CS689 Final Project: Text Genre Classification in the $n << p$ Regime

Sam Anzaroot and David Belanger

December 9, 2011

1 Introduction

An overwhelming amount of text data becomes available every second. The text comes in chunks corresponding to articles, press releases, blog posts, tweets, etc. These can all be thought of in general terms as 'documents'. Suppose you are looking for particular information in a very large collection of documents. A common real world situation is that there are so many documents that the time cost of looking for the information in all of the documents is prohibitive. Many of these documents may be irrelevant, however. You aren't likely to find the latest information on the European economy by looking in the sports section of the Wall Street Journal, but looking in the main section would be fruitful. Here, there are underlying genres for documents given by the section of the newspaper that they appear in. In many settings, annotation of such underlying genres is not available explicitly, and we require automatic methods for discerning the genre of a do cument.

Text genre classification can be accomplished by first mapping the document to a numerical feature representation and then using a general-purpose classification algorithm that was trained on labeled examples. Certain characteristics of text data make classification particularly challenging. These include:

1. Time has shown that the best way to embed documents in feature space is to map to a space where each dimension corresponds to a word in the vocabulary. The value in dimension i for a document can be represented, for example, as the frequency of word i appearing in the document [need a source for this]. Due to Zipf's law, the distribution of word usages in most text is extremely heavy-tailed [need a source for this]. Therefore, the dimensionality of the embeddings for documents in feature space can be quite high, in the tens of thousands.

2. In a given document, most words appear 0 times. Given that the feature embedding of documents has a dimension for each word, most dimensions are zero for a particular document. Therefore, features for text data are often extremely sparse.

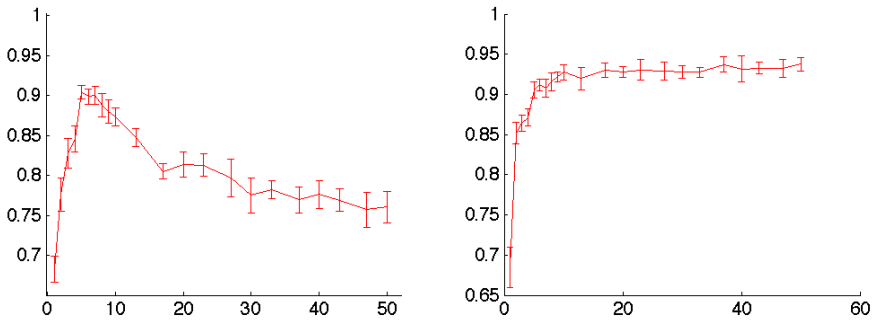
3. Data annotation can be expensive. Given the high dimensionality of feature embeddings for documents, we are often in the $n << p$ regime, where n refers to the number of distinct training points and p is the dimensionality of the feature space.

All three of these factors make classification difficult. Through experiments on the binary classification task of determining whether an article from Reuters is or isn't about corporate acquisitions, we help explain trends that occur when performing classification in this difficult regime.

2 Background

3 Experiments and Discussion

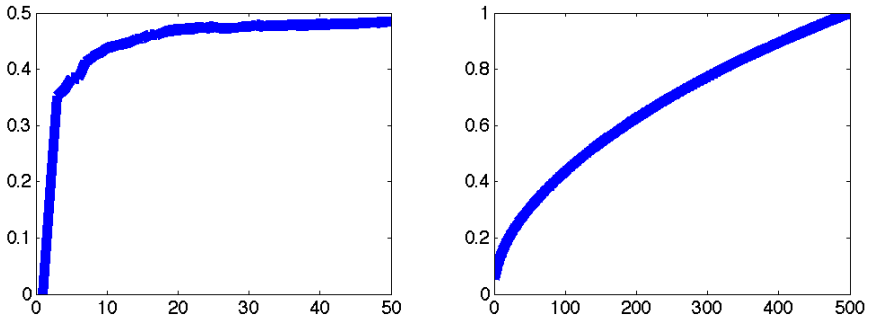
Outline: I: c3.m plots (acc and time) Observations: different algorithms have different behavior depending on size of training set. In figure 1 ...TODO. fix the figures too.



(a) Naive Bayes (b) SVM

Figure 1: Accuracy vs. dim. PCA Projection

In analysis, here, mention something about how with svm model complexity scales with number of training examples, not dimensionality of feature space. Also, point out that it makes sense to use linear kernel, rather than rbf, because of COD.



(a) Reduced Data Sparsity (b) Cumulative Covariance

Figure 2: Characteristics of PCA-reduced data

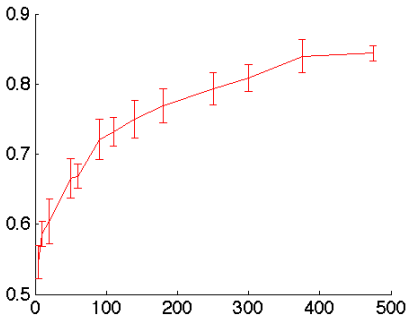


Figure 3: Naive Bayes Acc. vs. num dim Rand. Projection

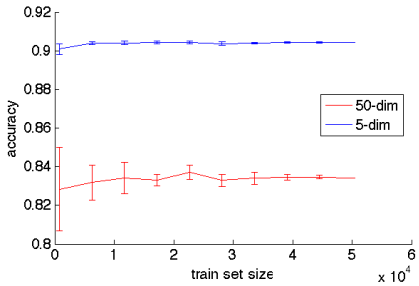
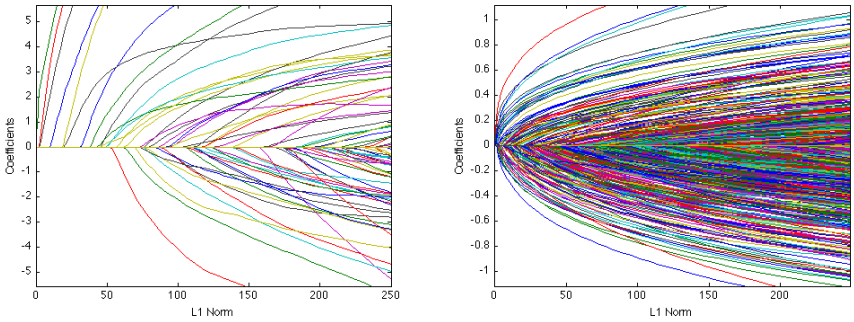


Figure 4: Naive Bayes Acc. vs. num dim Rand. Projection



(a) L1 regularization (b) L2 regularization

Figure 5: Coefficient Paths for Regularized Logistic Regression