Active Learning for Multi-Task Adaptive Filtering

Abhay Harpale AHARPALE@CS.CMU.EDU
Yiming Yang YIMING@CS.CMU.EDU
Carnegie Mellon Univeresity, 5000 Forbes Avenue, Pittsburgh, PA-15213, USA

Abstract

In this paper, we propose an Active Learning (AL) framework for the Multi-Task Adaptive Filtering (MTAF) problem. Specifically, we explore AL approaches to rapidly improve an MTAF system, based on Dirichlet Process priors, with minimal user/task-level feedback. The proposed AL approaches select instances for delivery with a two-fold objective: 1) Improve future task-specific system performance based on feedback received on delivered instances for that task, 2) Improve the future overall system performance, thereby benefiting other tasks in the system, based on feedback received on delivered instances for a particular task. Current AL approaches focus only on the first objective. For satisfying both goals, we define a new scoring function called *Utility Gain* to estimate the perceived improvements in task-specific and global models. In our experiments on standard benchmark datasets, we observed that global AL approaches that additionally take into account the potential benefit of feedback to other tasks in the system performed better than the task-specific approach that focused only the benefit of the current task.

1. Introduction

Adaptive Filtering (AF) (Robertson et al., 2001) systems monitor a stream of documents to filter out documents that are relevant to the particular task or user. For example, a stock analyst would like to filter out all the news about stocks in his portfolio, while a technology entrepreneur might be interested in news about latest technology startups. Each of these users can be considered a *task* from the AF system's perspective. AF systems model user interests based on the initial information request (e.g. a query) presented by the user, and the subsequent relevance feedback provided by the user for the results presented to the user. For example, one popular AF approach involves the use of Logistic Regression classifier (Jaakkola & Jordan, 1996) to classify documents into relevant and non-relevant categories for that particular user. The relevance feedback received from the user is used to supplement the training data to retrain the classifier for future predictions.

Most research in AF systems has focused on learning each task independently of other tasks. In this paper, we will refer to such approaches as Single-Task AF (STAF) approaches. In the initial stages of learning, when the feedback from each user is limited, these approaches suffer from data sparsity, leading to weaker models, and consequently poorer performance. Multi-task learning methods have shown significant success in mitigating this per-task data sparsity problem by sharing information across multiple tasks. For example, to learn the interest-model of a particular stock analyst, it could be useful to identify common important features for portfolio tracking based on the feedback received from other stock analysts. Irrespective of their portfolios, all stock analysts are usually interested in common news regarding corporate announcements, balance sheets, government regulations, and day-to-day stock market indicators pertinent to their stocks. In this paper, we will refer to such approaches, which leverage information from multiple tasks, as Multi-Task Adaptive Filtering (MTAF) approaches.

The future performance of an AF system relies on the feedback received on delivered items. If the system myopically focuses on delivering only (perceived) *relevant* documents (better immediate performance), the feedback received on these documents may not lead to the best learnt task models (in the future), thereby limiting the usefulness of such feedback for future predictions. In this paper, we present an Active Learn-

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

ing (AL) framework for MTAF that additionally takes into account the perceived benefit of feedback on items before making a delivery. In the MTAF setting, the AL system has a three-fold objective: 1) Provide relevant documents, 2) Feedback provided on a delivered document should maximally improve the task-specific performance in the future, 3) Feedback provided on a delivered document, should maximally improve the overall system (multiple tasks) in the future. The current AL approaches focuses only on the second objective, thereby narrowly focusing on task-specific performance improvements. In this paper, for satisfying the goal 1, we chose the Multi-Task Logistic Regression with Dirichlet Process priors (Blei & Jordan, 2006; Xue et al., 2007) for facilitating information sharing across tasks. For goals 2 and 3, we propose a novel AL framework based on a new scoring function called *Utility Gain*. This scoring function is inspired by the popular empirical risk minimization approaches in Active Learning (Melville et al., 2005). However, the current empirical risk minimization approaches only focus on minimizing the risk of one task, while we develop variants that selectively focus on one task, global model, or all tasks, as required during the various phases of learning. Consequently, our framework chooses instances that might lead to maximal (expected) gain in performance of the system (task-specific or global) for future predictions.

2. Adaptive Filtering Preliminaries

2.1. Single-Task Adaptive Filtering

We briefly describe a popular representative STAF approach based on the Logistic Regression (LR) classifier. The LR classifier estimates relevance of a document using the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ as follows:

$$P(y=1|\mathbf{x},\mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} \quad (1)$$

In the above equation, $\mathbf{x}$ is a feature vector representing a document, and $\mathbf{w}$ is the weight vector of regression coefficients. $\mathbf{w}$ is usually fit by maximum likelihood estimation on the available relevant and non-relevant training documents $\mathcal{D}$ for the task.

In this paper, for consistency among approaches, we will be focusing on the Bayesian Logistic Regression variant. In the Bayesian setting, the parameters $\mathbf{w}$ are usually drawn from a diffuse Gaussian prior distribution $\mathbf{w} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, to ensure regularization. For a fully Bayesian treatment, instead of using a point estimate of the parameters $\mathbf{w}$, one may integrate over the posterior distribution of $\mathbf{w}$ to obtain a Bayesian estimate of $P(y=1|\mathbf{x},\mathcal{D})$

$$P(y=1|\mathbf{x},\mathcal{D}) = \int_{\mathbf{w}} P(y=1|\mathbf{x},\mathbf{w})P(\mathbf{w}|\mathcal{D})d\mathbf{w} \quad (2)$$

Table 1. Important notation used in this paper.

Symbol	Description
M	Total number of tasks
$m \in \{1,\ldots,M\}$	A task
N_m	Total number of instances for task m
d	Dimensionality of feature space
$\mathbf{x}_{m,n} \in \mathcal{R}^d$	The n'th data instance for task m
$y_{m,n} \in \{0,1\}$	The label of instance $\mathbf{x}_{m,n}$
$\mathbf{w}_m \in \mathcal{R}^d$	LR parameters for task m
$k \in \{1,\ldots,\infty\}$	A group or cluster
$\mathbf{w}^*_k \in \mathcal{R}^d$	LR parameters of the k'th group
$\phi_{m,k} \in [0,1]$	Group k mixing proportion for task m

2.2. Utility of Adaptive Filtering

In the Adaptive Filtering setting, the performance of a system is usually measured in terms of the *utility* of documents delivered by that system. For example, in the TREC Filtering track, one popular utility metric is $T9U = \psi_1 R + \psi_0 N$, where R and N are the number of delivered results the user considered relevant and non-relevant respectively. ψ_1 and ψ_0 are the benefit achieved and loss incurred, by the user due to reading the relevant and non-relevant documents respectively. For TREC-9, $\psi_1 = 2$ and $\psi_0 = -1$.

From the system perspective, the expected utility of delivering a document can be computed as:

$$\mathcal{U}(\mathbf{x}|\mathcal{D}) = \sum_{y\in\{0,1\}} \psi_y P(y|\mathbf{x},\mathcal{D}) \quad (3)$$

Instances with $\mathcal{U}(x) > t$ are delivered, where t is the dissemination threshold learnt via cross-validation or set to 0 (Zhang et al., 2003). In this paper, we focus on the $T9U$ utility metric to guage the system performance, but the ideas are general and maybe extended to other evaluation metrics as well.

3. Our approach

3.1. Multi-Task Adaptive Filtering

Owing to the success of Logistic Regression (LR) classifiers in Adaptive Filtering, we choose a multi-task approach based on LR classifiers. Our chosen approach is depicted graphically in Figure 1. Table 1 lists the

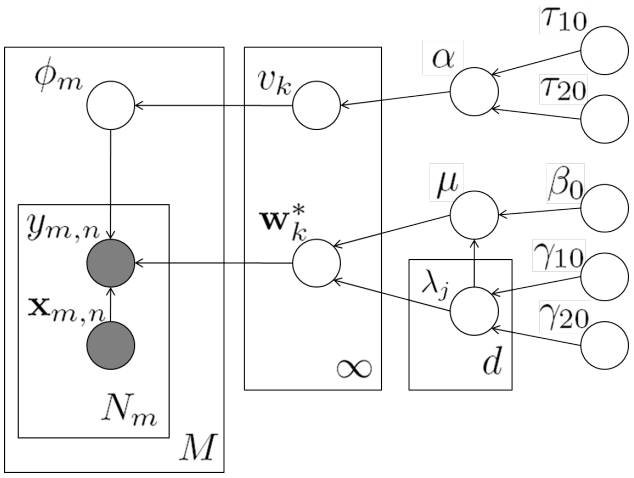

Figure 1. Graphical model representation of multi-task adaptive filtering based on Dirichlet Processes.

Algorithm 1 MTAF Generative Process

1: **Fixed diffuse hyperpriors:** $\tau_{10}, \tau_{20}, \beta_0, \gamma_{10}, \gamma_{20}$
2: Draw $\lambda_j \sim Gamma(\gamma_{10}, \gamma_{20}), \forall j = 1, \ldots, d$
3: Let Λ be a diagonal matrix with elements $\lambda_j, \forall j$
4: Draw $\mu \sim \mathcal{N}(0, (\beta_0 \Lambda)^{-1})$
5: Let $\Sigma = \Lambda^{-1}$
6: Draw $\alpha \sim Gamma(\tau_{10}, \tau_{20})$
7: Draw $v_k \sim Beta(1, \alpha), \forall k = 1, \ldots, \infty$
8: Draw $\mathbf{w}_k^* \sim \mathcal{N}(\mu, \Sigma), \forall k = 1, \ldots, \infty$
9: Let $\pi_k = v_k \prod_{i=1}^{k-1}(1 - v_i), \forall k$
10: **for** $m = 1$ **to** M **do**
11: Draw $\phi_m \sim Multinomial(1; \pi_1, \ldots, \pi_\infty)$
12: Let $\mathbf{w}_m = \sum_{k=1}^{\infty}(\mathbf{w}_k^*)\phi_{m,k}$
13: Draw $y_{m,n} \sim Binomial(1, \sigma(\mathbf{w}_m^T \mathbf{x}_{m,n})), \forall n = 1, \ldots, N_m$
14: **end for**

notation used in this paper. This approach is based on the Mutli-Task classification approach developed by (Xue et al., 2007). A similar approach, consisting of Support Vector Machines for individual tasks, has been used for content-enhanced collaborative filtering by (Yu et al., 2004). We summarize the generative process of the model in Algorithm 1. The approach clusters/groups related tasks by drawing the parameters for the related tasks from a mixture of Gaussians (as evident from the line 12 of the Algorithm 1). Intuitively, related tasks will share information by getting grouped into the same cluster(s). For example, features that are indicators of interests of a football fanatic (e.g. teams, scoreboards, win/loss decisions) may be inferred from the relevance judgments available from other sports enthusiasts. The grouping of related tasks ensures that unrelated tasks (e.g. stock-portfolio tracking vs sports-news filtering) do not contaminate, as such contamination might lead to poorer understanding of the individual tasks.

As the optimal number of groups is unknown apriori, a Dirichlet Process model is inferred over the tasks to discover the optimal number of groups pertinent to the tasks. The Dirichlet Process is essentially a mixture model with potentially *infininte* components. The actual number of components participating in the formation of the tasks is based on the parameter α, also known as the innovation parameter. Larger values of α lead to more participating components and vice-versa. The optimal value of α (and other parameters) can be inferred from the data using variational inference (Xue et al., 2007).

With the knowledge of the inferred parameters for a task m, the decision function for a new instance $\mathbf{x}_{m,\bullet}$ follows the distribution:

$$P(y_{m,\bullet} = 1 | \mathbf{x}_{m,\bullet}, \mathbf{w}_m) = \sum_{k=1}^{K} \phi_{m,k} \sigma(\mathbf{w}_k^{*T} \mathbf{x}_{m,\bullet}) \quad (4)$$

Again, like in Equation 2, for a fully Bayesian treatment, we integrate over the posterior distribution of $\mathbf{w}_k^*$, instead of using a point estimate:

$$P(y_{m,\bullet} = 1 | \mathbf{x}_{m,\bullet}, \mathbf{w}_m) = \sum_{k=1}^{K} \phi_{m,k} \int \sigma(\mathbf{w}_k^{*T} \mathbf{x}_{m,\bullet}) P(\mathbf{w}_k^* | \mu, \Sigma, D) d\mathbf{w}_k^* \quad (5)$$

The above integral does not have an analytical solution. (Xue et al., 2007) suggest the use of an approximate form of the integral based on (MacKay, 1992). Combining Equation 5 and Equation 3, one can compute the expected utility $\mathcal{U}(\mathbf{x}_{m,\bullet})$ of delivering the document $\mathbf{x}_{m,\bullet}$. A passive MTAF approach will deliver the document if $\mathcal{U}(\mathbf{x}_{m,\bullet}) > t$, where t is a dissemination threshold, usually set to zero, or maybe learned via cross-validation. We call this approach *passive*, because in choosing to deliver $\mathbf{x}_{m,\bullet}$, the system did not foresee the benefit of getting the feedback on that document. Consequently, the user effort in providing feedback on this item may go wasted as it may not result in better results (for the user) in the future. We remedy this situation in the next section by proposing our Active Learning framework for the MTAF approach.

3.2. Active Learning for Adaptive Filtering

In a classification setting, an AL approach typically selects instances that (if labeled) are expected to improve the classification performance the most. Along similar lines, in the MTAF setting, an AL approach could estimate the perceived benefit of delivering an item in three different ways. Firstly, does the delivered item $\mathbf{x}_{m,\bullet}$ lead to improved performance on that task m in the future (based on feedback received on $\mathbf{x}_{m,\bullet}$). Secondly, will the feedback on delivered item $\mathbf{x}_{m,\bullet}$ lead to improvements in the global model (e.g. α, μ, Σ)? Finally, does feedback on the delivered item $\mathbf{x}_{m,\bullet}$ lead to improvements in other tasks in the system? We propose an Active Learning solution for each of these objectives in the following sections.

3.2.1. Local Active Learning

In this section, we discuss an AL approach that scores instances based on the perceived future benefit of delivering these instances to the current task m. Consequently, we first define our notion of *future benefit* $\mathcal{LUG}_m(\mathbf{x}_{m,\bullet})$ of delivering an instance to the task m.

$$\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{LSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) - \mathcal{LSP}_m(\mathcal{D}) \quad (6)$$

In the above equation, $\mathcal{LSP}_m(\mathcal{D}')$ denotes the expected system performance on task m when trained on any data $\mathcal{D}'$. Intuitively, the benefit of delivering an instance $\mathbf{x}_{m,\bullet}$ is estimated as the *improvement* in the task-specific performance (for task m), if the feedback of that instance $(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ was added to the training set $\mathcal{D}$. In Equation 6, the system doesn't know the true label $y_{m,\bullet}$ for the instance $\mathbf{x}_{m,\bullet}$. As a result, we rephrase the equation as an expectation over the possible labels.

$$\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}) = \sum_{y\in\{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D)\mathcal{LSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y)) - \mathcal{LSP}_m(\mathcal{D}) \quad (7)$$

For the case of Adaptive Filtering, we define the system performance in terms of the expected utility of all potential instances the system will monitor. Consequently, $\mathcal{LUG}_m$ is the *localized* utility gain of task m.

$$\mathcal{LSP}_m(\mathcal{D}') = \int_{\Omega_{\mathcal{LU}_m}(\mathbf{x}_{m,\circ})} p(\mathbf{x}_{m,\circ})\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}')d\mathbf{x}_{m,\circ} \quad (8)$$

In Equation, 8, there are two important points to note. Firstly, the integral sums over $p(\mathbf{x}_{m,\circ})$, i.e. the task-specific population. It should be noted that each task in the system may have it's own sample population from which documents are being monitored/filtered. For example, a stock analyst may subscribe to news feeds from business magazines, while a sports enthusiast may subscribe to news feeds from sports magazines, leading to different $p(\mathbf{x}_{m,\circ})$ for these two tasks. (Ofcourse, the sports enthusiast is not interested in all sports news being served, but only those that match her interests. It is the goal of the AF system to identify those interests and filter the relevant news from the subscribed feed. Hence the system performance is evaluated over the sample population of the subscribed feed.). In this Equation, the domain of the integral is defined as $\Omega_{\mathcal{LU}_m}(\mathbf{x}_{m,\circ}) = \{\mathbf{x}_{m,\circ} : \mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}') > 0\}$. This means that the future system performance will only be calculated based on the instances that will be delivered; undelivered instances will be ignored from the system performance calculation due to the nature of the $T9U$ utility function.

The second subtle point in Equation 8, is the localized nature of the utility function $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}')$. For computing the expected future system performance $\mathcal{S}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$, we compute the utitlity $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$ based only on the updated *local* model $\mathbf{w}_m$ of the task m. This means, we do not consider the effect of updating the global parameters α, μ, Σ after observing the new training instance $(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$. If $\mathbf{w}_m$ is the current (trained on $\mathcal{D}$) local model, we can use the Bayes' rule to obtain the posterior distribution $P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet})$. We use the variational approximation method suggested by (Jaakkola & Jordan, 1996) to obtain the posterior. With the knowledge of the posterior distribution $P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet})$, the localized expected utility can then be defined as:

$$\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) = \int_{\mathbf{w}_m} \sum_{y\in\{0,1\}} \psi_y P(y|\mathbf{x}_{m,\circ}, \mathbf{w}_m)P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet})d\mathbf{w}_m \quad (9)$$

In the Adaptive Filtering setting, the system will then choose to deliver items that provide immediate utility $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D})$, or are potentially beneficial in improving the system in future iterations, as estimated by expected future utility $\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D})$. We express the joint objective as a linear combination:

$$\mathcal{LAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_L\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \quad (10)$$

Here, δ_L is the weightage given to the AL component

of the system. We discuss the nature of this weight in Section 3.2.5. Owing to the localized focus of this approach on the perceived improvement of the current task, we call this approach Local Active Learning (LAL). We expect LAL to perform well if the global model, consisting of α, μ, Σ, is already strong, and thus does not need to be updated. Thus, LAL is expected to be effective on new tasks that are added to an already well-performing MTAF system. We deal with the case of the improving the global model in the next section.

3.2.2. Global Active Learning

As mentioned in the previous section, LAL does not foresee the updates/improvements in the global model consisting of $\Theta = \{\alpha, \mu, \Sigma\}$. Global AL (GAL) fixes this by updating the global model for computing the expected utility gain in Equation 7. It should be noted that, just like LAL, the goal of GAL is still to improve the future performance of task m (the task to which the system plans to deliver instance $\mathbf{x}_{m,\bullet}$), albeit based on improvements in the global model. Consequently, we define the estimation of the *global* utility function $\mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$. This is a modification of localized utility $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$ in equation 9.

$$\begin{aligned}
&\mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) \\
&= \int_{\Theta} \sum_{y \in \{0,1\}} \psi_y P(y|\mathbf{x}_{m,\circ}, \Theta) \\
&\qquad P(\Theta|\mathbf{x}_{m,\bullet}, y_{m,\bullet}) d\Theta \\
&= \int_{\Theta} \sum_{y \in \{0,1\}} \int_{\mathbf{w}_m} \psi_y P(y|\mathbf{x}_{m,\circ}, \mathbf{w}_m) \\
&\qquad P(\mathbf{w}_m|\Theta)(P(\Theta|\mathbf{x}_{m,\bullet}, y_{m,\bullet}) d\mathbf{w}_m d\Theta
\end{aligned} \tag{11}$$

It is important to note that the posterior model global $P(\Theta|(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ has been trained on additional available feedback from the task m only. This is because, we are computing the expected global future utility if the instance is delivered to task m, that is if the feedback is received from task m. Replacing local utility $\mathcal{LU}$ in Equation 8 with expected future global utility $\mathcal{GU}$, we get the *globalized* system performance $\mathcal{GSP}_m$ of task m.

$$\mathcal{GSP}_m(\mathcal{D}') = \int_{\Omega_{\mathcal{GU}_m}(\mathbf{x}_{m,\circ})} p(\mathbf{x}_{m,\circ}) \mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}') d\mathbf{x}_{m,\circ} \tag{12}$$

Substituting the globalized system performance $\mathcal{GSP}_m$ from Equation 12 into Equation 7, we get the globalized utility gain

$$\begin{aligned}
\mathcal{GUG}_m(\mathbf{x}_{m,\bullet}) = &\sum_{y \in \{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D) \mathcal{GSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y)) \\
&- \mathcal{GSP}_m(\mathcal{D})
\end{aligned} \tag{13}$$

Finally, substituting the globalized utility gain $\mathcal{GUG}_m$ into Equation 14, we get the scoring function for the Global Active Learning GAL approach.

$$\mathcal{GAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_G \mathcal{GUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \tag{14}$$

3.2.3. Benevolent Active Learning

So far, LAL and GAL have focused on improved performance of the task m to which the system plans to deliver the instance $\mathbf{x}_{m,\bullet}$. But it is not clear if the feedback on that instance will lead to improvements in other tasks. To achieve this goal, we devise another AL score, called *Benevolent* AL, as the system tries to deliver instances to task m that might lead to improvements in other tasks. In this context, we revise the definition of $\mathcal{GSP}_m$ to $\mathcal{BSP}_m$, which sums over the performance of each task in the system, based on the feedback received on the instance $\mathbf{x}_{m,\bullet}$ delivered to task m.

$$\begin{aligned}
&\mathcal{BSP}_m(\mathcal{D}') = \\
&\sum_{m'=1}^{M} \int_{\Omega_{\mathcal{GU}_m}(\mathbf{x}_{m',\circ})} p(\mathbf{x}_{m',\circ}) \mathcal{GU}_m(\mathbf{x}_{m',\circ}|\mathcal{D}') d\mathbf{x}_{m',\circ}
\end{aligned} \tag{15}$$

It is important to note that inside the summation for each task m', the instances $\mathbf{x}_{m',\circ}$ are being sampled from the distribution $p(\mathbf{x}_{m',\circ})$ of incoming documents for that particular task m'. This stems from the fact that the distribution of incoming instances $P(\mathbf{x}_m)$ may be different for different tasks, as discussed earlier. Based on $\mathcal{BSP}_m$, for clarity and completeness, we provide the Equations of the corresponding utility gain $\mathcal{BUG}_m$ and AL scores $\mathcal{BAL}_m$.

$$\begin{aligned}
&\mathcal{BUG}_m(\mathbf{x}_{m,\bullet}) = \\
&\sum_{y \in \{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D) \mathcal{BSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y)) - \mathcal{BSP}_m(\mathcal{D})
\end{aligned} \tag{16}$$

$$\mathcal{BAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_B \mathcal{BUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \tag{17}$$

3.2.4. Analysis of LAL, GAL, and BAL

In Table 2, we summarize the major ideas from proposed AL approaches. Each approach focuses on improvements in different parameters of the model, thereby leading to different consequences. The LAL approach focuses on improving its own task. Such approach can be expected to perform well when the global model is already strong, and less susceptible to drastic change based on feedback on a new instance. The GAL approach restimates the global model, but in the process computes the future utility only based on the current task. This selfish strategy of modifying the global model without foreseeing the effect on other tasks in the system can be detrimental to the overall performance of the model, making this approach undesirable. The BAL approach updates the global model, but at the same time studies the effect of the update on the utility gain of other tasks in the system, thereby ensuring that the global model doesn't get biased towards a particular task. Consequently, we expect BAL to perform superior to other methods in the initial stages of AF when global model is weak.

Table 2. A comparative summary of AL approaches. The first column lists the AL score to decide delivery of instance $\mathbf{x}_{m,\bullet}$ to task m. The second column lists the parameters that will be (potentially) improved (for better future utility) if the system retrains on the feedback received on the instance delivered based on the corresponding AL score

Delivery criteria	Potentially improved parameters based on feedback
$\mathcal{LAL}_m$	$\mathbf{w}_m$
$\mathcal{GAL}_m$	$\mathbf{w}_m, \alpha, \mu, \Sigma$
$\mathcal{BAL}_m$	$\alpha, \mu, \Sigma, \mathbf{w}_{m'}, \forall m' \in \{1, \ldots, M\}$

3.2.5. Combined AL for MTAF

Based on the discussion in the previous section, each of the above methods can be combined to come up with a meta-AL system for AF that tries to satisfy multiple objectives at different phases of learning. We use weighted linear combination of the individual utility gain scores to come up with the Meta AL score

$$\mathcal{MAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_L \mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_G \mathcal{GUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_B \mathcal{BUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \quad (18)$$

The weight parameters δ_L, δ_G and δ_B should vary with the quality of the model, local as well as global. A stronger global model means lower δ_G. Similarly, δ_B should decrease so that the tasks can then focus on improving locally, thereby maintaining a higher value of δ_L. We test these hypotheses empirically in Section 4.3

4. Experiments

4.1. Datasets

We chose to use 2 datasets that are popular in the TREC filtering community, namely, RCV1 (84 categories, 810,000 documents) (Lewis et al., 2004) and 20 Newsgroups (20 categories, 18,846 documents) (Joachims, 1997). For both datasets, we modeled each category as a task. For experiments on both datasets, we start AF with only one known relevant document per task. It should be noted that RCV1 is a multi-labeled dataset, meaning each document belongs to multiple classes/tasks, thereby indicating some level of overlap/relatedness among tasks (e.g. tasks sharing the same documents). 20 Newsgroups is not multi-labeled and each document belongs to only one class. However, it is known that the categories can be grouped into 6 groups based on subject matter, namely, comp.*, rec.*, sci.*, talk.*, misc.* and others. Thus, the *relatedness* of tasks is in the feature-space, as documents are not shared among tasks. For example, the feature *computer* is a strong indicator of the tasks in comp.* group. We chose this dataset to see if the MTAF approaches are able to discover these hidden groups of related categories, even if the documents are not shared among tasks.

4.2. Methods

Primarily, we are interested in comparing the four Active Learning approaches LAL, GAL, BAL and MAL in the MTAF setting to see which approach rapidly improves utility, and how they perform in the various phases of filtering. We also wish to compare these 4 approaches to the passive version that decides to deliver an instance without taking any future utility gain into account. (i.e. delivery criteria is $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) > 0$).

4.2.1. STAF baselines

(Zhang et al., 2003) devised a strategy called *exploration and exploitation* to perform Active Adaptive Filtering, and our AL framework is inspired by their work. The *exploitation* component is a passive AF strategy that delivers an item if $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) > 0$. The *exploration* component is an Active Learning component based on a metric called *Utility Divergence*, similar to the *Utility Gain* score developed in this paper. Utility Divergence, however, doesn't com-

pute the actual gain in utility of the system based on feedback on the instance. It instead computes the difference between the expected utility of a hypothetical *true* model Θ^* and the model based on expected feedback $\Theta|(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ and instances are scored based on their potential to reduce the gap between utility of the true model and utility of the current model. In our results, we will refer to these approaches as STAF-passive (only exploitation), STAF-active-UD (exploitation and exploration using Utility Divergence score).

4.2.2. A NOTE ON IMPLEMENTATION

The various methods were implemented in MATLAB. For integrating over the posterior distribution of a variable (e.g. w_k^*), we used the Metropolis Hastings algorithm. Specifically, we draw several (S) samples from the posterior of the corresponding distribution of the variable, and averaged over the various outputs to obtain a probabilistic integral, similar to the approach described in (Zhang et al., 2003). For example, to implement Equation 3, we sample S samples from the posterior distribution $P(\mathbf{w}|D)$ and the integration was implemented as:

$$\mathcal{U}(\mathbf{x}|\mathcal{D}) = \frac{1}{S}\sum_{s=1}^{S}\sum_{y\in\{0,1\}} \psi_y P(y|\mathbf{x}, \mathbf{w}^{(s)}) \qquad (19)$$

In this specific case, the posterior distribution was derived by using the Laplace approximation method (Xue et al., 2007). Other posterior distributions for the MTAF case were based on the derivations available in (Xue et al., 2007). For integrating over $P(\mathbf{x})$, the samples were drawn from the data stream observed so far, consisting of delivered and undelivered items, to get a non-parametric estimate of $P(\mathbf{x})$.

4.3. Experimental Results and Discussion

First we compare the MTAF approaches (MTAF-Passive, LAL, GAL, BAL, and MAL) to test the hypotheses we made in Section 3.2.4. Figure 2 shows the trends (in terms of T9U utility) on the RCV1 dataset. AL approaches are typically most effective in the initial phases of filtering and so the Figure 2 shows performance upto the filtering of the first 5000 instances on the RCV1 dataset. The results validate our hypothesis that the LAL approach performs quite poorly in the initial stages of filtering, when the global model itself is quite weak. It can also be observed that BAL approach, that tries to improve the overall utility of all the tasks in the system performs the best initially. This is because improved utility of all tasks reflects in an stronger global model. LAL improves at a faster pace once the learnt global model is stronger (due to learning from more feedback). The GAL approach, that tries to improve the global model, without foreseeing improvements (or degradation) in other task performs inferior to the LAL approach. We believe that the GAL approach sometimes makes wrong decisions about instance selection, by selecting those instances that will lead to improvement in one task, but mostly degrade other tasks (because the utility gain in tasks is not taken into account). We found that setting $\delta_G = 0$ in Equation 18, i.e. ignoring the $\mathcal{GUG}_m$ component corresponding to GAL, in the MAL approach led to better performance of MAL. Consequently, the MAL approach, which combines the strengths of the LAL and the BAL approaches, outperforms all the other approaches.

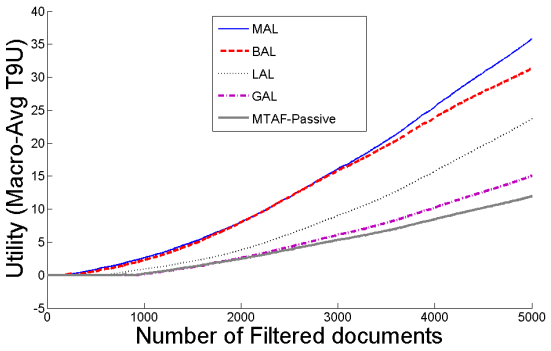

Figure 2. Comparison of the MTAF AL approaches on the RCV1 dataset (similar trends for 20 Newsgroups)

Next, we compare representative MTAF approaches (MTAF-passive and MTAF-MAL) to the STAF approaches to study the benefit of using a multi-tasking setup. We also compare the AL scoring functions: our *Utility Gain* (UG) criteria and the *Utility Divergence* (UD) criteria proposed in (Zhang et al., 2003). We call these variants of the STAF-active approach STAF-active-UG and STAF-active-UD respectively. In Figure 3, we observe that the MTAF approaches, active as well as passive, outperform their STAF counterparts. It can be observed that our best performing approach, MAL (T9U = 423), has a performance improvement of more than 20 percent over that of STAF-active-UD (T9U=348). A paired t-test shows strong statistically significant evidence (p-value = 0.0002) for the superiority of our approach over the current state-of-art STAF-active-UD. We also observe that the starting utility of the MTAF approaches is higher, due to information sharing between tasks, to overcome the per-task data sparsity problem in the STAF approaches.

Regarding the AL score, we observe that the performance of UG and UD is quite similar for STAF-active. Empirically, it seems that the goal of UD (reduce the gap between the utility of a hypothetical true model and the learnt model) is quite similar to the goal of UG (increase the future utility of the learnt model the most).

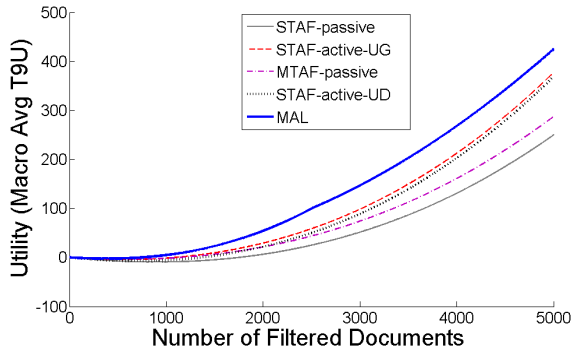

Figure 3. Comparison of MTAF and STAF approaches on the 20 Newsgroups dataset (similar trends for RCV1)

5. Conclusion and Future Work

In this paper, we have explored various Active Learning approaches to the Multi-Task Adaptive Filtering. To score the benefit of delivering an instance, we developed a new metric called *Utility Gain* that estimates the improvement in system performance (in terms of utility) if the system is re-trained on the feedback received for the delivered instance. In the MTAF setting, we compare the effect of selfish (local) AL approaches (that focus on improvements in one task) to a benevolent AL approach (that evaluates the benefit of labeling an instance across many tasks). Our empirical analysis demonstrates the superior performance of the benevolent approach in the initial phase of filtering, when the global model of the MTAF system is weak, while a more rapid improvement in the performance of the selfish approaches in later stages, when the global model is strong (i.e. global model may not benefit much from AL). We also demonstrate that a combined approach, called meta-AL, that combines the strengths of local and benevolent AL approaches, is superior to the individual approaches.

There are several areas for exploration in the future. The sampling approaches presented in this work can be infeasible in a large-scale adaptive filtering system with millions of tasks. In such an environment, it is necessary to first segregate the tasks so that each of the smaller AF systems can handle their respective tasks. It is also necessary to derive analytic solutions to the sampling strategies described here to come up with closed-form/quicker expected future utility evaluation schemes. Another challenge is the problem of spam. In a practical MTAF system, how does the MTAF system protect its genuine users/tasks from other malicious users. In a benevolent AL approach the feedback from the malicious users is potentially harmful to the system, and consequently to the genuine users. So how does a Benevolent AL approach safeguard againt malicious use?

References

Blei, David M. and Jordan, Michael I. Variational inference for dirichlet process mixtures, 2006.

Jaakkola, Tommi S. and Jordan, Michael I. A variational approach to bayesian logistic regression models and their extensions, 1996.

Joachims, Thorsten. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. pp. 143–151, 1997.

Zhang, Yi, Xu, Wei, and Callan, Jamie. Exploration and exploitation in adaptive filtering based on. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003*, pp. 896–903, 2003.

MacKay, David J.C. The evidence framework applied to classification networks. *Neural Computation*, 4: 720–736, 1992.

Stephen Robertson, Robertson, Stephen, and Soboroff, Ian. The trec 2002 filtering track report. In *In TREC-11*, 2001.

Xue, Ya, Liao, Xuejun, Carin, Lawrence, and Krishnapuram, Balaji. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:2007, 2007.

Yu, Kai, Tresp, Volker, and Yu, Shipeng. A nonparametric hierarchical bayesian framework for information filtering, 2004.

David Lewis, Yiming Yang, T. Rose and Fan Li. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5 (2004) 361-397.

Melville, P., Saar-Tsechansky, M., Provost, F., and Mooney, R. 2005. An Expected Utility Approach to Active Feature-Value Acquisition. In Proceedings of the Fifth IEEE international Conference on Data Mining (November 27 - 30, 2005).