Predicting U.S. president election result based on Google Insights

Yunzhu Chen, Fan Zhang, Yuanxi Yue

December 14, 2012

Abstract

In this report, we used Google Insights data to predict the U.S. president election results by states. To be specific, we selected keywords closely related to the president candidates policies, and used the searching frequencies in each states during the past 12 months as our data.

We applied supervised learning algorithm including Gaussian Discriminant Analysis, Support Vector Machine and Naive Bayes method, and unsupervised learning algorithm Principal Component Analysis. We also explained the results and discussed some interested findings which are useful for future study.

1 Introduction

It is commonly known that when people are interested and concerned about something, they are likely to search on the Internet. And there are already some studies using web search volume to make predictions. Google Insights (also known as Google Trends) is an application in Google to collect search history data and related location information.

We are interested in researching on the web searching data and exploring the features that could predict the US presidential election results by states.

This study could be useful because it provides a reliable tool to make prediction on the vote result, besides survey the voters. It can potentially facilitate the electoral team and the media to understand different concerns from different states, and the potential votes result. Also, this study could even help electoral team to strategize further campaign to their potential voters.

2 Data and Preprocessing

The data of keywords searching results are all collected from Google Insights. And the data are the normalized searching volume on Google by states with in 12 months (between Nov 2011 to Nov 2012). And we researched on the 2012 US presidential election result.

Specifically, we chose the keywords by selecting some most controversial topics between the two president candidates. The keywords cover area including environment, military, foreign policies and etc.

we denote for the i-th states, their search vector is $\mathbf{x_i} = (x_i^{(1)}, ...x_i^{(j)}, ..., x_i^{(m)}),\ i = 1, ..., n$. The j-th element is the searching volume for the j-th keyword in this state. m is the number of keywords we use. It also means that the dimension of the data is m. n is the number of cases, and in this particular problem, we have $n = 51$ as the number of states. Because the data are normalized by states, so $x_i^{(j)} \in [0, 100]$, where $x_i^{(j)} = 100$ means the i-th states searched the j-th keyword most frequently among other states.

And the labels for the states are the final voting result. We denote it as $\mathbf{y} = (y_1, ..., y_n)$. Because conventionally there are two president candidates, y_i only takes two values ($(1, 0)$ or $(1, -1)$ depends on particular method).

In order to select the features (keywords) that are most relevant to the election result, we calculate the mutual information between the keywords and voting result. From all the keywords, we finally decided to use $m = 63$ of them into our model.

3 Methodologies

Our goal is to find good algorithm that could reliably predict the election voting results. To achieve this, we try four different models, including Gaussian Discriminant Analysis, Logistic Regression, Naive Bayes and Support Vector Machine.

This problem is a supervised classification problem with all the keywords as features and states as cases.

4 Model Evaluation

4.1 10-fold cross-validation error rate

We randomly divide the states into 10 samples and conducted the 10-fold cross-validation. The error rate for the 4 classification models resulted from the 10-fold cross-validation is shown below.

Model	GDA	Logistic	Naive Bayes	SVM
Error Rate	0.26	0.28	0.26	0.18

Table 1: 10-Fold Cross-Validation Error Rates

Among the 4 models, SVM generated significantly smaller error rate comparing to the other 3 models. Logistic regression had the largest error rate among the models but the error rate is similar to that of GDA and Naive Bayes.

4.2 ROC Curve

We plotted the ROC Curve with 1/2 of the samples as the training set and 1/2 of the samples as the test set. We choose 1/2 for training and 1/2 for test because our sample size is relatively small and we would like to get smoother curves; thus we decided to include more data into the test set. The plot is shown below.

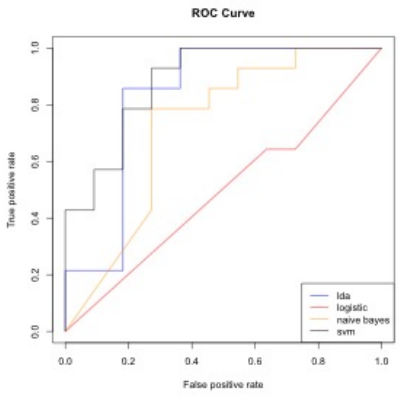

Figure 1: ROC Curve for the four model

Logistic regression showed little prediction power as its ROC curve lies almost on the 45-degree line. SVM and GDA showed relatively strong prediction power, while Naive Bayes had moderate performance.

4.3 Plots of training errors and test errors

Since the logistic regression has been recognized as the worst performed among the 4 models from the previous 2 comparisons, we further compare the 3 other models. We used 25 to 45 samples for the training set with the remaining as the test set. In this process, we got the training error and test error for different

training and test sample size, for GDA, Naive Bayes and SVM respectively. The plots of training errors and test errors are shown below.

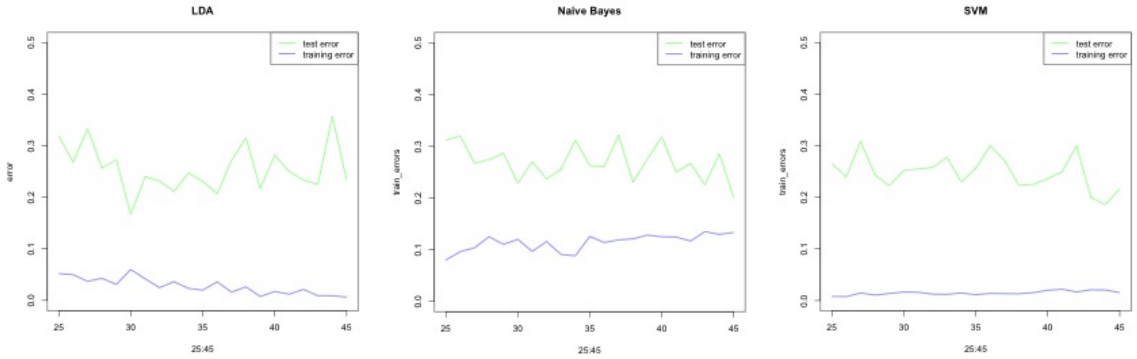

Figure 2: Training errors and test errors for the four models

Naive Bayes showed both high training error and test error with the tendency to converge. This indicates the Naive Bayes model has high bias or is under-fitting the data.

GDA showed low training error and high test error. The large gap between training error and test error indicates the GDA model has high variance or is over-fitting the data. In the plot, GDA even displays a slightly increasing trend, indicating GDA doesnt fit this case very well, possibly because the cases dont conform to a normal assumption.

SVM showed both low training error and test error. Although it shows high variance, but since our case number is limited, it could not be fixed for now. The error is acceptable, and it is the best among the 3 models. Therefore, we will choose it as a final model as the election voting prediction model.

5 Results with Keywords

We then conducted Principal Component Analysis to the data, to explore how the web searched keywords related to the result of voting.

In order to apply PCA, we need to reduce the number of keywords to less than 51 first, to satisfy the condition of using PCA. Once again, we calculate the mutual information between each of the keywords and voting result, and selected 21 most relevant keywords.

With the screeplot for PCA (Figure 3), we can tell that the first two components could explain most of the difference between voting. Therefore, we research on these two components, and the corresponding first two PCA scores for each state.

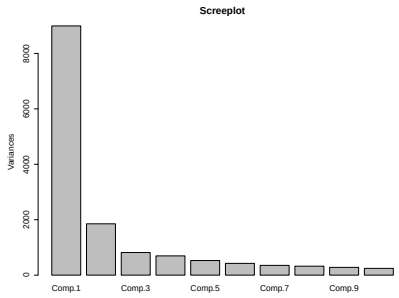

Figure 3: Screeplot for PCA

From the plot of first PCA scores (Figure 4 Left), we can see that there are two obvious clusters of states. And we could find the left cluster are more likely to vote for Obama. If without knowing the voting results, we simply classify the states to be Obama's and Romney's with the two cluster, the error

rate is around 25% We can also a rule for the left cluster, that with similar second score, a less first score for a state means that the state is more in favor for Obama; and with similar first score, a higher second score means favor for Romney. For the right cluster, there is not a very obvious structure.

In order to get a better understanding, we look at biplot (Figure 4 Right). It shows large negative contribution for each words in the first score. This well explained the cluster in scores, that the states in the left cluster are those web-search activate states. Also it shows some bias in our initial data selection.

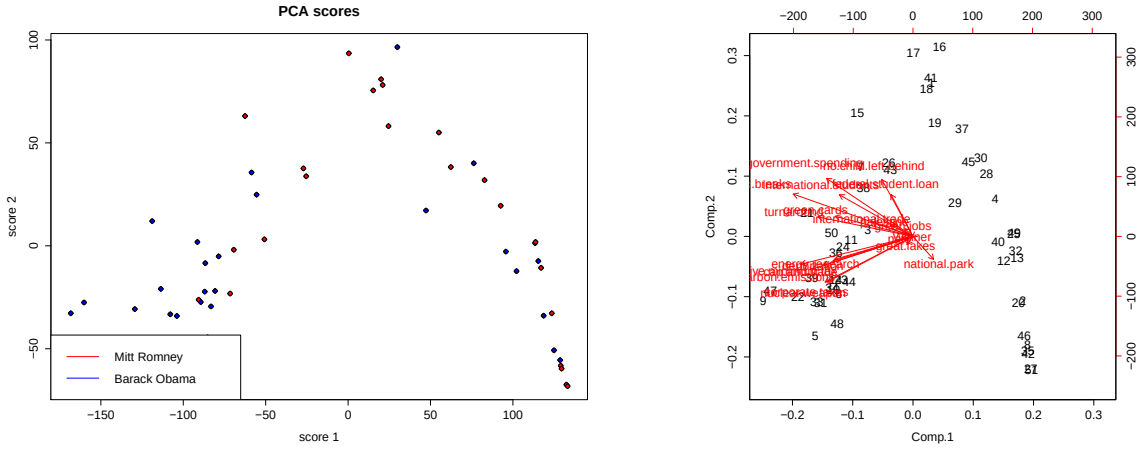

Figure 4: Left: First two PCA scores for each state. Right: Biplot - How does each keyword form the first two components and effects the first two PCA scores

The table of loadings (Table 2) shows how the PCA scores is calculated from the keywords. It listed the loadings of the top 5 keywords that contributed. With the plot and the ranks, we can tell a lot stories. People from different states have different concerns. Those who concerns a lot on the keywords with large negative loadings on component 1 are more likely to vote for Obama. Those who concerned about keywords with positive loading of Component 2 are more likely to be the supporter of Romney.

Component 1	Loadings	Component 2	Loadings
tax.breaks	-0.370	government.spending	0.421
alternative.minimum.tax	-0.347	no.child.left.behind	0.375
carbon.emissions	-0.311	nuclear.weapon	-0.330
cap.and.trade	-0.279	corporate.taxes	-0.319
nuclear.weapon	-0.273	tax.breaks	0.314

Table 2: Top keywords with largest absolute loadings for the first two components

6 Discussions

In this project, we tried to use different classification models to learn about the relationship between Internet searching volume for political keywords and the 2012 U.S. president election voting result in different states. We conclude that, among all the models, support vector machine with radical kernel is most effective.

However, we still have an error rate larger than 10%. It may on one hand due to the limited number of observations (i.e. the number of states), on the other hand due to the limited intrinsic explanation power of searching volume to election results. However, it is still possible that the prediction power could be enhanced if we select a more comprehensive set of keywords.

Even though the prediction is not extremely precise, it could serve as a complement approach for polling. At the same time, it also provides useful information for understanding some of the heterogeneity across states.

Furthermore, the results with Keywords can be instructive for further campaigns. For example, the political strategists in the campaign operation teams can apply the results generated by PCA to see what are the major concern of the voters.

One shortcoming is that the problem is an unsupervised problem when none of the states have released their election results. To tackle this shortcoming, the PCA part could serve as an unsupervised method to classification, which also has a well-controlled error rate. Moreover, the supervised algorithms could also be applied, without relying on released election results, by taking the safe states as the training set, since we could at most times safely assume the supported candidate in those states would win the states vote on the election day. Then, we could use the trained model to make prediction on the swing states, which are what people most likely to concern about.

References

[1] *Google Trends*, `http://www.google.com/trends/`.