Validating User Spam Reports in Chat Networks

Arjun Gopalan, Ashish Mathew, Prachetaa Raghavan {arjung, amathew9, pracheta} @stanford.edu

1. Abstract

In this paper, we consider the problem of validating user reports and detecting malicious users in chat networks. Bad behavior in chat networks is usually handled by a rather tedious system where the victim generates a report for the administrators to act on. However, since most of these reports are spurious, administrators have to manually validate each report before taking action. We analyze the effectiveness of Naive Bayes and Support Vector Machines using a couple of different kernels. Our experiments indicate that a SVM based algorithm using a polynomial kernel with the right features achieves an overall accuracy of 98% - 99%. The recall for the positive and the negative classes are also over 95%.

2. Introduction

This problem falls under the broad domain of moderating users in online chat rooms/networks. This is an active area of development and there is an ever increasing need for better solutions. The particular problem of focus here is validating user spam reports and detecting malicious users in the context of 'Chatous', a specific online chat network. The dataset was provided by the company. The current state of the banning system in Chatous involves administrators having to manually view the conversations corresponding to the reports and then decide whether or not to ban a given user. This is rather cumbersome as the number of reports keeps increasing over time. This necessitates a good machine learning algorithm that would automatically validate user generated reports as well as predict probable spammers. We describe our analysis using Naive Bayes as well as SVMs. The code was written in Python and most of the standard algorithms used were from the *sklearn* package **[1]**. Section 3 describes the framework for the system including details about the dataset and definition of positive and negative classes. Section 4 motivates feature selection and briefly describes the Chi-2 statistic. Section 5 describes our implementation and analyses the accuracy of Naive Bayes and SVM. We conclude in Section 6 and mention some future directions in Section 7.

3. Framework

3.1 Dataset

There are two data sets - the profiles dataset that contains the age, sex, location, screen name and “About Me” section for each user and the conversations data set that consists of over 10 million *conversation* logs. A *conversation* is a persistent entity that corresponds to the entire history of communication between a pair of user profiles. There are over 330,000 registered user profiles. This *conversation* log includes the time when they first talked to each other, time when they last talked to each other, whether the conversation is still alive and if not, who deactivated it, whether either of the two was reported by the other and finally the bag of all words used in the conversation.

3.1.1 Preliminary approach

Our preliminary approach was to assume that the “reported” field in conversation logs is accurate. Conversations where someone was reported are labeled +1 and -1 otherwise. This is a naïve way to define class labels and unsurprisingly, it resulted in a prediction accuracy of around 5% with Naive Bayes, thus validating the premise of our project that most reports are spurious.

3.2 Class definition

Since we did not have official data that tagged users as clean and malicious, we had to select a reasonable way to define our class labels. The training set consisted of two types of conversation logs chosen as follows. For each user profile we keep a count of reports against him. The top ***K*** profiles with the most reports against them are assumed to belong to *malicious* users. Profiles that have never been reported are assumed to belong to *legitimate* users. A conversation where at least one participant is *malicious* is labeled +1 and a conversation where both participants are legitimate is labeled as -1(or 0).

4. Feature Selection

It is not surprising that the number of words in the dictionary is close to 1 million. In chat networks, one can expect to have many typographical errors. The motivation for feature selection for words is primarily driven by the following reasons:

i) Commonly occurring words in the English dictionary were part of the dictionary in our dataset.

ii) Too many features usually results in over fitting. This means that the training error will be small whereas the testing/generalization error will be significantly larger.

iii) Irrelevant features may negatively impact the performance of the learning algorithm.

One can also imagine “pruning” the features to remove commonly occurring words prior to feature selection. A brief literature survey suggested that the Chi-2 statistic is a good approach for feature selection for text classification problems. Fortunately, this was also implemented in the sklearn package. We now briefly describe the Chi-2 feature selection algorithm. We assume that the readers are aware of the Chi-2 probability distribution.

Chi-2 is a statistical feature selection approach that captures the divergence from the expected distribution, if one assumes that the occurrence of a feature is independent of the class value **[2]**. In effect, it measures the lack of independence between a class label 'c and a term 't'.

The $\chi 2$ statistic (CHI-2) as defined by Yang and Pedersen in **[3]** is given by the following expression

$\chi 2(t, c) = [N \times (AD - CB)^2] / [(A + C) \times (B + D) \times (A + B) \times (C + D)]$

where N is the number of documents, A is the number of documents of class c containing the term t, B is the number of documents of other class (not c) containing t, C is the number of documents of class c not containing the term t and D is the number of documents of other class not containing t. This is comparable to a Chi-2 probability distribution with one degree of freedom.

Chi-2 was performant enough for bringing down the number of features (words in this case). Hence, we did not experiment with different feature selection algorithms. One experiment we however did regarding this was to vary the number of chosen features and measure its impact on the overall accuracy of the learning algorithm. Figure 4 in Section 5.2.1 plots the results of this experiment..

Additionally, we added a few extra features that significantly aid the learning algorithm. These include, the presence of a report in a conversation, the number of reports against a user, the number of disconnects against a user, age, sex, the 'about me' section. Some of these features like the number of reports/disconnected against a user had to be made binary values because absolute count values were large and the algorithm (SVM) did not converge. We do this by considering a relative threshold. For example, if a user has been reported in 65% of all the conversations he was involved in, the feature value would be + 1 and -1 otherwise. There was other information that we expected to be useful but actually degraded the accuracy of prediction. A few of these are, the location of both the users in a given conversation – whether or not both users were from the same region, whether or not the conversation ended in a friendship and whether or not the conversation is ongoing. In Chatous, there is exactly one conversation thread between a pair of users.

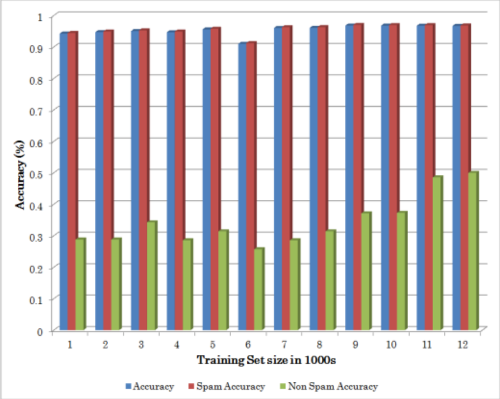
5. Implementation and Analysis

We started out with the basic evaluation using Naïve Bayes so as to get a general idea of classification and then proceeded onto different SVM kernels for classification.

5.1. Naïve Bayes

We implemented Naïve Bayes using multinomial distribution by using all features (which include word vectors of around 950,000, sex of the user, user location, duration of the conversation and a few other features). This resulted in a very bad recall percentage for spam - around 35 % on an average as shown in Fig 1. When we performed feature selection using

Chi-2 and included relevant features as mentioned in Section 4 (we included 1000 features for the run in Fig 2), we observed a significant drop in the spam recall accuracy as shown in Fig 2. This is due to the fact that Naïve Bayes assumes independence of features and we observed after pre-processing of the data that the features are in fact not independent and therefore, with feature selection, we lose some features leading to a loss in the recall percentage for spam. The x-axes in both graphs below represents the training set size and the y axis represents accuracy.



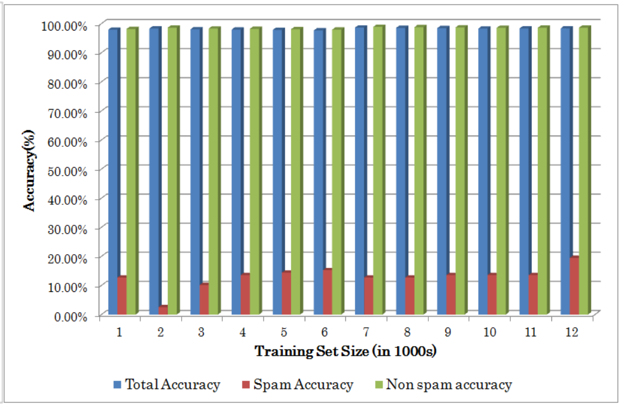


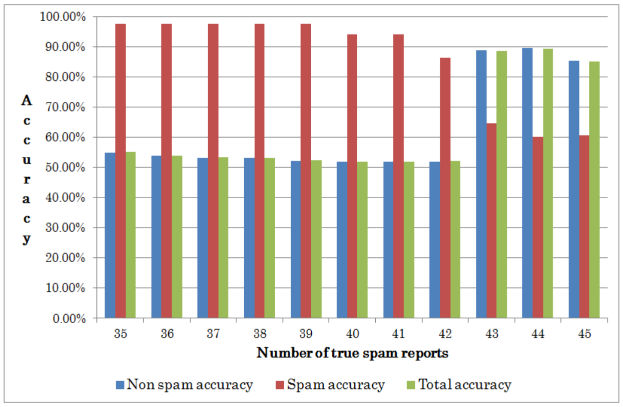
Fig 1.Naïve Bayes without any feature selection

Fig 2.Naïve Bayes after feature selection using Chi-square

5.2 SVM

From feature selection, we observed that we did not have independence of features and therefore, mapping the given set of data points to a higher dimension would definitely help. However, we needed to choose the right features and the right kernel for good results. We explain two kernels that gave us good results, one is the Gaussian kernel and the other is a polynomial kernel with degree 3. All the experiments involving SVMs were regularized. The reason is that even after feature selection, we are still susceptible to over fitting. Regularization of SVMs would smoothen the final learning curve. This was particularly the case when we experimented with polynomial kernels with high degrees.

5.2.1 Gaussian Kernel



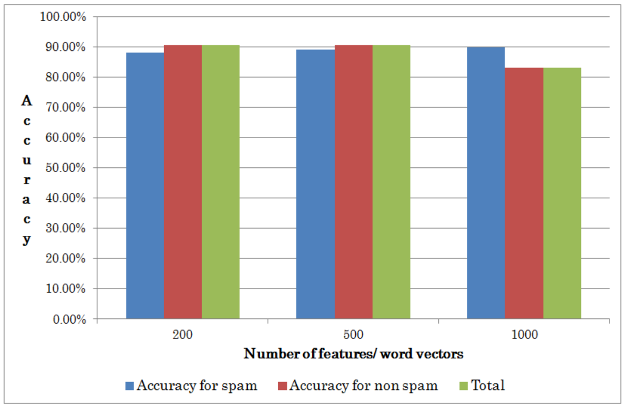


Fig 3. Plot showing accuracy vs. number of true spam reports using a Gaussian kernel

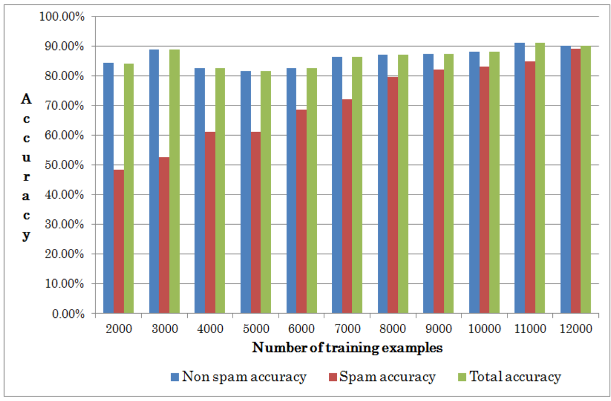
Fig 4. Plot showing accuracy vs. number of features using Chi-2 feature selection and Gaussian kernel

Fig.3 is a plot of accuracy vs. number of true spam reports that were considered during training. We observe that using a Gaussian kernel, our accuracy of prediction is not satisfactory. We observe that our recall for spam is around 98 % for values of 35 to 42 (taking 35 to 42 top reported users as the true spam reports**).** However, our non-spam accuracy is pretty low at around 55 %. Due to the fact that our data set is mainly biased towards non-spam reports (which include no reports too), we have a really low total accuracy. This showed that mapping to an infinite dimensional space is not helping us much.

One other heuristic that we considered was aimed at removing words occurring frequently in both classes of conversations. The threshold for the calculation was varied and we observed that the accuracy improved but was still performing lower than feature selection using Chi-2.

5.2.2 Polynomial kernel

We now consider a polynomial kernel of degree 3 for the support vector machine. As before, we use the Chi-2 test for feature selection. Fig. 4 shows our variation of accuracy vs. number of features included using the Chi-2 test (with the additional features included again) and we observe that with 1000 features included from the Chi-2 test, we have a very good recall for spam - around 92 % and a total accuracy of about 98 %. Fig. 5 shows an increasing trend for accuracy as the training set size increases and we observe that at a training set size of 12000 examples, we have a significantly high accuracy of 98% and a recall of around 93 %. The goal is to find the minimal training set size that would result in a high prediction accuracy. Fig. 6 shows that as we increase the number of true spam reports for our training, the recall for spam slightly reduces and therefore, we use 36 as our optimal value for all experiments.



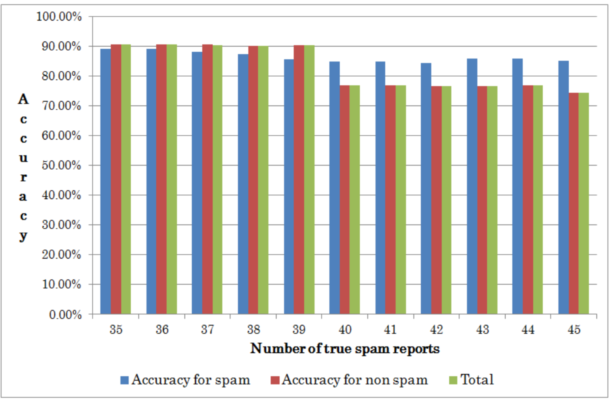


Fig 5. Accuracy vs. Training set size

Fig 6. Accuracy vs. number of true spam reports

Fig. 7 shows that as the testing set size increases from 10000 to 50000, we observe that our recall for spam is always high at around 94 %. Furthermore, our non-spam accuracy and the total accuracy are constantly high. This shows the predictions are accurate and are independent of the test size. This experiment was performed on a random test data set. We tested the same on a balanced data set where 50 % of the samples were labeled +1 and the other 50% labeled as -1. We observed similar results.

Confusion matrix (test size of 40000)	**Spam**	**Non – Spam**
Spam	109	9
Non – spam	745	39137

Table I: Confusion matrix for a polynomial kernel (1000 features after Chi-2 feature selection)

Fig. 8 shows the **F_1 score** (also **F-score** or **F-measure**) which is a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score: *p* is the number of correct results divided by the number of all returned results and *r* is the number of correct results divided by the number of results that should have been returned. The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and worst at 0. We observe that for the polynomial kernel, our F score is pretty high (around 95 to 96 %) whereas for a Gaussian kernel, we observe an F score of 70 % showing that our SVM using polynomial kernel performs well.

Table I shows the confusion matrix for a test size of 40000. Note that most of the spam users are caught with only 9 spammers being wrongly classified as non-spammers. This suggests that we do not miss out on many malicious users in the chat network. However, we do have a significant number of predictions of Non-spam as spam and would like to bring that down to a nominal number as a part of our future work.

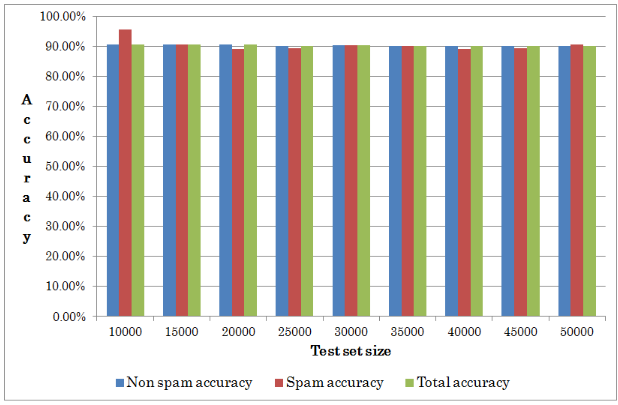


Fig 7. Accuracy vs. Testing set size

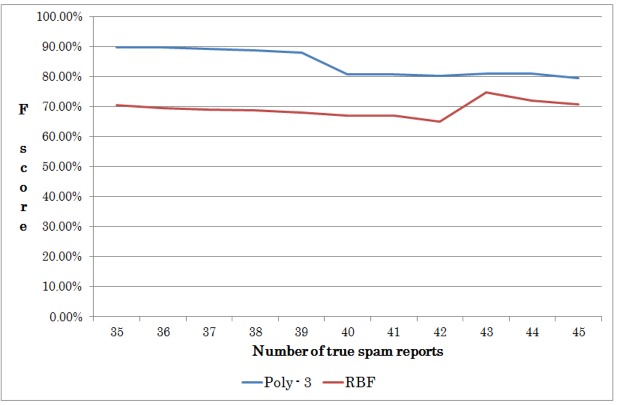


Fig 8. F-score vs. number of true spam reports

6. Conclusion

Anonymous chat networks are now an important form of anonymous conversations. However, there exist malicious users who cause disruptions in these networks. To validate user spam reports, manual inspection is not the feasible solution. Our work shows that feature selection is extremely important and in particular, including the right set of features in the learning algorithm is paramount to achieving good prediction accuracy. We used the Chi-square feature selection to reduce the number of features from more than 950,000 to 1000 features and manually add a few important features. We have shown that using an SVM with a polynomial kernel of degree 3 and with regularization, we can achieve a recall spam accuracy of 94 % and a non-spam accuracy of around 98%. These results are promising and we believe our system can be used effectively validate user spam reports and thereby ban users automatically without manual inspection.

7. Future work

We recently received a gold dataset tagging users as malicious or clean. We can use that data set to improve on the implementation described in this paper. One interesting extension would be to build an online learning system that proactively predicts users as malicious, which will help in immediately banning malicious users in chat networks. One of the features we might consider here is the number of reports received by each user over a period of time. This can be used to assign weights/goodness scores to users which will then be used in the online algorithm. Finally, we will push for the adoption of our system in Chatous. That will be the ultimate real time test of our learning system.

8. Acknowledgements

We would like to thank Kevin Guo, co-founder of Chatous for guiding us through the project and providing us with the data set without which none of this would have been possible.

9. References

[1] http://www.scikit-learn.org/

[2] Z. Zheng, X. Wu, R. Srihari, Feature Selection for Text Categorization on Imbalanced Data, SIGKDD Explorations, 2002, pp. 80-89

[3] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In Proc. the Fourteenth International Conference on Machine Learning (ICML-97), pages 412–420, 1997.