Project 2: Twitter Trolls and the Tweeters who Love them

Anonymous

1. Introduction

Twitter[1] is a popular microblogging website where users can post short text messages called tweets (Lee et al., 2011). This paper aims to analyze whether tweet text can help us to identify trolls on Twitter using Machine Learning methods.

The abbreviation for the following paper is declared below (see Table 1).

NOMENCLATURE	
Acc	Accuracy
Pre	Total Precision
Re	Recall
F-M	F-Measure
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
RF	Random Forest
L	Left Troll
R	Right Troll
O	Other

Table 1. The Nomenclature Table

2. Dataset

The whole dataset comprises 3 million Russian troll tweets, issued by 2848 different users[2]. Due to personal computer restriction, the dataset I use contains 122,637 tweets (the medium one). The dataset was portioned into 3 sets – training set, development set and test set. In which the training set is used to build a model, development set is used to evaluate the model and the test set is used to predict labels (left or right trolls) with the model I built.

The best-50 training dataset pre-processed and recorded the term of frequency for the terms with the greatest mutual information and chi-square values. Because of the User Id and Twitter Id in the dataset may contain some interference information for identifying trolls, I removed them in the pre-processing stage.

3. Methodology

3.1. Baseline

I used One-R (1-R) to create a baseline (create a rule for each attribute in the training data, then selected the rule with the minimum error rate as its one rule, discretizing numeric attributes). The accuracy of classified instances is 43.95%.

3.2. K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm is a simple and effective nonparametric classification method (Guo et al., 2003). I chose K=1, 2, 3, 4, 5 and 10 as parameters and Euclidean distance as the distance function. The result of different parameters with different distance weighting is shown as follows (see Figure 1).

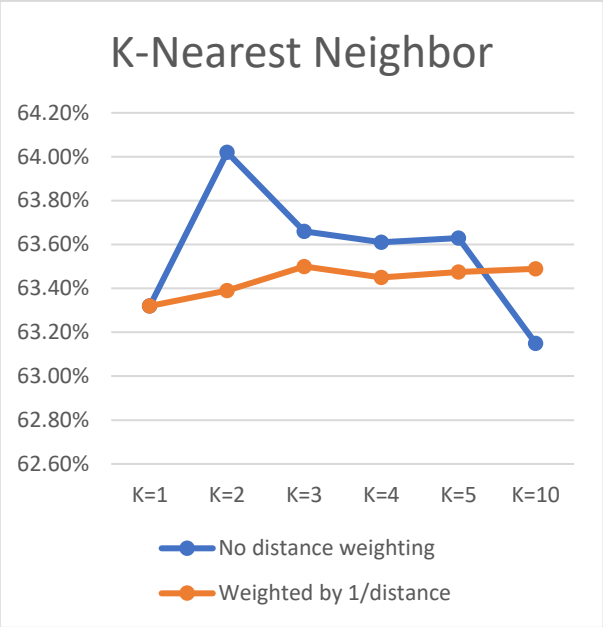

Figure 1. K-Nearest Neighbor Algorithm using Euclidean Distance

From the data, it can be seen that when there is no distance weighting, the KNN algorithm has a good performance with a 64.02% accuracy in distinguishing left/right trolls on Twitter compared to the One-R baseline. When K=2, the result is the highest.

3.3. Support Vector Machine

A Support Vector Machine (SVM) tries to find an Optimal Separating Hyperplane (OSH) between classes by focusing on training cases near the edge of class distribution and discard other training classes (Mathur and Foody, 2008). The parameter "c" represents the cost value for error cases. On one hand, increase the c value may improve the predicting performance. On the other hand, setting c value too high may lead over-fitting of models. I

[1] http://www.twitter.com

[2] https://github.com/fivethirtyeight/russian-troll-tweets/

tried c=1, 5 and 10 and the result is shown as follows (see Figure 2).

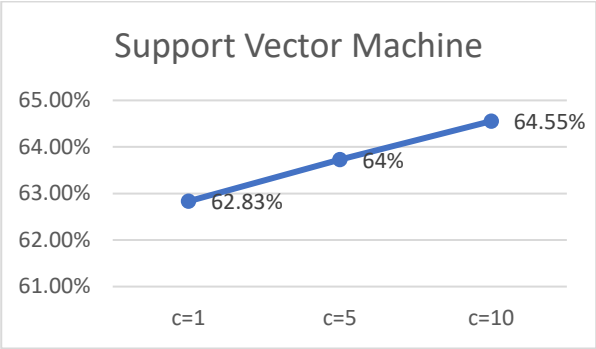

Figure 2. Support Vector Machine Algorithm

From the figure, we can see the accuracy of the system has slightly increased but is not very satisfying. Mostly is because of the number of attributes in this dataset is large and we should use the kernel to reduce the dimension of the dataset.

3.4.Random forest

Unlike previous methods, rather than changing the parameters of algorithms, I did some pre-processing on attribute selection. The dataset I use is 'medium best 50', which means there are fifty attributes given which have more or less mutual information. I use InfoGainAttributeEval[3] in Weka for attribute selection, ranking with 10-fold cross-validation. This gives me an average rank of attributes as follows (see Figure 3):

average merit	average rank	attribute
0.052 +- 0	1 +- 0	99 the
0.032 +- 0	2 +- 0	102 to
0.025 +- 0	3 +- 0	52 is
0.021 +- 0	4.8 +- 0.75	6 and
0.021 +- 0	4.8 +- 0.87	1 a
0.021 +- 0	5.4 +- 0.66	15 breaking
0.02 +- 0	7 +- 0	64 maga
0.018 +- 0	8 +- 0	101 this
0.017 +- 0	9 +- 0	93 rtamerica
0.015 +- 0	10.6 +- 0.8	78 of
0.015 +- 0	10.6 +- 0.49	92 rt
0.015 +- 0	11.8 +- 0.4	90 retweet
0.014 +- 0	13 +- 0	103 trump
0.011 +- 0	14.2 +- 0.4	97 tcot
0.011 +- 0	15.8 +- 0.87	98 that
0.011 +- 0	16 +- 1.48	13 black
0.011 +- 0	16.9 +- 0.94	12 beeth
0.011 +- 0	17.2 +- 0.98	71 news
0.01 +- 0	19.5 +- 0.67	79 on
0.01 +- 0	19.7 +- 0.78	46 i
0.01 +- 0	21 +- 0.89	10 bbsp
0.01 +- 0	21.7 +- 0.46	107 we

Figure 3. attribute selection

After attribute ranking, I use the random forest algorithm with the less rank 0, 5, 10 attributes deletion. The result is shown as follows (see Figure 4).

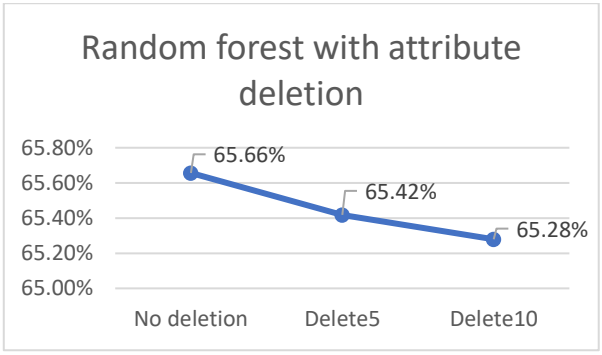

Figure 4. RF's Accuracy with Attribute Deletion

From the figure, we can see that the accuracy drops as the attribute deletion increases. One of the reasons for this phenomenon might be all of the attributes in 'bestXX' dataset contain mutual information more or less, and each of them has a determinable impact on troll prediction.

4. Analysis

To analyze all the algorithms mentioned above and figure out an algorithm with the best performance, I made a table for comparing the best results between different algorithms (see Table 2).

		Pre	Re	F-M	Acc
KNN	L	0.595	0.499	0.543	0.640
	R	0.678	0.417	0.517	
	O	0.648	0.969	0.777	
SVM	L	0.574	0.609	0.591	0.646
	R	0.832	0.268	0.406	
	O	0.621	0.980	0.761	
RF	L	0.627	0.538	0.579	0.657
	R	0.715	0.427	0.535	
	O	0.650	0.972	0.779	
1-R	L	0.516	0.271	0.355	0.439
	R	0.417	0.000	0.001	
	O	0.425	0.994	0.595	

Table 2. Algorithms Comparison Table

In the KNN algorithm, the highest accuracy for the evaluation on dev dataset is 64.02% with the parameter of K=2 and no distance weighting. In the SVM algorithm, the accuracy increases while the c (cost) value increases, and the highest accuracy among testing is 64.55%, but too high for the error value c may cause over-fitting. In the Random forest algorithm, the accuracy has the highest among testing, reaches 65.7%. Each indicator in this algorithm has a good performance in predicting troll users. The Random Forest algorithm can run efficiently on large databases and can handle thousands of input variables

[3] http://weka.sourceforge.net/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html

without attribute deletion, so delete attributes using this algorithm may cause negative impacts.

In this tweets text dataset, although the given attributes are the best XX attributes, there still may be some attributes which have mutual information missing, but Random Forest algorithm has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing. Furthermore, it generates an internal unbiased estimate of the generalization error as the forest building progresses.

What's more, all of the algorithms mentioned are better at identifying the Other class compared to left troll users or right troll users. The main reason is the attributes in the given dataset contain mutual information, and no matter users are left troll or right troll, they are likely to talk about these words. So that the class which doesn't mention these words would have a high prediction to the class 'Other'.

5. Conclusion

In conclusion, the statistics of the precision, recall and F-Measure are well-satisfying. i.e. more than a half of the troll users can be identified using machine learning methods. Due to the limitation of the computer device and time, the dataset running with these algorithms is small. But as long as the instances are big enough, the accuracy of identifying troll users by Tweet text can be increased.

References

[1] Lee, K., Palsetia, D., Narayanan, R., Patwary, M., Agrawal, A. and Choudhary, A. (2011). Twitter Trending Topic Classification. *2011 IEEE 11th International Conference on Data Mining Workshops*.

[2] Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K. (2003). KNN Model-Based Approach in Classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pp.986-996.

[3] Mathur, A. and Foody, G. (2008). Multiclass and Binary SVM Classification: Implications for Training and Classification Users. *IEEE Geoscience and Remote Sensing Letters*, 5(2), pp.241-245.