Classification of Optional Pratical Training (OPT) comments using a Naive bayes classifier

Anand
a3anand@ucsd.edu

Sampath
svelaga@ucsd.edu

Jorge Garza
jgarzagu@ucsd.edu

Adithya
akaravad@ucsd.edu

ABSTRACT

This paper aims to classify the optional practical training comments using a naive bayes classifier. We demonstrate the effectiveness of the Naive bayes approach and further enhance its performance using a simplified form of an expectation maximisation algorithm. We explore how sentiments change over time, and also provide preliminary results that help in understanding how sentiments vary with ethnicity.

1. INTRODUCTION

OPT is a scheme in which students with F-1 visas are permitted by the United States Citizenship and Immigration Services (USCIS) to work for at most one year on a student visa towards getting practical training to complement their field of studies. On April 2, 2008, the department of homeland security(DHS) announced an extension to the OPT which was passed by USCIS as an interim rule. This rule allows who graduate in a Science, Technology, Engineering or Math majors can get at OPT extension for upto 17 months. In August 2015, a US federal court gave its verdict on a lawsuit challenging the 17-month OPT STEM extension. The court has decided that the interim rule was deficient as it was not subjected to public notice, comments and opinions. The court vacated the 2008 rule allowing the 17-month extension. However, a stay was put in place until February 12, 2016. DHS will have until then in order to take action regarding the fate of the STEM extension program. This rule was open to public comments for a one month duration, ending on Nov 18th. The comments are publicly available at http://www.regulations.gov/#!docketDetail;D=ICEB-2015-0002

2. INFORMATION ON THE DATA

2.1 Data collection

2.2 Data visualisation/Exploratory analysis

2.3 Dataset Labeling

Since the original dataset is unlabeled, we manually labeled the first 900 comments as *support* or *oppose*. Out of these, the first 600 were used for training, comments from 601-700 constituted the validation set and 700-900 were used for testing . We used validation set to pick the best possible model from amongst a pool of possible models.

2.4 Predictive task

Our main goal here was to classify whether a given comment is supporting or an opposing. In addition, based on the classifier we obtained, we also examined how the proportions of supporting and opposing reviews varied with time. Finally, we tried to examine the trends on an ethnicity basis. The main idea of this was to check if the pattern folows the hypothesis that most Americans oppose OPT extension, while people from other ethnicities support it.

2.5 Dataset Preprocessing

As a preprocessing step, we removed all the punctuations from the words. We also changed all words to lower case letters, although a more rigorous model could make use of the caps information to identify stronger sentiments. Finally, all the common stop words were removed as they convey little meaning.

3. RELATED LITERATURE

4. ALGORITHMS AND MODELS TRIED FOR CLASSIFICATION

Broadly speaking, there are two classes of algorithms that could be tried to classify the comment labels - supervised and unsupervised.For unsupervised learning, we tried clustering based on the *tf-idf* features extracted from the text with the Eucledian distance metric. . Hierarchical clustering runs in time $O(n^3)d$, where n is the number of datapoints and d is the number of dimensions of the feature vector, making it very slow for large datasets. Therefore, we implemented K-means clustering which is much faster. However, no useful clusters were identified and the accuracies were no better than those of a random classifier. This is to be expected because there is no coherent structure across the different comments - they are of varying lengths and contain different kinds of vocabulary to express the same sentiment, thus rendering Eucledian distance as a very bad distance measure.

Naive bayes performs particularly well for text classification despite the aggressive assumption it makes about independence. The reason for this is thought to be because,

© 2015 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

although naive bayes fails to produce good estimates of the probabilities, we do not require the absolute values of these, but only the relative ordering to estimate the MAP estimate. Reports by cite Vikesh suggested that Naive Bayes indeed performs well on this dataset. There are at least two popular versions of Naive Bayes - Multinomial and Bernoulli. Bernoulli naive bayes makes the assumption that each document belonging to a class contains osscurences of some words that are described by the probability distribution of the words belonging to that class. The Probability of the document given the class can then be modeled by:

$$P(doc|class) = \prod_{w \in doc} P(w|class) \prod_{w \notin doc} (1 - P(w|class)$$

On the other hand, Multinomail naive bayes assumes that the document of a particular class is generated by the following generative process - First, the length is chosen according to some distribution(which we don't care, as the length does not depend on the class labels). Then, every word in the document is generated by a multinomail distribution over the words belonging to that class. In this case, the corresponding probability can be modeled by:

$$P(doc|class) = P(|length(doc)|) \prod_{w \in doc} P(w|class)$$

We implemented both multinomail and bernoulli naive bayes, but we considered only the multinomial model for further analysis because the run time was better whilst the performance was similar.

Multinomial models using just unigrams, just bigrams, and using both unigrams and bigrams were considered. Intial results showing the accuracies on the training set, test set and validation set are presented below. INSERT TABLE HERE From the table, it is clear that:

1. The bigram only model overfits to the training data
2. The unigram+bigram model is performing almost the same as the unigram only model

Based on these, and the obvious speed of unigram only over unigram+bigram, we considered the unigram only model for all further experiments.

4.1 Semi supervised estimation

It has been suggested by the authors in CITE, that in cases where the number of training examples is small, the performance of the naive bayes classifier can be improved by combining it with an expectation maximisation algorithm. In short, the authors suggest to do this :

1. Predict the class probabilites $P(class|data)$ for all examples in the dataset
2. Retrain the model based on *class probabilities* estimated in the previous step

The first step above is an expectation step in disguise, and the second step corresponds to the maximisation. Although the second step requires us to retrain the model based on the probabilities in the previous step, we relax this step as follows: Relaxed expectation maximisation:

1. Predict the class probabilites $P(class|data)$ for all examples in the dataset
2. Retrain the model based on *class labels* estimated in the previous step

This algorithm, which we'll refer to as "classification maximisation" algorithm is a convenient approximation to the more rigorous expectation maximisation. What this means, is that we use the predicted labels as the actual labels and retrain the model based on these labels until convergence. These iterations significantly improve the accuracy of the naive bayes model by incorporating the knowledge from the large pool of unlabeled examples. Refer to FIGURE to see how the accuracy of the model changes with the iteration of the EM algorithm.

5. RESULTS AND DISCUSISION

6. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LaTeX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

7. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

8. REFERENCES

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

9. MORE HELP FOR THE HARDY