NET NEUTRALITY LANGUAGE ANALYSIS

Li Tao, Xinyi Xie

Department of Electrical Engineering, Stanford University
Email: ltao2@stanford.edu, xinyixie@stanford.edu

***Abstract* — The Federal Communications Commission (FCC) has been seeking public comments about net neutrality. It is found that a great majority of these comments were written from format letters potentially provided by several campaign groups. Principle Components Analysis (PCA) was first used to visualize the data set and it is found that the data are obviously clustered. Based on this fact, several classifiers were built up based on k-means clustering, logistic regression, and Na ïve Bayes event model to identify if a certain comment is written from format letters.**

Keywords: net neutrality, classification, Principle Components Analysis, k-means clustering, logistic regression, Na ïve Bayes event model

I. INTRODUCTION

Net neutrality is the principle that Internet service providers and governments should treat all data on the Internet equally, not discriminating or charging differentially by user, content, site, platform, application, type of attached equipment, or mode of communication.

The Federal Communications Commission (FCC) recently launched a notice of proposed rule-making (NPRM) seeking public comments on how best to protect and promote an open Internet. As its largest-ever public comment collection, the Commission announced the bulk release of all the comments received from the public. The comments are being reviewed now and the FCC is expected to make their final decision by the end of the year.

In this project, we try to make use of this rich dataset. Specifically, we aim to use natural language processing techniques and topic modeling methods to identify the most relevant keywords in these comments. Then, we can make use of these keywords to group the comments. Once grouped, some of the hidden features behind the comments can be exploited. For example, we can classify the comments into agree and disagree categories. We can also identify if a certain comment is written from any templates provided by organized campaigns.

The results of our study will potentially be considered by the FCC to help them to make the final decisions, which would greatly influence all giant Internet companies, the whole Internet industry, and everyone's life as long as he or she uses the Internet.

II. DATA PROCESSING

The original comments released by the FCC serve as our raw data and we process this data set with several Java programs designed. Our first goal is to use the original data to build up a Lexicon containing the most frequent and relevant words in our training set. Secondly, we use the raw data and the Lexicon to get a statistical result that records how many times

each word in the Lexicon appears in each comment. Below is a detailed description.

A. Building the Lexicon

Our first Java program is designed in order to build the Lexicon. In this part, the program scans the whole file which contains all the selected comments in the training set and extracts every non-duplicate word and then records the times that each of these words appears. The first 5 most frequent words are: "the", "and", "internet", "to", "a".

Leaving out the words with no special meanings, we manually select out the first 53 most relevant words based on the result above and that comes our Lexicon, with the first 5 words: "choice", "common", "open", "protect", "use".

B. Analyzing Original Comments with Lexicon

Our second Java program is designed to count how many times each word in the Lexicon appears in each comment. We store the results in a 2-D array (our data matrix) where each row represents one comment and each column represents one word from our Lexicon. The i, j-th element stands for how many times word j appears in comment i.

We manually label our data set with two methods. First, just two labels are used – 1 for formatted comments and 0 for unformatted comments. In addition, since we know exactly that 5 different templates showed up in our training set, we then use a total of 6 labels – 1, 2, 3, 4, 5 for the 5 different templates we identified and 0 for unformatted comments.

C. Training Set and Test Set

We randomly select out 598 comments for the training set and 257 for the test set. Both sets are made sure to contain both formatted and unformatted comments.

III. MODELS AND RESULTS

To visualize the data points, we first implement Principle Components Analysis (PCA) method to map the data onto a three-dimensional space. After that, k-means clustering, logistic regression, and Naïve Bayes event mode are used to build the classifier for the data.

All methods are implemented in the same ways as on CS229 lecture notes and problem sets. We will not talk about the math in details in this report.

A. Principle Components Analysis

First, we want to visualize the data points to get preliminary knowledge of how the data points are clustered. PCA method was used in this process. The distribution of the data points after being mapped to a three-dimensional space is shown in Figure 1.

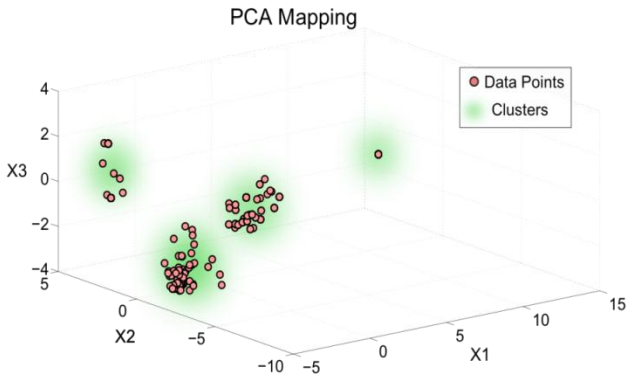

Figure 1. PCA mapping of data points onto 3D space

The PCA result shows an obvious clustering of the data points to several centroids. This can justify our assumption that the comments are written based on several templates. It also gives us ground to use unsupervised clustering to analyze the data.

B. K-means Clustering

Based on the PCA result, we then used k-means method to cluster our data. During this process, we first used just two centers – one for formatted comments and one for unformatted comments. We then used a total of 6 centers – 5 for the 5 different templates we identified and 1 for unformatted comments. The comparison of the training error and test error of these two clustering methods is shown in Figure 2.

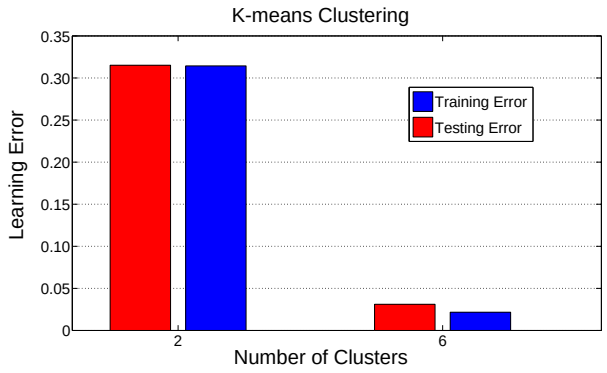

Figure 2. Comparison of k-means method with different number of centers

From the comparison, we can see that using the correct number of centers greatly reduces both training and test errors. Therefore, in the process of unsupervised clustering, identifying the most appropriate number of centers used should be the very first step.

C. Logistic Regression

In logistic regression, we first simplified our data labeling. We used to label the comments according to the 5 different templates they were written from. In logistic regression and Naïve Bayes event model, we kept a simplified labeling for the data without distinguishing different templates, i.e., we labeled 1 for all the formatted comments and 0 for all the unformatted comments.

We used gradient ascent to maximize the log-likelihood when implementing logistic regression algorithm.

Since each example shows the frequencies of each Lexicon word showing up in a certain comment, the number of features used in logistic regression will be equal to the total number of words in our Lexicon, which is 53 in this case. This large number of features will result in severe overfitting of our model and we need to select out just a few features that can contribute the most to our classification problem. Therefore, we first did our feature selection using forward search method. The change of training error and test error during the forward search process is shown in Figure 3.

Figure 3. Change of training and test errors during forward search process

We can see that by using the first 5 most relevant features we can already achieve satisfactory performance. Adding more features will not decrease the training error or test error any further. Therefore, we chose to use only the 5 features with indices 5, 11, 49, 1, and 6 as our features in logistic regression.

D. Naïve Bayes Event Model

In Naïve Bayes event model, we used the same data matrices and labeling method as in logistic regression, but we took into account all

the 53 features, corresponding to the 53 words in our Lexicon. We also made use of Laplace smoothing in our algorithm.

In this final step, instead of just learning a classifier, we would like to study the effect of the training set size on the training and test errors. Therefore, we generated a learning curve to show the relationship between the size of our training set and the training and test errors. The learning curve is shown in Figure 4.

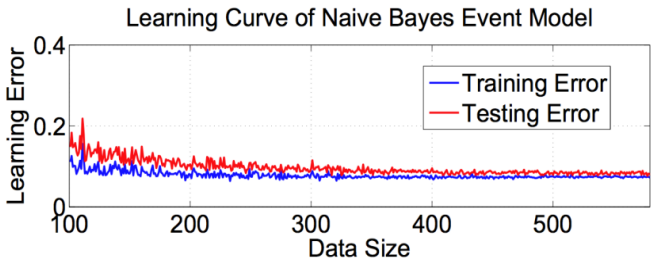

Figure 4. Learning curve generated with Naïve Bayes event model

From the learning curve, we can see that the training error and test error will converge to a similarly low level with increasing training set size. Our model selects a training set of around 600 examples, which corresponds to the point where the two errors have already converged. This will guarantee a satisfactory performance of our model.

IV. Discussion and Conclusion

From the PCA visualization and our analyses of the data, we can see that a great majority (more than 60%) of the comments collected by FCC are written based on format letters provided by certain campaign groups.

We develop several algorithms which successfully classify the data points. The comparison of the methods we used is summarized in Table 1.

Table 1. Comparison of methods

Model	Training Error	Test Error	Speed
K-means	2.17%	3.11%	fast
Logistic Regression	0.17%	0.39%	slow
Naïve Bayes Event Model	7.53%	9.73%	fast

From the comparison, we can see that logistic regression after feature selection gives us the smallest training and test errors. This is related to the fact that logistic regression is a more general model and relies less on the assumptions of the data set. Given enough training examples, logistic regression can generate a classifier that can best represent the data set.

The errors of Naïve Bayes event model are the largest. This might result from the fact that our data do not obey the Naïve Bayes assumptions perfectly, i.e., the consecutive words in each comment are actually related to each other rather than independent.

Based on our preliminary attitude analysis (not shown in details here), more than 99% of the comments agree with net neutrality, which makes format letter analysis more meaningful and feasible than attitude analysis.

V. Future Work

Given more time, we would like to proceed with the following aspects:

1. Collect more data into the training set and test set.
2. Exploit different feature sets to classify the data.

3. Implement semi-supervised learning with the data.
4. Study more aspects of the data set in addition to identifying formatted or unformatted comments, such as the relationship between the content of a certain comment and the age, educational background, and/or geographic location of its writer.

ACKNOWLEDGEMENT

The authors would like to thank our mentor Professor Dan Jurafsky for providing us with such an interesting topic. We would also like to thank Professor Andrew Ng and all the CS229 staff for a great course.

REFERENCES

[1]. CS229 lecture notes and problem sets.
[2]. FCC website: http://www.fcc.gov/guides/open-internet
[3]. Wikipedia: http://en.wikipedia.org/wiki/Net_neutrality
[4]. Leticia Miranda, The FCC's Net Neutrality Proposal Explained, *The Nation*, May 21, 2014.
[5]. Bob Lannon, Andrew Pendleton, What can we learn from 800,000 public comments on the FCC's net neutrality plan?, *Sunlight Foundation*, Sep 02, 2014.