NLP Assignment 1 Mini-report

SVM-based Sentiment Detection of Reviews

Zébulon Goriely, Queens', zg258

Thursday 31st October, 2019

Word Count: 500[1]

1 Introduction

In a 2002 paper[2], Pang et al. considered the problem of classifying reviews according to positive or negative sentiment. In this report, we reimplement two of the machine learning systems described: the Naive Bayes (NB) classifier (with smoothing) and the Support Vector Machine (SVM) classifier.

These methods are applied to a dataset of movie reviews which were given to us in the framework of a course in NLP.

2 Background

Both NB and SVM use a bag-of-features representation of the documents. In this report, we examine three of the seven features used in (Pang et al, 2002):

- unigrams,
- bigrams,
- unigrams and bigrams.

Each document d is represented as a vector $\vec{d} := (n_1(d), n_2(d), \ldots, n_m(d))$ where $n_i(d)$ is the occurrences f_i of document d.

2.1 Naive Bayes

The first classifier derived in the paper is the *Naive Bayes* (NB) classifier. A given document d is assigned to the class $c^* = \arg\max_c P(c|d)$. By decomposing Bayes' rule:

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

It is 'Naive' because we assume that the features f_i are conditionally independent in order to calculate $P(d|c)$ needed to apply Bayes' rule.

Following the paper, we use relative-frequency estimation using add-one smoothing.

2.2 Support Vector Machine

The third classifier derived in the paper is the *Support Vector Machine* classifier. Letting $c_j \in 1, -1$ (corresponding to positive and negative) be the correct class of document d_j, the procedure finds the hyperplane represented by vector $\vec{w}$ which divides document vectors into two classes.

Once trained, classification proceeds by determining which side of the hyperplane each d_i falls on. We use Joachim's (1999) SVM^{light} package [3], using default parameters.

2.3 Feature frequency versus feature presence

During evaluation of these two systems, Pang et al. also experimented with binarising the document vectors, setting $n_i(d)$ to 1 iff feature f_i appears in d and running NB and SVM^{light} on these

[1] *texcount docs/assignment1/report.tex*, using just the text in each section

[2] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of EMNLP

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

[4] Note: Pang et al. used arbitrary language here, it may be worth referring to the Multinomial Naive Bayes and the Bernoulli Naive Bayes which are related to this notion of feature frequency and presence.

	Features	frequency or presence?	NB	SVM
(1)	unigrams	freq.	**81.6**	73.1
(2)	unigrams	pres.	82.9	**85.8**
(3)	bigrams	pres.	**83.7**	83.4
(4)	unigrams+bigrams	pres.	84.7	**87.1**

Table 1: Average ten-fold cross-validation accuracies, in percent. Dataset: 2000 stemmed movie reviews. Boldface: best performance for a given setting (row).

System A	System B	p value
NB,freq,unigrams	SVM,freq,unigrams	7.59×10^{-7}
NB,freq,unigrams	NB,pres,unigrams	0.53
NB,freq,unigrams	NB,pres,bigrams	0.32
NB,freq,unigrams	NB,pres,unigrams+bigrams	0.138
SVM,freq,unigrams	SVM,pres,unigrams	2.97×10^{-9}
SVM,freq,unigrams	SVM,pres,bigrams	9.57×10^{-7}
SVM,freq,unigrams	SVM,pres,unigrams+bigrams	6.24×10^{-7}

Table 2: The p-values from two-tailed sign-tests with a significance level of $k = 0.01$.

new vectors[4]. We have reproduced this as shown in Table 1.

3 Method

We reimplemented both systems using Python[5]. Before running these classifiers, we used an implementation of the Porter Stemming Algorithm[6] on the dataset.

When training the models on 90% of the data, we get 33000 unigram features and 383000 bigram features[7]. For the sake of simplicity, we have not cut out any features.

4 Results

We ran 8 different classifiers as detailed in Table 1, using different combinations of features and using frequency or presence-based document vectors. For each classifier, we calculated the accuracy using ten-fold cross validation[8].

As can be seen by line (1) of Table 1, NB performs significantly better than SVM^{light} with frequency-based unigram features with an average accuracy of 81.6% compared to 73.1% using SVM^{light}. Significance is determined by a sign test as detailed in Table 2.

As can be seen by lines (2), (3) and (4), by accounting only for features presence we achieve better performance for SVMs, giving average accuracies of 85.8%, 83.4% and 87.1% respectively compared to the 73.1% for feature frequency.

The best result achieved by all settings was the 87.1% when running SVM^{light} on presence-based document vectors composed of unigrams and bigrams.

Conclusion

Through this report we confirmed the results of (Pang et al, 2002), finding that Naive Bayes performs better than SVMs when using frequency-based unigram vectors and that significant gains in performances are achieved using SVMs when switching to a presence-based document vector with more features to choose from.

[5]Our implementation can be accessed online at https://github.com/ZGoriely/cambridge-nlp
[6]The Stemmer used was the PorterStemmer from the nltk.tokenize package.
[7]These values are rounded to the nearest 1000 as they vary between folds
[8]The folds were chosen using a deterministic round-robin algorithm, modular 10.