



Twitter Sentiment Analysis Report

CS 354N - Computational Intelligence
Under the guidance of : Dr. Aruna Tiwari

Pranay Munda : 160001045
Biplab Kumar Sahoo : 160001015

Introduction

Sentimental analysis is the task to identify an e-text , in the form of comment, review or message, to be positive or negative.

Instead of spending times in reading and figuring out the positive and negative of text we can use automated techniques for sentiment analysis.

Sentiment Analysis is used in opinion mining. We will be using the following model for sentiment analysis:

- Naive bayes (MultinomialNB)



Before we start

We need dataset:

For this we are taking the tweet dataset from the kaggle. Which is crawled and labelled as positive/negative (1&0).

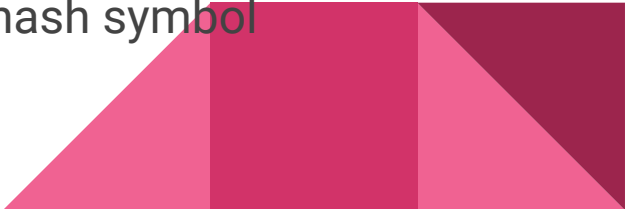
Dataset present in it is a Raw data containing bunch of emoticons, usernames and hashtags. which are required to be processed and converted into a standard form.

Data descriptions:

Training dataset is a csv file of type tweet_id, sentiment, tweet.



Pre-processing

- Convert the tweet to lower case.
 - Replace 2 or more dots (.) with space.
 - Strip spaces and quotes (" and ') from the ends of tweet.
 - Replace 2 or more spaces with a single space.
 - URLs in tweets with the word URL.
 - We replace all user mentions with the word USER_MENTION
 - We replace the matched emoticons with either EMO_POS or EMO_NEG depending on whether it is conveying a positive or a negative emotion.
 - We replace all the hashtags with the words with the hash symbol
- 

Feature Extraction

We extract two types of features from our dataset, namely unigrams and bigrams. We create a frequency distribution of the unigrams and bigrams present in the dataset.



Sparse Vector Representation

The feature vector for a tweet has a positive value at the indices of unigrams (and bigrams) which are present in that tweet and zero elsewhere which is why the vector is sparse.

presence In the case of presence feature type, the feature vector has a 1 at indices of unigrams (and bigrams) present in a tweet and 0 elsewhere.



frequency In the case of frequency feature type, the feature vector has a positive integer at indices of unigrams (and bigrams) which is the frequency of that unigram (or bigram) in the tweet and 0 elsewhere

| doc/ words | w1 | w2..... | class(+,-) |
|------------|----|---------|------------|
| Doc 1 | | | |
| Doc 2 | | | |



Naive bayes

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$c^* = \arg \max_c P(c \mid d)$$

Training

$$\phi_{k|label=y} = P(x_j = k \mid label = y)$$

$$\phi_{k|label=y} = \frac{\sum_{i=1}^m \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \text{ and } label^{(i)} = y\} + 1}{(\sum_{i=1}^m 1\{label^{(i)} = y\} n_i) + |V|}$$

$$P(label = y) = \frac{\sum_{i=1}^m 1\{label^{(i)} = y\}}{m}$$

Testing

Log of probabilities are taken for the laplacian smoothing

$$Decision1 = \log P(x | label = pos) + \log P(label = pos)$$

$$Decision2 = \log P(x | label = neg) + \log P(label = neg)$$



Confusion table

| | Positive | Negative |
|----------|----------------|----------------|
| Positive | True Positive | False Positive |
| Negative | False Negative | True Negative |

Accuracy

Precision (P) = $TP / (TP + FP)$

Recall(R) = $TP / (TP + FN)$

Accuracy(A) = $(TP + TN) / (TP + TN + FP + FN)$



output

Generating feature vectors

Processing 100000/100000

Extracting features & training batches

Processing 1/1

Testing

Processing 1/1

Accuracy: $7878/10000 = 78.7800\%$