NLP Assignment 2

SVM-based Sentiment Detection of Reviews

Anik Roy, Christ's (ar899)

December 3, 2019

Word Count: 999[1]

1 Introduction

Support vector machines are an ML model which can be used to classify vectors in a vector space. This method can be applied to the task of classifying documents by representing each document as a vector. In this report, I use a neural model, doc2vec, introduced by Mikolov and Le [2], in order to generate feature vectors. We also qualitatively show that the vector space produced by the doc2vec model is meaningful, by examining the behaviour of document embedding generation by doc2vec.

I use a large corpus of 100,000 movie reviews in order to train doc2vec, as well as a smaller set of 2,000 reviews (1000 of each classification), given in the framework of an NLP course.

2 Background

In a previous report, I used a bag of n-grams representation for documents (BOW), and used an SVM to classify these feature vectors. Each document is represented by a feature-count vector $(n_1(d), \ldots, n_m(d))$, with $n_i(d)$ being the number of occurrences of f_i in d. I also trained on a presence representation, setting $n_i(d)$ to 1 if f_i appeared in d, and 0 otherwise.

2.1 Support Vector Machines

SVMs are supervised learning models which are used to classify feature vectors in an n-dimensional space. Training consists of finding a hyperplane which separates the two classes with the largest margin. Classification takes place by measuring the distance of feature vectors to the plane.

[1]Using texcount

2.2 Doc2Vec

Doc2Vec is an unsupervised model for learning document embeddings which can be used as feature representations. It tries to overcome two flaws in BOW - word order is not being taken into account, and semantically similar words being equidistant. Doc2Vec produces fixed-length vectors for documents of any length. An important feature of these vectors is that they can't be directly interpreted.

Doc2Vec extends word2vec [4] , a method of learning word embeddings. There are two doc2vec architectures, distributed bag of words (DBOW) and distributed memory (dm).

3 Method

The doc2vec implementation used was gensim [6]. I trained the doc2vec model on 100,000 movie reviews from the Stanford Large Movie Review Dataset [3]. The SVM classifier is then trained on the documents found in the dataset used by pang et al. [5]. To tune parameters for the doc2vec model, I use a 10% validation set to evaluate different models, then report accuracies using 10-fold cross validation over the remaining 90%. I employed a nave search strategy, starting from the parameters used by Lau and Baldwin [1].

I compare the doc2vec based SVM model to two baseline svm models using a BOW representation.

4 Results

The results of the cross-validation (not using the validation set), show that the doc2vec representation is significantly better than either of the bag of words representations.

	BOW - freq.	BOW - pres.	Doc2vec
Doc2vec	0.014	0.002	-
BOW - pres.	0.001	-	
BOW - freq.	-		

Table 1: p-values, using a monte-carlo permuation test with $\alpha = 0.05$

	Representation	Accuracy
A	BOW - frequency	
B	BOW - presence	
C	Doc2vec	

Table 2: Accuracies, using ten-fold cross validation over 1800 reviews

5 Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6 Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

References

[1] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[2] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[3] 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.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques.

[6] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.