Selecting a Proper Training Set

We simplify our problem to one of a binomial classification for each article: verified or unverified. Of course, this is far from the "truth" whatever it may be but makes it infinitely easier to train a classifier.

For our set of verified articles, we turned to the Alexa rankings. Alex categorizes websites into different categories including Computers, Society, Sports, and many others. We chose the top US-only news websites from each category for collecting our verified set. We had a total of 250 sites total and the number of sites from each category was weighted proportionally to the total web traffic visiting the category. We manually verified that each site we chose does in fact produce news which is generally considered to be accurate.

For our unverified articles, we turned to a post by Columbia Professor in which he categorized a host of sites he reasonably believed to publish fake content. We eliminated sites he characterized as biased as we didn't feel that these sites (such as Breitbart) were "fake."

Having compiled a list of sites publishing verified and unverified content we were happy with, we proceeded to scrape all articles published by these sources on April 7th, 2017 as a training dataset. We noticed a significant imbalance between verified (8472) and unverified sources (1625) which we accounted for in our machine learning algorithm through weightings and adjusting priors appropriately.

Multinomial Naïve Bayes for Classification

The Multinomial Naïve Bayes on full text proved to work the best given several models we tried (Bernoulli Naïve Bayes, Linear SVM, and Logistic Regression). 10-Fold Cross Validation accuracy was 90.91%

The general equations for multinomial Naïve Bayes is as follows:

$$\log p(C_k \mid \mathbf{x}) \propto \log\left(p(C_k) \prod_{i=1}^{n} {p_{ki}}^{x_i} \right)$$

$$= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki}$$

$$= b + \mathbf{w}_k^\top \mathbf{x}$$

where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$.

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Essentially each x_i is a feature found in a document. The “features” are the number of all unigrams and bigrams of all words from the training text of all documents found in a specific document. For training text, we tried both the full title of the article and the full text: full text always worked better except in the case of Bernoulli Naïve Bayes. The algorithm is trained on a vector of x_i for all the documents in the training set. A given document is called “fake” or “verified” based on the weights on the sum of the weights of its specific unigrams and bigrams.

The second equation is the Bayesian “likelihood”: the probability of a given vector given a classification class. The first equation is the actual formula for determining the posterior probability of a classification of an article given its feature vector.