Recognizing SMS spam

EECS349 Machine Learning
Northwestern University
Teke Xu, Yuzhu Wang, Anqi Xing, Xi Wu
Anqixing2015@u.northwestern.edu

Abstract

Intelligent phones are widely used recently with the rapid development of requirement of communication among human-beings. People receives tons of messages everyday which makes them waste time on reading un-useful texts and influences them obtaining important information. The main goal of our project is recognizing spam or ham messages by using appropriate algorithms and machine learning skills. Messages can be classified by using message length, key words in text etc. as numerous special attributes. This project took advantage of several machine learning algorithms and achieved automatically filtering spam messages and keep ham messages with high accuracy.

1. Dataset

1.1 Data source

The dataset of this project is collected from messages of group members manually and three internet public resources of SMS data. The links of public datasets are shown as follows:

1.http://www.kdnuggets.com/2011/06/sms-spam-collection-data.html

2.http://www.dit.ie/computing/research/resources/smsdata/

3.https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

1.2 Dataset Pre-processing

Considering the number of ham messages in all datasets is larger than spam messages in all datasets which matches reality, preprocessing data is indispensable and important. The whole process of data preprocessing is shown in fig.1.

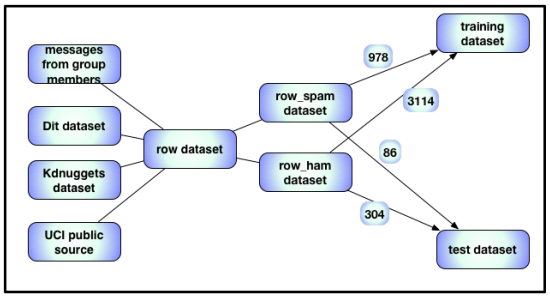

Fig.1 Process of preprocessing dataset

From fig.1, first, getting an unbiased dataset, combining all four datasets and saving it as "raw_dataset.txt". Then, we wrote python code to split the dataset "raw_dataset.txt", according to "ham" and "spam" label, saved as "ham.txt" and "spam.txt". Since we have a large dataset of ham and spam and the difference between the number of them is huge, we randomly selected the training dataset from "raw_dataset.txt". Choosing 978 spam messages and 3114 ham messages and merged them to be the final training dataset. As for obtaining independent testing dataset from training dataset, 304 ham messages and 86 spam messages were chosen, then merged them as testing dataset. After preprocessing and gaining both training dataset and testing dataset, the ratio of ham messages and spam messages is 3 which is a good proportion. We tried to use the original dataset in our mid-process of project, the result shows not as good as the new dataset with new ration of ham and spam messages.

2. Features Extraction

1) Wrote a python program to import the training dataset (which is .txt form) and found out the frequencies of all the separated words in the training dataset, that is, the number of each word existing in the

training dataset. To be more specific, if the number of a word is large, it means this word is more commonly used such as "you", "me", "to", etc.
2) Actually, we did the step 1 separately in "ham.txt" and "spam.txt". The result we got is shown in fig.2.

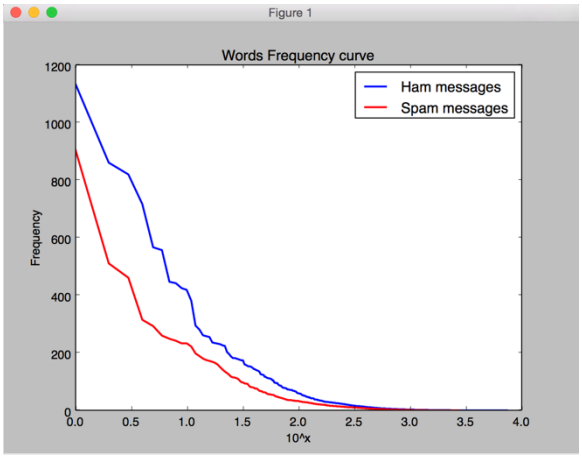

Fig.2 Words frequency curve

3) According to the large difference of words' frequency in whole dataset, we selected the smooth part of the whole dataset. In fig.2, both ham messages' curve and spam messages' curve are shown smooth when the frequency of words is between 10 to 1000. Therefore, choosing attribute words from 10 to 1000 (we then adjust the attributes later to get best the outcome) to test whether this method works. Then the number of attributes is 990 * 2 minus repeated ones and 1687 attributes are remaining in total. The original attributes are:

Link to attributes:
angelanki.github.io/ml.proj/attribute.html

4) At the last of the attribute vector, we add the length of each sample message as an attribute.

3. Training & Test

1) According to the attribute vector we have acquired, for each attribute is a word, so we search each word in every message. For one certain word, if we can find it in an instance (text message), then the value of the corresponding attribute is "1" otherwise "0".
2) Wrote code to realize step 1 as well as generated "training_set.csv" and "test_set.csv" for further analyzing.
3) Used classifiers in "weka" to train and test it after Step 2. We used several algorithms to get relatively high-accuracy classifiers, and finally 3 classifiers to be considered:

Table1: validation accuracy of different algorithms

	10-fold Cross Validation	Test Validation
Naive Bayes	93.91%	94.10%
Decision Tree	94.06%	94.87%
Random Forest	97.39%	98.46%

The learning curves of 3 classifiers we chose:

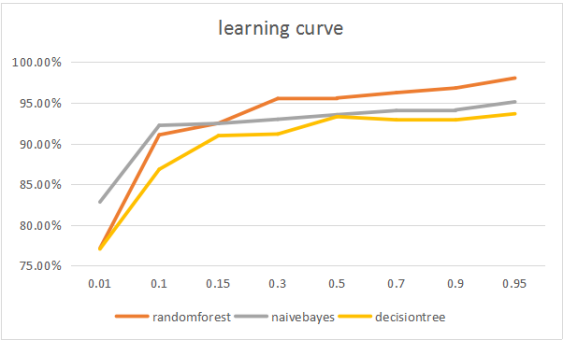

Fig.3 Learning curve of three algorithms

4) From the result above, "Random Forest" can be the best classifier so far. For more information on validation result, table 2 shows the detailed accuracy of Random Forest. In addition, in table 3, the confusion matrix shows that for all 86 spam messages, 2 are falsely classified as ham; and for all 304 hams, 4 are falsely classified as spam.

Table2. Detailed Accuracy by Class

class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
spam	0.955	0.007	0.977	0.955	0.966	0.996
ham	0.993	0.045	0.987	0.993	0.99	0.996
weighted Ave	0.985	0.037	0.985	0.985	0.985	0.996

Table3. Confusion Matrix

a	b	<--classified as
84	4	a=spam
2	300	b=ham

4. Conclusion and future works

We have now achieved several high accuracy algorithms to recognize spam messages, of which Random Forest is the best. In the future, we can update the attribute vector monthly or annually since spam messages are always being generated. Also, it is feasible to write an app including this algorithm to block spam messages.

5. Work allocation

Our group discuss together about the data collection and how to process the dataset, then discuss the learning algorithms and using "weka" to analyze the processed dataset.

- Anqi Xing: collect and process the dataset and build the website
- Yuzhu Wang: process the dataset and train and test the processed dataset
- Teke Xu: train and test the processed dataset and analysis and write the report
- Xi Wu: train and test the processed dataset and analysis and write the report