Sentiment Classification using Machine Learning Techniques Reimplementation

Karen Sarmiento - Natural Language Processing - Task 1

1 Introduction

This paper is a re-implementation of Pang et Al's[1] paper which aims to explore the effectiveness of machine learning techniques when classifying document sentiments. We implement the Naive Bayes and the Support Vector Machine classifiers. Then, we run these on larger data sets and compare this with Pang et Al's findings.

2 Machine Learning Methods

2.1 Naive Bayes (NB)

The NB classifier[2] uses Bayes' rule to determine which sentiment (positive or negative) is most likely to reflect the review given the words that are present within the review. Assuming all words are independent and word order doesn't matter, the sentiment is given by the following equation:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \{\log P(c) + \sum_{i=1}^{n} \log P(f_i|c)\}$$

where $C = \{POS, NEG\}$ is the set of possible classes, $\hat{c}$ is the most probable class, and f is an observed token.

2.2 Support-Vector Machines (SVM)

The SVM classifier works by representing each document as a feature vector and plotting these in n-dimensional space. Then, we find the hyperplane which divides these points into their classifications, such that the margin (separation) between the classes is maximised.

This was implemented using the SVM$^{\text{light}}$ library[3].

3 Method

We use documents from an IMDB movie-review corpus, containing 1000 positive and 1000 negative documents. Note that this data set is much larger than Pang et Al's. Since we are using bag-of-words techniques, each document is represented by a multi-set of tokens. A token can be a word or punctuation symbol, as these may both point to the sentiment of that document. All tokens have been stemmed using Porter Stemmer[4].

Cross validation has been used: all documents have been split into 10 stratified folds. We deem a result to be statistically significant if the p-value obtained from a two-tailed signed test, is 5% or smaller

4 Results

4.1 Stemmed vs non-stemmed

No statistically significant results have been found between stemmed and non-stemmed results.

4.2 Frequency vs Presence

The SVM classifier performs significantly better when the feature vectors are used to describe the presence of words in the documents, as opposed to their frequency (insignificant p-value). In contrast, this does not hold for NB.

4.3 Unigrams vs Bigrams

In test (4), bigrams (consecutive pairs of words) are used as tokens. The NB classifier yields an improvement ($p = 4\%$) with bigrams, in comparison to treating unigrams as features. This improvement also holds for when both bigrams and unigrams are used($p = 3\%$) in test (3).

The SVM classifier does not yield any statistically significant results.

4.4 NB vs SVMs

When our feature vectors describe the frequencies of unigrams appearing in documents, NB significantly outperforms SVM. This disagrees with Pang et Al's findings.

Additionally, when describing presence, SVM outperforms NB when using unigrams and bigrams separately. However, the converse holds for when we use unigrams and bigrams together.

[1] Pang, Bo, and Lillian Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004.

[2] https://github.com/KarenSarmiento/NLP-Practicals

[3] http://svmlight.joachims.org/

[4] https://www.nltk.org/py-modindex.html

	Features	# of features	frequency or presence?	NB	SVM
(0)	unigrams (not stemmed)	1,656,748	freq.	**81.4**	73.6
(1)	unigrams	”	freq.	**82.4**	73.4
(2)	unigrams	”	pres.	82.6	**85.1**
(3)	unigrams+bigrams	3,313493	pres.	**86.2**	83.6
(4)	bigrams	1,656,747	pres.	86.0	**87.8**

Table 1: Average ten-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row).

5 Conclusion

To summarise, we find that SVM performs significantly better when run with presence, and in particular when run using bigrams. However, NB is still strong and still outperforms SVM in some cases, particularly when run with both unigrams and bigrams. We have noticed some differences in findings when compared to those of Pang et Al's. This may be attributed to the use of different data sets.