Predicting the Credibility of News Articles
MS&E 226 Fall 2017

Madison Coots
Management Science
and Engineering
Stanford University
mcoots@stanford.edu

Gabriel Rodrgiguez
Management Science
and Engineering
Stanford University
grod@stanford.edu

Abstract

We propose an regularized logistic regression classifer for the purpose of classifying news articles as either credible or incredible based on their textual contents. Our classifier achieves classification accuracy of 0.90.

1. Task Definition

Our problem is one of constructing a binary classifier that outputs a label: credible or incredible, for a given natural language body of text. One of the considerations to be kept in mind in construction of the model is to employ a form of model regularization that will reduce the dimensionality of the data corpus so as to reduce the possibility of overfitting as well as to improve computational efficiency.

2. Data

This data set contains in total 6,335 articles that are tagged as either real or fake. The data was obtained from a repository on Github. The individual who originally amassed this collection of articles did so by scraping various online news outlets for articles deemed as fake or real. Each row in the dataset represents one article and contains the articles title, text, and binary tag of REAL or FAKE. One concern about the composition of the dataset is that it is unclear how the sources of the various articles were selected. That is to say, it is unknown how the collector compiled a list of trusted news sources as well as fallacious, untrusted sources in order to confidently be able to level each article as real or fake. Additionally, the distribution of the article topics is unknown. This could be problematic if, as an extreme example, all of the real articles were on politics and all of the fake articles were on travel. In that case, the prediction of real or fake might be more dependent on the topic of the article rather than its credibilty.

2.1. Data Processing

Before attempting to build models, it was necessary to vectorize the articles through the construction of a document term matrix based on the training data (5068 articles of the original 6335). In a DTM, entry A_{ij} corresponds to the number of times word j appears in article i. We also transform the data so that entry A_{ij} of the DTM is 0 if word j appeared 0 times in article i, and 1 is word j appeared one time or greater. Performing this transformation prevents regression coefficients from obtaining extremely large magnitudes in the event of separation in the data.

Pruning words present in all or most articles, such as the, and, but, as well as words that only appeared in less than 0.05% of the articles or less than 200 times reduced the dimension of the matrix to 5068 x 1262. These filtration criteria were intentionally non-restrictive so as to allow for later covariate selection through a Lasso regression. This way, rather than eliminating a large proportion of the words merely based on their frequency, Lasso could select the words (covariates) that had the best predictive value. Articles with an NA/NULL title, text body, or label will be removed from the dataset since the richness of this dataset is highly contingent upon the values of these covariates. Initially, the dataset contained an untitled column with some integer value–it is unclear what this column was intended to represent. Given that the values do not exceed a value of 6335, it is likely that these numbers serve as some sort of article identification number. However, this adds little to no value to the information contained in the dataset, therefore it was removed.

3. Methods

3.1. Covariate Selection

From the set of filtered covariates, we ran a Lasso regression over a set of twenty different values of lambda ranging from 0.001 to 0.02 by increments of 0.001. We used

cross-validation for glmnet (cv.glm) to return that resultant lambda that yielded the lowest mean cross-validated error for cross-validation with ten folds. The minimizing lambda was 0.002. We then ran Lasso again on our DTM with this value of lambda to extract the 521 covariates that Lasso selected. We then filtered out all other unselected covariates from the DTM to produce a matrix with reduced dimensions 5068 x 521.

3.2. Logistic Regression

We trained a logistic regression model with the objective of minimizing the 0-1 loss. Regarding the covariates used to generate the model, we used the set of 521 covariates selected by the Lasso regression. When we used our model to make classification predictions on the training data, it obtained a classification accuracy of 0.9298. Model performance values are tabulated at the end of this section.

3.3. K-Nearest-Neighbors

We also trained another classification model using the K-Nearest Neighbors algorithm with $k = 5$, and $k = 10$. The objective for this classification model was again to minimize 0-1 loss (maximize accuracy). For both values of k, the models yielded similar values of prediction accuracy on the training set. Summaries of the model predictions are tabulated below along with the prediction accuracies on the training set.

Interestingly, the prediction accuracy on the training set for both models is very similar. However, this is probably more coincidental than statistically significant. That is to say, if we were to perform this same analysis on different training subsets of the data, we would expect to see, on average, accuracies that differ more due to differences in the bias and variance of both models.

Model	Accuracy
Logistic Regression	0.9298
KNN, K = 5	0.796
KNN, K = 10	0.797

Table 1: Prediction accuracy on training set

4. Model Selection and Evaluation

From the outset of the generation of our various predictive models, we established that our primary goal was to maximize the prediction accuracy of our models. We decided that interpretability was not a priority because our models were generated using a substantial number of covariates. Therefore, we will select a model based solely on its performance.

As an estimate of the model of prediction error on unseen data we performed cross validation on the training data for each model. The resultant errors are tabulated below.

Again, the logistic regression model emerges as the model that maximized prediction accuracy (minimized error). Therefore, moving forward this will be our selected model.

Model	5-Fold Cross Validation Error
Logistic Regression	0.1095107
KNN, K = 5	0.224
KNN, K = 10	0.234

Table 2: Estimate of out-of-sample prediction error

5. Test Prediction Error

CV Estimate of Test Error	0.1095107
Test Error	0.1018153

Table 3: Prediction error on test set

When our logistic regression model is used to make predictions on the test data, the actual test error is slightly lower than the estimate for test error from cross-validation on the training set. This is surprising, but is likely due to the data in the test being particularly well-suited for the model by chance.

	True FAKE	True REAL
Predicted FAKE	546	71
Predicted REAL	58	592

Table 4: Logistic regression confusion matrix

Looking at the confusion matrix we see that the percentage of misclassified REAL and FAKE articles are about the same.

6. Inference

6.1. Analysis of the Statistical Significance of Model Coefficients

Since we have a large amount of significant covariates, we will be discussing only a small portion of these and speak generally on the implications of statistical significance. A few of the predictors with significant coefficients include: closed, remarks, specific, contributed, revealed, alternative, joe, please, nor, coverage, completely, via, expect, meet, project, and corporate. These predictors showed three star significance in the R output - the respective p values were less than 0.001. To put it in words, the probability that the coefficients are actually 0 is less than one in a thousand; we can reject the Null Hypothesis that states that the true value of the coefficients is 0. We believe the results for

a couple of reasons. Our predictions were fairly accurate (89%) and if the coefficients for the significant predictors were 0, were confident that the accuracy would decrease. Secondly, by analyzing some of the words we can derive that they do have an effect on the veracity or mendacity of an article. For example, from personal experience we see that the predictor via is normally used in FAKE articles so we expect it to have a negative effect on the veracity of an article. While we do believe that the coefficients are significant, we dont reject that the actual level of significance could be different. More robust and better models would take into account the interactions of various words which would have a great influence on the prediction capabilities of a model. If interactions and n-grams were taken into account, we would be able to see a better representation of the true value and significance of the coefficients.

6.2. Analysis of Significant Coefficients Generated by Fitting Model on Test Data

Originally, we used word counts per article as the values taken to build our model. When this was the case, we saw that all our coefficients were significant both on the model made from the train data and the model made from the test data. This form of the training data likely lead to separation of our data which made some words very indicative of an article being classified as REAL or FAKE. As mentioned earlier, this is also likely why our coefficients were so massive in magnitude.

After factorizing our training data and training our model on it, there were some significant coefficients with varying significance; unfortunately, when we created our model on our test data no coefficient was significant and all p values were virtually 1. We ruminated about the possible causes of this phenomenon and came to the conclusion that there could be a problem with either a lack of data or excess of predictors or problems from lack of model convergence, but we are still wary of this idea because our un-factorized models had fairly accurate predictions. According to the common statistical practice rule of ten, we would need at least 10 data points per each predictor - 5230 in our case. Having this many predictors could lead to overfitting and a misrepresentation of the effect of certain words.

6.3. Using Bootstrap to Estimate Confidence Intervals for Regression Coefficients

After using the bootstrap to estimate normal confidence intervals for each of our regression coefficients we see that there is a difference between that and the values given by R in the standard regression output for all intervals created. The confidence intervals for the standard regression output are much larger in range than that of the bootstrap. We believe that these results might differ because of the amount of data used to calculate the intervals; bootstrap generates new samples assuming that our data is the population so from the bootstrap perspective we have 5068 data points for 500 universes - replications of sampling with replacement - while the standard regression output is generated by our actual train data of 5068 data points. (We should also note that we chose to generate normal confidence intervals over quantiles because histograms of the bootstrapped coefficient distributions revealed that they were fairly normal.)

6.4. Comparison of Significant Coefficients Between Models Built on All Data and Lasso Subset

In light of the fact that we ended up two logistic regression models for prediction (one on the factorized training data, and one on the un-factorized data), we elected to discuss the changes we saw in the significance of the covariates between models trained on both versions of the training data.

Using the model fit on the un-factorized training data, yes, when we fit our logistic regression model on all of the available covariates rather than on just the subset selected by Lasso, the significant coefficients did change. In the model fit on the Lasso subset, all covariates were designated as highly significant (***). Interestingly, some of the coefficients that were designated as being highly significant in the model including only the Lasso subset were no longer as significant or marked as significant at all in the model using all available covariates.

One possible explanation for this is possible collinearity in the data. That is to say, perhaps two of the covariates (two words) were highly correlated like white and house. If this phrase has high predictive value, but Lasso zeros out one of these covariates, the remaining word is going to carry high statistical significance. However, in the presence of this words correlated counterpart, the model will not see this word as significant. Therefore, it is possible that there were several words in our dataset that were highly correlated with another word removed by Lasso, and therefore were reported as being less significant in the model with all of the covariates.

Using the model fit on the Lasso subset of the factorized data, we obtained more interpretable coefficients that had varying levels of coefficients. But oddly, when we fit the logistic regression model to the entire factorized training dataset, none of the resulting coefficients are designated as significant. This is especially puzzling given the results yielded from performing the same model fits on the un-factorized data. It would therefore be unlikely that we see these reductions in significance due to ubiquitous collinearity. It is also likewise unlikely that is due to a dimensionality problem since we obtained reasonable results from performing the same model fits on the un-factorized data. We suspect that these odd results most likely arise from either the logistic regression models being unable to converge, lack of

sufficient data. Though this did not prove problematic for prediction purposes, it seems to be an obstacle for fruitful statistical inference.

6.5. Commentary on Potential Issues

Given the bizarre results obtained from a comparison of our logistic regression model fit to various subsets of the data (Lasso training subset, all training, Lasso test subset), it is highly likely that there are some underlying issues with the model fitting process that are inhibiting our ability to attempt to make inferences about the data and significant relationships between covariates and or the outcome. As mentioned earlier in the section D, it is possible that there is collinearity in the data with words that are strongly correlated with one another. It is difficult to make a suggestion on how applying the Bonferroni correction would change our interpretation of significant coefficients given that we were impeded from making any substantive interpretations from the aforementioned anomalies in coefficient significance.

Furthermore, applying the Bonferroni correction makes the significance very conservative; our significance level (0.05) would be divided by 523, which would greatly decrease our significant coefficients. With regard to potential sources of bias in determination of significant coefficients, if we had actually been able to make reasonable inferences, one source could be that we ultimately selected a model trained on a subset of all of the data. With this selection, we could have excluded some potentially even more significant covariates because perhaps Lasso isnt sufficiently robust at selecting statistically significant covariates.

6.6. Commentary on Interpretations of Significant Relationships

If we had been able to identify significant relationships within our data, we would still have some qualms with interpreting them as causal relationships. For one, given that all of our covariates are words, it is unlikely that the presence or absence of certain words cause other words to appear or cause an article to be fake or real. These are more likely correlations. The very nature of this analysis is more correlative. That is to say, the goal of this endeavor was not so much to understand what makes an article fake or real, as it was be able to accurately classify real and fake articles based on correlative relationships between words, and article veracity.

7. Discussion

7.1. Practical Use of Models

In a practical setting, we would expect our models to be used to give a prediction for news articles from questionable sources consumed on the internet. For example, during the 2016 U.S. Presidential Election, there were numerous fallacious articles from dubious, unreliable news outlets that were widely circulated on social media sites such as Facebook and Twitter. Perhaps these social media sites could install a feature that would use this model to offer a prediction of whether or not a given article is likely to be real or fake. This way, the consumers could read the articles contents with a heightened degree of skepticism if it was predicted to be fake. Potential pitfalls of using the model to make these kind of judgment calls could arise if the model is consistently or rampantly misclassifying articles–this would potentially increase the spread of misinformation which is exactly what this model aims to curb.

7.2. Robustness of Models

If this model were to be used to predict the veracity of articles published on the internet, it would probably need to be trained once a week or a few times a week. Ideally, if there was a way of ascertaining after a given period of time the veracity of newly published articles, the model should be refitted as those new data points become available. As world events develop and as people begin to consume news with a more discerning eye, the nefarious actors who generate fake news articles will likely adapt their methods in writing these articles. The only way this model would be able to maintain any semblance of robustness in classifying new articles, it would need to be retrained regularly on new content.

7.3. Strategic Choices in Data Analysis of Interest to Future Model Users

There are several choices made in our data analysis that any user of the model would need to be aware of. With regards to data cleaning, we removed words that appeared less than 200 times total throughout all the articles because we felt that words that had few total appearances would be misrepresented. We also excluded words that appeared in more than 50% of documents in order to remove common words that normally appear in real or fake news. We removed word that appeared in less than 5% of the documents because the lack of data could lead them to show extreme false significance in determining if an article is real or fake. Given the fact that we used 523 predictors and we only had 5068 data points in our training set, it is possible that we over fit the model on the training data, but our results on the test data were still strong.

7.4. Desired Improvements of Data Collection Process

From the outset of this project, there was not much transparency with how the data was collected. It was unspecified where these articles came from, when they were scraped, or what the articles were about. This being said, it would

have likely been useful to have had more metadata about the articles in addition to the textual contents.

For example, perhaps certain media outlets produce more fallacious content than others. Perhaps fake news was more prevalent during certain time ranges than during others. Or perhaps it is more likely to have a fake news article about politics than about entertainment. Including covariates with this metadata would have likely improved classification is these variables had any true predictive value.

7.5. Proposed Changes in Strategy for Future Analyses

If we were attacking the same dataset again, we would take several different actions that could lead us to a better result. Firstly, we would add the interaction of terms to our logistic regression model. Due to the complexity and rigorousness of having interactions when there are over 500 covariates we were unable to pursue this method in our project. We would also use probabilistic language models, such as n-grams, to build a database of text sequencing to make our model more robust. We would also look into boosting, using splines, and using a subset approach on the logistic model. With regards to data manipulation and modification, we ran into some problems as we were wrapping up the project. All of the p values for the models made from the test data and the train data with all predictors were 0. We speculate that this was because we had too little data for the amount of predictors we had, so to fix this we would potentially try removing covariates to eliminate the computational exhaustiveness or partitioning the training and test sets differently.