Learning to Classify Documents
TJHSST Senior Research Project
Computer Systems Lab 2009-2010

Edwin Zhang

April 6, 2010

Abstract

This project uses a Bayesian method to classify documents into certain categories. A set of training data will be used to to derive a formula for probability. A set of features (words) specific to a certain topic and the conditional probability of the apperance of these features (the formula), will be are used to determine the classification of documents of unknown categories.

Keywords: Bayesian probability, document classification

1 Introduction

In this project, I will be using the Nave Bayes Classifier to classify documents based on content. The Nave Bayes Classifier computes the conditional probability p(T—D) for a given document D for every topic T and assigns the document D to the topic with the largest conditional probability. The Nave Bayes Classifier then converts the calculation of the conditional probability into a formula that can be easily calculated using Bayes rule.

I expect that initially, the program may have trouble classifying documents into the correct cateogry but as the program learns more and improves its formulas, it will get better at classifying documents into the correct categories.

2 Development

The program consists of two major steps: Learning and Prediction. The Learning part makes use of training documents to develop a formula for conditional probability, to be based on the probability that certain features appear in documents of similar topic. We will go through the training documents and look at how often a certain feature appears in a document that is about a certain topic. For example, if our topic is "tennis" and our feature if "winner" we would go through all the documents and see how often "winner" occurs in documents about tennis and other documents. The Prediction part uses the results from the Learning portion to predict and classify the topic of an unknown document.

Right now, I am starting with only two categories: tennis and other. Once my program predicts correctly for two categories, then I will add more categories and keep testing. So far in my program, I have obtained my training documents for both categories and I have read them in. I created 3 new classes: Category, Documents, and Terms. My Category class deals with the different categories and holds all the training documents for specific categories. My Documents class deals with documents and all the terms that are in every document. My Terms class deals with terms, as well as the number of times each term appears in documents of different categories and assigns each term a score based on the term's counts in each category.

I have also created an array of categories. Each category has an array of documents, which each has an array of terms. In addition, I have an array of Terms containing every term that appears in all of my training documents and have the correct counts for each term. I have given each term a score for each category by dividing the number of times it appears in that category plus one over the number of times it appears in all the other categories combined plus one to avoid dividing by 0. I have also sorted my array of terms by the score for each term.

Next, I chose features for each category based on my array of Terms and the corresponding scores. I sorted the array for each category based on the score and took the first twenty-five terms that appeared. That finished the Learning portion of the program.

After that, I will have my program read in a document that I do not know the topic off and see if my program correctly predicts the category of the document.

3 Expected Results

I expect that the more training documents I have, the better my program will perform. I also expect the different methods of assigning scores will produce different results, so once I get my program running, I may experiment with that a little bit. I also expect that adding my categories will affect my results slightly.

4 Discussion

I am not done with my program yet, so I do not have results to discuss yet. However, I have finished the Learning part of my project, so once I finish the Prediction part, with only two categories, I will discuss my results.

References

[1] Chai, Kian Ming Adam, Hai Leong Chieu, and Hwee Tou Ng. *ACM Portal.* Assocation of Computing Machinery, 2002. Web. 14 Jan. 2010. ¡http://portal.acm.org/citation.cfm?id=564376.564395coll=Portaldl=ACMCFID=70884224C

[2] Eyheramendy, Susana, and David Madigan. "A Flexible Bayesian Generalized Linear Model for Dichotomous Response Data with an Application to Text Categorization", *Lecture Notes-Monograph Series*, 54 (2007): 76-91. JSTOR. Web. 25 Oct. 2009. ¡http://www.jstor.org/stable/20461460¿.

[3] Lavine, Michael, and Mike West. "A Bayesian Method for Classification and Discrimination." *Canadian Journal of Statistics* 20.4 (1992): 451-461. JSTOR. Web. 14 Jan. 2010. ¡http://www.jstor.org/¿.