Learning from Data - Assignment 1 - Naive Bayes Classification

Remko Boschker
master student of information science at the Rijks Universiteit Groningen
s1282603, r.boschker@student.rug.nl

Abstract

This experiment uses a naive Bayes classifier on a tf-idf vector representation of a corpus of six thousand product reviews achieving an average f-score of 0.78 on predicting whether the review was positive of negative and an average f-score of 0.91 on predicting if the review was on one of six topics.

Good. Be more definite on what the experiments. Start with "this study" instead of "this experiment", as you are running two experiments.

1 Introduction

This experiment uses a naive Bayes classifier to predict if a review is either possitive or negative and to predict to which topic the review belongs. The predictions will be made on the basis of word frequencies in the review text.

Introduction should normally be longer than abstract. Say more about the task perhaps?

2 Related Work

-

Remove whole section if you're not using it

3 Data

The experiment uses a corpus six thousand product reviews. Each review consists of a label indicating whether the review is possitive or negative and a label indicating to which of the followings six topics it belongs: books, camera, dvd, health, music or software. The review also contains a file reference and the actual text of the review.

topic	cnt	%	pos	%	neg	%
books	993	16.5	471	47	522	53
music	1027	17.1	531	52	496	48
dvd	1012	16.9	490	48	522	52
health	986	16.4	470	48	516	52
software	994	16.6	502	51	492	49
camera	988	16.5	504	51	484	49
total	6000	100.0	2968	48	3132	52

Table 1: counts of topic and sentiment

Each topic is included in the corpus more or less equally ±0.5% as is the positive and negative sentiment within a topic and in total ±3% as you can see in table 1.

- This symbol usually means "positive of negative" not "approximately"
- "%" is "per cent", so you mean 50%, not 0.5%

I don't understand what you're trying to say here.

4 Method/Approach

The experiment uses the Python Scikit Learn module. The review texts are converted to a vector representaton using the term frequency - inverse document frequency statistic for each word in the document. This statistic is proportional to how ofter a word occurs in the document, but it is inversely proportional to how often the word occurs in the whole corpus to take into account words that simply occur very often without them being specially relevant for the document in question. The dataset is split into three quarters of the review used for training and one quarter used for evaluation.

Next the experiment uses the Scikit naive Bayes classifier for multinomially distributed data to build a probability model for the training data. The classifier then uses this model to predict the label for a review in the test data selecting the label with the maximum log likelihood. This is called the maximum a posteriori (MAP) decision rule. The experiment is run using the sentiment labels and using the topic labels.

5 Results

This first paragraph should be part of the previous section.

The experiment was run twice. Once fitting the classifier for the topic labels and predicting them for the test set and once for the sentiment labels. The results for both experiments can be found in tables 2 and 3 respectively.

Talking about two experiments is clearer. Saying "The experiment was run twice" is wrong, as you have different settings for each run.

topics

The scores are higher for the multi-class classification task. Although the prediction of positive sentiment labels has a high accuracy, the recall is low and the reverse is the case for the negative sentiment.

label	precision	recall	f1-score	support
books	0.94	0.91	0.93	233
camera	0.83	0.94	0.88	258
dvd	0.88	0.91	0.89	242
health	0.97	0.79	0.87	243
music	0.96	0.95	0.95	260
software	0.89	0.93	0.91	264
avg / total	0.91	0.91	0.91	1500

Table 2: scores for predicting the topic labels

label	precision	recall	f1-score	support
neg	0.71	0.93	0.81	731
pos	0.91	0.64	0.75	769
avg / total	0.81	0.78	0.78	1500

Table 3: scores for predicting the sentiment labels

6 Discussion/Conclusion

neg	pos
0.338	0.661
0.656	0.343
0.486	0.513
0.358	0.641
0.538	0.461
0.401	0.598
0.498	0.501
0.640	0.359
0.400	0.599
0.750	0.249

Table 4: posterior probabilities for the prediction of the sentiment labels of the first ten reviews

The prior probabilities for the topic labels are roughly 0.167 and for the sentiment labels 0.5. Analyzing the posterior probabilities, a sample of which is shown in tables 4 and 5, indicates that the word frequencies give a clearer prediction of the topic than of the sentiment labels in the sense that the difference between the maximum posterior probability for the correct label and the others is bigger for the predictions of the topic labels than for the sentiment labels, on estimate a value between 0.17 and 0.5 versus a value between 0.01 and 0.3.

This sentence is too long and unclear. . Why is the difference between the maximum and the others interesting? How did you calculate this?

It would seem that the reviews contains words that are more specific to the topic than to the sentiment. It could also be that the term frequency inverse document frequency vectorization works against the prediction of the sentiment as in the case of sentiment words with a high frequency in the whole corpus rather might be more relevant for the sentiment prediction although they are less document specific.

This is an interesting idea. Do you know how you could test it?

books	camera	dvd	health	music	software
0.544	0.041	0.241	0.037	0.088	0.045
0.214	0.111	0.381	0.099	0.111	0.081
0.037	0.790	0.056	0.041	0.033	0.040
0.088	0.060	0.606	0.039	0.143	0.061
0.114	0.056	0.223	0.389	0.143	0.072
0.108	0.028	0.172	0.029	0.606	0.055
0.035	0.782	0.040	0.059	0.040	0.040
0.093	0.026	0.726	0.018	0.111	0.024
0.072	0.150	0.066	0.054	0.046	0.609
0.075	0.484	0.112	0.120	0.074	0.133

Table 5: posterior probabilities for the prediction of the sentiment labels of the first ten reviews

Tables on the left are too big

7 Answers to additional questions

1. Why should one not look at the test set?

   One should not look at the test set, because if a system is developed with the test set in mind it may well become too specific for that dataset. It can score well, but perform poorly on other (unseen) data. ✓

2. What happens with cross-validation?

   Cross-validation means splitting the data set in a number of segments and using each segment in turn as the test data and the other segments as the training data. ✓

3. What baselines could you use for the binary sentiment classification and for the six- class topic classification? What would be their performance?

   Because the topic and sentiment labels are almost distributed equally in the dataset it does not really matter which label you choose to always select for a baseline system. For the binary classification always choosing negative would for predicting the negative sentiment correctly give a precision of 0.52 and a recall of 1 resulting in a f-score of 0.684.

   Choosing the most frequent class is one possible baseline approach, but you should describe the approach rather than talking about choosing a specific label. Nice calculation of F-Score. Why nothing about the 6-class problem?

4. What features are used in the classification task you ran?

   The features used in the task are the word frequencies in the review text. They are used in a feature vector calculating the tf-idf statistic for each word. ✓

Code: good comments. Nice to add argument parser, but not great use of it. You have use_sentiment and args.use_sentiment, but only use one of them. Not easy to work out how to run the code for the different settings. Some comments (such as “# Calls read corpus with trainset.txt as a filename and using the sentiment labels” are simply English versions of the code. Comments should focus on bigger picture.