Exploration and Exploitation in Adaptive Filtering Based on Bayesian Active Learning

Yi Zhang* YIZ@CS.CMU.EDU
Wei Xu** XW@CCRL.SJ.NEC.COM
Jamie Callan* CALLAN@CS.CMU.EDU
* Language Technology Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA USA

** NEC Laboratories America, 10080 North Wolfe Road, SW3-350, Cupertino, CA 95014 USA

Abstract

In the task of adaptive information filtering, a system receives a stream of documents but delivers only those that match a person's information need. As the system filters it also refines its knowledge about the user's information needs based on relevance feedback from the user. Delivering a document thus has two effects: i) it satisfies the user's information need immediately, and ii) it helps the system better satisfy the user in the future by improving its model of the user's information need. The traditional approach to adaptive information filtering fails to recognize and model this second effect.

This paper proposes *utility divergence* as the measure of model quality. Unlike the model quality measures used in most active learning methods, utility divergence is represented on the same scale as the filtering system's target utility function. Thus it is meaningful to combine the expected immediate utility with the model quality, and to quantitatively manage the trade-off between exploitation and exploration. The proposed algorithm is implemented for setting the filtering system's dissemination threshold, a major problem for adaptive filtering systems. Experiments with TREC-9 and TREC-10 filtering data demonstrate that the proposed method is effective.

1. Introduction

An information filtering system monitors a document stream to find the documents that match information needs specified by user profiles. The most difficult task for filtering is on-line time-critical filtering, where the value of a document decays rapidly with time. In this case, the potentially relevant documents must be delivered immediately, thus the system has no time to accumulate and rank a set of documents. *Adaptive information filtering* systems receive periodic feedback from the user about which delivered documents were relevant, which provides training data for learning.[1]

	Relevant	*Non-Relevant*
Delivered	R^+/A_R	N^+/A_N
Not Delivered	R^-/B_R	N^-/B_N

Table 1. The values assigned to relevant and non-relevant documents that the filtering system did and did not deliver.

User satisfaction is typically modeled with a linear utility function. The function used in recent TREC Filtering track evaluations is shown below (Robertson & Hull, 2001).

$$Utility = A_R \cdot R^+ + A_N \cdot N^+ + B_R \cdot R^- + B_N \cdot N^- \quad (1)$$

This model corresponds to assigning a positive or negative value to each element in the categories of Table 1, where R^-, R^+, N^-, and N^+ correspond to the number of documents that fall into the corresponding category, and A_R, A_N, B_R and B_N correspond to the credit/penalty for each element in the category. B_R and B_N are usually set to zero because i) user satisfaction is mostly influenced by what has been seen, and ii) N^+ and N^- are usually too big. For example, the TREC 2000 Filtering Track Utility was $T9U = 2R^+ - N^+$; the TREC 2001 and 2002 Filtering Track used a normalized version of $T9U$ for evaluation.

The traditional methods of deciding whether to deliver a document try to maximize the estimated im-

[1] In most prior research the feedback is received immediately after the document is delivered (Robertson & Soboroff, 2002).

mediate utility of the decision. That is, a document is delivered if and only if the expected immediate utility of delivering it is greater than the expected utility of not delivering it. The expected immediate utility can be calculated from the probability that the document is relevant. For example, for the utility functions used by the TREC9, TREC10 and TREC11 Filtering tracks, participants usually set the threshold where $P(relevant|document) = 1/3$ because the expected utility at that point is 0 (Zhang & Callan, 2001a; Arampatzis & Hameren, 2001). In fact, delivering a document iff $P(relevant|document) >= 1/3$ was written explicitly in the guidelines of some recent TREC adaptive filtering tracks (Robertson & Hull, 2001).

The delivery criterion above tries to optimize the immediate satisfaction of the user. It does not consider the possibility that the system can improve its knowledge about the user's information need based on the feedback from the user so that it can better serve the user in the future. Especially in the early stage of filtering, when the system's knowledge about the user's information need is very limited, the potential gain from improving the user model can be substantial.

The work described in this paper is based on considering the value of longer-term exploration along with the immediate reward of delivering a document when setting decision boundaries. We propose to use utility divergence, which will be defined later, as the measure of the model quality. Unlike measures of model quality used in most active learning methods, utility divergence has the advantage of having the same scale as the traditional utility model adaptive filtering systems try to optimize. Thus we can combine the expected immediate utility with the expected model quality to get a single quantity that measures the short-term and long-term value of a document in the document stream. This combined measure is the basis for deciding whether to deliver the document.

The following sections describe our research on exploration and exploitation while filtering based on Bayesian theory. Section 2 describes the general framework of optimizing the utility based on the trade off-between exploration and exploitation using Bayesian active learning. Sections 3 and 4 describe our experimental methodology and results. Section 5 discusses related work and section 6 concludes.

2. Exploration vs. Exploitation Based on Bayesian Theory

In order to maximize the overall utility of the system, we propose to have two modules for the system: the exploitation module and the exploration module. We also propose to use utility as the measure for both modules: U_1, the direct utility gain of delivering a document for the exploitation module, and U_2, the expected utility gain of knowing the document label for the exploration module. Thus we have unified control of the exploration and exploitation trade-off, and deliver a document if the combined utility is above 0.

For simplicity, we will use Bayesian logistic regression as our learner, and the objective function is to maximize $Utility = A_R \cdot R^+ + A_N \cdot N^+$. [2]

2.1. Exploitation Using Bayesian Inference

As mentioned in Section 1, the direct gain/loss on utility for delivering a document can be estimated based on the probability of relevance for that document. Suppose we have a model parameterized as θ to estimate the probability of the relevance of a given document. The prior distribution of the model parameters is $p(\theta)$. After seeing data $D = \{(x_1, y_1), \cdots, (x_k, y_k)\}$, the posterior distribution $p(\theta|D)$ is

$$p(\theta|D) = \frac{P(D|\theta)p(\theta)}{\int P(D|\theta)p(\theta)d\theta} \tag{2}$$

where $P(D|\theta)$ is the likelihood of the user feedback given delivered documents and θ:

$$P(D|\theta) = \prod P(y_i|x_i, \theta) \tag{3}$$

The Bayesian average of the immediate utility gain of delivering a document is:

$$U_1(x|D) = \int_\theta \sum_y A_y P(y|x, \theta)P(\theta|D)d\theta \tag{4}$$

where A_y is the utility of delivering a document with the true label y. According to Table 1, $A_y = A_R$ if the true label is "relevant", and $A_y = A_N$ if the true label is "non-relevant".

2.2. Exploration Using Bayesian Active Learning

For exploration, we need to define the quality of learned model. In the active learning framework proposed by (Tong & Koller, 2000), if we choose to use model $\tilde{\theta}$ and the true model is θ, we incur some loss $Loss(\theta||\tilde{\theta})$. Although we do not know the exact value of θ, $p(\theta|D)$ represents our beliefs about the distribution of θ given the evidence. Thus the expected loss of

[2] Most prior adaptive filtering research assumed user satisfaction is influenced only by what is seen, so $B_R = B_N = 0$ (Equation 1). We make the same assumption.

using model $\tilde{\theta}$ is given by:

$$\begin{aligned} Loss(D,\tilde{\theta}) &= E_D(Loss(\theta,\tilde{\theta})) \\ &= \int_\theta p(\theta|D) Loss(\theta||\tilde{\theta}) d\theta \end{aligned} \quad (5)$$

where $p(\theta|D)$ is the posterior distribution of different model parameters. The quality of a model after we have seen data set D is

$$Loss(D) = Loss(D,\theta^*_D) \quad (6)$$

where $\theta^*_D = arg\min_{\tilde{\theta}} Loss(D,\tilde{\theta})$.

Smaller $Loss(D)$ means a better model. For active learning, $Loss(\theta,\theta^*_D)$ needs to capture the notion of uncertainty of the model. One commonly chosen metric is Kullback-Leibler divergence (KL-divergence):

$$KL(\theta||\theta^*_D) = \int p(x) \sum_y p(y|x,\theta) \log \frac{P(y|x,\theta)}{P(y|x,\theta^*_D)} dx \quad (7)$$

where $p(x)$ is the distribution of input x, which is independent of θ and is usually given or learned from unlabelled data.

The usefulness of knowing the label of a document is measured by its potential to lower $Loss(D)$. However, in information filtering, our ultimate goal is to optimize some utility function in the long run. It is unnatural to combine KL-divergence with the expected immediate utility credit/loss together to get a single quantity on which the decision of whether to deliver a document can be made. Also, it is unclear how a smaller KL-divergence relates to higher utility. In stead of using KL-divergence, we propose to use the difference between the best possible utility and the actual utility, which we call *utility divergence*, as the function $Loss(\theta,\tilde{\theta})$ to measure the model quality:

$$UD(\theta||\tilde{\theta}) = U(\theta,\theta) - U(\theta,\tilde{\theta}) \quad (8)$$

where $U(\theta',\theta'')$ is the expected utility if we choose to use model θ'' and the true model is θ'. A well-defined $U(\theta_1,\theta_2)$ should have the following property

$$\forall \theta',\theta'', U(\theta',\theta') \geq U(\theta',\theta'') \quad (9)$$

which essentially says that the expected utility of an incorrect model cannot exceed the expected utility of the correct model. It is worth noting that KL-divergence is a special case of *utility divergence*. If we choose to use loglikelihood as utility, then $U(\theta',\theta'')$ is:

$$U(\theta',\theta'') = \int p(x) \sum_y p(y|x,\theta') \log P(y|x,\theta'') dx \quad (10)$$

which shows that when using loglikelihood as utility, $UD(\theta||\tilde{\theta}) = KL(\theta||\tilde{\theta})$.

Using utility divergence as the loss function, $Loss(D)$ can be rewritten as

$$\begin{aligned} Loss(D) &= Loss(D,\theta^*_D) \\ &= \int p(\theta|D)(U(\theta,\theta) - U(\theta,\theta^*_D)) d\theta \end{aligned} \quad (11)$$

For information filtering, the goal is to maximize the utility (Equation 1), so we use the following as utility:

$$U(\theta',\theta'') = \int_{S(\theta'')} p(x) \sum A_y P(y|x,\theta') dx \quad (12)$$

where $S(\theta'') = \{x | \sum_y A_y P(y|x,\theta'') > 0\}$ is the space of x where model θ'' "thinks" delivering x has immediate positive utility.

The expected reduction on utility divergence of knowing the label of a document x is:

$$U_2(x|D) = \sum_y P(y|x,D) Loss(D \cup (x,y)) - Loss(D) \quad (13)$$

Suppose there are N_{future} future documents, then the expected utility of delivering a document x is :

$$U(x|D) = U_1(x|D) + N_{future} U_2(x|D) \quad (14)$$

We deliver a document x if and only if $U(x|D) \geq 0$

2.3. Logistic Regression to Determine Dissemination Threshold

Information filtering systems based on statistical retrieval models usually compute a numeric score that indicates how well each document matches each profile. Documents with scores above profile-specific dissemination thresholds are delivered. Optimal dissemination thresholds are usually difficult to determine a priori, so they are often learned during filtering, using relevance feedback about disseminated documents. In this section we apply the theory discussed above to the problem of determining the dissemination threshold. We assume we already have a separate module that can compute the score of document. The input to the algorithm is the score x of a document; the problem is to determine whether to deliver the document given its score.

The algorithm proposed in this paper can be applied to high dimensions and used to update the scoring function that calculates x. But considering the extremely

small number of training data, especially in the early stage of filtering, our first implementation algorithm is only in one dimension, that is for setting the dissemination threshold.

We use logistic regression to model the conditional probability of user feedback y given the document score x:

$$P(y=1|x,\theta) = \frac{1}{1+\exp(-w_0 - w_1 x)} \tag{15}$$

where the prior distribution $p(w)$ equals the Gaussian distribution $N(w; m_0, v_0)$. m_0 is the mean of the Gaussian and v_0^{-1} is the covariance of the Gaussian.

In information filtering the explicit goal is to find a decision boundary, in this case the dissemination threshold t, to optimize the linear utility function. t corresponds to the boundary of $S(\theta'')$ in Equation 12. Thus the quality of the model should be quantified by the expected utility of using $t_{\theta^*_D}$ as the decision boundary that defines $S(\theta'')$ in Equation 12

$$S(\theta'') = [t_{\theta''}, \infty) \tag{16}$$

where $t_{\theta''}$ is the threshold model θ'' "thinks" is the best.

$$\begin{aligned} & U(\theta', \theta'') \\ & = U(\theta', t_{\theta''}) && (17) \\ & = \int_{t_{\theta''}}^{\infty} \left(A_N + \frac{A_R - A_N}{1+\exp(-w_0' - w_1' x)} \right) p(x) dx && (18) \\ & Loss(D, \tilde{\theta}) \\ & = \quad Loss(D, t_{\tilde{\theta}}) \\ & = \quad \int p(\theta|D)(U(\theta, t_\theta) - U(\theta, t_{\tilde{\theta}})) d\theta && (19) \end{aligned}$$

To calculate $Loss(D)$, we can find $t_{\theta^*_D}$ by solving

$$\frac{dLoss(D, t_{\theta^*_D})}{dt_{\theta^*_D}} = 0 \tag{20}$$

and

$$Loss(D) = Loss(D, t_{\theta^*_D}) \tag{21}$$

2.4. Computational issues

Computation of U_1 in Equation 4 and $Loss(D)$ in Equation 11 involve integration over posterior $p(\theta|D)$. However, the posterior distribution $p(w|D)$ for logistic regression is quite complicated and the integration cannot be calculated in closed form. Our strategy is to use a Monte Carlo method to get an approximate solution. We generate K random samples θ_i using the Metropolis-Hastings algorithm (Tanner, 1996). Then U_1 and $Loss(D)$ can be approximated by:

$$U_1(x|D) \approx \frac{1}{K} \sum_{i=1}^{K} \sum_y A_y P(y|x, \theta_i) \tag{22}$$

$$Loss(D, t_{\tilde{\theta}}) \approx \frac{1}{K} \sum_{i=1}^{K} (U(\theta_i, t_\theta) - U(\theta_i, t_{\tilde{\theta}})) \tag{23}$$

In order to generate random samples from the posterior $p(\theta|D)$ efficiently, we apply Laplace's method to use a Gaussian distribution $N(w; \theta_{MP}, v)$ to approximate the posterior, where θ_{MP} is the maximum a priori estimation of θ and v is Hessian matrix of the log-likelihood $\log(P(D|\theta)p(\theta))$ at θ_{MP}. Then we use this Gaussian approximation to generate candidate values for the Metropolis-Hastings method.

We summarize the computational procedure for determining the dissemination threshold in Table 2. A document is delivered when its score is above the threshold. When a document is delivered, the dissemination threshold is recomputed based on the scores and labels of *all* of the delivered documents.

```
FUNCTION : Calculate dissemination threshold
INPUT : D = (x_1, x_2), ..., (x_k, y_k)
OUTPUT : dissemination threshold
LOOP binary search for x such that :
  U(x|D) = U_1(x|D) + n_future · U_2(x|D) = 0
  where
    U_1 is computed using Equation 22
    U_2 is computed using Equation 13
return x

FUNCTION : Calculate Loss(D)
INPUT : D = (x_1, x_2), ..., (x_k, y_k)
OUTPUT : Loss(D)
Calculate the MAP estimation θ_MP
Calculate the Gaussian approximation of P(θ|D)
Generate K samples using Metropolis Algorithm
Calculate t_{θ*_D} using Equations 20 and 23
Return Loss(D) = Loss(D, t_{θ*_D})
```

Table 2. Pseudo code for determining threshold.

3. Experimental Methodology

The algorithm described in Table 2 was tested experimentally, using the methodology described below.

3.1. Datasets

Two different text corpora were used in the experiments: the OHSUMED dataset used in the TREC-9 Filtering Track, and the Reuters 2001 dataset used in the TREC-10 Filtering Track. As required by TREC adaptive filtering track, for each user profile, the system begins with two identified relevant documents and a natural language description of the information need, which is the title and the description field of the corresponding topics provided by NIST. The two datasets have rather different properties, as described below.

3.1.1. OHSUMED Data

The OHSUMED dataset contains 348,566 medical abstracts published from 1987 to 1991 (Hersh et al., 1994). It was used by the TREC-9 Filtering Track (Robertson & Hull, 2001). 63 OHSUMED queries were used to simulate user profiles. The relevance judgments were made by medical librarians and physicians based on the results of interactive searches. In the TREC-9 Filtering Track, it is assumed that the user profile descriptions arrived at the beginning of 1988, so the 54,709 articles from 1987 can be used to learn word occurrence (e.g., idf) and corpus (e.g., average document length) statistics. The average number of relevant articles per profile in the testing data is 51.

3.1.2. Reuters 2001 Data

The Reuters 2001 data is a collection of about 810,000 Reuters English News stories from August 20, 1996 to August 19, 1997. It was used by the TREC-10 Filtering Tracks (Robertson & Soboroff, 2002).

In TREC-10, 84 Reuters categories were used to simulate user profiles. The average number of relevant articles in the testing data is about 9,795 documents per profile, which is much larger than in the OHSUMED dataset. However, the filtering system only begins with 2 positive and zero negative training data, thus this is considered a very difficult dataset.

3.2. Evaluation Methodology

Utility was measured by the macro average of $T9U = 2 \cdot R^+ - N^+$ and a normalized version of $T11SU$.[3]

The "Normal-Exponential" threshold setting algorithm described in (Arampatzis & Hameren, 2001) was used as a baseline. It uses a normal distribution to fit the relevant documents' scores and an exponential distribution to fit the non-relevant documents. This algorithm was a component of the most effective system tested in the TREC9 filtering track (Robertson & Hull, 2001).[4]

One problem with using generative models such as the Normal-Exponential model for learning adaptive filtering thresholds is that although the training data is assumed to be representative, it is in fact biased because the system only gets relevance feedback for documents it delivers (i.e., documents with scores above the dissemination threshold). (Zhang & Callan, 2001a) proposed a Maximum Likelihood Normal-Exponential (ML N-E) algorithm to explicitly compensate for this sampling bias. This algorithm was used as a second experimental baseline.

For the Bayesian approach, we also did some experiments without active learning, which we call "Bayesian Immediate Learning". It was used as a third experimental baseline.

The algorithms cannot learn when the threshold is too high to let any documents be delivered, so the filtering system gradually decreased the threshold in such cases.

3.3. Filtering Environment

The YFilter adaptive information filtering system was used our experiments (Zhang & Callan, 2001b). Documents were processed by removing symbols such as punctuation and special characters, excluding stopwords, and stemming terms with the Porter stemmer. Processed documents were compared to each profile using a variant of the BM25 tf.idf formula (Callan, 1996) to measure the similarity of the document and user profile. The Bayesian active learner's input was the score that indicates the similarity; its output was a threshold. Documents with similarity scores above the threshold were delivered. Relevance judgements were provided immediately for delivered documents, which enabled the system to update user profiles.

For each topic, the filtering system created initial profiles using terms from the TREC topic Title and Description fields. Because the first two relevant documents were sampled according to $P(x|y = 1)$ instead of $P(x)$, we cannot use it for training the discriminative model. A heuristic set the initial threshold to allow the highest-scoring documents (top 1%) in the training data to pass. Once the system had at least 1 positive and 1 negative feedback in the testing document stream, the proposed algorithm was used to set

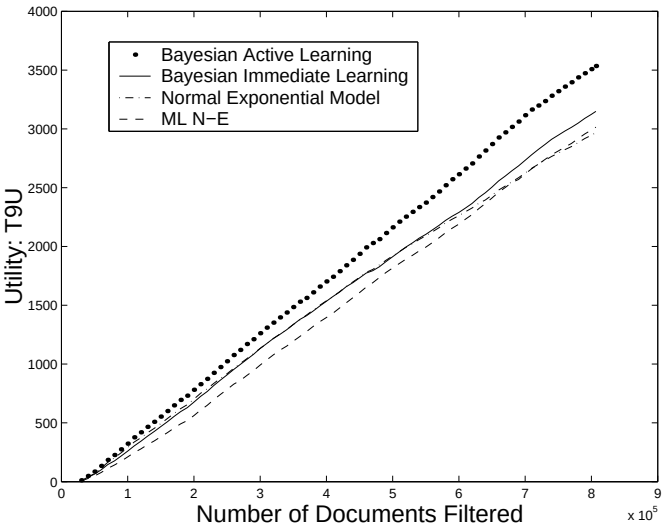
[3] $T11SU = \frac{max(\frac{T11U}{MaxU}, MinNU)}{1-MinNU}$, where $MaxU = 2 * (R^+ + R^-), MinNU = -0.5$.

[4] Experimental results reported below for the Normal-Exponential algorithm are not directly comparable to prior results, because the YFilter profile-learning (term and term weight) module was used in all of our experiments.

Table 3. Comparison of four threshold-learning algorithms on the TREC-10 Filtering data. Reuters dataset.

Metrics	Bayesian Active	Bayesian Immediate	Norm. Exp.	ML N-E
T9U	3,534	3,149	2,969	3,015
T11SU	0.448	0.445	0.436	0.439
Precision	0.463	0.481	0.464	0.496
Recall	0.251	0.234	0.227	0.212
Docs/Prof	4,527	3,895	2,792	3,380

Figure 1. Comparison of threshold-learning algorithms on the TREC-10 Filtering data over time. Reuters dataset.



dissemination thresholds.

Our algorithm also needs to model $P(x)$, the distribution of document scores. We used a simple exponential model to fit $P(x)$ in our experiments. We set the number of documents in the future $N_{future} = \alpha \cdot \frac{R^+ \cdot N_{new}}{N_{old}}$, where R^+ is the number of relevant document delivered, N_{new} is the number of expected documents in the future, N_{old} is the number of filtered documents, and α is a constant that controls the exploration rate. α was set arbitrarily to 200 in our experiments. This is a conservative estimate, and is similar to the discounted future rewards used in reinforcement learning.

4. Experimental Results

Our first experiment compared the threshold setting algorithms on the TREC10 Reuters corpus. This is considered a relatively difficult corpus because the initial training data is very limited and not particularly representative of the variety in the large number of relevant documents in the test data. Table 3 and Figure 1 summarize the experimental results.

Table 4. Comparison of Bayesian active learning and Bayesian immediate learning on Profile 83.

Metrics	Bayesian Active	Bayesian Immediate
T9U	10,488	8,712
T11SU	0.296	0.246
Precision	0.682	0.644
Recall	0.578	0.509
Docs/Prof	10,017	9,342

Table 5. Comparison of threshold-learning algorithms on TREC-9 Filtering data. OHSUMED data, OHSU topics.

Metrics	Bayesian Active	Bayesian Immediate	Norm. Exp.	ML NE
T9U	11.32	11.54	6.59	11.79
T11SU	0.353	0.360	0.329	0.362
Precision	0.300	0.325	0.256	0.339
Recall	0.231	0.203	0.264	0.177
Docs/Prof	31	25	46	20

Active learning was very effective compared to the baseline methods. T9U utility was higher, T11SU utility was slightly higher, Precision and Recall were comparable, and many more documents were delivered without hurting utility. As expected, the Maximum Likelihood Normal-Exponential method, which compensates for sampling bias, outperformed the basic Normal-Exponential method. Sampling bias is not a problem for discriminative models such as the Bayesian active and Bayesian immediate methods. Bayesian active learning outperformed the other models at all times during the experiment (Figure 1).

When we compared the performance of Bayesian active and immediate learning on profiles where active learning significantly improved performance we found that active learning increased both Recall and Precision (e.g., Table 4). This improvement is partly due to the profile (term and term weight) learning algorithm, which also benefits from the additional training data generated by the active learner. Our thresholding algorithm did not consider the benefit of an improving profile, so it was suboptimal (although effective). For simplicity we have focused only on threshold learning in this paper, however the active learning algorithm (Section 2) is not restricted to problems of low dimensionality; a higher dimensionality version of the algorithm could also incorporate profile learning.

The threshold-setting algorithms were also tested on the TREC9 dataset, which contains a relatively small percentage of relevant documents (0.016%); the test

data consists of more than 300,000 documents, but only 51 documents per profile are relevant, on average. The OHSUMED topic descriptions are well-written, which provides relatively accurate initial profiles. One might expect that on this dataset exploitation would be much more important than exploration, thus active learning might be detrimental. If the threshold is set too low, the system delivers thousands of non-relevant documents, hurting utility. In TREC9 Filtering Track evaluations some participants reported negative $T9U$ utility on this dataset (Robertson & Hull, 2001).

The experimental results are summarized in Table 5. As expected, active learning did not improve utility on this dataset. More importantly, it did not hurt utility, either. Consequently these results rank in the top 2 when compared with results from the 9 systems that participated in the TREC9 Adaptive Filtering track.[5]

Bayesian immediate learning can be viewed as an active learner that only selects documents on the positive side of the decision boundary for exploration; Bayesian active learning also samples on the negative side of the decision boundary. The comparable performance of the Bayesian active and Bayesian immediate learners indicates that the active learner recognized that the relatively good initial profiles, the limited number of relevant documents in the stream, and the penalty for delivering non-relevant documents collectively made exploring the negative side of the decision boundary a poor choice. Active learning did not hurt accuracy, even in a test where exploration was a risky choice.

5. Related Work

There has been much work on active learning in the machine learning and IR research communities (Joachims, 1998; Lewis & Catlett, 1998; Freund et al., 1997; McCallum & Nigam, 1998). For example, the "Query by Committee" algorithm selects the next query according to the principle of maximal disagreement between a committee of classifiers (Freund et al., 1997). (McCallum & Nigam, 1998) modified the "Query by Committee" algorithm with a Naive Bayes model, together with unlabelled data for active learning. (Lewis & Catlett, 1998) introduced "uncertainty sampling" to choose the instance that current classifiers are most uncertain about. (Tong & Koller, 2000) provided a theoretical framework for active learning and applied it to Support Vector Machine classifiers.

Unfortunately, most of the prior active learning research focused on interactive retrieval tasks, thus it did not address the trade-off between exploitation and exploration. It cannot be applied easily to adaptive filtering, where the system is evaluated based on utilities such as $T9U$, and the direct cost/reward of delivering the document (exploitation) is as important as the improvement in model estimation (exploration).

There is also much prior research on adaptive filtering, especially on setting dissemination thresholds. However, as discussed in Section 1, none of the previous research addressed the trade-off between exploration and exploitation. The common approach is to consider only exploitation. (Chai et al., 2002) used the information gain between a document and the model to select documents, however the task was batch filtering instead of adaptive filtering, and the method did not provide a framework for combining information gain with the immediate cost/reward.

Prior research in related areas influenced our work. The Bayesian framework, on which our algorithm is based, was used for active learning by (Tong & Koller, 2000) and (McCallum & Nigam, 1998). Our approach also matches the risk minimization model described in (Lafferty & Zhai, 2001). The work described in this paper differs from prior work in two major aspects.

- This research uses *utility divergence* to measure the quality of the model for exploration. If the objective function for a classification system is to maximize the likelihood of the data, KL divergence can be used to measure the quality of a model. However, in adaptive filtering user satisfaction is usually modelled by utility (e.g., Table 1), and the system is evaluated using linear utility. Our algorithm uses *utility divergence* to measure the quality of the model, thus it optimizes utility directly, based on the Bayesian framework.

- This research considers the direct and indirect cost/reward for delivering a document. An adaptive filtering system must consider the immediate cost/credit for a request, that is, it gets credit A_R for delivering a relevant document and penalty A_N for delivering a non-relevant document. Previous work in active learning did not consider this cost/credit, which affects the system performance significantly. Previous work in adaptive filtering is focused entirely on this cost/credit, disregarding the future benefit.

To summarize, exploitation and exploration are combined in a single, unified framework based on utility divergence.

[5] This comparison is intended to be descriptive only, because the research community now has greater experience with this dataset than TREC9 participants had.

6. Conclusion

This paper provides a framework, based on Bayesian active learning, for modeling the trade-off between exploitation and exploration in adaptive information filtering. It provides a quantitative measure of the immediate cost/reward and future reward of delivering a document when the objective is maximizing a user-defined utility measure. We believe that this is the first work to study the trade-off between exploration and exploitation in adaptive information filtering.

The experimental results demonstrate that a combination of exploration and exploitation can improve results, for example, when the initial profiles are poor and the document stream contains many relevant documents. The results also demonstrate that it can have little effect, for example, when the initial profiles are good and the document stream contains few relevant documents. Bayesian active learning handles both of these situations well, exploring only when it is useful to do so. When the algorithm does not explore, it is as good as or better than three competing methods.

There are several directions for future work. The research reported here only applied the new framework to setting dissemination threshold, however it could also be applied to problems of higher dimensionality, for example to learn the importance of features such as words or phrases. We didn't use high dimensional Bayesian logistic regression, due to model complexity, because the recent TREC Filtering track methodology is to begin with only 2 relevant and zero non-relevant documents for training. If the initial conditions are relaxed, experiments are possible with problems of higher dimensionality. Our model of $P(x)$ and the way we calculate the expected number of future documents is also weak; additional research in this direction is needed.

7. Acknowledgements

We thank Tom Minka for suggestions on some of the work in this paper.

References

Arampatzis, A., & Hameren, A. (2001). The score-distribution threshold optimization for adaptive binary classification task. *Proceedings of the 24th ACM SIGIR Conference.*

Callan, J. (1996). Document filtering with inference networks. *Proceedings of the 19th ACM SIGIR Conference.*

Chai, K. M. A., Chieu, H. L., & Ng, H. T. (2002). Bayesian online classifiers for text classification and filtering. *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM.

Freund, Y., Seung, H. S., Shamir, E., & Tishby., N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning* (pp. 133–168).

Hersh, W., Buckley, C., J.Leone, T., & Hickam, D. (1994). OHSUMED: An interactive retrieval evaluation and new large test collection for research. *Proceedings of the 7th ACM SIGIR Conference.*

Joachims, T. (1998). Text categorization with support vector machine. *Proceedings of the European Conference on Machine Learning.* Springer-Verlag.

Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. *Proceedings of the 24th ACM SIGIR Conference.*

Lewis, D., & Catlett, J. (1998). Heterogeneous uncertainty sampling for supervised learning. *Proceedings of the Eleventh International Conference on Machine Learning.* Morgan Kaufmann.

McCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proceeding of the International Conference on Machine Learing.*

Robertson, S., & Hull, D. (2001). The TREC-9 Filtering track report. *The Ninth Text REtrieval Conference (TREC-9)* (pp. 25–40). National Institute of Standards and Technology, special publication 500-249.

Robertson, S., & Soboroff, I. (2002). The TREC-10 Filtering track final report. *Proceeding of the Tenth Text REtrieval Conference (TREC-10).* NIST, special publication 500-250.

Tanner, M. A. (1996). *Tools for statistical inference.* Springer. 3 edition.

Tong, S., & Koller, D. (2000). Active learning for parameter estimation in bayesian network. *Neural Information Processing Systems (NIPS).*

Zhang, Y., & Callan, J. (2001a). Maximum likelihood estimation for filtering thresholds. *Proceedings of the 24th ACM SIGIR Conference.*

Zhang, Y., & Callan, J. (2001b). Yfilter at TREC-9. *Proceedings of the Ninth Text REtrieval Conference (TREC-9).* NIST, special publication 500-249.