An Analysis of the Coupling between Training Set and Neighborhood Sizes for the kNN Classifier

J. Scott Olsson
Dept. of Math., University of Maryland, College Park
olsson@math.umd.edu

ABSTRACT

We consider the relationship between training set size and the parameter k for the k-Nearest Neighbors (kNN) classifier. When few examples are available, we observe that accuracy is sensitive to k and that best k tends to increase with training size. We explore the subsequent risk that k tuned on partitions will be suboptimal after aggregation and re-training. This risk is found to be most severe when little data is available. For larger training sizes, accuracy becomes increasingly stable with respect to k and the risk decreases.

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: Miscellaneous

General Terms: Experimentation, Measurement

Keywords: text classification, k-Nearest Neighbors, parameter tuning, parameter stability

1. INTRODUCTION

Before applying text classification to real world problems, practitioners generally carve up the available labeled data to evaluate the system's accuracy and to tune classification parameters. At the same time, it is widely believed that the most consistent way to improve performance in general is to simply add more training data. Thus, real world systems evaluated and tuned on partitioned data typically aggregate all the available data and retrain before application. This calls into question the stability of the classification parameters being extrapolated onto the post-aggregation problem.

We consider the parameter k in the well known kNN classifier [2], where k is the number of training examples (neighbors) used to determine a test document's labels. kNN is conceptually simple, scales well [3], and has performed strongly on several well studied test corpora [1],[2].

Our aim is to understand whether the best choice of k is dependent on the training size, where we consider the optimal choice to be that k which produces the largest R-precision. R-precision is a well understood metric from the ranked retrieval community which, for the classification problem, can be seen as a measure of the utility of a ranked list of hypothesized labels for a user. Formally, R-precision is the average proportion of labels correctly assigned to a document, where a document with R true labels has R labels hypothesized by the classifier.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

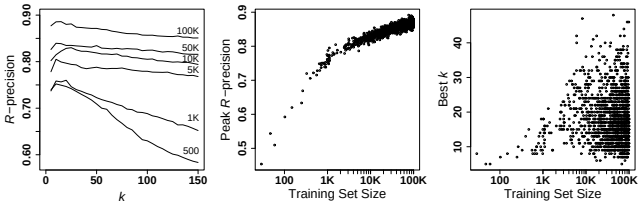

Figure 1: (a) R-precision *vs.* k for several example training set sizes; sizes are noted on curves. (b) Best R-precision and (c) best k from Setting II trials.

2. EXPERIMENTS

Our implementation of kNN uses symmetric Okapi term weighting, $w(tf) = \frac{tf}{0.5+1.5(\frac{dl}{avdl})+tf}$, where $w(tf)$ is the computed term weight, tf is the term frequency of a word in a document, dl is the length of the document the term appears in, and $avdl$ is the average document length. Term weights are multiplied by their inverse document frequency, $idf(t) = \log\left(\frac{N-df(t)+0.5}{df(t)+0.5}\right)$, where $df(t)$ is the document frequency of term t and N is the total number of training documents. The label scores contributed from the k nearest neighbors are weighted by the inner product of the test and neighbor document vectors.

Our data consists of 200K documents from the RCV1-v2 newswire corpus [1] which we examine in two distinct settings. Setting I considers 500 trials at each of the fixed sizes: 100, 500, 1K, 10K, 50K training documents randomly sampled. This narrow view allows us to examine variance in best k. In Setting II, for each of 1500 trials, we sample D training documents, where D is a random variable uniformly distributed between 1 and 100K. This broad view hopes to reveal trends over a range of set sizes. In both settings, 1K disjoint testing documents are sampled per trial.

For each trial, we search for the optimal k: beginning at $k = 1$, we increment k by one until fifteen *consecutive* incrementings have failed to improve upon the best R-precision yet seen. This approach requires the R-precision *vs.* k curve to be somewhat smooth, which we experimentally validated and illustrate in Figure 1a. R-precision tends to monotonically increase, peak, and then decrease as k increases—a trend exhibited regardless of the training set size. Our re-

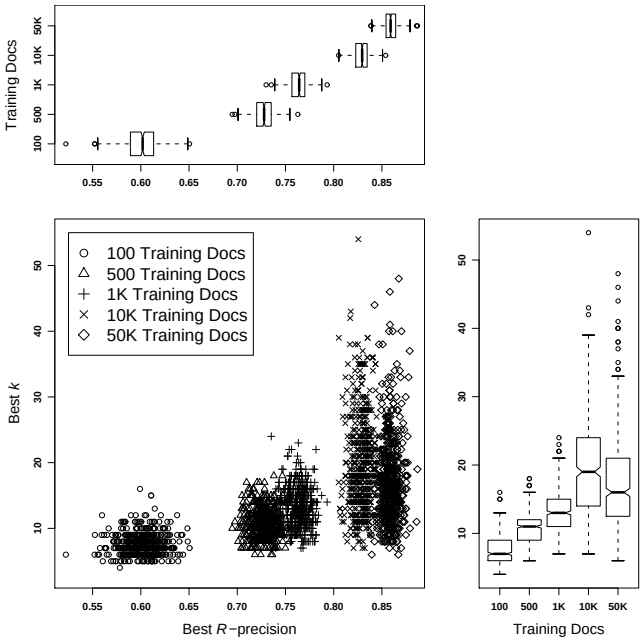

Figure 2: Setting I trials. Top and side boxplots show R-precision and k for each training size. Boxplot notches show 95% CIs for the mean.

quirement that fifteen consecutive increments of k fail to improve the R-precision additionally mitigates the risk that a particular curve will peak in R-precision a second time. We also observe from Figure 1a that curves become increasingly flat as set size increases; that is, for larger training sets, the *risk* of poorly choosing k decreases. Figure 1b plots the peak R-precision obtained for each Setting II trial *vs.* training set size. Observe that, after roughly 1K training documents, increasing R-precision by 10% requires the training set be enlarged by a factor of nearly 100. The need for so much data to improve R-precision motivates the common practice of aggregating all available data and retraining before the classifier is applied. R-precision increases faster below 1K training documents. At the same time, with fewer documents, the risk of poorly choosing k is increased.

Figure 2 plots best k *vs.* R-precision for each of the Setting I trials. Trials with equal amounts of training data form clusters, and, unsurprisingly, the average best R-precision improves (with statistical significance) for each increase in training set size. Note that the mean optimal k, as well as the variance in optimal k, also increases at each step from 100 to 10K training documents, before dipping again at 50K. Suppose only 1K labeled documents were available for an evaluation. Figure 2 roughly tells us that if we partition those 1K documents into two equally sized training and testing pools, search for an optimal k using the 500 training documents, and then aggregate all 1K documents for a real problem using that same k, this k will typically be much smaller than the true optimal k for the aggregated training set. And as we saw in Figure 1a, small deviations from optimal k can result in large deviations from best R-precision on small training set problems. Unfortunately, this deficiency will go undetected as, in such a case, we would have no labeled data left for further evaluation.

We suspect the optimal k at first increases because, for very little training data, a large k means a large proportion of training documents will be used for labeling (i.e., the smoothing will be very aggressive); accordingly, the best k (and the variance in best k) must be small. As the training set size increases initially, this small data problem is relaxed, and the best k tends to increase, dependent on some other property of the training data (e.g., perhaps the separability of documents having distinct labels). Variance in best k likely also increases because, as seen in Figure 1a, as training sizes increase, near peak R-precision is sustained over a broader range of k. On the other hand, if our training space were to be densely populated by example documents, we might expect the optimal k to be roughly $k = 1$. That is, if each test document had an identical document in training, that one nearest document in the training space would presumably have the appropriate labels attached to it (inconsistencies in human judgments might increase this limiting k slightly). However, because the number of documents needed to densely fill the space grows exponentially in the number of features (i.e., dimensions) and because we would expect the document space to be "semantically anisotropic" (i.e., more "meaning distance" is traversed along some dimensions than others), this theoretical behavior of best k will never be observed in real, high-dimensional, problems.

To determine the extent to which k may continue to depend on training set size for larger set sizes, we considered trials from Setting II, in which up to 100K training documents per trial were investigated. As before, we found that best k increases with training size for smaller set sizes. For larger set sizes, however, k and training set size appear to be completely uncorrelated. It is therefore unclear whether aggregating large amounts of training and testing data poses a risk due to the instability of best k (it is possible that the pre-aggregation training data will sufficiently effect the aggregated statistics so as to prevent the optimal k from drifting far). Future work could include experimental trials of this scenario. At the least, it is clear that best k could only remain fixed through aggregation if the yet unknown properties which determine best k themselves remain mostly unchanged. This suggests future work might explore the effects of partitioning strategies on parameter stability.

3. CONCLUSION

We have seen that we must be cautious to assume classification parameters tuned using a partition of available data will be optimal after aggregation. The choice of k can significantly depend on training size, particularly for problems with little available data. When more training data is used, it remains unclear what principally determines the optimal choice of k, although we have seen this is generally of little concern since R-precision becomes increasingly stable with respect to k as training sizes increase.

4. ACKNOWLEDGMENTS

Thanks to Douglas W. Oard for his helpful feedback. This work was funded in part by NSF IIS 0122466 (MALACH).

5. REFERENCES

[1] D. D. Lewis, et al., RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5, 2004.
[2] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1, 1999.
[3] Y. Yang, et al., A Scalability Analysis of Classifiers in Text Categorization. In *SIGIR*, 2003.