Michael Baldwin

Josh Engelsma

Adam Terwilliger

February 16, 2016

CIS 678 – Machine Learning

Project 2

**Abstract**

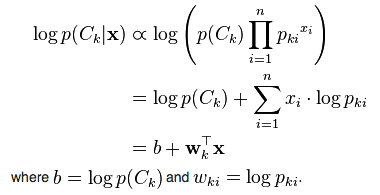
We look to further expand our abilities in CIS 678 – Machine Learning, through learning and implementing the Naïve Bayes algorithm for document classification. Using Python, we were able to develop a supervised learning model that uses probabilities from Bayes Theorem. As such, we abstracted our code in a way that allowed for modularity to train and test two different types of datasets: forums and twitter. Using three different validation approaches, we found classification rates over 80% for forum data (guessing would be 1/20 = 5%) and 75% for twitter data (guessing would be ½ = 50%). Our final analyses looked to demonstrate the predictive abilities of our classifier by predicting the sentiment of nearly 30 twitter users’ last 3000 tweets, and as such, we obtained a “positivity” rating for each user.

**Implementation details**

Our program is written in Python 2.7 and bash scripting in Unix. These programs were executed locally on each member’s respective Macbook Pro (2012), as well as, testing on eos23 and okami.

**Summary of Problem**

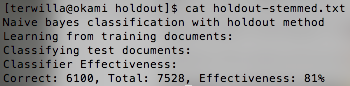
Linear regression is the foundational method when first understanding supervised learning. In supervised learning, we look to characterize a set of variables or model that can predict a known response variable. In this instance, we have just a single explanatory variable (simple linear regression), time in hours since the release date, and we are predicting amount of downloads as the response. A simple linear regression line can be represented with a slope and intercept, as seen in Figure 1.



**Figure 1. Formula of Bayes Theorem and Naïve Bayes log transformation**

**Results**

Our model correctly classifies over 81% of forum documents stemmed with Porter’s Stemmer using a 60/40 training/test holdout validation method, as we note in Figure 2.



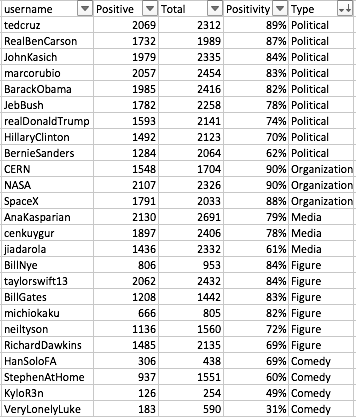
**Figure 2. Sample validation output using original holdout split**

Using the k-fold validation approach, we found 12-fold (92/8) with 86% classification rate offered the most promising results. We find Figure 2, maximizes over the k iterations with the maximum over 2 through 15-fold validation landing around 80%.

alidation methods like k-fold and randomized, we found 86% and 89% classification rates using 12-fold (92/8) and 78/22 for k-fold and randomized validation; respectively. We explored a sentiment analysis case-study training our Naïve Bayes classifier using over 1.5 million tweets pre-labeled with positive or negative sentiment. Using the randomized validation approach for this twitter data, we observed a classification rate of nearly 76% using an 87/13 training/test split.

**Figure 2. Sample Output from Python Regression program**

We can observe the performance of our models through R2 values in Table 1, and visually in Figure 3. As we uncovered, the linear fit to the data explains only approximately 42% of the variation in amount of downloads. However, as it appears in Figure 3, we see that the quadratic and cubic fits to the data show great promise explaining approximately 67% and 75% of the variation in amount of downloads; respectively.



**Table 1. R-Squared values for First, Second, and Third order models**

**Figure 3. Scatterplot of Time vs. Downloads with Models and Missing Value Imputation**

**Discussion**

One interesting feature we find in Figure 3 is missing value imputation. In our original dataset, 7 of the 744 total data points were missing. As such, we imputed these values with the quadratic predicted values for number of downloads. We choose this model for the imputation over the cubic with the principle of balancing model simplicity with the amount of variation explained.

We began to extract features of the dataset in Time of Day, Day of Week, and Day of Month as seen in Figures 4, 5, and 6; respectively.

**Figure 6. Histogram of Day of Month vs. Average Amount of Downloads**

Figure 4 provides little to no additional information, as we can infer that Time of Day would not be a valuable feature in a multiple regression model due to equal variance throughout the day with only a slight peak around 4/5 pm. Additionally, Figure 5 shows a great peak on days 1, 2, and 3 (we did not have day markers i.e. Sunday, Monday, etc.). However, this is a result of having 1 additional day contributing to the average downloads for the day, with the first three days showing the effect of the rise in downloads at the end of the month, as seen in Figure 6.

Our final note is with regards to avoiding overfitting the model as we may be encouraged by a higher order model explaining more of the variation in downloads; however, in future work, we should apply appropriate machine learning techniques of training and test sets to avoid this issue.

**Future Work**

Due to time constraints, we were not able to implement the histograms for each respective feature in D3.js. We value these data-driven libraries as we appreciate the customization and future forms of interactivity and animation that are crucial in exploratory data analysis.

We would also be interested in separating our data into training and test sets to better validate our model.

**Credits**

- [Simple Explanation of Naive Bayes](http://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification)

- [Where to start with text mining](http://tedunderwood.com/2012/08/14/where-to-start-with-text-mining/)

- [Intro to Topic Modeling](http://journalofdigitalhumanities.org/2-1/topic-modeling-a-basic-introduction-by-megan-r-brett/)

- [Naive Bayes Time Complexity](http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html)

- [K-fold Cross Validation](https://www.cs.cmu.edu/~schneide/tut5/node42.html)

- [Python - Time Complexity of Operations](https://www.ics.uci.edu/~pattis/ICS-33/lectures/complexitypython.txt)

- [Python Progress Bar](https://github.com/WoLpH/python-progressbar)

- [Python K-fold Cross Validation](<http://stackoverflow.com/questions/16379313/how-to-use-the-a-10-fold-cross-validation-with-naive-bayes-classifier-and-nltk)>

- [Twitter for Python] (<https://gist.github.com/yanofsky/5436496>, <http://www.tweepy.org/)>

- [Twitter-sentiment] (<https://github.com/yogeshg/Twitter-Sentiment)>

- [Stemming] (<https://bitbucket.org/mchaput/stemming)>