Report of project2

October 13, 2019

1 Introduction

This is the report of project 2 for COMP90049. In this project, I try to use the data set of some tweets to predict the location of the new tweets. At first, I analyze some algorithms by 10-fold cross validation. Then I try some algorithms with good performance in cross validation to predict the tweets in test set and try to develop accuracy.

2 Literature review

The general introduction of statistical classification and machine learning are based on the content of lectures. Many important methods like decision trees and Boosting are according to some literatures from Hastie and Friedman, Duda and G and Mitchell.A comprehensive introduction to text categorization methods and results can be found in the leterature(Sebastiani).The method of how to use weka to achieve the models is based on a web page from LarryNLPIR.

3 Data set

The basic data sets come from tweets with user-id, the content of tweets and the label of locations. The training data sets I used have 96585 tweets. And the non-alphabetical characters have been moved. And the top 10, 20, 50, 200 terms according to document frequency are chosen as attributes. And we also have data with best 10, 20, 50, and 200 terms which are chosen by method of Mutual Information. The develop data set have 34028 instances and the test sets have 32977 tweets. And They also have the information with same attributes.

4 Analysis

4.1 Cross validation

By reading some articles, I choose some methods to predict the location of test tweets. At first, I use 10-fold cross validation to test the performance of these models on part of training set.(Figure 1)
We can find out naive bayes Multionmial,Multilayer Perceptron and SMO have higher accuracy with more attributes(from best 10 to best 200).But naive bayes, J48, Random tree and Random forest and lower accruacy with more attributes. Comparing accuracy of data set with 'best' attributes and data set with 'most' attributes. We can point out the predictions are more accurate if we have related attributes rather than the words with high frequency. These results are based on cross validation. So we can reference these performance if our test sets don't have obvious difference with training set.
Comparing different methods. We can find out bayes Multionmial have best performance with more 'good' attributes. And performance of SMO is good too. J48, random and random forest have good performance with fewer attributes.

4.2 Test on developing set

Then we use the model trained by training data set to test developing data set.(If we use best-10 training data set to build the model, we need test it on best-10 developing data set)And there is the result

below.(Figure 2)
We can find out J48, random tree and random forest only have good performance with data set of fewer attributes. And SMO have best performance on data set best-200. And bayes Multionmial and one-R also have good performance on this data set too.

4.3 Time Analysis

There is the time of building model(Figure 3)and time of taking to test model on supplied test set(Figure 4) with different methods and different data set. We can find out there is no big difference with time of testing model by different methods. But the time of build model by SMO, J48 and Random forest will become very large with more attributes.

5 Prediction

5.1 Using traditional model

According to my analysis above. I decided to use SMO with best 200 attributes to predict the tweets in test data set. The accuracy is about 60%. Then I tried bayes Multionmial, one R, J48 and random forest with same data set. The highest accuracy is 62.407%, based on prediction of naive bayes multionmial
I also tired prediction by J48, random tree and random forest with best 10 data set. The highest accuracy is 61.679% based on random forest.

5.2 Model combination

Because using traditional machine learning model to predict location of tweets is hard. I haven't got good results. So I try to use the better results to combine these models. Considering four better results, I want to decide the location of a tweet is NewYork if more models think it's NewYork. I used python to complete this judgment.(If there is same amount of locations in the results,I will choose the first one. Because the first model got the highest accuracy.)The accuracy I got is 62.549%. And I also try the combination of 2,3 and 5 models with high accuracy. But the best one is the combination of 4 models.

6 Discussion of Development

6.1 Feature engineering

I think the models I use can't predict location well. Because the highest accuracy is only 62.549%.The reason may be many words as attributes I used can't judge the location.
So, I think maybe I should get feature engineering again. I tried to extract the words which are name of a location in three states by python. But I found out that only few tweets have these words. And it's hard to get all the names of landmarks. Maybe we can use TF-IDF to complete feature engineering. Because it can help us find out the important words we need in the judgment and classify.

6.2 Methods choice

Another reason why I can't get higher accuracy may be I didn't use correct models. If I choose models of deep learning can get better results. Like xgboost and lightgbm Weitiao. I used models combination. And the advantage is can avoid the mistakes made by one model. But the disadvtage is that it may miss the right prediction by one model if other models make a wrong prediction.

References

Duda, P. E. H., Richard O., & G, D. (2000). Pattern classification. *Stork*, 264£343.

Hastie, R. T., Trevor, & Friedman, J. H. (2001). The elements of statistical learning: Data mining, inference, and prediction. *Springer*, 264£265,291,292,391.

LarryNLPIR. (2012). Guide of weka. *CSDN*, https://blog.csdn.net/yangliuy/article/details/7589306.

Mitchell, T. M. (1997). Machine learning. *McGraw-Hill*, 264.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 264.

Weitiao. (2017). Deep learning. *CSDN*, https://zhuanlan.zhihu.com/p/29769502.

	best-10	most-10	best-20	most-20	best-50	most-50	best-200	most-200	Average
zero-0	62.96%	62.96%	62.96%	62.96%	62.96%	62.96%	62.96%	62.96%	62.96%
naivebayes	61.91%	61.88%	60.83%	60.09%	58.83%	56.41%	57.15%	55.71%	59.10%
oneR	63.27%	62.96%	63.27%	62.96%	63.27%	62.96%	63.27%	62.71%	63.08%
naivebayesMultionmial	63.87%	62.96%	64.16%	62.87%	64.66%	62.80%	65.29%	62.71%	63.66%
SMO	63.81%	62.96%	64.13%	62.96%	64.60%	64.60%			63.84%
J48	63.75%	62.95%	64.03%	62.59%	64.11%	0.00%			52.91%
Randomtree	63.75%	62.00%	63.64%	59.24%	61.87%	55.57%			61.01%
Randomforst	0.637521		0.637004		62.56%	60.18%			0.62549825

Figure 1: The result of cross validation

	best-10	most-10	best-20	most-20	best-50	most-50	best-200	most-200	Average		
zero-0	64.41%	64.41%	64.41%	64.41%	64.41%	64.41%	64.41%	64.41%	64.41%		
naivebay	63.15%	63.13%	62.08%	61.09%	60.09%	57.61%	58.21%	56.69%	60.26%		
oneR	64.68%	64.41%	64.68%	64.42%	64.68%	64.42%	64.68%	63.68%	64.45%		
naivebay	65.13%	64.40%	65.31%	64.34%	65.44%	64.23%	65.40%	63.68%	64.74%		
SMO	65.13%	64.41%	65.37%	64.41%	65.49%	64.41%	65.73%	64.67%	64.95%		
J48	65.14%	64.41%	65.32%	64.02%	65.15%	62.33%	63.39%	61.25%	63.88%		
Randomtr	65.11%	63.37%	65.01%	60.13%	62.91%	56.42%	58.19%	53.95%	60.64%		
Randomfo	65.12%	63.35%	64.98%	61.46%	63.62%	61.48%	62.92%	62.66%	63.20%		

Figure 2: Using model on developing set

	best-10	most-10	best-20	most-20	best-50	most-50	best-200	most-200
zero-0	0.01	0	0.03	0	0.01	0.02	0.01	0.04
naivebayes	0.34	0.12	0.6	0.28	1.42	0.6	5.2	3.77
oneR	0.27	0.1	0.58	0.27	1.65	0.68	6.41	0.19
naivebayesMultionmial		0.01	0.09	0.02	0.08	0.1	0.22	0.12
SMO	11.41	3.43	33.42	4.23	331.65	356.25		
J48	16.72	6.19	62.39	23.89	381.3	94.96		
Randomtree	2.27	0.85	4.64	1.69	10.04	3.07		
Randomforst	133.36		273.41	110.8	529.4	162.53		482.25

Figure 3: Time of building models

	best-10	most-10	best-20	most-20	best-50	most-50	best-200	most-200
zero-0	82.96	77.7	53.59	70.84	24.46	100.1	33.25	62.58
naivebay	77.18	64.73	57.05	69.11	30.43	64.81	32.32	43.48
oneR	62.43	101.6	48.23	97.89	28.4	0.67	43.59	26.64
naivebay	84.77	92.95	54.35	107.66	28.45	56.77	52.8	0.17
SMO	92.77	53.94	62.31	59.25	25.71	143.19	39.13	42.83
J48	40.87	69.58	23.29	66.38	22.55	63.73	42.67	50.58
Randomtr	62.31	63.11	24.13	83.17	33.64	27.79	90.89	27.6
Randomfo	146.19	100.23	53.59	39.78	82.91	51.69	76.83	62.97

Figure 4: Time of taking to test models on supplied test set