Tweets Sentiment Analysis on American Airlines

Zhu Yifan

Abstract

In this report, I suggest that ensemble and boosting does not necessarily improve precision for classification problems. This report investigates the utility of using different base classifiers and evaluates the impact of ensembles and boosting on tweet sentiment classification accuracy. The usefulness of features that capture negation information for detecting the sentiment of Twitter comments is also investigated.

1 Introduction

Sentiments are not only presented in the reviews of products information pages nowadays. In the recent decades, there has a huge growth in using microblogging services such as Twitter to express opinions about brands and products. These platforms offer information about what people think and feel about the products and services. Automatic tools can help decision makers to come up with solutions to the problems raised in a more efficient way. Under this perspective, the focus of my work is on the sentiment analysis of tweets.

I consider Tweets sentiment analysis as a classification problem. If tweets contain sentiment, then it is either polar (negative or positive) or neural.

Then how useful are the categorical and numerical features given in the dataset? How useful is the text itself? Besides, when a single model can not achieve a high accuracy, will ensemble contribute to a higher accuracy score? If not, can boosting improve the accuracy of models? When there are not many features, can polynomial features always help? In this report, I began to investigate these problems.

Another problem I want to address is how much negative information would contribute to the improvement of model accuracy when predicting the labels.

2 Related work

Many researchers have focused on the single usage of single classifiers, such as Naive Bayes, Support Vector Machines, Gradient Descent to solve to this classification problems. With many features given in the datasets, sometimes it is possible that a single model cannot achieve a ideal accuracy score. Da Silva et al. (2014) suggests that classifier ensembles for tweet reduces the overall risk of making a poor selection of the classifiers to be used with new data.

For sentiment analysis, some researchers use a tweet as a whole to identify whether it is a positive, negative or neural sentiment. Other researchers target at a marked term within the tweet, and determine the polarity of the whole sentiments by the polarity of the term. Zhu et al.(2014) proposes to use separate sentiment lexicons for negated contexts when training models.

Novakovic et al. (2014) suggests when with appropriate base estimators, adaboost could contribute to the increase of reliability of classification for datasets.

3 Classifiers, ensembles and boosting

The base classifiers used in this experiments are Logistic regression, Stochastic gradient descent, Support Vector Machines, the k-nearest neighbors and Decision Tree classifier. These models could analyze data used for classification.

Ensemble methods train multiple learners to solve the same problem. In contrast to single models, which construct one learner only from the training data, ensemble methods construct a set of learners and combine them.

In this report, ensemble is realized by voting classifiers and random forest. In the experiments, both hard voting and soft voting methods have been evaluated. Hard voting uses a voting classifier which includes all the single classifiers to calculate the accuracy score. Soft votings feed each base learner different data. There are two main methods, bagging and pasting. Pasting splits the training data into as many train/test data sub-split as the number of base learners. Bagging is basically the same, but it is sampled with replacement. This means that it has more variance than bagging, and even if two splits contain the same data points, the proportion of data points of each type might be different.

Boosting checks estimators' errors and adapts each new tree to focus on the fitting errors of existing trees. The primary method for this boosting is Ada boosting. Gradient boosting does not train each successive tree on the previous one. Instead, it acquires errors by calculating the distance between predictions and target. Based on gradient boosting, XGboost parallelizes multiple branches of trees. All the mentioned three boosting methods are evaluated in the experiments.

3.1 Data

The dataset I use in the experiments is from https://data.world/socialmediadata/twitter-us-airline-sentiment. This dataset contains 14640 tweets comments on US airlines. After splitting, 80% of it becomes the training data, and the rest 20% becomes the test data.

3.2 Features

The dataset contains 20 columns. But half of them does not have a direct connection with the label, thus they are not regarded as features in the experiments. For example, it is hard to me to see how Twitter names and the last time they posted their judgments could affect their opinions on US Airlines.

I classified the features used in the experiments into three different categories. First, numerical features. They include the number of the trusted judgments the tweeter posted before, their comments confidence in the airline sentiment, the confidence of their negative reasons and their retweet count. The second category is the categorical features, in the first experiment, I just include the state of the tweet, the airline company the user chose and time zone of the user. In the later experiments, I also include the 'negative reason' as a categorical feature. The tweet text is considered as a text feature.

Considering there are only nine features in the dataset, I also evaluated the usefulness of polynomial features to see if it could improve the accuracy score. Polynomial features could create more features without adding much burden to calculating the features and it is a common strategy to solve non-linear problems.

3.3 Approach

To compare the results of single classifiers with ensemble and boosting. I first evaluate the usefulness for several single classifiers. Then I use voting methods and boosting methods imported from Scikit-learn to realize ensembles and boosting.

For negation reasons, there are in total 10 reasons presented in the dataset. I treat them as a categorical feature and adding it to the model to evaluate their usefulness.

3.4 Preprocessing

For categorical features, I use get_dummies from pandas to encode the features. Some classifiers I used in the experiments are not scale-invariant, so I use standard scalar to scale all the numerical features. For text features, I encode them with two ways, TF-IDF and Bag of Words. And I created two versions of texts for encoding. Their difference is if they were prepocessed or not. In the prepocessed version, stop words, punctuation are dropped, and the morphological form is not kept. While in the other version, all elements in the original tweets are retained.

4 Experimental evaluation

I have several goals for these experiments. First, I want to evaluate the contribution of the negation words compared to just using the categorical, numerical features and the tweet text itself. Secondly, I want evaluate whether ensemble and boosting could improve the accuracy score. Model performances are evaluated by their accuracy score and F1 score.

4.1 Baselines

I used the majority label to calculate the baseline. The most frequent label in the airline is

'positive'. By utilizing this label as the predication for all the tweet comments, the accuracy for the training data is only 0.16.

4.2 Experiment scores

In the first experiment, I evaluate all the single classifiers using only the categorical features and numerical features (excluded negation reasons). For different models, the results vary. Decision Tree Classifier has the highest accuracy score, 83.26% . It is followed by KNN, which gets 80.43%. For the rest classifiers, the difference is not much. SGD has the lowest score. The accuracy score is only 66.53%.

Classifiers	Accuracy Score
Logistic regression	0.6714
KNN	0.8043
Linear SVC	0.6721
SVC	0.7076
Decision Tree Classifier	0.8326
Stochastic gradient descent	0.6653

Table 1: Accuracy Score for single classifiers using categorical features (negative reasons excluded) and numerical features.

Then, to test the usefulness of negation words, I used the 'negativereason' feature. This feature provides linguistics elements providing negative information such as 'not', 'no', 'can't'. Considering the number of these elements in the tweet text is not big, I treated this feature as a categorical feature and used one-hot to encode the elements in the feature.

Classifiers	Accuracy Score
Logistic regression	0.8446
KNN	0.8289
Linear SVC	0.8425
SVC	0.8358
Decision Tree Classifier	0.8313
Stochastic gradient descent	0.8391

Table 2: Accuracy Score for single classifiers using categorical features (negative reasons included) and numerical features.

The accuracy score has a noticeable improvement, especially for logistic regression, SGD and SVM. The improvement is over 15% for the three classifiers. For KNN and Decision Tree Classifier, there is also slight improvements.

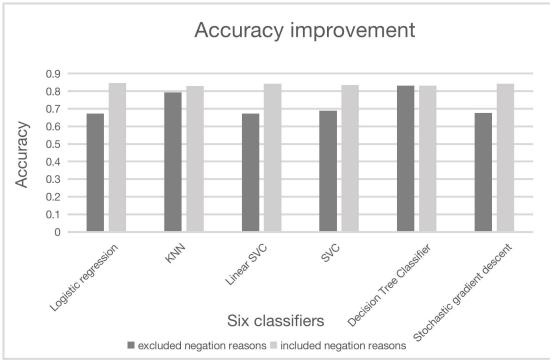

Figure 1: Accuracy improvement by adding negation reasons.

The results suggest that KNN and Decision Tree Classifier do not rely on linguistics elements much to classify the comments, but for the other classifiers, negative information helps them make decisions.

In order to increase the accuracy score, I ran a small experiment to test the usefulness of polynomial features. The estimators I used is logistic regression. The degree I set is 3. The accuracy is 0.8439, which is lower than the predication accuracy score from logistic regression without polynomial features. It seems that in this task, polynomial feature cannot improve model performance much.

It is also noticeable that learners trained from data with the negation information beats those only trained from tweet texts. Considering the logistic regression has the best performance in learning from the categorical and numerical features, I use this classifier to learn from tweet texts. For experiments, I prepared two versions of text for the model to learn, in one version, stopwords and punctuations are removed, all words are in their stem form. In the other version, everything in the original text is kept.

Vectors	preprocessed	Accuracy score
TF-IDF	yes	0.8029
	no	0.8019
Bags of words	yes	0.8154
	no	0.8070

Table 3: accuracy score of logistic regression learner learned from tweet text.

The table shows that first, the learner gains the best score when it learns from the original tweet texts, and the Bag of Words is a better encoder for tweets text. Besides, logistic regression learning

from categorical features and numerical features is 3% better than the best accuracy score gained by the same learner learning from text feature only.

From the experiment result, I could see that negation elements contribute to a considerable improvement to accuracy score for logistic regression, SGD and SVM .

The second experiment is to evaluate the performance of ensembles.

For hard voting, I include all the classifiers I used in the first experiments. For soft voting, I used the logistic regression as the base estimator considering it is the best single estimator in the first experiment.

Voting Types	Accuracy Score
Hard voting	0.8408
Soft voting (bagging)	0.8425
Soft voting (pasting)	0.8443

Table 4: Accuracy Score for voting classifiers using categorical features (negative reasons included) and numerical features.

Comparing the accuracy score of voting classifiers and single classifiers, the soft voting classifiers are slightly better than most single classifiers. But logistic regression still has the highest accuracy among all the classifiers.

In addition to the above voting classifiers, I also compare the accuracy score of Decision Tree Classifier and random forest classifier, which is considered as Decision Tree Ensemble, extreme resistant to overfitting. The results shows that Random Forest Classifier is 5% better than the Decision Tree Classifier.

Classifiers	Accuracy Score
Decision Tree Classifier	0.8313
Random Forest Classifier	0.8361

Table 5: Accuracy Score for Random Forest Classifier and Decision Tree Classifier

The evaluation scores presented above shows that ensemble classifiers could slightly improve accuracy score, but they cannot beat all the single classifiers such as logistic regression in this project.

My third experiment is to see if boosting could contribute to better label predictions.

Compared to the accuracy of ensemble classifiers, boosting has a better score. But the improvement is still not very big. And the accuracy score of it is still lower than that of logistic regression when all the categories and numerical features are learned by estimators.

Boosting Types	Accuracy Score
Gradient Boosting	0.8432
Ada Boost	0.8432
XG boost	0.8402

Table 6: Accuracy Score for boosting classifiers using categorical features (negative reasons included) and numerical features.

Based on the above experiments, I could see that ensembles and boosting could contribute to better label predication to some extend, but they cannot beat all the single classifiers.

To test my hypothesis generated from the above experiments, I conducted the final experiment, using all the features to train models to see the performance difference of different estimators. In this experiment, I used Gradient Boosting, logistic regression and pasting to predict labels and collects the accuracy score considering they gain the best score in boosting classifiers, single classifiers, and voting classifiers respectively. In this experiment, texts are transformed into Bag of words for its performance beats TF-IDF.

classifiers	Accuracy Score
Gradient Boosting	0.92
Logistic Regression	0.94
Soft voting (pasting)	0.94

Table 7: Accuracy Score for Gradient Boosting, Logistic Regression, and pasting (all features included)

classifiers	F1 score in negative	F1 score in positive	F1 score in neural
Gradient Boosting	1	0.73	0.83
Logistic Regression	1	0.81	0.85
Soft voting (pasting)	1	0.77	0.87

Table 7: Accuracy Score for Gradient Boosting, Logistic Regression, and pasting (all features included)

The experiment results show that Logistic regression and pasting(with logistic regression as the base estimator) both has 94% accuracy, but they have different F1score for different label predictions. When I use Decision Tree Classifier as the base estimator, the accuracy is 93%. This supports that hypothesis that boosting and ensembles do not necessarily has a higher score that a single estimator in this project.

5 Conclusion

The accuracy score baseline of this project is 16%, by adding numerical features and categorical features, the accuracy score could be improved. My experiment shows that negation information could largely improve model performance for logistic regression, SGD and SVM. The ensembles and boosting cannot improve the accuracy much, and they cannot beat the logistic regression in accuracy score.

Acknowledgments

I would like to thank my lecturer, Elizabeth Merkhofer and Teaching Assistant Caroline for their assistance during the whole semester, which not only helps me with my assignments but also largely support me during the final project time.

References

Da Silva, N., Hruschka, E., & Hrushka, E. 2014. Tweet sentiment analysis with classifier ensembles. *Decision support systems,* 66, 170-179, http://doi.org/10.1016/j.dss.2014.07.003

Maynard, D., & Greenwood, M. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. Retrieved from http://eprints.whiterose.ac.uk/130763/1/sarcasm.pdf

Novakovic, J. D., & Veljovic, A. 2014. Adaboost as classifier ensemble in classification problems. Proc. Infoteh-Jahorina, 616-620.