2017S2 KT Project 2 Reviews

Print Options:

☑ Include Questions & Answers ☑ Include Comments ☑ Include All Reviews ☑ Include File Info

Print

JIANGYUE YAN'S PEERMARK REVIEW OF ANONYMOUS'S PAPER
(0% COMPLETED)

ASSIGNED QUESTIONS

1. Please describe what the author has done, in a couple of sentences
2. Please indicate what you think the author has done well, and why
3. Please indicate what you think could have been improved, and why (we would prefer you to focus on the content, rather than the language, structure, or style)

COMMENTS LIST

No comments added

SUBMITTED FILE INFO

file name	Assignment2.pdf
file size	608.9K

"ASSIGNMENT 2" BY ANONYMOUS

COMP90049 Report: Project 2

1. Introduction

This report is aimed to solve a binary classification problem: whether a tweet contains an ADR or not. To achieve this goal, some knowledge about supervised machine learning will be introduce in this report. This report mainly focusses on Zero-R and Naive Bayes Classifier. Because they are simple and easy to understand, which will be helpful when we make analyses.

2. Dataset [1]

The source data contains three different part:
(1) train – it has a certain class for each instance in this file and can be used for training the machine
(2) dev – it has a certain class for each instance in this file and can be used for evaluation
(3) test – a file contains instances need to be predicted.
And all the part contains two types of files:
(1) .txt - contains the raw tweets
(2) .arff – contains the data after processing which can be used in WEKA (a machine learning tool, will be introduced later).

3. Classifier

3.1 Zero-R

Zero-R classifier is one of the simplest classifiers, this method is based on the historical data and selects a probability of the largest category as unknown sample classification results. That is, for any unknown sample, the classification results are the same. Zero-R classifier simply uses the category of majority as the predicted value. So, for Zero-R classify, no attribute will be used. Although this classifier does not have any predictive ability, it can be used as a contrast classifier with other classifiers. Therefore, this classifier is always regarded as a baseline to measure other classifiers.

3.2 Naive Bayes

The core of this classify is Bayes Theorem:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

And Naive Bayes' core idea is: *models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.* [2]
The steps to get a result are:
a) Choose some attribute from instance and regard them are independent
b) Calculate each attribute's probability in the class (YES and NO)
c) Multiply the all the probabilities in the same class, the larger one is the result.

4. Tool

Waikato Environment for Knowledge Analysis (Weka) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It is free software licensed under the GNU General Public License. [3]

5. Result

5.1 Result Data Presentation

Zero-R Prediction

		Predicted	
		Y	**N**
Actual	**Y**	962	0
	N	114	0

ACC: 89.4052%

Naive Bayes Prediction

		Predicted	
		Y	**N**
Actual	**Y**	829	133
	N	59	55

ACC: 82.1561%

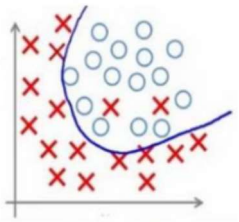
5.2 Analysis

5.2.1 Zero-R

Because most instances in train data is yes, so the classify just regard all the instances in dev data are yes and none of provided attribute makes contribute to the result, as in this model, the classify only care about the number of each class's instance.
Just thinking about ACC, it seems that this classify is not bad in this situation, but it has a limitation that if in this experiment, the NO class is the one we want to get, this classify will be quite useless. And another big problem is that if the probability of Yes class and NO class are close (like 49% and 50%), in that case, the result from this classify will be not credible.

5.2.2 Naive Bayes

To find the problem the classifier, I output all the predictions from WEKA and find the error one the original tweet text.
And I found the error can be concluded into these two types:
(1) After calculation, the false class's probability is indeed larger than the true one. This is inevitable because just as we learned in the lecture, the diagram below



Screenshot from lecture slide [13]
The model should focus on the essential patterns in the data, so it is allowed to ignore some special situation for prediction.

(2) No attribute is found in a tweet, so the result class is certainly NO but the actual is Yes. This is mainly because the coverage of our attribute is not large enough, we miss some important attribute in the text. To avoid this, we should find more valid attributes.
The basic different between (2) and (1) is that in the first situation, our attributes cover the instance, the reason to get a wrong answer is because of noise, but in the second it is out of our attribute coverage, so it should be regarded as a third type of class, however there are only two types classes, so the system regards it as NO class.

(3) Another hidden factor is that in this classifier, it considers all the attributes are independent, however, in actual, there are always relations between two words: for example, attribute "caused/causing" and "allergic" have same probability in YES class and NO class, which means they have no influence to the final result based on this classifier, but when we consider them together, it seems that when they appears at the same time, there will be high probability that a tweet contains ADR.

5.2.3 Comparison and Improvement

Compared with these two classifiers, we can find that even Naive Bayes seems more complex than Zero-R but its ACC is lower than Zero-R. The reason causes this is the choices of attribute.
So, to improve Naïve Bayes Classifier, I suppose that we can find attribute in that way:
(1) Find all the attribute which might have influence on the final result (through WEKA it is easy to distinguish this and if have no, we can delete it latter)
(2) Build relationship manually, just as mentioned before, we can create attribute like "cause& allergicBothAppear"

6. New Feature Selection

6.1 Introduction

I think when people want to say something when they have ADR, they will always mention the drugs they have used, so when a drug appears in a tweet, there are high probability this tweet is say something after they use a drug.
To achieve I add a new attribute named "inDrugList" for the data, which has two values:1 for yes and 0 for no. And then I create a list of drugs' names and compare it with tweets text to give value for the new attribute. The drug list is from an online website [5] which has a statistic for Top 40 Drug Searches.

6.2 Result

Naive Bayes Prediction

		Predicted	
		Y	**N**
Actual	**Y**	825	137
	N	59	55

ACC: 81.7844%

6.3 Improvement

Compare the result with the original one, ACC has a slightly drop. Back to the original tweets text, I found that sometimes when people mentioned a drug they may also want to say the drug has some good effect, so it is not an ADR. This situation also can be observed in WEKA Classifier Output:

inDrugList		
mean	0.2431	0.1984
std. dev.	0.429	0.3988
weight sum	2793	373
precision	1	1

Screenshot from WEKA
To solve this problem, I think we can check the existence of some side effect words, like sleepy, tired, feel bad and so on. And then, if a drug and bad reaction words appears together, there should be high probability that a tweet contains ADR.

6.4 Result

Naive Bayes Prediction

		Predicted	
		Y	**N**
Actual	**Y**	829	133
	N	59	55

ACC: 82.1561%

Back to initial ACC, and from the output:

inDrugListsbadReaction		
mean	0.0125	0.0214
std. dev.	0.1667	0.1667
weight sum	2793	373
precision	1	1

Screenshot from WEKA

The instance contains my attribute are really small, so I think if add more words to the list, the result will better.

7. Conclusion

Overall, we can find that Naive Bayes rely heavily on the choice of attributes, and whether attributes are independent or not should also be considered when we make a choice. So, to improve the ACC for this classifier, I think the rules I mentioned in 5.2.3 can be considered as a option, and we should not except that we can find a perfect model in the first time, we should update and compare again and again and we can get a better model at last.

8. Reference

[1] Abeed Sarker and Graciela Gonzalez. (2015) Portable automatic text classification for adverse drug reaction detection via multi-corpus training. Journal of Biomedical Informatics, 53: 196-207.

[2] Naive Bayes classifier (2017, Sept.). Retrieved from https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[3] Weka (machine learning) (2017, Oct.). Retrieved from https://en.wikipedia.org/wiki/Weka_(machine_learning)

[4] Nicholson, Jeremy, Justin Zobel and Karin Verspoor (2017). COMP90049 Knowledge Technologies [Lecture slides]. The University of Melbourne, Sept & Oct 2017.

[5] Drug Index (2017, Oct.). Retrieved from https://www.drugs.com/drug_information.htmlstatistics