Notes on a Formal Framework for Adaptive Filtering

Paul Kantor & David Madigan

December 1, 2002

We consider algorithms for the following text filtering problem. Documents arrive one at a time and the task is to identify "interesting" documents from the stream of documents. As each document arrives, the algorithm has to decide whether or not to present the document to an oracle. If the algorithm presents document i to the oracle and if document i is indeed interesting, the oracle returns $y_i = 1$ and rewards the algorithm r utiles. If the algorithm presents document i to the oracle and if document i is not interesting, the oracle returns $y_i = 0$ and penalizes the algorithm c utiles. The goal is is to maximize expected utiles over a fixed horizon K. We refer to this problem as "adaptive filtering."

This is a sequential decision making problem. Such problems have attracted much attention in medical literature (see, for example, Lavori and Dawson, 2000, and van der Laan, Murphy, and Robbins, 2002) and the engineering literature (see, for example, Kaelbling, Littman, and Cassandra, 1998).

Our goal in this present work is to gain some understanding of the interrelations among different aspects of adaptive filtering by building a formal model of the filtering process as a multistage decision problem. The hope is that this will lead to insights that will prove useful as we develop practical methodologies and algorithms for monitoring message streams.

In what follows we consider two versions of this problem. The first version considers a simplified setup where document feature space is discrete and all interesting

documents share the same features. The second version is more general but involves modeling assumptions.

1 Discrete Features, Single Bin

Suppose the filtering algorithm represents document i using a discrete-valued feature x_i that takes N possible values ("bins"), that is, $x_i \in \{1, \ldots, N\}, i = 1, \ldots, K$. Further suppose that all the interesting documents reside in one bin, that $Pr(x_i = j) = 1/N, i = 1, \ldots, K, j = 1, \ldots, N$, and that the stream produces documents that are independent and identically distributed according to this distribution. Denote by G the index of the single bin containing interesting documents. Assume that $Pr(G = i) = 1/N, i = 1, \ldots, N$ and:

$$Pr(y_i = 1 | x_i, G = g) = \begin{cases} p & \text{if } x_i = g \\ 0 & \text{otherwise,} \end{cases}$$

where p is assumed known. Figure 1 shows a probabilistic graphical model representation.

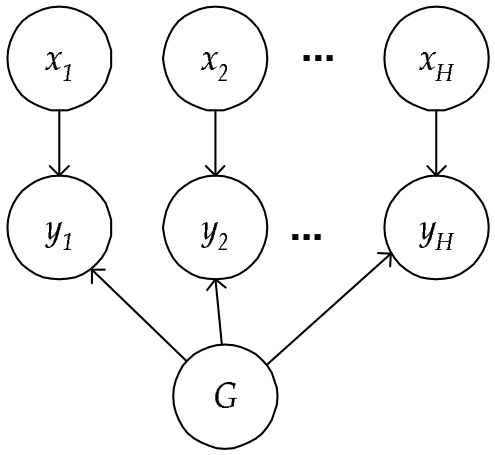

Figure 1: *Model for discrete features, single bin.*

Let $a_i = 1$ if the algorithm presents document i to the oracle and $a_i = 0$ otherwise. Let Y_i denote the reward associated with document i, so $Y_i = r$ if $a_i = 1$ and $y_i = 1$, $Y_i = -c$ if $a_i = 1$ and $y_i = 0$, and $Y_i = 0$ if $a_i = 0$. In general we use a bar

over a variable to denote that variable and all past values of the same variable, so $\bar{a}_j = (a_1, \ldots, a_j)$.

The goal is to choose the series of decisions $a_1, \ldots, a_K$ that maximizes the expected value of $Y \equiv \sum_{i=1}^{K} Y_i$. Backward induction (dynamic programming) can find the optimal decisions. These arguments are usually expressed as follows (Bather, 2000; Murphy, 2002). Set:

$$J_0(\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1}) = \sup_{a_K} E\left[Y|\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1}, a_K\right] \quad (1)$$

$$d_K^*(\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1}) = \arg\sup_{a_K} E\left[Y|\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1}, a_K\right] \quad (2)$$

and then for each j:

$$J_{K-j}(\bar{y}_{j-1}, \bar{x}_j, \bar{a}_{j-1}) = \sup_{a_j} E\left[J_{K-j-1}(\bar{y}_j, \bar{x}_{j+1}, \bar{a}_j)|\bar{y}_{j-1}, \bar{x}_j, \bar{a}_{j-1}, a_j\right] \quad (3)$$

$$d_j^*(\bar{y}_{j-1}, \bar{x}_j, \bar{a}_{j-1}) = \arg\sup_{a_j} E\left[J_{K-j-1}(\bar{y}_j, \bar{x}_{j+1}, \bar{a}_j)|\bar{y}_{j-1}, \bar{x}_j, \bar{a}_{j-1}, a_j\right]. \quad (4)$$

The optimal decisions are $a_1 = d_1^*, \ldots, a_K = d_K^*$.

The key input to the computation of (1) is $Pr(y_K = 1|\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1})$. Denote by $i_1, \ldots, i_m$ the indices of the documents sent to the oracle and $j_1, \ldots, j_l$ the indices of the documents not sent to the oracle. Note that $\{i_1, \ldots, i_m\} \cup \{j_1, \ldots, j_l\} = \{1, \ldots, K-1\}$. Now,

$$\begin{aligned}
& Pr(y_K = 1|\bar{y}_{K-1}, \bar{x}_K, \bar{a}_{K-1}) \\
= \; & Pr(y_K = 1|\bar{x}_K, y_{i1}, \ldots, y_{im}) \\
= \; & \sum_{g=1}^{N} \sum_{z_1=0}^{1} \cdots \sum_{z_l=0}^{1} Pr(y_K = 1, y_{j1} = z_1, \ldots, y_{jl} = z_l, G = g|\bar{x}_K, y_{i1}, \ldots, y_{im}) \\
= \; & \sum_{g=1}^{N} \sum_{z_1=0}^{1} \cdots \sum_{z_l=0}^{1} Pr(y_K = 1|G = g, x_K) Pr(y_{j1} = z_1, \ldots, y_{jl} = z_l|G = g, \bar{x}_K) \\
& \times Pr(G = g|\bar{x}_K, y_{i1}, \ldots, y_{im}) \\
= \; & \sum_{g=1}^{N} Pr(y_K = 1|G = g, x_K) Pr(G = g|\bar{x}_K, y_{i1}, \ldots, y_{im}) \\
\propto \; & \sum_{g=1}^{N} \left(Pr(y_K = 1|G = g, x_K) Pr(G = g) \prod_{k=1}^{m} Pr(y_{ik}|G = g, x_{ik}) \right)
\end{aligned}$$

so that,

$$Pr(y_K = 1|\bar{x}_K, y_{i1}, \ldots, y_{im}) = \frac{pPr(G = x_K) \prod_{k=1}^{m} Pr(y_{ik}|G = x_K, x_{ik})}{\sum_{g=1}^{N} Pr(G = g) \prod_{k=1}^{m} Pr(y_{ik}|G = g, x_{ik})}$$

Example. Suppose $N = 3$, $K = 6$, $\bar{y}_5 = (0, 0, 0, 0, 0)$, and $\bar{x}_6 = (1, 1, 2, 3, 3, 3)$. Then:

$$Pr(y_6 = 1 | \bar{y}_5 = (0, 0, 0, 0, 0), \bar{x}_6 = (1, 1, 2, 3, 3, 3)) = \frac{p(1-p)}{3-2p}.$$

If $p = 1/2$, for example, then this is $1/8$.

Example. Suppose $N = 3$, $K = 6$, $\bar{y}_5 = (0, 0, 0, 0, 0)$, and $\bar{x}_6 = (3, 3, 3, 3, 3, 3)$. Then:

$$Pr(y_6 = 1 | \bar{y}_5 = (0, 0, 0, 0, 0), \bar{x}_6 = (3, 3, 3, 3, 3, 3)) = \frac{p(1-p)^5}{2 + (1-p)^5}.$$

If $p = 1/2$, for example, then this is $1/130$.

Example. Suppose $N = 3$, $K = 6$, $\bar{y}_5 = (0, 0, 0, 0, 0)$, and $\bar{x}_6 = (1, 1, 1, 2, 2, 3)$. Then:

$$Pr(y_6 = 1 | \bar{y}_5 = (0, 0, 0, 0, 0), \bar{x}_6 = (1, 1, 1, 2, 2, 3)) = \frac{p}{1 + (1-p)^3 + (1-p)^2}.$$

If $p = 1/2$, for example, then this is $8/22$.

Example. Suppose $N = 3$, $K = 6$, $\bar{y}_5 = (0, 0, 0, 0, 1)$, and $\bar{x}_6 = (1, 1, 1, 2, 2, 3)$. Then:

$$Pr(y_6 = 1 | \bar{y}_5 = (0, 0, 0, 0, 1), \bar{x}_6 = (1, 1, 1, 2, 2, 3)) = 0.$$

Example. Suppose $N = 3$, $K = 6$, $\bar{y}_5 = (0, 0, 0, 0, 1)$, and $\bar{x}_6 = (1, 1, 1, 2, 3, 3)$. Then:

$$Pr(y_6 = 1 | \bar{y}_5 = (0, 0, 0, 0, 1), \bar{x}_6 = (1, 1, 1, 2, 3, 3)) = \frac{p^2}{p} = p$$

If $p = 1/2$, for example, then this is $1/2$.

2 Heuristic Algorithm for the Discrete Features, Single Bin Model

In spite of the sheer complexity of the formal model and dynamic programming approach, there is hope for a "forward heuristic". This stems from the fact that the decision made at a given step k in the decision process can depend only on the information at hand. This consists of the "distance to go" $K - k$ and the present information about the potential of each of the several bins to be the one containing

the relevant documents. Note that (this is a special feature of our simplified model) as soon as a bin is found to contain a positive document, all uncertainty is removed. And when uncertainty is removed the dominant strategy is clear: submit a document for judgment if and only if it comes from the bin labeled G.

The distribution of score resulting from this strategy, if the "good bin" is found at step k is the sum of $K-k$ i.i.d Binomial random variables taking the value r with probability $\pi = p/N$. We write this variable as

$$\mathbf{Y} = \sum_{\mathbf{l=1}}^{\mathbf{l=K-k}} \mathbf{X_l}$$

The characteristic function of this variable is defined in terms if the characteristic function $\psi_{\mathbf{X}}(z) = \pi z^r + (1-\pi)z^{-c}$ and takes the value

$$\psi_{\mathbf{Y}}(z) = \psi_{\mathbf{X}}(z)^{K-k}$$

The difficult problem is determining the distribution of the state of affairs at the point when the correct bin is discovered. We can begin an attack on this problem by reasoning as follows. At any point in the process the information available to the decision maker consists of precisely two things: (1) the number of steps to go $(K-k)$, which refer to for the time being as the "working horizon" H, and the history of which bins have already accumulated some number of negative judgments.

Since the bins are indistinguishable this latter information can be completely summarized for our purposes by an "information state" given by $s(n)$ where $s(n)$ is the number of bins that have accumulated exactly n negative judgments so far. Note that at step k there have been $k-1$ judgments, and so $\sum ns(n) = k-1$. As noted above, the effect of a negative judgment is to rescale the probability that a bin is the correct one, by a factor $1-p$. Therefore, we need only consider a modified distribution, in which each number of counts is replaced by its excess over the minimum number of counts observed. This transformation controls the complexity when the number of bins is small. When the number of bins is large it will take longer to have any effect. [Intuitively, the expected waiting time to submit the correct bin for judgment is N, and the waiting time to submit it and get a useful (i.e. positive) judgment is N/p].

For simplicity, in this exposition we do not use the modified distribution. Now, it seems reasonable that at any step the optimal strategy will take this form: if the probability that the bin at hand is the correct bin exceeds some threshold probability p_t then it will be submitted for judgment. Otherwise we will "pass" and wait for a better bin to come along.

Let $n(b)$ be the number of negative judgments accumulated by bin b at the present instant (that is, when we are about to take decision k). Since $p(b)$ is proportional to $(1-p)^{n(b)}$, there corresponds to p_t a threshold value of $n = n_t$. Bins b with $n(b) < n_t$ are precisely those with $p(b) > p_t$. Hence the probability that the next bin to be presented will have probability greater than p_t of being the right one is precisely $w(p_t) = \sum_{n<n_t} s(n)/N$. We shall also need the average probability that a bin is the correct one, for those bins whose probability of being correct exceeds p_t. We denote this by $p(p_t)$

As we proceed, the distribution of the numbers $s(n)$ will change. For the moment (this is our "heuristic step") we set aside this important complication. We ask therefore: what is the probability that we will have discovered the correct bin, if we keep the threshold fixed at p_t, assume the distribution $s(n)$ does not change, and submit for judgment only those bins having $p > p_t$?

The probability to have the correct bin at exactly the $h-th$ step from now, and having accumulated exactly m negative judgments (the m is for $m$isses) is given by

$$\phi(h,m) = w(p_t)p(p_t)\begin{pmatrix} h-1 \\ m-1 \end{pmatrix}(1-p(p_t))^{m-1}(1-w(p_t))^{h-m}$$

That is, the $m-1$ misses are scattered anywhere in the previous $h-1$ trials, and the remaining trials are bins not worth judging.

The expected value of following the strategy $\mathbf{S(p_t)}$ defined by p_t, assuming no change in the distribution, is

$$\mathbf{ES(p_t)} = \sum_{\mathbf{h,m}} \phi(\mathbf{h,m})\left(-\mathbf{mc} + \mathbf{r} + \mathbf{E(H-h|know_G)}\right)$$

We can introduce a characteristic function ψ with two arguments z and y, defined by

$$\psi(y,z) = \sum_{h \leq H; m \leq h} \phi(h,m) y^h z^m$$

This can be reduced to elementary forms, by simple algebra.

Then, we can use the fact that the characteristic function for the conditioned process that includes finding G at (h,m) and then behaving optimally is given by composition of the characteristic functions. This yields a cumbersome expression from which the expected value of the strategy $\mathbf{S}(\mathbf{p_t})$ may be calculated directly. The single parameter p_t may be scanned to find the optimum. It is, as shown by the general argument preceding, a function only of the remaining horizon $\mathbf{H}$ and the current state of knowledge, as summarized by $s(n)$.

2.1 Anticipated problems and refinements

Note: there are some problems with doing this, since it is not intuitively clear that it will enable us to get started in situations where the accumulation of information plays a vital role in being able to beat the limited horizon. If there are indeed such situations. These are situations in which the strategy of submitting every documen for judgment is not as effective as one which begins to withhold documents from bins not likely to be the correct one. Perhaps a heuristic can be found to identify these situations.

A complete analysis would require also that we know how the state of knowledge $s(n)$ changes as a result of new knowledge. In this heuristic model that information is lost, because we do not know which of the bins the document sent for judgment came from. Thus we do not know the precise new (updated) distribution $s(n)$.

Numerical simulations are being performed to explore the importance of this deficiency.

3 Influence Diagrams and Learning Curves

Influence diagrams (Howard and Matheson, 1984) are compact representations of decision problems under uncertainty. Local computation algorithms for solving influence diagrams exist in various forms - see, for example, Shachter (1996), Jensen *et al.* (1984), and Lauritzen and Nielsen (2001). Here we propose an influence diagram model for the adaptive filtering problem.

Influence diagrams are acyclic directed graphs (ADGs) with three types of nodes. *Chance nodes*, displayed as circles, represent random variables. *Decision nodes*, displayed as squares, correspond to alternative choices available to a decision maker. Finally, *value nodes*, displayed as diamonds, represent additive components of the joint utility function.

Figure 2 shows an influence diagram representation of the adaptive filtering problem. In this influence diagram, y_i represents the true state of the ith document. That is, $y_i = 1$ if document i is "interesting" and $y_i = 0$ otherwise. The chance node $\hat{y}_i$ represents the predicted value for y_i and also takes values zero and one. We assume that some predictive model such as Naive Bayes or probit regression outputs $\hat{y}_i$. The decision node d_i represents the decision to send ($d_i = 1$) or not to send ($d_i = 0$) document i to the oracle. The value node u_i depends on d_i and y_i and is either $+r$, zero, or $-c$. The chance node s_i, which may be multivariate, represents the "state" of the model before processing the ith document.

Note that if the prediction for document i is known to be perfect, that is, $Pr(\hat{y}_i = y_i) \equiv 1$, then $d_i = 1$ if and only if $p > \frac{c}{r+c}$. In general, however, predictions will be fallible. Furthermore, we assume that the predictive model updates itself as documents are sent to the oracle. Hence the the quality of the predictions, i.e., $Pr(\hat{y}_i = y_i)$, will vary over time.

The essential idea of the influence diagram model of Figure 2 is to model $Pr(\hat{y}_i = y_i)$ as a function of s_i, where s_i represents summary statistics pertaining to $\bar{y}_{i-1}, \bar{x}_i$, and $\bar{\hat{y}}_i$. Note that $\bar{y}_{i-1}$ will only contain values for those documents sent to the oracle. One simplistic version of this setup would be to let $s_i = \sum_{j=1}^{i-1} d_j$, that is, the number of documents sent to the oracle. Figures 3 and 4 show two empirical analyses plotting $Pr(\hat{y} = y)$ versus sample size. In both cases, the horizontal axis is analagous to the

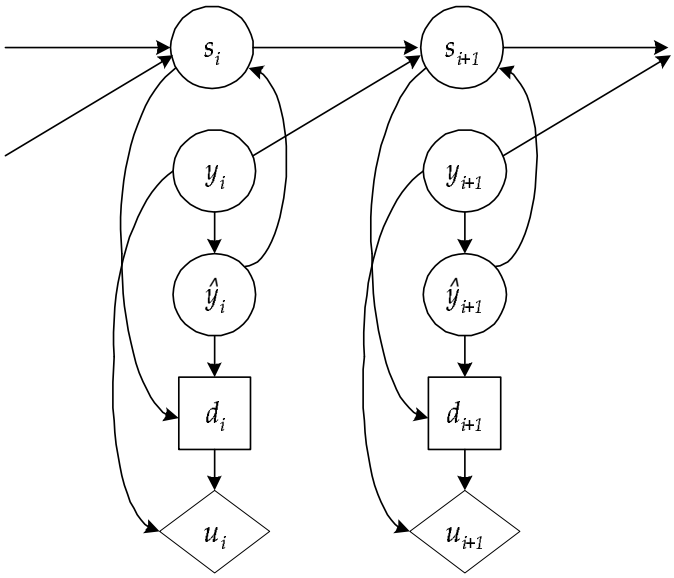

Figure 2: *Influence diagram model for adaptive filtering.*

number of documents sent to the oracle and the vertical axis is analagous to $Pr(\hat{y} = y)$ for a randomly selected future document.

The literature contains many examples of such curves. Given a fully specified learning curve model, standard influence diagram algorithms (e.g., Lauritzen and Nilsson, 2001) can solve the resulting decsision problem through local calculations.

Acknowledgements

We are grateful to Fred Roberts for useful discussions.

References

Bather, J. (2000). *Decision Theory: An Introduction to Dynamic Programming and Sequential Decisions.* Chichester, U.K.: Wiley.

Howard, R.A. and Matheson, J.E. (1984). Influence diagrams. In: *Redings in the Principles and Applications of Decision Analysis.* R.A. Howard and J.E. Matheson

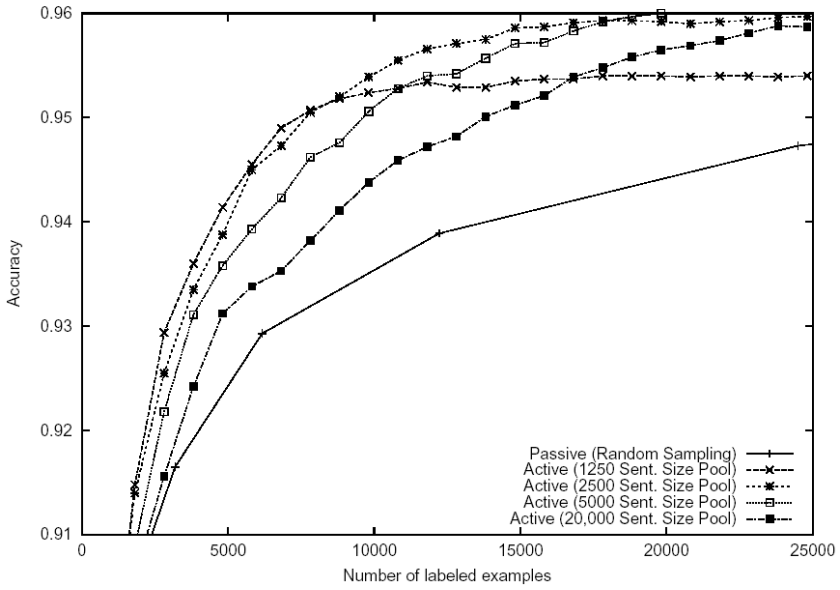

Figure 3: *Learning curves from Sassano (2002).*

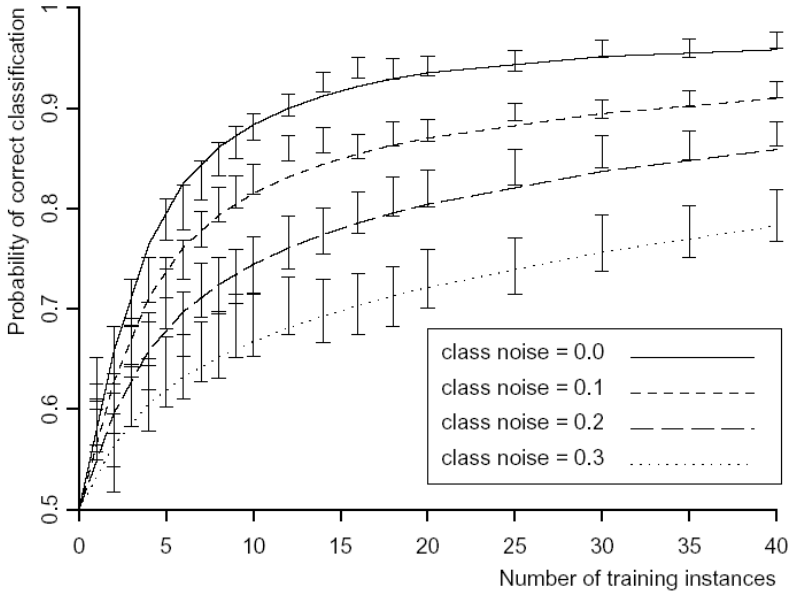

Figure 4: *Learning curves from Langley and Sage (1999).*

(Editors), Strategic Decisions Group, Menlo Park.

Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, **101**, 99–134.

Langley, P. and Sage, S. (1999). Tractable average-case analysis of Naive Bayesian classifiers. In: *Proceedings of the 16th Annual Conference on Machine Learning*, 220-228.

Laurizten, S.L. and Nilsson, D. (2001). Representing and solving decision problems with limited information. *Management Science*, **47**, 1235–1251.

Lavori, P.W. and Dawson, R. (2000). A design for testing clinical strategies: biased adaptive within-subject randomization. *Journal of the Royal Statistical Society (Series A)*, **163**, 29–38.

Murphy, S.A. (2002). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society (Series B)*, to appear. http://www.stat.lsa.umich.edu/ samurphy/papers/optimal.pdf

Sassano, M. (2002). An empirical study of support vector machines for Japanese word segmentation. In: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 505–512.

Shachter, R. (1986). Evaluating decision diagrams. *Operations Research*, **34**, 871–882.

van der Laan, M.J., Murphy, S.A., and Robbins, J.M. (2002). Analyzing dynamic regimes using structural nested mean models. *Journal of the American Statistical Association*, to appear.