

# SVM-based Sentiment Detection of Reviews

Anik Roy, Christ's (ar899)

November 7, 2019

Word Count: 499<sup>1</sup>

## 1 Introduction

This report considers the problem of the positive or negative sentiment classification of reviews. We reimplement a subset of techniques used in Pang et al. (2002)<sup>2</sup>, specifically support vector machine (SVM) and Naive Bayes (NB) classifiers. The data used was a set of movie reviews given in the framework of an NLP course.

## 2 Background

A bag of n-grams representation is used to classify documents. Pang et al. also choose some combinations of parameters to investigate the effect on accuracy of the final system, for example using unigrams or bigrams. The effect of considering the frequency of words compared to just their presence is also considered.

### 2.1 Naive Bayes

The NB classifier assigns documents to a class based on the likelihood of it being in that class  $c^* = \arg \max_c P(c|d)$ .  $P(c|d)$ , is derived by:

$$P_{NB}(c|d) := \frac{P(c)(\prod_{i=1}^m P(f_i|c)^{n_i(d)})}{P(d)}$$

<sup>1</sup>Using texcount, excluding footnotes

<sup>2</sup>Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. *Thumbs up?: sentiment classification using machine learning techniques*.

<sup>3</sup>Available at both <https://github.com/anik545/Part2-NLP> and at /home/ar899/NLP on MCS machines

<sup>4</sup><http://svmlight.joachims.org>

<sup>5</sup>[https://www.nltk.org/\\_modules/nltk/stem/porter.html](https://www.nltk.org/_modules/nltk/stem/porter.html)

making the naive assumption that features are dependent only on class.

### 2.2 Support Vector Machines

SVMs are a type of classifier which treat each document as a vector of features. Training consists of finding a hyperplane to separate the two classes, and classifying documents by measuring distance to the hyperplane.

## 3 Method

The systems described in Pang et al. were reimplemented in Python<sup>3</sup>. We use Joachim's (1999) SVMlight package<sup>4</sup> as our SVM implementation.

We consider several different types of features - unigrams, bigrams as well as both unigrams and bigrams. We also perform stemming on words before converting to n-grams, and consider the effect of stemming on the performance of the classifiers. The porter stemmer implemented in the nltk package<sup>5</sup> was used. No feature cutoff was implemented, and no stoplists were used. We also do not consider the position of

Features	Frequency or presence?	NB	SVM
Unigrams	frequency	<b>81.0</b>	73.3
Unigrams	presence	82.8	<b>86.3</b>
Unigrams + Stemming	presence	82.3	<b>85.6</b>
Bigrams	presence	<b>85.5</b>	83.0
Unigrams + Bigrams	presence	85.6	<b>87.4</b>

Table 1: Accuracies of NB and SVM systems with different feature types, averaging over 10 fold cross validation, in percent

System A	System B	P value
NB, Unigrams	SVM, Unigrams	<i><math>6.20 \times 10^{-4}</math></i>
SVM, Unigrams + Frequency	SVM, Unigrams + Presence	<i><math>5.09 \times 10^{-9}</math></i>
NB, Bigrams	NB, Unigrams	0.236
SVM, Unigrams	SVM, Bigrams	0.140
NB, Unigrams + Frequency	NB, Unigrams + Presence	0.421
NB, Unigrams + Presence	SVM, Unigrams + Presence	0.117
NB, Unigrams + Bigrams	SVM, Unigrams + Bigrams	<i><math>2.33 \times 10^{-6}</math></i>
NB, Unigrams + Presence	NB, Bigrams + Presence	<i>0.0418</i>

Table 2: Systems compared for system B significantly outperforming system A, using two-tailed sign test, with p-values under 0.05 in italics

words in sentences, nor do we carry out parts-of-speech tagging.

We use 10-fold stratified cross-validation, and dividing the data using round robin splitting. With 2000 documents (1000 positive, 1000 negative), and training on 90% of the documents, the system is trained on 446506 bigrams and 52556 unigrams - the first 9 folds.

## 4 Results

The accuracies produced by the systems under test are shown in Table 1.

The SVM classifiers perform significantly<sup>6</sup> better when frequency is not taken into account. While we observed an improvement in the accuracy of Naive Bayes when using

presence over frequency, it was not significant.

Just using bigrams did not produce significantly better results, for both SVM and NB.

We did not observe an improvement in accuracy when using stemming. For both classifiers, the best results were observed when using both unigrams and bigrams. In particular, SVM was significantly better than NB in this case.

## 5 Conclusions

Our reimplement of Pang et. al shows, as in the original paper, that machine learning methods can be successfully applied to the sentiment classification problem.

---

<sup>6</sup>When using a paired sign test with  $\alpha = 0.05$