A Statistical Approach to Filtering My Inbox

Bryant McArthur
BYU
Ling 580R
Dr. Earl Brown and Dr. Joey Stanley

In this paper, I present a statistical approach to filtering my inbox while also evaluating the effect of the subject's part of speech on my behavior.

1 Introduction

This project started off with my desire to create a good email spam classification filter. Gmail's spam classification model works well on a general setting. However, I would like something more customized to my needs. I want to include in my model all of my emails that are not necessarily "spam" but that I never bother to open like social media notifications.

For the sake of this paper I will frequently refer to these unopened emails as "spam" and my opened emails as "ham", even though that's not exactly the case because I receive several emails that I want to continue receiving even though I never open them, like receipts.

I then had the question of whether part of speech in each subject line will correlate to my spam or not spam ("ham") emails.

Several people have built their own spam detection algorithms, but I found only one paper that claims to use part of speech tagging to build a k-means algorithm for email spam detection. (Parsaei) However, in the kmeans algorithm it appears that it also uses a vectorized dictionary of word labels as well. There is no comparison to a model that uses only words or only part of speech and it does not present the effect of part of speech tagging on the predicted classification. It is still very possible that his part of speech tags were completely insignificant and irrelevant in determining whether it was spam.

This is why I will evaluate whether there is evidence to reject the null hypothesis that part of speech usage does not differ in the subjects of emails that I read and emails that I leave unopened.

Finally, I will attempt to build a classification model that can accurately predict my behavior whether or not I will open an email to read it. I will use part of speech tagging if it is a significant predictor variable. To do this I will test different classification models, parameters, and features to see which combination will perform the best.

2 Data and Methods

The data I used to evaluate Part of Speech correlation and train and test my model was my own emails from the past 2 years.

I used takeout.google.com to download a .mbox file of all my "Opened", and "Unread" emails. I ended with 17,391 emails; 4,478 were in my "Opened" inbox and 12,913 were in my "Unread" inbox.

Second, I converted all my emails to .txt files while simultaneously getting rid of attachments, and images etc.

Third, I used regex expressions to extract my main features: Sender, Subject, Reply-To, and Content of the email.

Next, I used spacy to tag the part of speech of the subject and content words and counted the occurrences and frequency for each email.

Finally, I created my data frame. My Dependent (Response) Variable was Opened or Unread emails labeled either as Spam or 0 for Unread and ham or 1 as Opened.

My Independent (Predictor) Variables were: From, Subject, Reply-To, Content (Just the first couple lines that I would see), Part of Speech count and frequency for Subject and Content separately. The parts of speech that I included were, adjectives, adverbs, nouns, proper nouns, numbers, symbols, verbs, and interjections.

I created bar plots of the parts of speech count for both ham and spam emails next to each other. I then created a single bar plot with geom_bar(position=position_fill()) in order to see the difference in proportion of each part of speech for ham and spam emails.

In order to figure out if the part of speech had a significant correlation to my behavior of reading an email or leaving it unopened, I had to run a chi-squared test because I had several categorical variables. A p-value of less than .05 would be small enough to reject the null hypothesis that there is no difference in distribution between the several parts of speech.

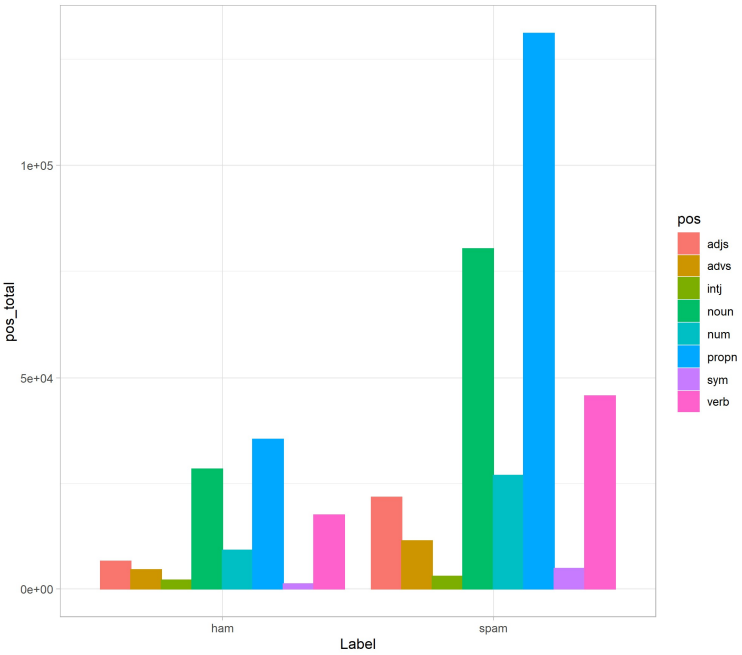
If the part of speech distributions were significantly different then they would inherently be a significant predictor in my behavior if I opened the emails or left them unread and I would therefore add them to my model. Otherwise, I'd build a spam classification filter traditionally without the part of speech.

I decided to build two different classification models to compare their accuracy in predicting my behavior. I compared the difference between discriminative and generative models for this task.

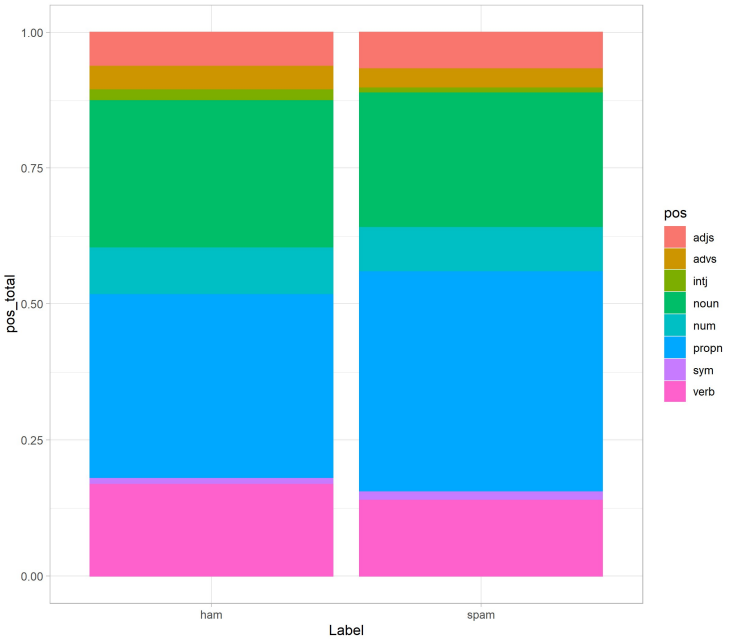
My discriminative model of choice was a random forest because they are fast and robust, and tend to not overfit nearly as much as a simple logistic or softmax regression especially with many more features than data points. The generative model I used was a Naïve Bayes model. A naïve bayes model trains much faster than any QDA model by making the "naïve" assumption that the data is independent of each other and creating a diagonal covariance matrix.

3 Results

I first compared the total number of each part of speech for ham and spam emails as seen below. Just by looking at this graph it would be easy to assume that there is a huge difference for almost every part of speech between the two labels.



However, this graph is misleading because there were way more spam emails than ham from the start. We need a graph that will more accurately compare the proportion of each part of speech for the two groups.



In this second figure we can see that for the most part a lot of the parts of speech have a similar ratio between spam and ham emails. However, a couple of the parts of speech appear to differ. For example, it appears that there are more proper nouns (blue) in spam emails on the right than ham emails, and there is a higher proportion of verbs (pink) in ham emails on the left.

In order to see if these differences are actually significant, I had to run a chi-squared test. The test gave me a **p-value of 0.382**. Therefore, we cannot conclude the distributions of the parts of speech are different enough for ham and spam emails to be significant.

Since there was no significant difference in the part of speech distributions for the emails I opened and left unread, I did not include that information in my classification models.

I first built and tested my discriminative random forest model. Using SciKit Learn's method I performed a grid search to build several random forest models in order to find the parameters that would give me the best accuracy score. My resulting RF that had the highest accuracy contained 250 trees, with a maximum depth of 5 (5 feature splits), and a minimum training instances per leaf of 3. The accuracy was **69.9%** from a test train split evaluation.

Next, I used those optimal parameters in order to perform another grid search in testing different parameters for vectorization. These parameters included different text normalization techniques, stop words dictionaries, ngram ranges and limiting the dictionary size. I found with the vectorization including up to trigrams and limiting the dictionary to 1000 words I was able to increase my random forest's accuracy up to **74.7%**, an almost 5% increase.

Next, I built and tested my Naïve Bayes generative model. I wrote my own class algorithm for my Naïve Bayes classifier because SciKit Learn's NB function couldn't do everything I wanted it to.

Right off the bat, my Naïve Bayes algorithm performed at **77.8%** accuracy with a test train split. I went ahead and used my algorithm to test the accuracy if I included the part of speech tagging as predictor variables. The accuracy dropped to **73.1%** almost a 5% drop. This confirmed my decision based off the chi-squared test to not include that feature in the model. Because the variables were insignificant my model was overfit on the parts of speech from the training data which caused to poorly perform on the test set.

Just as I did with my RF I did a grid search for my NB model to test the optimal parameters for vectorization. As opposed to limiting the

dictionary size like the discriminative model preferred, this model performed much better with an unlimited dictionary size, a matrix that included weights for every word that showed up. My vectorization didn't include any new parameters that I wasn't doing before except the possibility to include up to trigrams, however, my accuracy remained the same at **77.8%.**

4 Discussion and Conclusions

What results mean with respect to research question, further questions and research, any reflections on data collection, analysis, and interpretation processes.

My research question had two parts to it. First, I wanted to find the correlation between the parts of speech of the words used in the subject of an email and if I opened it or not. Second, I wanted to build two different models to compare the performance of generative vs. discriminative models for an email classification task.

I found overall that although certain parts of speech seem to appear more frequently in certain categories, the difference was not significant enough to suggest a correlation. This was confirmed when the accuracy of my model went down when I used the part of speech in training and testing as opposed to excluding it altogether.

I also found that my generative model of choice (Naïve Bayes) performs much better with a higher accuracy at classification than my discriminative (Random Forest) model with scores of 77.8% and 74.7% respectively. That difference may seem insignificant but with 17,000 emails, that's over 500 more emails that are classified correctly by Naïve Bayes.

One reason I think the Naïve Bayes classifier worked better is because it uses an assumption that the emails are normally distributed and is able to calculate a joint distribution between the features and the labels that holds much more information than just the single distribution of the conditional probability.

A discriminative model solely uses the information given in the training data to fit its model which gives a probability distribution from $P(y|x,\theta)$ where y is the label (spam or ham), x is the vector of all the features or predictor variables (weighted dictionary of words), and θ is the coefficients the model must decide. However, the joint distribution which the generative model uses is $P(x,y|\theta)$. Basic rules of probability give us:

$$P(x,y|\theta) = P(x|y,\theta)P(y|\theta)$$
$$= P(y|x,\theta)P(x|\theta)$$

We can see that the joint distribution includes the information of the discriminative model (highlighted above) plus so much more that the discriminative model can't find on its own without the assumptions on the prior distribution of the data.

5 Follow-Up Work and Questions

There are still more questions that can be asked from this research, and more questions that still remain unanswered.

First, the chi-squared test I performed was dependent on my intuition of which parts of speech I thought could be correlated with the spam and ham emails. However, I could lower the degrees of freedom of the chi-squared test by only testing a couple of the parts of speech that appeared to be most proportionally different from my graph above. By lowering the degrees of freedom and only focusing on those parts of speech that are most different my chi-squared test would be able to tell me more

accurately if those specific parts of speech have a significant effect even if the other ones don't.

I suspect that this is the case with maybe 2 or 3 of the parts of speech, and if so, I would like to see how a model trained on just the number of these 2 parts of speech would perform.

Second, my grid search in finding the optimal parameters for my random forest were limited in my computational power. Particularly, I tested the maximum depth of the trees for [1,2,3,4,5]. My optimal parameter I got out was 5. With every additional split added to a single tree it takes much longer to train the entire forest, which is why I limited the max depth to 5. However, I believe that I could continue to increase the maximum depth at a computational cost and achieve a higher accuracy. Traditionally trees are used when the number of features is low, however in this case when the number of features is very high, we may still be able to allow for more splits without overfitting.

If I had more time to train deeper trees, I suspect that a Random forest may be able to outperform the Naïve Bayes model but it would take much longer and cost a lot more.

In this case, the reason the Random Forest may be able to outperform the NB model is because the RF is an ensemble really comprised of 250 different discriminative models as opposed to a single generative model. An ensemble of several weaker models can outperform a single strong model. It would be interesting to test the tradeoff.

Overall, it is obvious that most parts of speech had no correlation with my behavior, and that generative models do perform better for email classification tasks than discriminative models. It still may be possible that a select few parts of speech may be correlated with my opened or unopened emails and could improve the accuracy of a model. It is also possible that an ensemble of several discriminative models, a Random Forest, may be able to outperform a single generative Naïve Bayes model, but it would come at a heavy cost.

References

M. R. Parsaei and M. Salehi, "E-mail spam detection based on part of speech tagging," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015, pp. 1010-1013, doi: 10.1109/KBEI.2015.7436182.

Appendix

````
### Plots
```{r}
# Plots
plot = data %>%
  ggplot(aes(Label, y = pos_total, fill = pos, color = pos)) +
  geom_col(position=position_dodge()) +
  theme_light()
plot
ggsave("visualization_dodge.png", height = 7, width = 8, dpi = 300)
plot = data %>%
  ggplot(aes(Label, y = pos_total, fill = pos, color = pos)) +
  geom_col(position=position_fill()) +
  theme_light()
plot
ggsave("visualization_fill.png", height = 7, width = 8, dpi = 300)
```
````

````
## Chi-squared test

```{r}
df_tbl = data %>%
  xtabs(~Label + pos_total, data = .) %>%
  print()
```

      pos_total
Label  1168 2075 2984 4519 4843 6557 9127 11334 17675 21814 26905 28415 35335 45529 80465 130934
  ham     1    1    0    1    0    1    1     0     1     0     0     1     1     0     0      0
  spam    0    0    1    0    1    0    0     1     0     1     1     0     0     1     1      1

```{r}
rstatix::chisq_test(df_tbl)
```
````

A tibble: 1 × 6

n <int>	statistic <dbl>	p <dbl>	df <int>	method <chr>	p.signif <chr>
16	16	0.382	15	Chi-square test	ns

1 row

Words: 2315 (Not including the Appendix)