Learning Query Bias for Improved On-Line Document Classification

Ron Papka
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, MA 01003
papka@cs.umass.edu
http://www.cs.umass.edu/~papka

The following work describes two threshold selection algorithms for the document tracking task. The tracking task was defined by the Topic Detection and Tracking (TDT) research initiative, which is a DARPA sponsored effort comprising research groups from several commercial and academic sites. The task is a form of supervised learning and it is analyzed based on a system's ability to classify a test set of documents as relevant or non-relevant to a set a pre-specified topics derived from news events. This task is similar in nature to the document filtering track for the Text Retrieval Conferences (TREC) with three salient exceptions: 1) the information domain is a temporally ordered stream of broadcast news ; 2) solutions to tracking must make classification decisions on-line; and 3) classifiers are trained using only 1,2,4,8, or 16 relevant training documents from each topic. Much of the related research in this area has been explored using TREC's filtering methodology which focuses on training classifiers for general topics with one to a few thousand relevant training documents from heterogeneous sources including news, web, email, and other types of documents. What has emerged as the classifier of choice in the research is a query represented as a vector of features comprising stemmed words and associated weights (Figure 1). Query words and weights are determined through statistical and learning techniques utilizing inter- and intra-document word occurrences in the training data. The most common document representation is a vector of *tf*idf* weights co-occurring with the words in the query. Queries are compared to documents using a similarity function, which in many instances, is an extension of the inner-product of two vectors. This representation has the nice property of being extensible to linear classification using both supervised and unsupervised learning approaches. In what follows, we define and illustrate threshold estimator bias for document classification based on this representation. We discuss two methods that produce bias-reduced estimators resulting in improved classification accuracy.

```
q24= #WSUM( 1  0.69 crash   0.49 usair   0.49 plane   0.40 pittsburgh   0.38 flight   0.38 investigate   0.34 area
0.31 737   0.30 427   0.29 airport   0.26 wood   0.26 board   0.25 cause   0.24 safety   0.24 scene   0.24 site
0.23 piece   0.22 aircraft   0.22 pennsylvania   0.21 even);
```

Figure 1: 20 feature query formulated from the first 16 relevant documents from "Crash of US Air Flight 427"

Recently, tracking systems from various research sites were tested on the TDT2 evaluation corpus (~22K documents, 30 topics) [1]. Our system compared favorably to the best systems which used similar information retrieval representations to the one described above. An important component of our tracking system is a threshold estimator for queries formulated for the Inquery retrieval engine. We used an estimator $\hat{u}$ such that $\hat{u} = 0.4 + \theta (b_{optimized} - 0.4)$, where θ is a global system parameter, 0.4 is the minimum belief value produced by Inquery , and $b_{optimized}$ is the belief value resulting from the query that, when applied to the topic's labeled training documents, optimizes a pre-specified utility function. The utility function used for this evaluation was defined by TDT and is a cost function based on error rates for false positives and false negatives produced by a system.

Using data from several experiments on the TDT1 corpus (~16K docs, 25 topics) and the TDT2 train and development corpora (~40K docs, 66 topics), it was determined that when fewer relevant training documents were used, $b_{optimized}$, our estimator $\hat{u}$ when $\theta = 1.0$, was consistently below the parameter u it was trying to estimate i.e. the optimal threshold for the testing data of a particular topic, or simply $b_{optimal}$. In what follows, we assume the bias in our estimator $\hat{u}$ is the quantity $b(\hat{u}) = E[\hat{u}] - u$.

We observed, as we tested various values for θ, that bias decreased when more relevant training documents were used to formulate queries. We also observed similar but less significant increases in bias when more features were used. The observation that increasing training instances reduces the bias of an estimator, in general, is not surprising. James, for example, proves that estimates move toward the true population values when training instances are increased for data assumed to have multivariate-normal distributions [2]. He also proposes that once found, it may be possible to reduce the bias using a linear transformation. We test James's theory in the following experiments. We define a threshold estimator $\hat{o}$, such that $\hat{o} = b_{optimized.}$ We then define a new bias-reduced threshold estimator $\hat{e}$, such that $\hat{e} = m\hat{o} + b$. We then compare the effectiveness of the query thresholds produced by our original estimator $\hat{u}$, which is bias-reduced using near optimal values for θ, to those produced by $\hat{e}$, with parameters m and b learned through linear regression. In what follows, we explain these two methods that learn query bias for estimator $\hat{o}$.

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

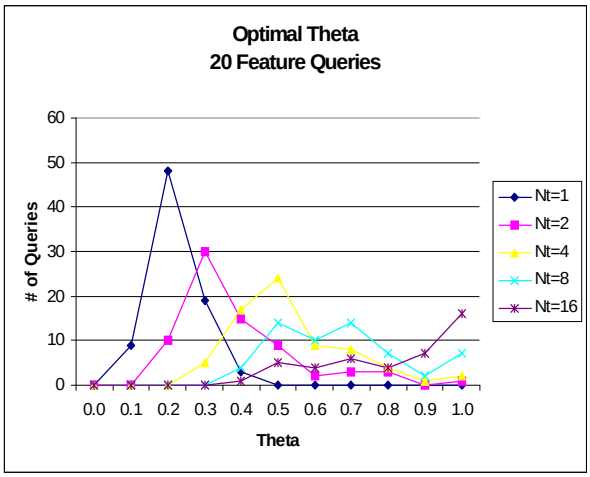

Figure 2: Histogram approach for estimator û.

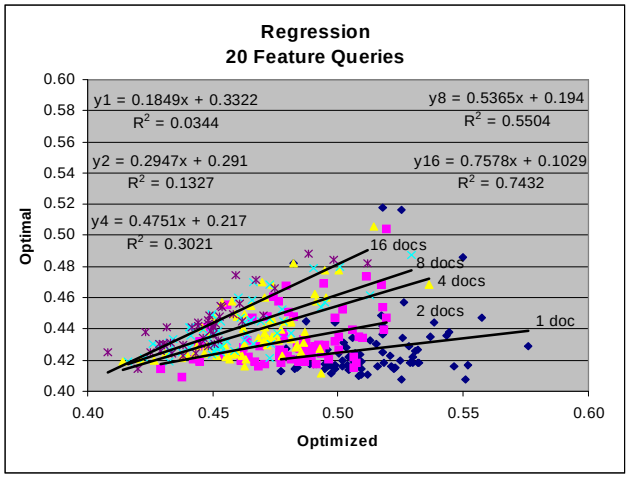

Figure 3: Regression approach for estimator ê.

Both approaches trained on 91 topics and associated labeled documents from the TDT1 and TDT2 train and development corpora. Training and testing are conducted independently over varying numbers of relevant training documents (Nt) and varying numbers of query features. The first method, which utilizes threshold estimator û, is illustrated using 20 word queries in Figure 2. Histograms of optimal values for θ are collected for each value of Nt and for queries formulated with 10, 20, 50,100, 200, 600, and 10000 features. Using this method on the training data determined that for 1 relevant document (Nt=1) and 20 feature queries, 49 out of 91 queries had optimal cost when θ = 0.2. For each pair of Nt and number of features, we calculate E[θ] from the corresponding histogram, and use that value as the parameter for estimator û when conducting evaluation experiments using the same pair. From the data in Figure 2 it is evident that as Nt increases E[θ] increases. This observation coupled with the definition of estimator û implies that $b_{optimized}$ is closer to $b_{optimal}$ when more relevant training examples are used.

The linear regression method used to train estimator ê is illustrated in Figure 3. Instead of collecting histograms, points represents by $b_{optimized}$ and $b_{optimal}$ are fitted using a line for each value of Nt and number of features . The slopes and intercepts of lines produced by the various regressions are subsequently used as parameters m and b for estimator ê. This method learns the same tendencies in threshold estimator bias as the approach for estimator û. As the number of relevant training documents increases, the slope of the resulting regression line approaches 1. As the slope approaches 1, $b_{optimized}$ approaches $b_{optimal}$ which implies that higher values of Nt give rise to less estimator bias in the training data.

TDT2 Evaluation corpus (30 events)					
	# of Relevant Training Documents (Nt)				
# of Features	1	2	4	8	16
10	8.2%	-20.4%	-13.2%	-21.9%	-2.9%
20	27.5%	-17.0%	-32.1%	-23.4%	-1.6%
50	5.0%	53.2%	0.0%	-4.0%	-8.6%
100	1.1%	12.7%	4.5%	-6.0%	-20.3%
200	3.3%	-7.5%	-8.2%	2.0%	-16.4%
600	27.8%	-6.0%	-7.3%	-8.0%	-1.8%
10000	27.8%	-6.0%	-7.3%	-8.0%	0.0%
Average	14.4%	1.3%	-9.1%	-9.9%	-7.4%

Figure 4: % Cost increase caused by replacing estimator û with estimator ê.

From the recent TDT experiments evaluated by NIST, we knew estimator û generalized and provided relatively low cost on the TDT2 evaluation corpus, so in the following experiment, we use it as a baseline to compare the effectiveness of estimator ê. We used the same query formulation and tracking processes and measured the percent increase in cost resulting from tracking with estimator ê instead of estimator û. Since we are trying to minimize average cost across a set of queries, an increase in cost implies a decrease in classification accuracy. The results are listed in figure 4. They suggest that the histogram method works better for Nt=1 and marginally so for Nt=2. However, for Nt > 2 the regression method appears to reduce cost consistently for most of the query sizes tested. This relationship suggests a mixture of approaches where estimator û is used for Nt <= 2 and estimator ê otherwise. We are, however, still evaluating these data, but we anticipate that estimator û is not significantly more effective than estimator ê and vice versa, which would suggest that both methods are likely to do well on the currently available TDT corpora. We plan to continue investigating these approaches as other data sources with document assessments become available.

References:

[1] To appear in *Proceedings of the 4th Annual DARPA Broadcast News Workshop*, February 1999.
[2] Mike James, *Classification Algorithms*, John Wiley & Sons, New York, 1985.