L90 Assignment 2: Improving Sentiment Classification of Movie Reviews Using Support Vector Machines and doc2vec

Jennifer White (jw2088)

December 9, 2019

Abstract

This works aims to improve work from the previous L90 Assignment on sentiment classification of movie reviews by using Support Vector Machines (SVMs) and doc2vec. It compares the performance of the SVM with and without the use of doc2vec to create document embeddings to use as input. It also compares a variety of models created with `doc2vec` and asses the impact that each one has on the performance of the classifier. It finds that [FINDINGS].

1 Introduction

[DONE]

Sentiment classification is a common task within Natural Language Processing (NLP). It involves analysing a comment on a product or service, analysing it and classifying the nature of the sentiment it conveys, commonly into the binary classification of "positive sentiment" or "negative sentiment". Thanks to online movie review websites, a large number of movie reviews created by real people are freely available online, along with star ratings that tell us whether they had broadly positive or negative feelings about the movie. As a result, sentiment classification of movie reviews is a common task with easily available datasets.

In the previous L90 Assignment, this task was approached using a Naive Bayes model with Laplace smoothing. Different configurations of the model were tested and evaluated in order to find the model that gave the best performance.

However, Naive Bayes is a simple model and is unlike to allow us to achieve the best possible performance on this task. Additionally, the data for each review was being given to the model as a Bag of Words (BoW) vector either showing which words were present in the review or additionally providing the frequency of each word in the review. This does not allow the model to harness connections and similarities between any of the vocabulary words or between similar reviews. So it should be possible to improve on the results achieved previously by using a more sophisticated model and by feeding data into the model in a way that allows it to take advantage of this information.

This work investigates using Support Vector Machines (SVMs) for this task. SVMs aim to learn a hyperplane to separate the data such that the minimum distance from any point to the hyperplane is minimized [CHECK]. SVMs also allow the use of a kernel so that the model can learn to separate data that is not linearly separable. In this work, the choice of kernels was investigated in order to find which performed the best for each type of model tested.

Additionally, `doc2vec` is used to learn embeddings of the documents. These are then used as input to the SVM model. This work investigates whether this can improve the performance of the model and which parameters are model choices result in the most useful embeddings.

1.1 Previous Work

[DONE]

In the previous L90 assignment, this task was attempted using a Naive Bayes model. The highest accuracy that was achieved by the best form of the Naive Bayes model was 84.5%. This was achieved with a model that combined both unigrams and bigrams, took as input a vector indicating presence of a unigram or bigram in a document and used Laplace smoothing. This work builds on the work from the previous assignment and attempts to improve this accuracy by using SVMs and using doc2vec to create document embeddings to use as input to the model.

2 Data

2.1 Movie Reviews

[DONE]

The data used for training and testing of the SVM consisted of 2,000 IMDB Movie Reviews, of which 1,000 were positive, and 1,000 were negative. These reviews were available both as raw text and with additional Part of Speech tags. I chose to use the untagged reviews as Pang et al (2002) [3] found that using tags did not result in a statistically significant improvement for this task.

Of this data, 10% was reserved to be used as an unseen test set. 9-fold cross-validation was performed using the remaining data.

2.2 Data for Training doc2vec

[DONE]

In order to train doc2vec to produce suitable document embeddings, a corpus of 100,000 IMDB movie reviews was used [2]. This dataset was available as raw text. It contained some HTML tags which were stripped before use.

3 Method

[TO FINISH]

The code used for this work can be found on my user area on the MPhil machines (user: `jw2088`) in the file `file`[FILE NAME][1]. The models produced by `doc2vec` that were used to obtain the results discussed are also available on the MPhil machines [WHERE?][2].

3.1 Support Vector Machines

[TO FINISH]

The `scikit-learn` [4] implementation of SVMs was used in this work. [WHICH ONE?]

[PARAMETERS?]

The SVM model was tested in various configurations. It was tested using unigrams as vocabulary items, bigrams as vocabulary items and with a vector containing information about both unigrams and bigrams. It was tested with Bag of Word (BoW) vectors. Tests were performed both with BoW vectors that contained information about the frequency of the i^{th} vocabulary item in position i and with vectors that contained a 1 or 0 in position i in order to indicate the presence or absence of the i^{th} vocabulary item.

A frequency cutoff was used meaning that vocabulary items that appeared fewer than 5 times were not included in the BoW vector. Words that were outside of this cutoff were encoded as unknown vocabulary items.

3.1.1 Kernels

[TO FINISH]

SVMs allow for a choice of kernel that allows the model to learn to sucessfully separate data that is not linearly separable. The implementation of SVMs used in this work allowed for a choice of 4 kernels: `rbf`, `linear`, `poly` and `sigmoid`, which are as follows:

- `rbf`: [DESC]
- `linear`: [DESC]
- `poly`: [DESC]
- `sigmoid`: [DESC]

3.2 doc2vec

[TO FINISH]

[1] Alternatively, it can be viewed at `URL`

[2] Or they can be downloaded from `URL`

Table 1: Table of parameters used for `doc2vec` models

	Training epochs	**DM/ DBOW**	**Embedding Dimension**	**Frequency Cutoff**
dbow	50	DBOW	100	3
dm	50	DM	100	3
dbow100	100	DBOW	100	3
dm100	100	DM	100	3
dbowlarge	50	DBOW	200	3
dmlarge	50	DM	200	3
dbow5cutoff	50	DBOW	100	5
dm5cutoff	50	DM	100	5
dbow1cutoff	50	DBOW	100	1
dm1cutoff	50	DM	100	1

Doc2vec [1] was used to create document embeddings to be used as input to the SVM.

[DISCUSS DBOW VS DM]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac euismod sapien, ut varius metus. Maecenas vulputate lorem ac erat ornare, sed auctor eros lacinia. Donec faucibus euismod condimentum. Donec elit urna, consectetur sit amet felis sit amet, commodo iaculis massa. Sed non imperdiet augue. Duis gravida, urna id tempus congue, tortor tortor dictum dui, nec vulputate sem massa eu purus. Donec eget laoreet tellus, at mollis enim. Proin eu nibh nec neque hendrerit rutrum nec sed tellus. Vivamus dignissim neque sed augue viverra semper. Duis posuere ex sed justo iaculis consectetur.

Aliquam arcu erat, elementum fermentum maximus et, rhoncus et velit. Curabitur fringilla arcu sed nisl finibus, quis gravida mauris aliquet. Duis fermentum mi sed interdum congue. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur dolor velit, posuere vel aliquam ac, commodo ac tellus. Quisque eget ligula fermentum sapien euismod pretium. Pellentesque facilisis quam vel convallis pharetra. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut non purus bibendum, ullamcorper sem id, consectetur justo. Ut vehicula arcu quis leo pretium, tincidunt semper elit porttitor. Integer tincidunt lectus quis lorem auctor, quis bibendum leo venenatis.

Morbi at placerat dui. Praesent bibendum ullamcorper augue. Sed volutpat mauris et congue sodales. Donec vel dui non ante finibus aliquam. Nam accumsan nulla ut quam feugiat cursus. Integer et suscipit lorem. Nullam quis nisi justo. Nunc volutpat blandit dolor quis commodo. Nam ut massa non velit sagittis vehicula ut a nulla. Sed at euismod massa, at auctor neque. Ut id ullamcorper ligula. Cras ut finibus tortor.

4 Investigation and Evaluation of doc2vec models

[TO FINISH]

Some investigation was performed as to how each `doc2vec` performs when classifying documents.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac euismod sapien, ut varius metus. Maecenas vulputate lorem ac erat ornare, sed auctor eros lacinia. Donec faucibus euismod condimentum. Donec elit urna, consectetur sit amet felis sit amet, commodo iaculis massa. Sed non imperdiet augue. Duis gravida, urna id tempus congue, tortor tortor dictum dui, nec vulputate sem massa eu purus. Donec eget laoreet tellus, at mollis enim. Proin eu nibh nec neque hendrerit rutrum nec sed tellus. Vivamus dignissim neque sed augue viverra semper. Duis posuere ex sed justo iaculis consectetur.

Table 2: Table of accuracy achieved on the blind test set for non-`doc2vec` models

	rbf	**linear**	**poly**	**sigmoid**
unigram, presence	76.0%	83.5%	72.0%	44.0%
unigram, frequency	84.0%	84.5%	73.5%	83.5%
bigram, presence	56.5%	81.0%	55.3%	53.0%
bigram, frequency	81.0%	80.0%	55.0%	84.0%

Table 3: Table of accuracy achieved on the blind test set for `doc2vec` models

	rbf	**linear**	**poly**	**sigmoid**
dbow	90.0%	89.5%	89.5%	90.5%
dm	81.5%	80.5%	81.5%	79.5%
dbow100	88.0%	88.5%	88.5%	88.0%
dm100	90.0%	89.5%	89.0%	87.5%
dbowlarge	89.0%	88.0%	89.5%	91.0%
dmlarge	81.5%	82.0%	81.5%	83.0%
dbow5cutoff	89.5%	90.5%	91.0%	91.5%
dm5cutoff	84.0%	83.0%	82.0%	83.5%
dbow1cutoff	90.0%	90.5%	87.5%	91.0%
dm1cutoff	81.5%	82.0%	79.5%	74.0%

Aliquam arcu erat, elementum fermentum maximus et, rhoncus et velit. Curabitur fringilla arcu sed nisl finibus, quis gravida mauris aliquet. Duis fermentum mi sed interdum congue. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur dolor velit, posuere vel aliquam ac, commodo ac tellus. Quisque eget ligula fermentum sapien euismod pretium. Pellentesque facilisis quam vel convallis pharetra. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut non purus bibendum, ullamcorper sem id, consectetur justo. Ut vehicula arcu quis leo pretium, tincidunt semper elit porttitor. Integer tincidunt lectus quis lorem auctor, quis bibendum leo venenatis.

Morbi at placerat dui. Praesent bibendum ullamcorper augue. Sed volutpat mauris et congue sodales. Donec vel dui non ante finibus aliquam. Nam accumsan nulla ut quam feugiat cursus. Integer et suscipit lorem. Nullam quis nisi justo. Nunc volutpat blandit dolor quis commodo. Nam ut massa non velit sagittis vehicula ut a nulla. Sed at euismod massa, at auctor neque. Ut id ullamcorper ligula. Cras ut finibus tortor.

5 Results

[TO FINISH]

The accuracy obtained on the blind test set for each classifier can be seen in Tables 2 and 3.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac euismod sapien, ut varius metus. Maecenas vulputate lorem ac erat ornare, sed auctor eros lacinia. Donec faucibus euismod condimentum. Donec elit urna, consectetur sit amet felis sit amet, commodo iaculis massa. Sed non imperdiet augue. Duis gravida, urna id tempus congue, tortor tortor dictum dui, nec vulputate sem massa eu purus. Donec eget laoreet tellus, at mollis enim. Proin eu nibh nec neque hendrerit rutrum nec sed tellus. Vivamus dignissim neque sed augue viverra semper. Duis posuere ex sed justo iaculis consectetur.

Aliquam arcu erat, elementum fermentum maximus et, rhoncus et velit. Curabitur fringilla arcu sed nisl finibus, quis gravida mauris aliquet. Duis fermentum mi sed interdum congue. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur dolor velit, posuere vel aliquam ac, commodo ac tellus. Quisque eget ligula fermentum sapien euismod pretium. Pellentesque facilisis quam vel convallis pharetra. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut non purus bibendum, ullamcorper sem id, consectetur justo. Ut vehicula arcu quis leo pretium, tincidunt semper elit porttitor. Integer tincidunt lectus quis lorem auctor, quis bibendum leo venenatis.

Morbi at placerat dui. Praesent bibendum ullamcorper augue. Sed volutpat mauris et congue sodales. Donec vel dui non ante finibus aliquam. Nam accumsan nulla ut quam feugiat cursus. Integer et suscipit lorem. Nullam quis nisi justo. Nunc volutpat blandit dolor quis commodo. Nam ut massa non velit sagittis vehicula ut a nulla. Sed at euismod massa, at auctor neque. Ut id ullamcorper ligula. Cras ut finibus tortor.

5.1 Permutation Test

[DONE]

The Permutation Test was the test used to test for significance. The permutation test takes a set of paired results from two systems and examines how swapping a random selection of these would affect their means. In particular, it examines whether these permutations would change which mean is the larger one and uses this to determine whether the difference is significant or if it is likely to have occurred by chance. For the results presented here, a difference in means is considered significant if $p < 0.05$.

The significance tests were performed using the paired results of the two systems being compared from the 9-fold cross-validation tests. The average accuracies across cross-validation sets, along with the variance of these accuracy values, is presented in Tables 4 and 5.

In some cases, a system that performed better than another system on the blind test set

Table 4: Table of accuracy achieved on the cross-validation test set for non-`doc2vec` models, with variance shown in brackets

	rbf	**linear**	**poly**	**sig**
unigram, presence	69.3% (0.083)	81.6% (0.055)	65.2% (0.084)	46.6% (0.159)
unigram, frequency	84.0% (0.050)	83.1% (0.067)	72.8% (0.073)	84.3% (0.061)
bigram, presence	55.4% (0.071)	77.9% (0.089)	53.7% (0.052)	49.4% (0.238)
bigram, frequency	76.7% (0.087)	77.1% (0.145)	54.1% (0.037)	78.9% (0.069)

Table 5: Table of accuracy achieved on the cross-validation test set for `doc2vec` models, with variance shown in brackets

	rbf	**linear**	**poly**	**sig**
dbow	88.4% (0.039)	87.8% (0.048)	88.1% (0.038)	87.7% (0.062)
dm	82.6% (0.042)	82.5% (0.036)	81.0% (0.049)	81.0% (0.036)
dbow100	88.0% (0.061)	87.7% (0.072)	87.3% (0.061)	86.8% (0.042)
dm100	87.6% (0.071)	87.3% (0.090)	87.4% (0.058)	86.3% (0.050)
dbowlarge	88.0% (0.048)	87.7% (0.032)	88.5% (0.036)	88.0% (0.042)
dmlarge	82.0% (0.056)	81.7% (0.077)	80.5% (0.035)	81.4% (0.051)
dbow5cutoff	88.4% (0.041)	88.1% (0.038)	88.2% (0.032)	88.2% (0.048)
dm5cutoff	82.1% (0.035)	82.4% (0.033)	81.0% (0.056)	80.8% (0.043)
dbow1cutoff	88.3% (0.071)	88.0% (0.088)	88.1% (0.115)	87.7% (0.060)
dm1cutoff	81.0% (0.031)	81.4% (0.061)	79.4% (0.105)	73.1% (0.088)

actually performed worse on average in average of the cross-validation accuracies, so these differences were not significant.

5.2 Statistically Significant Differences

[TO FINISH]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac euismod sapien, ut varius metus. Maecenas vulputate lorem ac erat ornare, sed auctor eros lacinia. Donec faucibus euismod condimentum. Donec elit urna, consectetur sit amet felis sit amet, commodo iaculis massa. Sed non imperdiet augue. Duis gravida, urna id tempus congue, tortor tortor dictum dui, nec vulputate sem massa eu purus. Donec eget laoreet tellus, at mollis enim. Proin eu nibh nec neque hendrerit rutrum nec sed tellus. Vivamus dignissim neque sed augue viverra semper. Duis posuere ex sed justo iaculis consectetur.

Aliquam arcu erat, elementum fermentum maximus et, rhoncus et velit. Curabitur fringilla arcu sed nisl finibus, quis gravida mauris aliquet. Duis fermentum mi sed interdum congue. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur dolor velit, posuere vel aliquam ac, commodo ac tellus. Quisque eget ligula fermentum sapien euismod pretium. Pellentesque facilisis quam vel convallis pharetra. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut non purus bibendum, ullamcorper sem id, consectetur justo. Ut vehicula arcu quis leo pretium, tincidunt semper elit porttitor. Integer tincidunt lectus quis lorem auctor, quis bibendum leo venenatis.

Morbi at placerat dui. Praesent bibendum ullamcorper augue. Sed volutpat mauris et congue sodales. Donec vel dui non ante finibus aliquam. Nam accumsan nulla ut quam feugiat cursus. Integer et suscipit lorem. Nullam quis nisi justo. Nunc volutpat blandit dolor quis commodo. Nam ut massa non velit sagittis vehicula ut a nulla. Sed at euismod massa, at auctor neque. Ut id ullamcorper ligula. Cras ut finibus tortor.

6 Discussion

[TO FINISH]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac euismod sapien, ut varius metus. Maecenas vulputate lorem ac erat ornare, sed auctor eros lacinia. Donec faucibus euismod condimentum. Donec elit urna, consectetur sit amet felis sit amet, commodo iaculis massa. Sed non imperdiet augue. Duis gravida, urna id tempus congue, tortor tortor dictum dui, nec vulputate sem massa eu purus. Donec eget laoreet tellus, at mollis enim. Proin eu nibh nec neque hendrerit rutrum nec sed tellus. Vivamus dignissim neque sed augue viverra semper. Duis posuere ex sed justo iaculis consectetur.

Aliquam arcu erat, elementum fermentum maximus et, rhoncus et velit. Curabitur fringilla arcu sed nisl finibus, quis gravida mauris aliquet. Duis fermentum mi sed interdum congue. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur dolor velit, posuere vel aliquam ac, commodo ac tellus. Quisque eget ligula fermentum sapien euismod pretium. Pellentesque facilisis quam vel convallis pharetra. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut non purus bibendum, ullamcorper sem id, consectetur justo. Ut vehicula arcu quis leo pretium, tincidunt semper elit porttitor. Integer tincidunt lectus quis lorem auctor, quis bibendum leo venenatis.

Morbi at placerat dui. Praesent bibendum ullamcorper augue. Sed volutpat mauris et congue sodales. Donec vel dui non ante finibus aliquam. Nam accumsan nulla ut quam feugiat cursus. Integer et suscipit lorem. Nullam quis nisi justo. Nunc volutpat blandit dolor quis commodo. Nam ut massa non velit sagittis vehicula ut a nulla. Sed at euismod massa, at auctor neque. Ut id ullamcorper ligula. Cras ut finibus tortor.

6.1 Possible Improvements

[DONE]

Further improvements for this task could be made by changing the model used to per-

form sentiment classification, by perhaps using a neural network architecture. However, this would require much more than 2,000 examples as a testing and training set. Though, as movie review sentiment analysis is a well-researched task, much larger datasets, such as the one used to train `doc2vec` in this work, are available.

Without the need for additional data, the result could potential also be improved by using an ensemble of classifiers.

Further investigation could also be performed into the ideal parameters for training the `doc2vec` model, and whether a better model could be created than those used here. The performance of the `doc2vec` model could also potentially be improved by performing additional preprocessing on the training data and movie reviews. For example, removing stopwords or stemming the words in the review, and thus removing additional data that is unlikely to provide additional information as to the sentiment of the review. Additionally, some approaches to sentiment analysis have involved preprocessing such as changing a word in a review to demonstrate that it has been negated if preceded by 'not', so that, for example, the word 'good' is not considered to be positive in the phrase 'the film was not good' [3]. [CHECK THIS]

Implementing some or all of these techniques could allow for the accuracy achieved in this work to be improved.

Word count: 927 words

References

[1] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.

[2] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.