Support Vector Machines and Feature Analysis

Remy Wang / Richard Scholtens
s2212781 / s2956586

Abstract

This research examines if it is possible to create a classifier which can be used for sentiment classification. A support vector machine classifier was used with count and tf-idf vectorizers as features. By comparing these results with the results of the Dummy classifier of Scikit Learn as baseline, there could be determined that this was indeed possible. For the sentiment classification our best model resulted in a F-score of 0.82 which is better than the baseline which resulted in a F-score of 0.51.

1 Introduction

There are many classifiers which work in different ways, use different parameters and therefor have different results. In this report the support vector machine (SVM) classifiers (Pedregosa et al., 2011) are examined by performing a binary classification task. This will be done by tuning with different parameters while explaining the results. For this research, a dataset will be used and processed by an SVC and LinearSVC classifier. These classifiers will also be tested to see which classifier yield better results. GridSearch has been implemented to see which parameters perform the best. This is a classifier which returns the best parameters for a classifier. Finally, the best classifier is proposed, and the results will be discussed.

2 Data

Machine learning models require input data which is used for training. Therefore, it is necessary to obtain a data set. In this inquiry a data set with 6000 reviews is used which is provided by the Rijksuniversiteit Groningen. The data set is distributed as a text file. Each line in the text file represents an instance and is structured in the following order: class, sentiment, document, and sentence. The sentences are classified by a total of six classes which include books, camera, dvd, health, music, and software. An instance can also be classified according to a positive or negative sentiment. The given document of an instance refers to the specific text file to which the sentence belongs. Pre-processing has been applied by tokenizing and lowercasing the data. This data will be used as training set.

Class	Sentiment	Document	Sentence
music	neg	575.txt	the cd came ...
dvd	neg	391.txt	this was a very ...
health	neg	848.txt	the braun ...
camera	pos	577.txt	when it comes to ...

Table 1: Sample data

Class	**Support**
negative	3032
positive	2968
books	993
camera	988
dvd	1012
health	986
music	1027
software	993

Table 2: Distribution of labels

3 Method/Approach

For this experiment the SVC and LinearSVC classifier will be used. This classifier will be used to classify the sentiment of reviews being either positive or negative.

A baseline is required to measure the performance of our classifier. For the baseline we use the dummy classifier provided by Scikit Learn (Pedregosa et al., 2011), where we use the stratified strategy which generates predictions by respecting the training set's class distribution.

Two vectorizers will be combined and used for our classifier for this task which is included in Scikit Learn pedregosa2011scikit. The first vectorizer that is used is the term frequency inverted document frequency (tfidf) vectorizer. This vectorizer gives a weight to each word using the equation shown in figure:

$$tfidf_{i,d} = tf_{i,d} \cdot idf_i$$

The second vectorizer that is used is the count vectorizer. This vectorizer simply counts the occurrence of each word.

For the evaluation of the performance of our tasks, precision recall and F-1 score will be used and compared with the baseline. The accuracy will also be calculated in both tasks. A confusion matrix will also be created for both tasks as it visualizes well how your system does. We will also use 10-fold cross-validation to evaluate our model. GridSearch (Pedregosa et al., 2011) will be used in our model to find the best parameters.

First, We will run a support vector machine with a linear kernel(cls = svm.SVC(kernel='linear',C=1.0) on the binary sentiment classification. By giving GridSearch different values for the C parameter it is possible to find the best parameters for this task. A radius basis function (rbf) kernel with the gamma parameter will also be used. Furthermore, LinearSVC (Pedregosa et al., 2011) will be used, and both implementations will be compared with eachother. Based on our observations and results on our cross validated data, a final decision on our kernel and parameters will be made, and be reported on.

4 Results

The results of classifier will be presented and discussed in this section. This will be done with the help of tables. In table 3 the results for the sentiment classification task are presented.

4.1 Default settings and C parameter

A SVC classifier using the default settings(cls = svm.SVC(kernel='linear', C=1.0) was run and validated by using 10-fold cross-validation in order to obtain the accuracy for this model. An accuracy of 0.833 has been acquired with these settings. The C parameter tells the SVM optimization how much you want to avoid miss-classification for each training example. If large values of C are used, the optimization will choose a smaller-margin hyperplane. When smaller values of C are used, the optimization will choose a larger-margin hyperplane. After tuning the C parameter,S the accuracy stays the same(C=0.1,0.2,0.3,0.5,1,2,3,4,5). This can be due to how our data set is distributed, and due to the volume of the corpus.

4.2 Radial basis function kernel

A radial basis function kernel has also been used to run our model. Cross-validation has been used to get the accuracy. GridSearch has been applied to find the best parameters for this model. The following parameters were given to grid search. For the kernel parameter the values were: 'linear' and 'rbf'. For the gamma parameter the values were: '0.1','0.5','1.0'. The best parameters are gamma = 0.1 and kernel = linear. It can be concluded that the linear kernel outperforms the rbf kernel in this classification task. An accuracy of 0.833 has been acquired with these settings using cross validation.

4.3 Implementation differences

In SciKit there are two SVM implementations models, namely the SVC and LinearSVC. By default scaling, LinearSVC minimizes the hinge loss(loss function) while the SVC minimizes the regular hinge loss. LinearSVC uses the one-vs-all multiclass reduction while the SVC uses the one-vs-one multiclass reduction. A big difference is that LinearSVC uses liblinear estimators, while SVC uses libsvm estimators. Liblinear estimators are optimized for a linear case, resulting in faster convergence on big amounts of data. Therefor LinearSVC needs less time to run than SVC.

4.4 Best model

The LinearSVC classifier combined with the two vectorizers(tfidf and count) proved to be our best model. The C parameter was set to 0.02. Different techniques like pos tagging, GridSearch, stopwords but we could not increase the score much. However, the module is robust due to the cross-validation.

Classifier	Precision	Recall	F1	Acc
Baseline	0.51	0.51	0.51	0.51
LinearSVC	0.82	0.82	0.82	0.82

Table 3: Results of the sentiment classifier

5 Feature contribution

It is possible to obtain the coefficient of the variation of the observations. The coefficient describes the level of variability within a population independently of the absolute values of the observations. Only when the absolute values are similar it is possible to compare populations using their standard deviations. However, when the values differ extremely or are from different variables it is wise to us a standardized measure. A measure like te coefficient of the variation. The standard deviation of the observations divided by the mean is the coefficient of the variation for the sample. The coefficient of the variation can be used to asses the precision of a model. When the standard deviation is proportional to the mean it can be used as a measure of variability. It can also be used as a means to compare variability of measurements made in different units. In our model it the coefficient points out that the value for the gamma parameter influences our model the most.

6 Discussion/Conclusion

This research set out to find an answer whether it was possible to build a good performing SVM classifier for a binary classification task. By using a support vector machine classifier a model was created which used review data as input. The previous section presents the results of our best model in tabel 1. For the sentiment classification our best model had an F1-score of 0.82. The baseline resulted in a F-score of 0,51.

This means it is possible to create a SVM classifier which can perform pretty well on this binary classification task. However, it is still possible to improve the classifier by adding more features or using more data.

References

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.