A Survey of Document Classification Problem Using Modified Naïve Bayes and E-M Algorithm*

Michael Peven
Johns Hopkins University
100 Bleecker Street, 27A
New York, NY 10012, USA
mpeven1@jhu.edu

Christian Reotutar
Johns Hopkins University
3700 N. Charles St.
Baltimore MD, USA
creotut1@jhu.edu

Abstract

Document classification is an important machine learning and natural language processing topic that has applications in spam filters and summarization techniques. In this typically multi-class problem, the Naïve Bayes classifier is a popular model with results rivaling that of the Support Vector Machine classifier with shorter training time. In this survey, we seek to find improvements in the Naïve Bayes classifier such that accuracy is improved and compare this to another classifier used in the problem space.

1 Introduction

The classification of documents, spanning from text documents and articles to images and videos, has been a prominent problem in information and computer science. The problem is to assign a category or classification to a given document that gives some indication to relationships between documents or information about the document itself.

1.1 Problems in Document Classification

Many problems in the

*This document has been adapted from the instructions for earlier ACL and NAACL proceedings, including those for NAACL-HLT-09 by Joakim Nivre and Noah Smith, for ACL-05 by Hwee Tou Ng and Kemal Oflazer, for ACL-02 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats. Those versions were written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence*.

1.2 Topic Classification

This survey deals with the problem of document classification in relation to summarization of documents. That is, given a text document, we want to predict the topic of the document based on a bag-of-words approach. We do this in the domain of Twitter messages: given a Tweet body, we wish to predict the associated hashtag for the Tweet.

2 Data Description

Here we talk about Twitter data and how it relates to our problem

3 Machine Learning Techniques

Here give a broad overview of what we're comparing, what techniques we used and why

3.1 Naïve Bayes

http://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf http://www2.hawaii.edu/ chin/702/sigir99.pdf
The Naïve Bayes Classifier is a probabilistic model based on Bayes' theorem for conditional probabilities but with a naïve independence assumption. The basic idea is to find the probability of a classification given the data (features). This is calculated with probabilities based on the entire training set. The independence assumption assumes the features conditionally independent. For example, features like time of day written and timezone are independent of each other and the presence of one value does highly correlate to the presence of another value. This assumption simplifies calculations and often makes Naïve Bayes a quickly trained classifier.

As it relates to our project, we use a bag of words as our features and a single word as the classification for the bag of words. We assume presence of a word is independent of other words.

3.2 Modified Naïve Bayes

Here we give background into the classifier + word2vec library

3.3 SVM

Here we give background into the classifier

4 Methods

4.1 Data Description

Here we talk about the filtering done on the data, the importance of feature vector formation and attribute picking, and the problems we came upon (e.g. language, more than one hash tag).

4.2 Code for Naïve Bayes

In this survey, we use the multinomial Naïve Bayes model used by [Yang and Liu, 1999]

4.3 Code for Modified Naïve Bayes

4.4 Code for SVM

5 Results

Here we give a broad overview of our results

5.1 Comparisons

Lots of tables and graphs

5.2 Explanations

It is important that we relate our results to the classifiers we used and other ML topics.

5.3 How we can improve

Any improvements for the classifiers we did not have time to implement

6 Conclusion

7 Comparison to Proposal

- A list of what we did differently from proposal

Acknowledgments

We would like to acknowledge Ilya Shpitser for giving feedback on our project and offering suggestions. Ilya for the skeleton for the code.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.