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

Jennifer White (jw2088)

November 8, 2019

Abstract

This works aims to improve on the success of work from the previous 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

This work examines the task of sentiment classification of movie reviews. It aims to investigate the accuracy of Support Vector Machines (SVMs) on this task, as well as investigating how the accuracy is affected by choices made about the model. Additionally, it investigates whether the use of doc2vec to create document vectors that can then be used as input to the SVMs can further improve the accuracy of the model, and which parameters and model choices create the most useful embeddings to use.

1.1 Previous Work

In the previous 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 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

The data used for 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 rest of the data.

2.2 Data for Training doc2vec

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

The code used for this work can be found on my user area on the MPhil machines (user:

`jw2088`) in the file `file`[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

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?

3.2 doc2vec

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

[DISCUSS DBOW VS DM]

4 Results

4.1 Statistically Significant Differences

5 Discussion

5.1 Possible Improvements

Word count: words

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

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

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.