CS-534 Machine Learning

Implementation Assignment 2

Mohammand Velayati ($33.\bar{3}\%$) Lucas Wells ($33.\bar{3}\%$) Chirag Shah ($33.\bar{3}\%$)

Introduction

In this assignment we are given a set of tweets from Hillary Clinton's and Donald Trump's Twitter accounts. We will train a Naïve Bayes classifier to predict the account that a new tweet comes from. Throughout this document we use the variable C to represent the account that a given tweet comes from, Hillary's or Trump's, and $\mathbf{x}$ represents a tweet, such that each x_i is a single word in the tweet $\mathbf{x}$ of class C. Also, let V be the vocabulary of unique words found in all tweets of the training set.

The Naïve Bayes classifier

The Naïve Bayes classifier is based on Bayes' Theorem,

$$p(C|\mathbf{x}) = \frac{p(C)p(\mathbf{x}|C)}{p(\mathbf{x})} \propto p(C)p(\mathbf{x}|C), \tag{1}$$

that states that the probability of of observing a class C, given a set of features $\mathbf{x}$, is equal to the prior probability of observing class C times the likelihood of the feature begin in that class divided by the evidence $p(\mathbf{x})$. As the denominator does not depend on the class we can state that the probability of observing a class is proportional to the numerator. This is equivalent to the joint probability $p(C, x_1, \ldots, x_n)$. In practice this computation on high dimensional features is prohibitive since we must estimate $k \times (2^d - 1)$ parameters. Naïve Bayes relaxes the conditional probability model and assumes that each feature is conditionally independent from all other features. Thus, we can write the conditional distribution over C as

$$p(C|x_1, \ldots, x_n) \propto p(C) \prod_{i=1}^{n} p(x_i|C). \tag{2}$$

Combining the Naïve Bayes probability model with the Maximum A Posteriori (MAP) decision rule we get

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i|C_k). \tag{3}$$

We will compare the performance of two Naïve Bayes event models, Bernoulli and Multinomial, on our Twitter classification problem.

Bernoulli event model

In the Bernoulli event model we represent tweets as binary vectors, so each word $x_i \in \{0, 1\}$ and $|\mathbf{x}| = |V|$. Thus, $x_i = 1$ means that the word V_i is contained in the tweet at least once, and $x_i = 0$ means that word V_i is not in the tweet $\mathbf{x}$. In the Bernoulli model we use

$$p(\mathbf{x}|C) = \prod_{i=1}^{|V|} p_{x_i|C}^{x_i} (1 - p_{x_i|C})^{(1-x_i)} \tag{4}$$

to compute the likelihood that a given tweet $\mathbf{x}$ is from account C. And the likelihood that word x_i is found in class C_k is estimated by

$$p_{x_i|C_k} = \frac{n_k(x_i)}{N_k}, \tag{5}$$

where $n_k(x_i)$ denotes the number of occurrences of word x_i in class k and N_k is the number of tweets from class k. Since the product of many probabilities could result in underflow, we compute the log of the likelihoods by

$$\log p(\mathbf{x}|C) = \log\Big(\prod_{i=1}^{|V|} p_{x_i|C}^{x_i} (1 - p_{x_i|C})^{(1-x_i)}\Big) \tag{6}$$

$$= \sum_{i=1}^{|V|} [x_i \cdot \log p_{x_i|C} + (1 - x_i)\log(1 - p_{x_i|C})]. \tag{7}$$

Combining equations (1) and (7) we perform MAP estimation in log space with

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; \log p(C_k) \sum_{i=1}^{|V|} [x_i \cdot \log p_{x_i|C_k} + (1 - x_i)\log(1 - p_{x_i|C_k})]. \tag{8}$$

The priors for each class are calculated using Maximum Likelihood Estimation

$$p(C_k) = \frac{N_k}{N}, \tag{9}$$

where N is the total number of tweets in both classes.

Multinomial event model

In the Multinomial case we represent tweets as integer vectors, so each word $x_i \in \mathbb{Z}$, where each element x_i denotes the frequency of that word in the tweet. The likelihood is given by

$$p(\mathbf{x}|C) = \prod_{i=1}^{|V|} p_{x_i|C}^{x_i}, \tag{10}$$

and the probability $p_{x_i|C_k}$ is the relative frequency of word x_i in tweets of class k divided by the total number of words in the tweets of that class. In log space we get

$$\log p(\mathbf{x}|C) = \log\Big(\prod_{i=1}^{|V|} p_{x_i|C}^{x_i}\Big) \tag{11}$$

$$= \sum_{i=1}^{|V|} [x_i \cdot \log p_{x_i|C}]. \tag{12}$$

Finally, combining equations (1) and (12) we perform MAP estimation with

$$\hat{y} = \operatorname*{argmax}_{k \in \{1,\ldots,K\}} \log p(C_k) \sum_{i=1}^{|V|} [x_i \cdot \log p_{x_i|C_k}], \tag{13}$$

where the priors are calculated using equation (9).

Implementation

1. (5 pts) Please explain how you use the log of probability to perform classification
2. (5 pts) Report the overall testing accuracy (number of correctly classified documents over the toatl number of documents) for both (Bernoulli and Multinomial) models.
3. (5 pts) Whose tweets were confused more often than the other? Why do you think this is?
4. (5 pts) Identify, for each class, the top ten words that have the highest probability.

Priors and overfitting (20 pts)

Identifying important features (20 pts)

Bonus