Spam Filtering using Association Rules and Naïve Bayes Classifier

Tianda Yang, Kai Qian, and Dan Chia-Tien Lo
Department of Computer Science
Kennesaw State University
Marietta, GA, USA
{tyang4, kqian, clo}@spsu.edu

Kamal Al Nasr
Department of Computer Science
Tennessee State University
Nashville, TN, USA
kalnasr@tnstate.edu

Ying Qian
Department of Computer Science
East China Normal University
Shanghai, China
yqian@cs.ecnu.edu.cn

***Abstract*—E-mail service is one of the most popular Internet communication services. Thousands of companies, organizations and individuals use e-mail every day and get benefit from it. However, an amount of spam emails always hang around us and bring down our productivity. We urgently need a spam filtering to clean up our network environment. A spam filtering using Association Rule and Naïve Bayes Classifier is recommended here. Instead of focusing on increasing spam precision rate, we try to preserve all non-spam emails as the first priority. In the real world applications and services, that's what we should do. In this paper, we also provide the comparison between using both Association Rule and Naïve Bayes Classifier algorithms and just using Naïve Bayes Classifier.**

Keywords—Spam; Association Rule; Naïve Bayes Classifier; Spam filter; Apriori algorithm; Hadoop

I. INTRODUCTION

An electronic message is spam if the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; and the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent [1]. E-mail spam is one of the electronic spams and it can be defined as unsolicited bulk email (UBE).

Whether the email message is an advert, a free product or service, a porn picture, a scam, or irrelevant content, then the email can be considered as spam. Spam is a serious problem because it costs time, money and resources. All the cost is borne by recipients and services providers. Let's look at the first quarter of 2015 statistics [2]. The proportion of spam in email traffic was 59.2%, which is 6 percentage points lower than in the previous quarter, but still large numbers.

Although there are many types of spam emails, the underlying characteristics of spam cannot be changed. We keep track of words in each email to start our filtering. For any text-based big data analysis, data pre-processing is the most important step. This step helps us to remove meaningless words and text symbols. This step also includes Natural Language Processing (NLP). NLP is the ability for a computer to understand human languages and it can help us to classify text.

978-1-4673-9088-0/15/$31.00 ©2015 IEEE

In many approaches, Naïve Bayes Classifier always does surprisingly well, so it has been widely used in several spam filtering. Naïve Bayes has two popular models: Bernoulli model and Multinomial model. For Bernoulli, the significant difference from Multinomial is not only because it does not take into consideration the number of occurrences of each word, but also because it takes into account the non-occurring terms within the document [3]. Thus, we use Multinomial Naïve Bayes model. After data pre-processing, each distinct word in the dataset is defined as an attribute and the value of the attribute is the English word found in the dataset.

In addition, this paper try to use association rule mining to improve Naïve Bayes spam filtering. Association rule finds relationship between seemingly unrelated data among a large set of data items. Therefore, association rule always bring surprise to us. An apriori algorithm is one of the influential algorithms for frequent itemset mining and association rule learning. The classic example is the famous "Beer" and "Diaper" association problem that is often mentioned in data mining books and tutorials.

Besides, we use Enron's spam/ham email dataset [4] and during training stage, we prepare to analyze nearly one million original words, so Hadoop training is the best way for us. Hadoop is an open source framework for distributed processing of very large datasets. The reason why Hadoop is well suited to big data analysis is because Hadoop works by breaking the data into pieces and assigning each "piece" to a specific node for analysis [5]. Therefore, Hadoop framework is widely used in data training and big data analysis. And also, Hadoop Training and Certification Courses are becoming more and more popular with each passing year.

In this paper, a brief introduction about the techniques and algorithms was given in the paragraphs above. Related work is discussed in section II. The work implementation in this paper is illustrated in Section III, include pre-processing, Hadoop training and Naïve Bayes Classifier. The work improvement by association rule and comparison is explained in Section IV. Lastly, the conclusions and future work discussions are presented in section V. Section VI is the acknowledgements.

II. Related Works

Electronic message has become more popular and significant for internet social networks. E-mail is one of the electronic message services, and people use emails for personal and business communication every day. According to the 2013 Email Marketing Benchmark Report [6], 247 billion emails are sent every day. That's one email every 0.00000035 seconds. However, E-mail spam is an unfortunate problem to hamper the development of E-mail, about 60% spam emails appear in the email traffic [2].

The good news is many researchers have already proposed several solutions to the spam problem involve detection and filtering by machine learning algorithms. Naïve Bayes Classifier was suggested to give good results and used in many filtering software. The first mail-filtering program using Naive Bayes Classifier is Jason Rennie's ifile program. And then, the first scholarly publication on Bayesian spam filtering was proposed by Sahami et al. [7]. In 2002, Graham decreased the false positive rate to use as a single spam filter [8] [9]. A number of solutions use cluster as a part of spam detection, such as KNN algorithm [10] and SVM classification [11]. There are not many research people using association rule in spam detection and filtering, so association rule may help us to get a breakthrough.

Spam emails may have different styles. Spammers always create special format to avoid detection. Some spams include an embedded images, links or attachments, while others may contain space or strange characters in a word. Those spams caused researchers work hard to create their filtering. Ketari's work illustrates the image spam filtering techniques [12]. Image spam filters use algorithms such as SIFT, TR-FILTER and NDD. Deshmukh proposed a spam filtering system using Sobel operators and AOCR [13] for filtering text and image based emails.

Since research for spam needs a large dataset, some techniques and frameworks such as Hadoop, MapReduce and HDFS are become popular. Tran Ho proposed fingerprinting techniques combine with sim-hash algorithm to detect spams. This is a novel similarity-based method that implements by using Hadoop framework [14]. We apply the method of association rule into spam filtering system, and this is a frequency-based method that also implements on Hadoop framework.

III. Implementation using Naïve Bayes Classifier

Our experiment has three phases: pre-processing, training and testing. The main task in the pre-processing phase is to remove zero contribution words and replace words to the same category. In the training phase, we count the total number of words and their occurrences to calculate the contribution probability to each distinct word or category. At the last phase, we classify and calculate some new email dependent upon the training result to get the precision rate.

A. Pre-Processing

We use Enron's dataset [4] for training and testing. There are 6000 emails consisted of 1500 legitimate (ham) emails and 4500 spam emails. Emails have already been labeled as spam or ham. Messages in a sample email are shown in Fig.1.

```
Subject: buy cheap viagra through us .
hi ,
we have a new offer for you . buy cheap viagra through
our online store .
- private online ordering
- no prescription required
- world wide shipping
order your drugs offshore and save over 70 % !
click here : http : / / aamedical . net / meds /
best regards ,
donald cunfingham
no thanks : http : / / aamedical . net / rm . html
```

Fig. 1 Sample email format

Since we apply each distinct word in the dataset as an attribute, emails should be clean and categorized. We consider the following steps to clean up our dataset.

- Find out links, URLs and email addresses in emails. Transform links and URLs to word "url", and transform email addresses to word "email". Transform date or time to the word "date". Date format can be "Jul 10, 2015" or "7/10/2015". Time format can be "12:20:35", "12:20", "12:20 pm" or "2 pm".
- Remove punctuation symbols and other text symbols except "$" and "%". Transform "$" to word "price", only if there are digitals after "$", e.g. "$70" or "$ 70". Transform "%" to word "percentage", only if there are digitals before "%", e.g. "70%" or "70 %". If there are digitals with neither "$" nor "%", transform to word "number".
- Split words by "space". Remove preposition [15] and common conjunction words, e.g. "for", "and", "but", "yet", "nor", and other zero contribute words like "the".
- Use Stanford NLP [16] for English words' lemmatization. For example, "am", "is" and "are" should be transform to the same word "be". In addition, if a word has entity tag, transform the word to its entity tag. For example, word "Lenovo" has the entity tag "ORGANIZATION" and word "Washington" has the entity tag "LOCATION".
- Remove words with less than three letters, e.g. "a", "it", "be". In case of any omission, especially higher contribution features such as URLs, links, transform words "www", "com", "net", "http", "org", "edu" to word "url".

We finished the pre-processing by the above steps and we get our training dataset to start our next phase — training.

B. Training

After pre-processing, we first run a word count application on Hadoop MapReduce framework to get total number of words and their occurrences.

The unique storage method in Hadoop is based on distributed file system called HDFS. Our training data are saved in HDFS and we propose to use MapReduce framework to handle the amount of words. MapReduce approach is to use <key, value> pairs and the groups that will be received in the reduce function will be grouped by the key.

Map: $<k_1, v_1> \rightarrow list<\mathbf{k_2, v_2}>$

Reduce: $<\mathbf{k_2}, \mathbf{list\ (v2)}> \rightarrow list\ (k_3, v_3)$

Then, we can get fast and accurate results by this framework, and the procedure of MapReduce is shown in Fig.2.

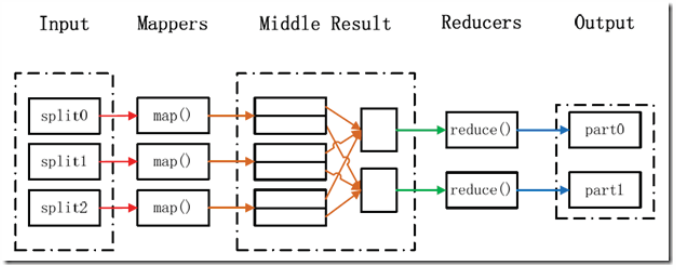

Fig. 2 Procedure of MapReduce

To apply Naïve Bayes Classifier, each distinct word in the training set is defined as an attribute and the value of attribute is the number of occurrences. Multinomial Naïve Bayes Classifier to Text Classification is given by:

words ← all words in email dataset

$$V_{NB} = \arg\max P(V_j) \prod_{i \in words} P(x_i \mid V_j) \qquad (1)$$

To simplify the above equation, we list all the necessary parameters and equations to calculate whether an email is spam.

First, we need to calculate the probability for spam emails and ham emails, and we assign the probability P(spam) and P(ham) to:

P(spam) = (No. of spam words) / (No. of words) (2)

P(ham) = (No. of ham words) / (No. of words) (3)

Then, we need to calculate the contribution probability for each word in spam/ham email. We use the following equation:

$$P(\text{word} \mid \text{spam/ ham}) = \frac{n_k + 1}{n + vocabulary} \qquad (4)$$

where, n_k = number of occurrences of a specific word in spam/ham emails, n = number of words in spam/ham emails, vocabulary = number of distinct word in all training dataset.

After calculating all the word contribution probabilities for spam and ham emails, save it and we are finished our training phase and able to classify a new email by the following equation:

$V(spam) = P_1(word_1|spam) * P_2(word_2|spam) *\ldots* P_n(word_n|spam) * P(spam)$ (5)

$V(ham) = P_1(word_1|ham) * P_2(word_2|ham) *\ldots* P_n(word_n|ham) * P(ham)$ (6)

where, $P_1, P_2, \ldots, P_n$ are the contribution probabilities in spam or ham and they are calculated by equation (4). If V(spam) > V(ham), the email is considered as spam. Otherwise, the email is ham.

Our experiment result for training phase is shown below:

Number of words (total): 963625

Number of spam words: 708033

Number of ham words: 255592

Vocabulary: 57114

We can get P(spam) = 73.476% and P(ham) = 26.524% based on the equation (2) and (3). Also, we record our training result to further use. Fig.3 shows partial training result for ham email.

```
…
willingness        3.8374703395521674E-5
bacon    6.395783899253612E-6
antisocial         3.197891949626806E-6
boundlessness      3.197891949626806E-6
braunlegal         6.395783899253612E-6
candle   6.395783899253612E-6
report   0.0016469143540578052
tjuxf    3.197891949626806E-6
coverall 3.197891949626806E-6
hilarium 3.197891949626806E-6
surfing  3.197891949626806E-6
annyong  3.197891949626806E-6
saviour  3.197891949626806E-6
poindexter         1.598945974813403E-5
bruzj    3.197891949626806E-6
diore    3.197891949626806E-6
scramble           1.598945974813403E-5
islandium          6.395783899253612E-6
download           1.8867562502798157E-4
dirtiness 3.197891949626806E-6
coplanar 3.197891949626806E-6
insufflation       3.197891949626806E-6
disruption         3.8374703395521674E-5
…
```

Fig. 3 Training result for ham email

Words on the left side are the partial vocabulary of ham/spam email, and the numbers on the right side are the contribution probabilities for each word as ham/spam email.

C. Testing

We test three times to get a stable precision rate. If word in the testing emails cannot be found from the previous training result, we will use equation (4) and $n_k = 0$ to calculate its contribution probability for both spam and ham email. Table I shows the testing result.

TABLE I. TESTING RESULT

#	No. of Emails		Spam Emails		Ham Emails	
	Spam	*Ham*	*Classify as spam*	*Precision rate*	*Classify as ham*	*Precision rate*
1	350	250	333	95.14%	236	94.40%
2	400	200	379	94.75%	189	94.50%
3	450	150	430	95.56%	140	93.33%

The average of spam precision rate is 95.15%, and the average of ham precision rate is 94.08%. This result is good enough for just using Naïve Bayes Classification algorithm.

However, our purpose is to preserve ham emails as the first priority. We cannot be satisfied with the testing result shown below, so we propose to have an enhancing implementation to improve the ham precision rate.

IV. EXPERIMENT IMPROVED BY ASSOCIATION RULE

Association rule helps uncover relationships between seemingly unrelated data in a relational database. There are several algorithms of association rules. One of the most popular algorithms is Apriori algorithm, which is used to extract frequent itemsets from large database and get the association rule for discovering the knowledge.

Apriori algorithm is an influential algorithm for mining frequent itemsets. Since the Algorithm uses prior knowledge of frequent itemset it has been given the name Apriori. The most important step in our experiment is to find frequent itemset. Apriori uses a bottom-up approach, where frequent itemsets are extended one item at a time. Each of frequent itemsets will occur at least as frequently as a pre-determined minimum support count. The support is the percentage of task-relevant data transactions for which the pattern is true.

A. Execution of Apriori Algorithm

In our experiment, each distinct word is considered as an item in Apriori algorithm, so we define the minimum support equals to 10%. The dataset contains 4500 spam emails and 1500 ham emails. To effectively use the dataset, all the emails have been preprocessed.

We define the final frequent itemset, which contains the most items in one frequent itemsets. We propose to find a spam final frequent itemset and a ham final frequent itemset. The result in our experiment is shown in Table II.

TABLE II. RESULT OF FINAL FREQUENT ITEMSET

Spam Final Frequent Itemset	Ham Final Frequent Itemset
DATE	DATE
have	have
number	number
that	that
this	this
url	url
you	you
	DURATION
	PERSON
	enron
	gas
	please
	subject
	they
	will

After executing Apriori algorithm, we found spam final frequent itemset is a subset of ham final frequent itemset. The result indicates that spam emails have fewer similarities than ham emails. Based on this conclusion and above final frequent itemsets, we can design a new method to improve our spam filter by combining Association Rule and Naïve Bayes Classifier.

B. Combination and Comparison

Words in the final frequent itemsets called frequent words. To combine with Naïve Bayes Classifier, the first thing to do is calculate the contribution probability for all frequent words, equation is given by:

$$P_{\text{Apri}}(\text{word} \mid \text{spam/ ham}) = \frac{n_{\text{f}}}{n_{apri} + vocabulary} \quad (7)$$

Where, n_f = number of occurrences of a frequent word in a coming email, n_{apri} = number of occurrences of all frequent word in a coming email. vocabulary = number of words in the spam/ham final frequent itemset. The coming email has been preprocessed as well.

The purpose to calculate P_{Apri} is to emphasize the importance of words in final frequent itemsets. From previous spam and ham final frequent itemsets, we know that ham emails may have more frequent words then spam emails. To calculate P_{Apri} of frequent words may increase the ham precision rate.

For a coming email, we calculate the P_{Apri} of all frequent words by equation (7). And then, use the calculated results from equation (7) to replace the original value in the training result. After this operation, contribution probabilities of frequent words in the new training result should be increased. Based on the new training result, we are able to classify emails by equation (5) and (6).

In our experiment, comparison between using both algorithms and just using Naïve Bayes Classifier is shown in Table III. Both implementations use the same testing datasets.

TABLE III. COMPARISON

Naïve Bayes Classifier	No. of Emails		Spam Emails		Ham Emails	
	Spam	*Ham*	*Classify as spam*	*Precision rate*	*Classify as ham*	*Precision rate*
1	350	250	333	95.14%	236	94.40%
2	400	200	379	94.75%	189	94.50%
3	450	150	430	95.56%	140	93.33%
Naïve Bayes & Association Rule	**No. of Emails**		**Spam Emails**		**Ham Emails**	
	Spam	*Ham*	*Classify as spam*	*Precision rate*	*Classify as ham*	*Precision rate*
1	350	250	323	92.29%	249	99.60%
2	400	200	369	92.25%	198	99.00%
3	450	150	411	91.33%	149	99.33%

The average of spam precision rate is 91.96%, and the average of ham precision rate is 99.31%. The comparison result indicates that using two algorithms may bring down the spam precision rate but increase the ham precision rate. Spam precision rate still greater than 90% and we can accept. Ham precision rate is up to 99.31%, which is a very high probability in classification.

V. CONCLUSION AND FUTURE WORK

In this paper, firstly, we applied Naïve Bayes Classifier for spam filtering. And then, we proposed an enhancing implementation to combine Naïve Bayes Classifier and Apriori algorithm. The purpose for the enhancing implementation is to improve ham precision rate. Based on the comparison table, we achieve our goals and get a satisfactory conclusion.

However, there are also many problems occurred in our work. In pre-processing phase, the number of vocabulary may be so large. It's because spam emails may have many spaces, and one word may split by spaces to two words. Fortunately, Naïve Bayes Classifier can handle this problem. The other problem is in the classification part to calculate equation (5) and (6). A "double" sometimes cannot represent the calculating result, and we use some methods to bypass this problem.

Association rules always bring surprise to us. The main purpose for our research is to preserve ham emails in the classification. Even though spam precision rate is decreased in the enhancing implementation by combining Naïve Bayes Classifier and Apriori algorithm. We still believe the result is valuable.

In the future work, we will solve the problems occurred in this work and keep the spam precision rate as well by providing an appropriate equation to calculate P_{Apri}. Also, we should take much effort in pre-processing. Pre-processing has a great impact to training and classification results. If possible, we will combine some other algorithms to improve our filter.

VI. ACKNOWLEDGMENTS

The work is partially supported by the National Science Foundation under award: NSF proposal #1244697, #1438858, #1241651. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] The Definition of Spam. *The Spamhaus project.* https://www.spamhaus.org/consumer/definition/ [Jul. 15, 2015]

[2] S. Tatyana, V. Maria, D. Nadezhda, "Spam and Phishing in the First Quarter of 2015". https://securelist.com/analysis/quarterly-spam-reports/69932/spam-and-phishing-in-the-first-quarter-of-2015/ [Jul. 15, 2015]

[3] V. Vasilis, "Machine Learning Tutorial: The Naive Bayes Text Classifier". http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/ [Jul. 16, 2015]

[4] Enron-spam dataset, http://www.aueb.gr/users/ion/data/enron-spam/ [Jul. 16, 2015]

[5] P. Brien, "Hadoop clusters: Benefits and challenges for big data analytics". http://searchstorage.techtarget.com/tip/Hadoop-clusters-Benefits-and-challenges-for-big-data-analytics [Jul. 17, 2015]

[6] B. Mark, "8 email statistics to use at parties". Sponsored by: 2013 Email Marketing Benchmark Report. http://www.email-marketing-reports.com/iland/2009/08/8-email-statistics-to-use-at-parties.html [Jul. 17, 2015]

[7] M. Sahami, S. Dumais, D. Heckerman and E. Horvitz. 1998. "A Bayesian Approach to Filtering Junk E-mail". In *Proc. of the AAAI'98 Workshop on Learning for Text Categorization*, pp. 1048–1054.

[8] Brian Livingston (2002), Paul Graham provides stunning answer to spam e-mails. http://www.infoworld.com/article/2674702/techology-business/paul-graham-provides-stunning-answer-to-spam-e-mails.html [Jul. 21, 2015]

[9] Paul Graham (2003), Better Bayesian filtering. http://www.paulgraham.com/better.html [Jul. 21, 2015]

[10] Firte, L., Lemnaru, C. and Potolea, R. 2010. "Spam Detection Filter Using KNN Algorithm and Resampling," in *Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conferenc* , pp. 27 – 33.

[11] Kyriakopoulou, A. and Kalamboukis, T. 2006. "Text Classification Using Clustering," in *ECML-PKDD Discovery Challenge Workshop Proceedings.*

[12] L.M. Ketari, L.M. Chandra, and M.A. Khanum. "A Study of Image Spam Filtering Techniques." *4th IEEE Internet. Conf. Computational Intelligence and Communication Networks, 2012.*

[13] S.S. Deshmukh, P.R. Chandre, "Survey on: Naive Bayesian and AOCR Based Image and Text Spam Mail Filtering System". *International Journal of Emerging Technoogy and Advanced Enginerring.*

[14] P.T. Ho, H.S. Kim,S.R. Kim," Application of Sim-Hash Algorithm and Big Data Analysis in Spam Email Detection System". 2014 Conference on Research in Adaptive and Convergent Systems, pp. 242-246.

[15] "English Prepositions List", EnglishClub www.englishclub.com/grammar/prepositions-list.htm [Jul. 24, 2015]

[16] Stanford CoreNLP, The Stanford Natural Language Processing Group. http://nlp.stanford.edu/software/corenlp.shtml [Jul. 25, 2015]