(x) = sign(\Sigma_t \alpha_t h_t(x))$$

The reason for the high accuracy of our methods lies in two perspectives. The first reason is that the feature vector extracted from movie reviews contains not only the frequency of words but also the relation between adjacent words so that when the movie reviews are converted to feature vectors, the original information is preserved to the greatest extent possible. The second reason is that when choosing classification methods, we apply AdaBoost to combine a series of weak learning algorithm into a strong learning algorithm.

For further improvement, we think CNN may be useful to do word embedding since CNN may be able to preserve more information about the order of words in a document.

References

[1] Le, Q. V. , and T. Mikolov . "Distributed Representations of Sentences and Documents." JMLR.org(2014).

[2] Reimers, N. , and I. Gurevych . "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019).

[3] D Angelov. "Top2Vec: Distributed Representations of Topics." (2020).