Using Support Vector Machines and Doc2Vec to Classify Movie Reviews

Tristram Newman (tssn2)

Much work has previously been done in the area of sentiment classification within natural language processing. As such, there exist a wide variety of techniques, however this study focuses on the use of support vector machines and Le and Mikolov's doc2vec system.

The doc2vec system is an extension/modification to Mikolov's original word2vec system, which was able to associate a vector to each word in a dataset. The dimensionality of these vectors could be chosen by the user, as they are not related to the features or context words. In fact, these vectors represent the hidden layer of a neural network used to learn from the dataset. The significance of doc2vec is that it allows for arbitrary sequences of words to be associated with such vectors, and is agnostic to granularity. The implication of this is that we can now have a very compact and concise representation of documents to feed into a classifier.

Two datasets were used in this study: one of 2000 labelled movie reviews (used for the SVM classifier), and one of 100000 unlabelled movie reviews (used to train doc2vec). Classification was performed in two ways: one using the standard bag-of-words model for the SVM classifier, and one using the doc2vec model. Parameters for the BOW classifier included: which n-gram(s) to use, whether or not to stem each word (using an implementation of the Porter stemmer) and whether to truncate frequencies above 1 to just 1 (i.e. frequency vs presence). Parameters for the doc2vec model included: stemming (again), the size of the context window and the number of epochs to perform (for the model to converge).

To gauge the optimal parameter set for both classifiers, the 2000-item dataset was split into two folds: a training set (90%) and a validation set (10%). For each parameter configuration, the classifier was trained on the 90% and then tested and scored on the other 10%. Naturally, the parameter configuration that produced the highest accuracy was chosen as the optimal. Then, with this configuration selected, 3-fold cross-validation was performed on the 2000-item dataset (2 folds for training, 1 for testing) and the accuracy was measured.

For the BOW model, the best parameter configuration was using both unigrams and bigrams, with stemming, and using presence instead of frequency. This produced an accuracy of 86%. For the doc2vec model, the best parameter configuration was using no stemming, a context window of 5, and 20 epochs for training. This produced an accuracy of 79%. The code for this study can be found at https://github.com/TriCrose/nlp.