Special Topics – Machine Learning – Sentiment Analysis

Jovan Suvajac - 0982173

29 November 2019

Introduction

The major goal of this project is to select three different techniques from an existing machine learning package in order to study those techniques for the related problem. The topic this project is the issue of the binary classification of a labeled movie review data set using NPL techniques and machine learning.

The following three classification techniques were analyzed: Multinomial Bernoulli, Support Vector machine and Random Forest. Two feature sets were and 6 different feature sizes were used in the attempt to analyze the difference between trained classifiers and their performance across the 12 different configurations.

Technique descriptions

SVM – a discriminative classifier that attempts to optimally divide the plane in two parts. Its ability to classify can be improved with the adjustment of its regularization (shape) and gamma (distance for consideration

Random Forest – another classifier that is in fact a classifier of smaller classifiers (decision trees), the random decision property is used to correct for overfitting by the decision sub-trees

Multinomial Classifier – a classifier that is the extension of the binomial distribution, based on the idea of probability, it predicts the classification based on the number of occurrence of data seen before, fast, but less accurate that the others.

For the purpose of feature selection POS-tagging and stop words were used to filter out some of the unnecessary words, the main idea being that after the filtering, the dataset would contain a higher concentration of words useful for classification.

The given dataset was split into 3 portions, 15% of the data was reserved for cross validation, another 13% of the remaining 85% of the data was reserved for testing, and the rest was used for training of the classifiers. Cross validation was used as a more accurate means of measuring the success of the classifiers compared to just directly looking at the expected and generated sentiment values.

Additionally, the confusion matrix was used to guide development as it clearly show what the classifier is failing at.

Implementation highlights

In terms of feature selection, one set of features was derived from unigrams with removed stop words and the second feature set was derived from unigrams with stop words and filtered POS tags that correspond adjectives and adverbs, verbs and nouns.

major data structures and modules; assumptions and limitations; and important design decision

The project structure was broken int 3 files:

- Data_analysis.py
- Load_data.py
- Models.py

Data_analysis contains the statistics calculations required for that portion of the project, the results of which can be seen in the first set of tables in the results section below.

Load_data was designed to read in the entire dataset and unify it into 2 lists (positive and negative). of tuples, with the first tuple element being a document and the second tuple element being the correct classification of the review (positive or negative).

These lists of documents were the further processed by the removal of stop words and some pos tagging filtering, the output of which are the requirements for the two feature sets. The sets were serialized in memory to a file so that this data cleaning portion of the program can be decupled from the machine learning portion.

Models contains the entire machine learning portion of the program including the training and testing of the 2 feature sets with 3 models across all feature sizes. The file also contains all of the language data encoding.

Data, results, and analysis

The dataset used contained labeled movie review data consisting of 1000 positive and 1000 negative movie reviews. Each review was stored in its own file inside of the positive or negative directory.

Sentence Statistics

Category	Max	Min	Avg
Positive	113	6	33.937
Negative	113	2	32.783

Token Statistics

Category	Max	Min	Avg
Positive	2678	130	787.051
Negative	2181	17	705.63

Whole collection sentence

avg: 33.36

Feature set 1

Feature Size	Multinomial Bernoulli	Support Vector Machine:	Random Forest:
500	0.7612456747404844 0.6362359914791146	0.7577854671280276 0.6665749745299621	0.7335640138408305 0.7268593127720664
1000	0.7889273356401384 0.7201907937390016	0.7820069204152249 0.7268593127720664	0.7577854671280276 0.7298573677873483
1500	0.7854671280276817	0.8200692041522492	0.7716262975778547

	0.6666898212466426	0.690303788089284	0.6797934611466149
2000	0.7923875432525952 0.6567546540705751	0.7923875432525952 0.7469445216263778	0.7854671280276817 0.7469463739927756
2500	0.7923875432525952 0.7065295915532092	0.8477508650519031 0.6967518755209781	0.8166089965397924 0.7400889135871075
3000	0.8235294117647058 0.6501305918310643	0.8442906574394463 0.7103056404556821	0.8408304498269896 0.7099101602296934

Feature set 2

Feature Size	Multinomial Bernoulli	Support Vector Machine:	Random Forest:
500	0.7474048442906575 0.6171269797165879	0.7750865051903114 0.6204621654163194	0.726643598615917 0.6332397888302307
1000	0.7577854671280276 0.7028767250162082	0.7889273356401384 0.7234926368435677	0.7993079584775087 0.693099008983977
1500	0.7820069204152249 0.6633333333333333	0.7993079584775087 0.7033333333333334	0.8235294117647058 0.6966666666666667
2000	0.8096885813148789 0.6834361396684263	0.8339100346020761 0.7403927016763916	0.8408304498269896 0.7334472538668149
2500	0.7993079584775087 0.746700935445031	0.8512110726643599 0.73992312679448	0.7993079584775087 0.7165860887283504
3000	0.8339100346020761 0.680126887098268	0.8477508650519031 0.7269686023895525	0.8304498269896193 0.723301843104566

The cells in feature set tables above have the difference between the calculated and actual classification on the top and the cross-validation accuracy on the bottom.

It appears that as the feature size increased, the accuracy of all classifiers increased as well, the overall shift appears to have increased the accuracy by about 10% in both feature set and across classifiers when looking at the difference between the largest and smallest feature size measurements.

As for the differences between the classifiers, comparatively, they appear to be miniscule. SVM appears to consistently outperform the other two algorithms in terms of accuracy, Random Forest appears to outperform Multinomial in the larger feature sizes in set 1 but Multinomial appears to have more consistent and less variable accuracy.

It also appears to be the case that the second feature set (the one with additional POS-tag filtering) has achieved a higher degree of classification accuracy that feature set 1.

Conclusions and future work

As the results of the changes in features sets, feature sizes and classification method proved to not have a very significant effect on the effectiveness of the classifiers, it seems logical to explore other means of increasing the accuracy of classification. One such option is the algorithmic or even manual tweaking of classification parameters. In the case of SVM the change in regularization and gamma parameters could potentially have a positive impact on the success of the SVM classifier. This can be done for each classification method as each one has unique and adjustable parameters.

Another avenue for potential future work is the exploration of other well-known machine learning algorithms in addition to the ones used in this project. It is entirely possible that un-optimal classifiers were chose as there may exist another classifier that is much better suited to this dataset.

Alternatively, more work can be done on the pre-processing component of the project. Other filtering techniques could prove more fruitful in terms of classification accuracy and precision. The words in the dataset can be stemmed or filtered out using a different method.

Build/test guide

Installation guide:

As this project was written in Python3 and requires Python3 libraries in order to run. The library requirements are in the “requirements.txt” file in the root directory of the project and can be installed using the Python package manager (pip):

```
Pip3 install -r requirements.txt
```

Programs can be executed with:

```
Python3 programName.py
```

The order in which programs should be run:

- Data_analysis.py
- Load_data.py
- Models.py

This is the case as Data_analysis downloads data required generating stop words, which is a requirement for Load_data. Load_data partially filters the dataset and prepares the two feature sets. As these feature sets are serialized and save to a file, Load data must be run before Models.

It is recommended that the output of Models.py be diverted to a text file for easier analysis as it will output slightly more than 600 lines.

Testing

For the purposes of testing of the machine learning algorithms, parameters can be adjusted, and Modules can be run again to generate the classifiers and generate their output.

References

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP 02*. doi: 10.3115/1118693.1118704