Assignment1 Report

DV2578-Machine Learning

Rong Peng
19951120-7004
Blekinge Institute of Technology

I. Introduction

In this project aims at implementing a concept learner from Spambase dataset[1] to classify spam or ham. Firstly computing the size of the hypothesis space and the number of possible conjunctive concepts. Secondly realizing the Least General Generalisation(LGG) algorithm and verify that it works as expected. Thirdly report the statistics and the generated model.

II. Explanation

A. Dataset

Dataset was split into train and test datasets respectively. 60% of full data used as training and 40% of full data used as testing. For the 57 continuous variables, I used pandas *cut* method to divide them into three intervals, and labeled as 1/2/3. The least upper bounds (lub) are the minimum number in each column, and the greatest lower bounds (glb) are the maximum number in each column. Then calculated the averages for each column, for example, when spam==0 called the mean avg0, otherwise called it avg1, and comparing avg0 and avg1, if avg0>=avg1, then midlub is avg1, midglb is avg0. The parameter *bin* in *cut()* method should be [lub,midlub,midglb,glb], monotonically increase. Therefore, for each of the 57 attributes have three values.

B. Hypothesis Space and Conjunctive Concepts

According to the P.Flach[2], and combine this case, the dataset have 57 features, then the hypothesis space is 3^{57}, and given the possible condition each attribute could be 'ignored', the number of possible conjunctive concepts is 4^{57}.

C. LGG Algorithm Realization

I implemented the Algorithm4.1 and Algorithm 4.2. LGGSet algorithm repeatedly apply a pairwise LGG operation, and recursively call the *LGG-Conj* algorithm to find the least general generalisation of a set of instances.[2]

In the procedure, I faced two challenges, the first one comes from recursively calling LGG-Conj then how to save the changing temporary hypotheses. My solution is setting conditional judgement to determine whether it is the first call of LGG-Conj, if not,judge the y['spam'] is 0 or 1, then do the conjunction of all literals common to x(temporary hypothesis, now it is a dictionary) and y, if the y[features] not in x, then append value in x, otherwise, do nothing. The second challenges is how to deal the large number of zero. In my code, there are two python files, LGG1 and LGG2,the difference between them is the way dealing zero. LGG1 sees 0 as the boundary.LGG2 replace 0 as numpy.nan, then selected 12 features recovering as 0. LGG2 idea comes from the Dropout method used in neural network, some of the attributes 0 occupies most of it, if use NAN for all of the dataset, which invisibly deleted a lot of data. I used Excel Filter function to check which features values change greatly when switching spam to ham, adjusted and picked 12 features. Therefore I recovered the 12 attributes NAN back to 0. However, from LGG1 result, the accuracy and f1 score can even achieve 99%, which was overfitting. In the LGG2 it occurs under-fitting, the accuracy only achieve 12.66%.

And I found it was hard to improve the LGG2 accuracy, if I add or switch other attributes NAN recovered as 0, the classifier occurred overfitting or under-fitting again.

D. Results and Statistics

lub~midlub:1 midlub~midglb:2 midglb~glb:3

	spam=1		spam=0	
LGG1	telnet	1,3	415	1,3
	lab	1,2	857	1,3
	conference	1,3		
	857	1		
	cs	1,3		
LGG2	telnet	1,3	font	1,3
	lab	1,2	415	1,3
	people	1,3	857	1,3
	857	1		
	cs	1,3		

	Accuracy	Precision	Recall	F1-score
LGG1	99.84%	100%	99.60%	99.80%
LGG2	12.66%	17.32%	29.64%	21.87%

III. Discussion

In the future work, I hope to do three aspects improvements. The first one is the dataset itself, the full data has 4601 instances but only contains 1813 Spam instances, hope to balance the number of spam and ham instances, and do the data augmentation. And try to add some new words frequency as attributes. The second one I hope to discover the additional boundaries I didn't discuss, like less than lub(the minimum) or large than glb(the maximum).In addition, I can try to change the way I divided the intervals in this project. The third one is I found the classifier is very easy to occur overfitting or under-fitting, thus I am thinking to study the sparse dataset preprocessing knowledge, which is the significant initial step.

IV. Conclusion

From the calculation, can know the hypothesis space is 3^{57}, the number of possible conjunctive concepts is 4^{57}.

The accuracy for the classifier can even achieve 99.84%, f1-score can achieve 99.80%.

1/2/3 seperately represents features interval. The result of the conjunction for 'spam=1' should be (telnet=1,3)∩(lab=1,2)∩(857=1)∩(cs=1,3)∩(conference =1,3), for 'spam=0' should be (415= 1,3)∩(857=1,3).

REFERENCES

[1] "UCI Machine Learning Repository: Spambase Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase. [Accessed: 20-Nov-2018].

[2] P. Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, 2012. Page 108-113