Towards Higher Order Complexity Measures for Text Classification

KVS Dileep KVSDILIP@CSE.IITM.AC.IN
Sutanu Chakraborti SUTANUC@CSE.IITM.AC.IN
Indian Institute of Technology Madras, Chennai, 600036.

Abstract

We propose measures to estimate the complexity of classifying a document. We test different variations of our algorithm and measure the correlation with datasets of varying classification difficulty. We obtain promising results of high correlation with the classifiers, while overcoming some of the defects of previous measures.

1. Introduction

Consider classifying, say, documents from Economics vs Geography and IBM Hardware vs Mac Hardware. Intuitively we can see that it is easy to classify between Economics and Geography, while it is difficult to classify between IBM Hardware and Mac Hardware. Capturing the difference between these tasks with a quantitative measure is a non-trivial problem. A quantitative measure may help in deciding the right set of features and motivate to search for richer features, while classifying the given dataset. We try to come up with a reliable complexity estimator that overcomes the shortcomings of previous measures.

2. Alignment of Higher Orders

Current complexity measures take the dataset as a whole and measure the clustering tendency between document clusters and label clusters. An example of such measure is the Global Alignment MEasure(GAME)(Chakraborti et al., 2008). The tendency of clustering is explored in another complexity measure proposed in (Vinay et al., 2006). Another complexity measure proposed in (Massie et al., 2006) looks at the immediate neighborhood of the query document and estimates its complexity. We use this measure as the base measure as shown in Eq 1. But there arises a question as to how reliable the neighbors are in predicting complexity. Consider the example in Fig 1. The immediate neighbors and their labels predict the complexity to be low and thus easy to classify. But in reality, the classification with the current neighbor set would be wrong. Extending the neighborhood by including the neighbors of neighbors would give a more reliable estimate. And that motivates us to propose alignment of higher orders. Higher order neighbors refer to neighbors obtained through controlled expansion of neighbors from the query. First order neighbors

Algorithm 1 Algorithm for calculating alignment of higher orders

```
1. Given a collection of documents $C$, neighborhood order $n$ and the number of neighbors $k$
2. For each query $q$, do :
    (a) Find the neighbors of order n, denote it by $D$. There will be $N = k^n$ neighbors for the query.
    (b) For each neighbor $c \in k^n$ neighbors, find their alignment $align(c)$ using Eq 1. $DocSim()$ is cosine similarity.
    (c) $AlignVal(q) = \frac{\sum_{c \in D} align(c) * DocSim(q,c)}{\sum_{c \in D} DocSim(q,c)}$.
3. The mean of all the alignments is the alignment for the given dataset.
```

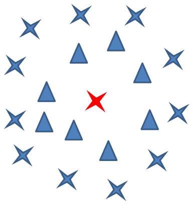

Figure 1. Example to motivate higher orders. Similar shapes correspond to similar labels.

Appearing in Proceedings of the 1^{st} Indian Workshop on Machine Learning, IIT Kanpur, India, 2013. Copyright 2013 by the author(s).

correspond to the immediate neighbors of the query document, second order neighbors correspond to the neighbors of the neighbors of query, and the same idea can be extrapolated to higher orders. If a neighborhood of size k is considered, then evaluating n^{th} order complexity would involve exploring k^n neighbors.

$$Align(q) = \frac{\sum_{c \in NN(q)} DocSim(q,c) * LabelSim(q,c)}{\sum_{c \in NN(q)} DocSim(q,c)} \quad (1)$$

Here $NN(q)$ refers to the neighbors of a query, $DocSim(c, q)$ is document similarity function(cosine similarity in our case), and $LabelSim(c, q)$ is label similarity function which returns 1 when labels are similar. Equation 1 measures the local complexity/alignment with a small neighborhood. The algorithm to calculate alignment of higher orders is described in Alg 1. It must be noted that alignment is the opposite of complexity, and hence more the alignment, lesser the complexity.

3. Experiments & Results

To show the performance of alignment of multiple orders, we took 6 datasets - Relpol, Hardware, Recreation , Science, Lingspam and Usremail. The first four datasets were constructed by dividing the 20 Newsgroups dataset into 4 categories based on the topics. The reason to divide such a way is to get datasets with a varying levels of classification difficulty. Our notion of difficulty is the accuracy on test set by various classifiers.

The experimental setup is as follows. We create disjoint sets, where each of the sets contain 20 % of the original corpus randomly chosen. For repeated trials, 15 such splits were created each containing 6 datasets of varying difficulty. We keep some documents aside as queries and use the rest of the corpus for calculating the higher order neighbors. Our aim is to show through experimentation that the proposed measure can predict the complexity of the dataset. This is shown by looking at the correlation between the calculated alignment and the accuracy of various standard classifiers. The alignment results across different trials for second order is shown in Fig 2. The graph clearly shows the distinction in the alignment values for different datasets.

The correlation results with the alignment scores for different classifiers are shown in Table 1. Different variations of Alg 1 have been tested on all the 15 sets. unwt1, unwt2 correspond to the unweighted version of first and second order, and wt1, wt2 correspond to the weighted versions. Weights correspond to the distance between the query. comb refers to combination of first and second orders, while prop refers to another variation where alignment propagates from higher to lower order neighbors and then to the query. It can be seen that prop version performs better, which is based on second order alignment.

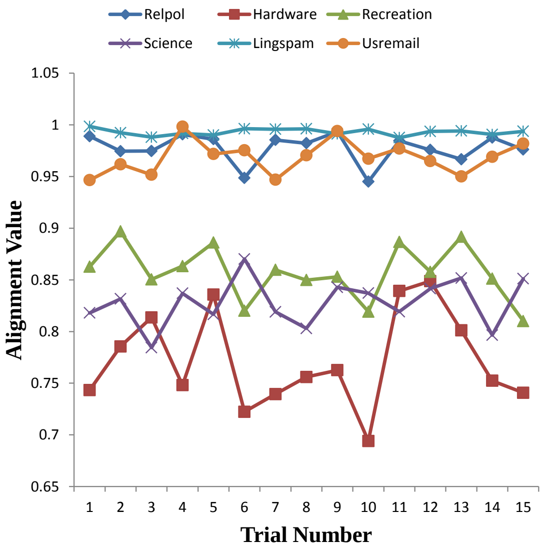

Figure 2. Second Order Alignment results across different datasets over 15 trials

	svm	nb	knn	crn	randforest
unwt1	0.95	**0.96**	**0.96**	0.65	0.80
wt1	0.95	0.96	0.96	0.64	0.80
unwt2	0.94	0.95	0.94	0.65	0.78
wt2	0.93	0.95	0.93	0.62	0.76
comb	0.95	0.96	0.95	0.65	0.79
prop	**0.95**	0.89	0.94	**0.75**	**0.91**

Table 1. Correlation values of the alignments scores with the classifiers - Support Vector Machine(svm) with linear kernel, Naive Bayes Classifier(nb),k-Nearest Neighbor(knn)with $k = 3$,Case Retrieval Net(crn),a spreading activation based classifier; Random Forest(randforest), an ensemble classifier for different datasets for one of the trials

4. Conclusion

We have proposed the alignment of higher orders as an attempt to achieve a more reliable complexity estimator. The results look promising, and we wish to investigate further into much higher orders of alignment. Convex combination of higher orders is another direction we wish to pursue along with the influence of higher order neighbors on lazy learning algorithms like knn.

References

Chakraborti, Sutanu, Beresi, Ulises Cervio, Wiratunga, Nirmalie, Massie, Stewart, Lothian, Robert, and Khemani, Deepak. Visualizing and evaluating complexity of textual case bases. In *Advances in Case-Based Reasoning*, volume 5239, pp. 104–119, 2008.

Massie, Stewart, Craw, Susan, and Wiratunga, Nirmalie. Complexity profiling for informed case-base editing. In *Proc of the 8th European Conf. on Case-Based Reasoning*, pp. 325–329, 2006.

Vinay, Vishwa, Cox, Ingemar J., Milic-Frayling, Natasa, and Wood, Ken. Measuring the complexity of a collection of documents. In *Proceedings of the 28th European conference on Advances in Information Retrieval*, ECIR'06, pp. 107–118, 2006.