Sentimental Analysis

Ruchin Patel

Introduction

- Text data can range from two lines about to entire paragraphs. All the various degrees of information in text are ultimately used to classify the subject matter in a single binary domain.
- Sentiment Classification is determining emotional tone behind a series of words thus gaining a very succinct understanding of the opinions expressed.
- Our project applies Sentiment Classification to reviews of 8 books from Amazon.com. (8 data sets with avg. 20,000 reviews each)
- Why Naïve Bayes algorithm? for text classification, it considers as if all the words in the sentence are independent of each other. While this assumption might not be true, this doesn't matter since the predicted probabilities only used to make decisions. As a result, this algorithm requires less time for execution when compared to other superior techniques such as boosted trees.

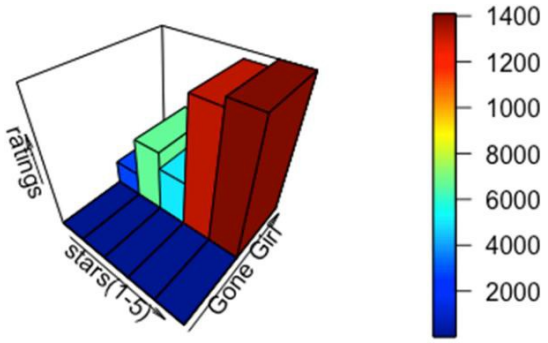


Figure 1 Graphical representation of the number of reviews of the book 'Gone Girl' that received ratings from 1-5 stars

- Review of a given book classified as ‘good’ if associated star rating is 4 or 5 on a 5-point scale, otherwise ‘bad’.
- Reviews are alphanumeric in nature. Words/tokens which do not help us in classifying a particular review as ‘good’ or ‘bad’ are removed. Vocabulary for a book consists of unique words from each review. Words with frequency less than 1% are also removed.
- Classifier can be described by the following formula:

$$c_{map} = \arg_{c \in C} max \left(P(c \mid r \ ) \right) = \arg_{c \in C} max \left(P(c) \prod_{1 \leq k \leq n_r} P(t_k \mid c) \right)$$

$$P(t_k \mid c) = \frac{T_{ct_k}}{\sigma_{t' \in V} T_{ct'}}$$

- To prevent floating point underflow in memory, logarithms are used in the above formulae.
- To prevent the computer from computing values such as $\log(0)$ Laplace Smoothing is used.

Algorithm for Training Classifier:

Calculate the probability $P(c)$ $\forall\, c \in \{'good\ review', 'bad review'\}$. We then calculate the conditional probability $P(t_k \mid c)$ which estimates the conditional probability of a particular word/token given a class as the relative frequency of term t in documents belonging to class c

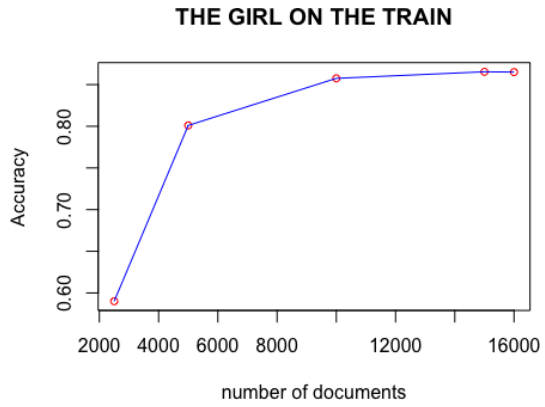
Algorithm for Test Data:

Using values computed during training phase, for words from the review which also belong to the vocabulary, we compute $\left(\log P(c) + \sigma_{1 \le k \le n_r} \log P(t_k \mid c)\right)$ for each class. The review then belongs to the class for which the above expression yielded the maximum value.

Results

- Naive Bayes classifier successfully implemented and tested. In process of classification, we estimated $P(c)$ which is marginal PMFs and also estimated $P(t_k \mid c)$, which is the ratio of joint PMF of (t_k, c) to the marginal PMF of c.

- The variation of accuracy of prediction of our classifier as the size of the data set increases is illustrated in the adjacent figure for one of the books



Conclusions

- As can been seen from the graphs, as the size of the dataset increases, the accuracy of our classifier also increases. Particularly, it can be concluded that the accuracy of our classifier increases as the absolute size of the training data set increases.

- For the limiting case when the biggest subset of the total number of available reviews was used to train the data set, for each of the eight books/data set, at minimum, the accuracy of our classifier is **greater than 80%**.

- Hence we may conclude that the predictions of our classifier are **good**.

P.S. : Graphs for all 8 books incorporated in report.