DIMACS Technical Report 2005-01
January 2005

Exploration Approaches to Adaptive Filtering

by

Dmitriy Fradkin
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08854

Michael Littman
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08854

DIMACS is a collaborative project of Rutgers University, Princeton University, AT&T Labs–Research, Bell Labs, NEC Laboratories America and Telcordia Technologies, as well as affiliate members Avaya Labs, HP Labs, IBM Research and Microsoft Research. DIMACS was founded as an NSF Science and Technology Center.

ABSTRACT

Exploration-exploitation trade-off is particularly important in the adaptive filtering setting, where feedback is received only on submitted documents and, therefore, is scarce. We examine the effects of certain exploration techniques on the performance of simple linear classifiers.

1 Introduction

Adaptive filtering is an important task for real-world systems that process thousands and millions of documents. It is possibly the only reasonable way to train systems for which it would be impractical to provide feedback on every example. It also matches the interactions that customers are likely to have with the system - they will provide feedback only on (some of) the documents that they see.

The adaptive filtering setting makes exploration-exploitation trade-off particularly important. Since the feedback is received only on submitted documents and, therefore, is scarce, the system must balance between obtaining sufficient amount of feedback to obtain a good model of user interests without overwhelming the user with “false alarms” and irrelevant information.

We examine several simple linear classifiers and the effect of certain exploration techniques. We examine effect of certain exploration techniques (deterministic, random, threshold-based and model-based exploration) on performance of simple linear classifiers.

2 Linear Classifiers

Given a vector representation d_i for a document i, a linear classifier labels the i-th document as relevant to category j iff d_i satisfies:

$$(d_i, w_j) + b \geq 0, \tag{1}$$

where w and b are the parameters of a classifier.

Linear classifiers are easy to interpret (each feature can be seen as having its own weight and affecting the decision accordingly). Below we discuss two popular linear classifier schemes that have an added bonus of being easy to train online/adaptively.

2.1 Multinomial Naive Bayes (MNB) Classifier

MNB is based on a generative model of text documents. We follow McCallum and Nigam [4] in our description of this method. Each document is viewed as a “bag of words”, with probability of each word in a document relevant for a topic independent of the others.

$$P(d_i|c_j;\theta) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j;\theta)^{N_{it}}}{N_{it}!}. \tag{2}$$

The model θ can be estimated by:

$$\hat{\theta}_{w_t|c_j} = P(w_t|c_j;\theta) = \frac{1 + \sum_{i=1}^{|D|} N_{it} P(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N_{is} P(c_j|d_i)}. \tag{3}$$

Given the estimates of the parameters, classification is performed by computing posterior probability of the class:

$$P(c_j|d_i;\hat{\theta}) = \frac{P(c_j|\hat{\theta})P(d_i|c_j;\hat{\theta}_j)}{P(d_i|\hat{\theta})} \tag{4}$$

Therefore:

$$\log P(c_j|d_i;\hat{\theta}) = \log P(c_j|\hat{\theta}) + \sum_{t=1}^{|V|} N_{it} \log P(w_t|c_j;\theta) + C \tag{5}$$

$$= \log P(c_j|\hat{\theta}) + (d_i, \hat{\theta}_j) + C \tag{6}$$

where C is a category independent (document specific) value.

A decision regarding relevance of d_i is based on the sign of:

$$\log P(c_j|\hat{\theta}) + (d_i, \hat{\theta}_j) - \log(1 - P(c_j|\hat{\theta})) - (d_i, \hat{\theta'}_j) = (d_i, \hat{\theta}_j - \hat{\theta'}_j) + (\log P(c_j|\hat{\theta}) - \log(1 - P(c_j|\hat{\theta}))) \tag{7}$$

Thus we can view MNB as a linear classifier, where $w_j = \hat{\theta}_j - \hat{\theta'}_j$ and $b = \log P(c_j|\hat{\theta}) - \log(1 - P(c_j|\hat{\theta}))$.

2.2 Centroid Method

Let $v(S) = \sum_{x \in S} x$. Also, let D^+ be a set of relevant documents, while D^- is the set of not relevant documents for a category. In the centroid method the score of a document is computed as:

$$s(d_i) = cos(d_i, v(D^+)) - cos(d_i, v(D^-)) \tag{8}$$

$$= \frac{(d_i, v(D^+)}{||d_i||||v(D^+)||} - \frac{(d_i, v(D^-))}{||d_i||||v(D^-)||} \tag{9}$$

$$= (\frac{d_i}{||d_i||}, \frac{v(D^+)}{||v(D^+)||} - \frac{v(D^-)}{||v(D^-)||}) \tag{10}$$

So if we represent a document by a unit-length vector in the same direction ($\frac{d_i}{||d_i||}$), the centroid method is linear, with $b = 0$ and $w = \frac{v(D^+)}{||v(D^+)||} - \frac{v(D^-)}{||v(D^-)||}$.

3 Experimental Setup

3.1 Data

In our experiments we used a well-known Reuters-21578 dataset []. The ModApte training test split has 9603 training and 3299 test documents. We selected 10 topics with largest number of relevant documents in the data and built binary classifiers for each of these topics.

Topic	Number of relevant documents
acq	2369
corn	241
crude	578
earn	3964
grain	582
interest	478
money-fx	717
ship	286
trade	487
wheat	283

Table 1: Topics used in experiments

In all the experiments, unless otherwise specified, documents were represented with "bag of words" model, using raw term frequency. In other words, a documents is seen as a vector with the weight of a word equal to the number of times the word occurs in the document.

3.2 Evaluation

Let T be the test set. Let T^+ be the set of all positive documents for a query in the test set. Let D be a set of all submitted documents, and let D^+ be the set of positive documents retrieved, and D^- be the set of nonrelevant documents retrieved. $|S|$ denotes the number of elements in a set S. The F1 measure is described by the following formula:

$$\mathrm{F1} = \frac{2|D^+|}{|D^+| + |D^-| + |T^+|} = \frac{2|D^+|}{|D| + |T^+|}$$

We used this measure to evaluate all of our results.

4 Exploration Methods

4.1 "Apple Tasting"

In the [3], the authors suggest 2 methods for transforming online algorithms into adaptive filtering ones with some theoretical guarantees on the performances of the transformations.

One method, STtoAP is applicable when the number of steps T is known in advance. This methods takes parameter M, intended to be a bound on the number of false negative mistakes, and works as follows:

- Execute the online algorithm A on d_i
- If A suggests submitting d_i, submit

- else, draw a random number r. Submit if $r \leq \sqrt{M/T}$.

If we do not know T in advance, this approach can be seen as ϵ-greedy exploration, with $\epsilon = \sqrt{M/T}$.

Another transformation method, STAP, works as follow:

- Initialize $m = 0$.
- Execute the online algorithm A on d_i
- If A suggests submitting d_i, submit
- else, draw a random number r.
- Submit if $r \leq \sqrt{(m+1)/i}$. If the submitted document is relevant, increment m.

Note that this algorithm is parameter free.

Both of the above algorithms are examples of random exploration - the decision a document that would otherwise be rejected is made without considering any property of the document itself.

4.2 Margin-Based Approach

In [2], the authors suggest a method for incorporating exploration into a linear classifier, by changing the intercept. A document is submitted iff

$$(d_i, w_j) + b + \tau \geq 0, \tag{11}$$

where $\tau = -\sqrt{(5 \ln i)/N_{i-1}}$ and N_{t-1} is the number of submitted documents among the first $i-1$.

This method also disregards any document-specific information. The exploration is achieved by decrease of a threshold that takes place when few relevant documents are found. Thus this is a deterministic exploration method.

4.3 Hoeffding-Bound Based Approach

4.3.1 Computing Similarity Bound

Let m be a true distribution over a finite vocabulary V, and let M be the empirical estimate of m based on N i.i.d. observations. (Note that since m, M are distributions, it follows that $||m||_1 = 1$, $||M||_1 = 1$.). Weissman et.al. [5] give the following bound:

$$P(||m - M||_1 > \epsilon) \leq (N+1)^{(|V|-1)} e^{(-N\epsilon^2)/2}. \tag{12}$$

Therefore, if we set $\delta = (N+1)^{(|V|-1)} * e^{(-N\epsilon^2)/2}$, we arrive at the following formula for epsilon:

$$\epsilon = \sqrt{(2/N) * (-\log(\delta) + (|V|-1)\log(N+1))} \tag{13}$$

We are interested in a bound on inner product of m and M. Using the fact that for any $x \in \Re^d$:

$$\frac{||x||_1}{\sqrt{d}} \leq ||x||_2 \leq ||x||_1 \tag{14}$$

and that:

$$||m - M||_2 = \sqrt{||m||_2 - 2(m, M) + ||M||_2} \tag{15}$$

we have:

$$\delta \geq P(||m - M||_1 > \epsilon) \geq P(||m - M||_2 > \epsilon) \tag{16}$$
$$= P(||m||_2 - 2(m, M) + ||M||_2 > \epsilon^2) \tag{17}$$
$$= P(\frac{1}{2}(||m||_2 + ||M||_2 - \epsilon^2) > (m, M)) \tag{18}$$
$$= P(\frac{1}{2}(||m||_2 + ||M||_2 - \epsilon^2) > \frac{cos(m, M)}{||m||_2||M||_2}) \tag{19}$$
$$= P(\frac{||m||_2||M||_2}{2}(||m||_2 + ||M||_2 - \epsilon^2) > cos(m, M)) \tag{20}$$
$$\geq P(\frac{||M||_2}{2\sqrt{|V|}}(\frac{1}{\sqrt{|V|}} + ||M||_2 - \epsilon^2) > cos(m, M)) \tag{21}$$

Let $\epsilon_c = \frac{||M||_2}{2\sqrt{|V|}}(\frac{1}{\sqrt{|V|}} + ||M||_2 - \epsilon^2)$. If a confidence interval $(1 - 2\delta) * 100\%$ is specified, we have $P(cos(m, M) < \epsilon_c) \leq \delta$.

4.3.2 Computing An Optimistic Model

Given ϵ_c, a document d_i and a model M (both now scaled to unit vectors) we can compute an "optimistic" model m, that maximizes $cos(d_i, m)$ subject to:

$$cos(m, M) > \epsilon_c \tag{22}$$

This is done as follows. Suppose $cos(d_i, M) \geq \epsilon_c$. Then d_i is within "exploration region" of M, and therefore the optimistic score is of d_i is 1 (i.e. $m = d_i$ and $cos(d_i, m) = 1$).

Suppose $cos(d_i, M) < \epsilon_c$. Then m lies between M and d_i:

$$m = aM + bd_i, \; a, b > 0 \tag{23}$$
$$(m, m) = 1 \tag{24}$$
$$(m, M) = \epsilon_c \tag{25}$$
$$\tag{26}$$

Let $s = (M, d_i)$. Combining 1st and 3nd of the above equations (keeping in mind that M and d_i are unit-length), we have:

$$a = \epsilon_c - bs \tag{27}$$

Combining 1st and 2nd:

$$a^2 + 2abs + b^2 = 1 \tag{28}$$

Substituting previous into this, we have:

$$\begin{aligned} 1 &= (\epsilon_c - bs)^2 + 2(\epsilon_c - bs)bs + b^2 &(29)\\ &= {\epsilon_c}^2 - 2\epsilon_c bs + (bs)^2 + 2\epsilon_c bs - 2(bs)^2 + b^2 &(30)\\ &= {\epsilon_c}^2 - b^2(s^2 - 1) &(31) \end{aligned}$$

And so:

$$b = \sqrt{\frac{1 - {\epsilon_c}^2}{1 - s^2}} \tag{32}$$

4.3.3 Practical Considerations

Notice that exact Hoeffding bounds can be very loose. Unless $|V|$ is much smaller than N and δ is large we are going to have $\epsilon > 1$ and therefore $\epsilon_c < 0$, implying that exploration region will include more than half of the space.

For text classification and filtering problems where vocabulary size grows rapidly with the number of documents (until reaching some saturation point). Our experiments, results of which are not included here, showed that using exact Hoeffding bounds leads to very poor performance.

4.4 Cosine Exploration

In the Hoeffding bounds method, we specify a confidence interval via δ and then compute a lower bound ϵ_c on similarity of models that we are willing to explore. A more direct approach, though one without a theoretical support is to specify ϵ_c as a parameter to the exploration method. An optimistic model is then computed as above.

This can allow us to avoid the problem of large exploration region that we would have when using the exact Hoeffding bounds.

Note that both Hoeffding-based and cosine exploration explicitely explore the space of models in the direction of a current document. This makes them significantly different from random exploration of "apple tasting" methods.

4.5 Threshold Exploration

In the previous section we described random exploration (exemplified in STtoAP and STAP) and model-based exploration, where uncertainty is allowed in the whole model.

Another method of exploration involves adjusting the threshold for accepting a document. While margin-based exploration (FIL) could be seen as threshold exploration, it is better viewed as a wrapper method that decreases existing threshold by some amount depending on number of relevant and non-relevant documents submitted.

A more sophisticated threshold exploration method would directly model our uncertainty about the current threshold or border of relevant score region.

In order to examine such methods we first need a method for setting the thresholds in a classifier. Many such methods are described in the literature.

The DIMACS method [1] recomputes scores of documents in history using the latest model and find the optimal threshold. (This approach is clearly expensive in memory and time, since whole documents have to be keep and rescored).

Other methods keep only scores of the documents in history.

Yang et al [7] use regression to fit a line to the positive and negative scores in history and then interpolate between these lines to set the threshold. This basic scheme is refined to account for situations with insufficient data or for values of threshold that are too high or too low.

Zhang et. al [9] modeled distribution of relevant scores with a Gaussian and that of non-relevant documents with and exponential distribution. After taking into account sampling bias they found Maximal Likelihood estimates of the parameters of distribution and solved for optimal threshold.

Zhai et. al [8] used beta-gamma method. The threshold is found by interpolating between optimal threshold estimate t_{opt} and zero-threshold estimate t_0 using several parameters:

$$\begin{aligned} t &= at_0 + (1-a)t_{opt} \\ a &= \beta + (1-\beta)\exp(-\gamma N) \end{aligned} \tag{33}$$

where β and γ are parameters and N is the number of available judged examples.

Some methods are extremely simple and reactive in nature. CAS [6] starts with fixed initial threshold values and adjusts them (by δ) only if too many negatives are submitted or if no submissions are made for a long time. Thus it only keeps track of its performance and the number of documents seen since the last submission.

$$t_C(i) = \begin{cases} t_C(i-1) + \delta, & \text{if utility is negative after a submission} \\ t_C(i-1) - \delta, & \text{if } z_{i-1} = L \\ t_C(i-1), & \text{otherwise} \end{cases}$$

where z_i is a counter, keeping track of the number of documents seen since the last submission:

$$z_i = \begin{cases} z_{i-1} + 1, & \text{if } d_{i-1} \text{ is not submitted} \\ 0, & \text{if } z_{i-1} = L \text{ or } d_{i-1} \text{ is submitted} \end{cases}$$

and L, the maximum length of period of no submission, is a parameter.

5 Random and Model-Based Exploration Experiments

5.1 MNB with Training Stage

The first set of experiments involved MNB classifier, which was trained on the training part and then evaluated (in different modes) on the test part. Such set up is likely to favor

Topic	Batch	Online	MNB	STtoAP				STAP	MNB-FIL
ϵ				0.01	0.05	0.1	0.3		
acq	0.826	0.853	0.840	0.823	0.773	0.725	0.572	0.733	**0.842**
corn	0.417	0.431	0.412	0.337	0.208	0.152	0.072	0.224	0.405
crude	0.808	0.838	0.839	0.770	0.607	0.476	0.257	0.659	0.831
earn	0.930	0.945	0.939	0.928	0.896	0.861	0.731	0.887	0.938
grain	0.736	0.757	0.738	0.668	0.504	0.388	0.207	0.512	0.735
interest	0.683	0.648	0.667	0.597	0.445	0.337	0.169	0.506	0.656
money-fx	0.703	0.719	0.716	0.667	0.540	0.432	0.243	0.591	0.710
ship	0.671	0.773	0.724	0.600	0.378	0.276	0.130	0.457	**0.729**
trade	0.529	0.553	0.551	0.497	0.361	0.276	0.145	0.407	0.532
wheat	0.576	0.616	0.573	0.480	0.321	0.224	0.106	0.350	0.564
Average	0.688	0.713	0.700	0.637	0.503	0.415	0.263	0.533	0.694
St. dev	0.021	0.021	0.022	0.028	0.040	0.045	0.042	0.034	0.023

Table 2: Results of MNB experiments with Training Stage

approaches with little exploration, since there is sufficient training. Results are in Table 2.

Note that, as expected Online does better than Adaptive which does better than Batch. STtoAP and STAP exploration performs worse than no exploration (MNB). However, MNB-FILE beats MNB on 2 topics, and is very close on the rest.

In exploration columns (STtoAP, STAP and MNB-FIL) the numbers in bold are where exploration beats greedy MNB. This happens only with MNB-FIL for "acq" and "ship" topics. The "acq" is the topic with most relevant documents, while "ship" is one of those with fewest. In general, exploration only hurt the performance (possibly because values of ϵ were too high).

5.2 Exploration with MNB Filtering

We programmed our classifiers (MNB and Centroid), when in Adaptive or Online mode, to submit all documents until two positive examples are obtained. This was done to provide some initial positive documents for model training.

Results of the experiments are in Table 3. Surprisingly, on some topics ("money-fx" and "trade"), MNB does extremely poorly. The reasons for that are unclear. Again, the numbers in bold are where exploration beats greedy MNB.

Here Online results are again better than any adaptive filtering results, which is to be expected. MNB-FIL performs comparably to MNB on all topics beating it slightly on some, and doing a bit worse overall.

However, STAP and STtoAP methods (in particular $\epsilon = 0.05$ and $\epsilon = 0.1$) show different behavior. They perform worse on the topics where MNB does well (acq, crude, earn), but much better where MNB does poorly (trade, money-fx, interest), especially for higher values of ϵ. These results suggest that the right amount of exploration may be extremely useful

Topic	Online	MNB	STtoAP						STAP	MNB-FIL
ϵ			0.0001	0.001	0.01	0.05	0.1	0.3		
acq	0.794	0.766	0.766	0.766	0.766	0.722	0.673	0.496	0.748	**0.769**
corn	0.493	0.312	0.293	0.291	0.174	0.122	0.090	0.068	0.241	0.282
crude	0.717	0.617	0.609	0.616	0.578	0.438	0.328	0.189	0.483	0.602
earn	0.915	0.885	0.885	0.884	0.881	0.854	0.819	0.695	0.868	**0.887**
grain	0.772	0.665	0.657	0.658	0.634	0.504	0.334	0.198	0.602	0.658
interest	0.528	0.222	0.222	0.218	0.165	**0.275**	0.216	0.134	**0.307**	0.212
money-fx	0.664	0.004	0.004	0.004	**0.142**	**0.394**	**0.294**	**0.215**	**0.424**	0.004
ship	0.668	0.552	0.552	0.552	0.476	0.269	0.161	0.090	0.219	0.540
trade	0.515	0.078	0.078	0.071	0.062	**0.226**	**0.179**	**0.129**	**0.217**	**0.080**
wheat	0.625	0.507	0.519	0.520	0.336	0.279	0.170	0.091	0.241	0.498
Average	0.669	0.461	0.459	0.458	0.421	0.408	0.326	0.231	0.435	0.453
St. Dev	0.017	0.078	0.079	0.080	0.074	0.048	0.051	0.037	0.050	0.079

Table 3: Results of Filtering MNB experiments

in getting a classifier out of bad region. While the average of STtoAP and STAP is lower than MNB, their st. deviation is also much lower. Thus it seems that exploration makes performance more stable across topics.

These results also suggest the following questions: What are the characteristics of "easy" and "hard" topics? Are they also indicative of whether exploration will improve results?

Looking back at the Table 1 with number of relevant documents per topic, there seems to be a some correlation between that and the results: greater number of positive documents leads to better performance. Since the size of the whole data set is 12902, topics such as "acq" and "earn" have every 5th or 4th document positive. For such topics, exploration intuitively should not be very advantageous. The smaller topics, for which relevant docs. are less frequent, on the other hand appear to benefit from STAP and STtoAP exploration.

There are exceptions on both sides though. MNB does poorly on money-fx, but relatively well on wheat, while STAP and STtoAP do much better on money-fx and worse on wheat.

5.3 Centroid Method

Surprisingly, for Centroid method adaptive approach (column Centroid in Table 4) does better than Online.

STAP, STtoAP and FIL methods overall performance is worse than no exploration. However, an exception is STtoAP with $\epsilon = 0.0001$. It performs comparably to greedy, doing better on "acq" topic.

Results of exploration with cosine method are in Tables 5 and 6. The first of these corresponds to cosine exploration only of positive model, while the second has results of exploration of the whole model The column with $\epsilon_c = 1$ corresponds to no exploration ("greedy").

Topic	Online	STtoAP				STAP	SFIL	Centroid
ϵ		0.0001	0.001	0.01	0.05			
acq	0.762	**0.862**	0.851	0.830	0.757	0.760	0.441	0.854
corn	0.328	**0.495**	**0.496**	0.445	0.240	0.315	0.070	0.494
crude	0.602	**0.682**	0.674	0.623	0.471	0.573	0.154	0.681
earn	0.875	0.870	0.869	0.867	0.829	0.830	0.629	0.870
grain	0.555	0.744	0.736	0.661	0.476	0.574	0.159	0.747
interest	0.364	0.506	0.500	0.469	0.362	0.407	0.130	0.506
money-fx	0.462	0.525	0.523	0.495	0.412	0.445	0.186	0.525
ship	0.338	0.621	0.603	0.501	0.301	0.440	0.083	0.623
trade	0.389	0.547	0.541	0.497	0.384	0.491	0.138	0.547
wheat	0.401	0.665	0.647	0.526	0.332	0.452	0.094	0.665
Average	0.508	**0.652**	0.644	0.591	0.456	0.529	0.208	0.651
St. dev	0.032	0.018	0.017	0.021	0.033	0.023	0.030	0.017

Table 4: Results of Centroid with exploration

While in both tables for all ϵ_c results are usually worse than those of greedy (except for values of ϵ_c close to 1), it is clear from these tables that some amount of exploration can almost benefit any topic. The difficult part seems to be to determine how much exploration is enough.

Exploration with the whole model seems to do better than exploration of the positive model only: the best results for each topic are better, and they beat greedy more often. It also gives better results for the same value of ϵ_c.

6 Threshold Setting for Centroid Method

Before studying threshold exploration schemes we wanted to see the effectiveness of simpler schemes for setting the threshold.

In Table 7 the first column is the topic. The second column is the best threshold, computed in the following way: we classifier is trained on all the documents, the scores are recomputed and the optimal threshold (with respect to F1 measure) is computed. The score is would be obtained by such a system with optimal threshold is in column S1 (this is "Gold Standard"). The scores achieved by the same system with default threshold 0 are in column S2. Column S3 contains results of online training (i.e. feedback is given for every documents) without any threshold adjustment. Column S4 has results of adaptive filtering without threshold adjustment. Surprisingly it does better than online in S3.

The next 3 columns attempt to find good thresholds in adaptive mode, but they do it in different ways. In S5, after every submission and feedback, the optimal threshold is recomputed (but the scores themselves are not). In S6 the scores of documents in history are also recomputed, as in DIMACS method [1]. And in S7 the threshold is adjusted similarly to

Topic	Cosine Exploration of Positive Model								
ϵ_c	0.99	0.995	0.999	0.9995	0.9999	0.99995	0.99999	0.999995	1
acq	0.376	0.441	0.645	0.734	0.851	**0.863**	**0.869**	**0.869**	0.854
corn	0.051	0.079	0.213	0.307	0.423	0.481	0.489	**0.502**	0.494
crude	0.117	0.178	0.473	0.559	0.638	0.657	0.669	0.672	0.681
earn	0.530	0.642	0.861	0.871	0.870	**0.874**	**0.877**	**0.872**	0.870
grain	0.119	0.176	0.440	0.555	0.706	0.733	0.740	0.741	0.747
interest	0.103	0.137	0.317	0.389	0.463	0.483	0.488	0.498	0.506
money-fx	0.152	0.193	0.403	0.490	**0.567**	**0.580**	0.516	0.521	0.525
ship	0.070	0.114	0.421	0.486	0.564	0.590	0.614	0.621	0.623
trade	0.113	0.160	0.387	0.452	0.519	0.528	**0.550**	**0.554**	0.547
wheat	0.072	0.116	0.335	0.436	0.567	0.617	**0.684**	**0.678**	0.665
Average	0.170	0.224	0.450	0.528	0.617	0.641	0.650	**0.653**	0.651
St. dev	0.022	0.028	0.030	0.025	0.021	0.018	0.019	0.018	0.017

Table 5: Results of Centroid with cosine exploration of positive model

Topic	Cosine Exploration (whole model)							
ϵ_c	0.999	0.9995	0.9999	0.99995	0.99999	0.999995	0.999999	1
acq	0.822	0.855	**0.870**	**0.873**	**0.863**	**0.862**	**0.863**	0.854
corn	0.339	0.434	0.468	0.469	**0.502**	**0.509**	**0.495**	0.494
crude	0.576	0.618	0.658	0.668	0.672	0.674	0.680	0.681
earn	**0.875**	**0.874**	**0.873**	**0.871**	**0.871**	0.870	0.870	0.870
grain	0.602	0.690	0.737	0.739	0.739	**0.749**	0.746	0.747
interest	0.429	0.447	0.479	0.485	0.497	0.500	0.502	0.506
money-fx	**0.531**	**0.547**	**0.587**	**0.600**	0.520	0.522	0.523	0.525
ship	0.445	0.504	0.567	0.590	0.614	0.618	**0.624**	0.623
trade	0.455	0.497	0.526	0.536	**0.554**	**0.556**	0.543	0.547
wheat	0.471	0.562	0.618	0.645	**0.679**	**0.681**	0.662	0.665
Average	0.554	0.603	0.638	0.648	0.651	**0.654**	0.651	0.651
St. dev	0.027	0.022	0.019	0.019	0.018	0.017	0.018	0.017

Table 6: Results of Centroid with cosine exploration of combined model

Topic	Best t	S1	S2	S3	S4	S5	S6	S7
acq	0.028	0.795	0.753	0.762	0.854	0.002	0.002	0.703
corn	0.094	0.534	0.339	0.328	0.494	0.033	0.087	0.115
crude	0.052	0.758	0.650	0.602	0.681	0.054	0.111	0.511
earn	0.036	0.881	0.872	0.875	0.870	0.001	0.001	0.870
grain	0.069	0.712	0.569	0.555	0.747	0.020	0.071	0.355
interest	0.142	0.660	0.660	0.364	0.506	0.009	0.009	0.338
money-fx	0.095	0.651	0.651	0.462	0.525	0.004	0.004	0.174
ship	0.087	0.728	0.306	0.338	0.623	0.013	0.013	0.657
trade	0.111	0.737	0.414	0.389	0.547	0.006	0.006	0.430
wheat	0.172	0.694	0.408	0.401	0.665	0.021	0.021	0.063
Average	0.089	0.715	0.516	0.508	0.651	0.016	0.033	0.422
St.dev	0.0431	0.0876	0.180	0.179	0.131	0.016	0.039	0.253

Table 7: Results of Threshold Exploration with Centroid method. See text for details

CAS method [6]: utility was twice the number of relevant minus the number of non-relevant documents, $\delta = 0.005$ and $L = 1000$.

As can be seen from these results, the threshold adjustment methods did not improve the performance. Analysis of poor results in columns S5 and S6 indicated that the thresholds get set to a very high value initially (i.e. after few examples) and don't recover.

Good (relatively) result in S4 is because 0 is a good threshold. But optimal thresholds are all greater than 0, and some are even greater than 0.1 (as shown in column 2).

We surmised that starting out with some training data may avoid situations where thresholds quickly become very high. To test this we decided to use the part of the corpus (7000) for training, and the rest (3000) for filtering evaluation. Results are in Table 8. Column S8 shows results of Batch mode, training on all documents (keeping threshold 0) and evaluating on test. Column S9 has results of online evaluation on test set, without threshold update. Column S10 shows results of adaptive approach without threshold update. In column S11 optimal threshold was computed but documents in history were not rescored for that. (In fact updating all document scores made the classifiers extremely slow and we did not run that experiment). The last column, S12 shows results of threshold updating using with CAS-style method (described above).

Here we can see that finding optimal-threshold based on history does better than not updating the threshold. The difference with the previous set of experiments is that, because of the training set, there was enough data to make good estimated of the optimal threshold. Note that this time CAS also does better than no threshold updating.

More experiments are needed to determine how much training data is sufficient and to find which threshold updating method is the best. Some of the methods described in literature were reported to compare favorable with simply estimating the optimal threshold.

However what we observed suggests that a lot of training data is needed for a thresholding scheme to be effective. Therefore we did not proceed with a study of threshold-based

Topic	S8	S9	S10	S11	S12
acq	0.846	0.849	0.853	0.869	0.436
corn	0.307	0.313	0.314	0.413	0.336
crude	0.682	0.695	0.705	0.812	0.735
earn	0.914	0.909	0.912	0.927	0.935
grain	0.552	0.550	0.559	0.662	0.649
interest	0.440	0.456	0.467	0.676	0.680
money-fx	0.507	0.522	0.535	0.680	0.462
ship	0.372	0.360	0.371	0.772	0.688
trade	0.365	0.365	0.376	0.667	0.627
wheat	0.400	0.400	0.396	0.614	0.594
Mean	0.539	0.542	0.549	0.709	0.614
St.d	0.199	0.199	0.199	0.138	0.1617

Table 8: Results of Threshold Exploration after Training with Centroid method. See text for details

exploration on this data.

7 Discussion

In this work we aimed to analyze a number of existing exploration schemes for adaptive filtering and to compare them to new Hoeffding and cosine exploration that we proposed.

However the results of our experiments show little benefit for any of exploration methods we tried. Only cosine method was able to produce improvements for most of the topics, but using different values of the parameter. This suggests that one possible direction of further work is to study how properties of the data and the topic relate to the degree of exploration that is beneficial to performance.

In general, it seems rather surprising to us that exploration and threshold setting did not provide greater benefits. This contradicts a number of existing studies, some of which were already mentioned in this report.

We see several potential (not mutually exclusive) causes for such results. One is that the dataset we used in the experiments is too small to allow the benefits of exploration to become fully visible. Second one is that the classification methods we chose may be too simple to allow the benefits of additional information to overweigh the costs. Finally, it is possible that the fault is not with the data or the classifiers but with the experimental setup: not providing a few training documents at the onset, without the "submit all" stage; or having too large if a training set with too small of a test set for the adaptive filtering stage.

We hope in the future to be able to conduct more rigorous study on the benefits of exploration methods and will learn from the our experience in conducting work.

8 Acknowledgments

The authors thank the KD-D group for its support through National Science Foundation grant number EIA-0087022 to Rutgers University. The views expressed in this article are those of the authors, and do not necessarily represent the views of the sponsoring agency.

The authors thank Paul Kantor and Peng Song for helpful suggestions and insights. This work was done as part of DIMACS Monitoring Message Streams Project.

References

[1] A. Anghelescu, E. Boros, D. Lewis, V. Menkov, D. Neu, and P. Kantor. Rutgers filtering work at trec 2002: Adaptive and batch. In *The Eleventh Text REtrieval Conference (TREC-11)*. NIST, 2002.

[2] N. Cesa-Bianchi, A. Conconi, and C. Gentile. Margin-based algorithms for information filtering. In *Proceedings of NIPS*, 2002.

[3] D. P. Helmbold, N. Littlestone, and P. M. Long. Apple tasting. *Information and Computation*, 161:85–139, 2000.

[4] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[5] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. Technical Report HPL-2003-97, HP Labs, June 2003.

[6] H. Xu, Z. Yang, B. Wang, B. Liu, J. Cheng, Y. Liu, Z. Yang, X. Cheng, and S. Bai. TREC-11 experiments at CAS-ICT: Filtering and web. In *The Eleventh Text REtrieval Conference (TREC-11)*. NIST, 2002.

[7] Y. Yang and B. Kisiel. Margin-based local regression for adaptive filtering. In *Proceedings of CIKM-03*, 2003.

[8] C. Zhai, P. Jansen, and D. A. Evans. Exploration of a heuristic approach in threshold learning in adaptive filtering. In *Proceedings of SIGIR-00*, 2000.

[9] Y. Zhang and J. Callan. Maximum likelihood estimation for filtering thresholds. In *Proceedings of SIGIR-01*, 2001.