Approximate Dynamic Programming in Knowledge Discovery for Rapid Response

Peter Frazier
Princeton University
pfrazier@Princeton.edu

Warren Powell
Princeton University
powell@princeton.edu

Savas Dayanik
Princeton University
sdayanik@princeton.edu

Paul Kantor
Rutgers University
kantor@scils.rutgers.edu

Abstract

One knowledge discovery problem in the rapid response setting is the **cost** *of learning which patterns are indicative of a threat. This typically involves a detailed follow-through, such as review of documents and information by a skilled analyst, or detailed examination of a vehicle at a border crossing point, in deciding which suspicious vehicles require investigation. Assessing various strategies and decision rules means we must compare not only the short term effectiveness of interrupting a specific traveler, or forwarding a specific document to an analyst, but we must also weigh the potential improvement in our profiles that results even from sending a "false alarm". We show that this problem can be recast as a dynamic programming problem with, unfortunately, a huge state space. Several specific heuristics are introduced to provide improved approximations to the solution. The problems of obtaining real-world data to sharpen the analysis are discussed briefly.*

1. Introduction

In KD for rapid response the signs or characteristics presented by a case (e.g., intelligence information, a traveler at a border or airport) are imperfectly related to the condition of interest (a threat, a smuggler, etc). A prompt decision must be made as to how to deal with the case at hand. In particular, we learn nothing about the true situation without some additional and expensive processing (detailed review by an analyst, a search an interview). For the expense of collecting this information, we learn something not only about a particular case, but also about how well a particular set of attributes correlates with a positive outcome (the cargo represents a threat). The decision to collect this information, then, must balance the cost (and, probabilistically, the value) of having the information now, and the value of the improving our ability to make better predictions in the future, even if we are wrong about this particular case.

Whatever the current state of our rule or algorithm, each case has some estimated immediate value, and some potential to improve our future decisions. We must balance the value of making the best decision now against the value of learning now to benefit future decisions. We address this using dynamic programming. Specifically, the state of the system is its state of knowledge relating the attributes of the event (cargo coming through a port, a website) to the likelihood of a positive outcome (the cargo is carrying dangerous cargo, the website has valuable information). Of course, the dynamic program in its full form is computationally intractable.

We need near-optimal policies that strike a balance between making good decisions now and learning for the future. We show some specific theoretical properties (asymptotic optimality, provable bounds), and methods that approximate the optimal policy as characterized by a rigorous dynamic program. We assess the *distribution of the effectiveness* of any specific procedure. This work is related to the "multi-armed bandit" problem, a special case that has been solved optimally in a practical way using Gittins indices. That literature, however, does not recognize that evaluating one case teaches you something about other cases with different (but related) attributes. The situation is comparable to the benefit of traveling one path over a stochastic network, and then ignoring what we learn about the cost of another, overlapping path. The best strategy may be to travel a path that overlaps with many other paths.

2. Previous related work

This work draws on prior work in approximate solution of dynamic programming problems, analysis of change point detection using Markov models, and assessment of the performance of detection systems, using ROC analysis, and specific models of the behavior of algorithms.

2.1 Prior work on optimal information collection

The problem of optimally collecting information is an old one, but progress has generally been made in narrow problem classes. Early work on optimal decisions in a statistical setting generally addressed problems such as sequential hypothesis testing (see [1] and [2] for reviews), ranking and selection [3] and sequential design of experiments [4]. Cohn et al. [5] provide a method for optimally collecting information in a machine learning setting. The classic bandit problems for making choices while learning from the experience of making each choice saw a resurgence in interest with the discovery of Gittins indices [6, 7]. There have been many attempts to generalize the basic model, but the results remain quite limited. Gittins indices can be applied heuristically, and we can use procedures such as the knowledge gradient [8] which has nice theoretical properties but is not generally optimal.

Information collection problems can be formulated as dynamic programs, but these cannot be solved to optimality. Duff & Barto [9] propose a method for discrete state, discrete action dynamic programs to learn optimally, but this strategy has not been widely adopted. We investigate the structure of optimal policies using classical theory, and we use approximate dynamic programming (ADP) to compute near-optimal policies, and to evaluate more convenient heuristics. ADP has proven successful in a range of resource allocation problems [10, 11, 12].

The simulation community has addressed the information collection problem in the context of choosing parameters to control a simulation (see [13] and [14]). Each particular parameter setting usually produces a noisy response, and the time required to measure the performance of a single parameter setting may be significant. The literature on optimal computing budget allocation (OCBA) has focused on finding methods for collecting this information as quickly as possible (see [15, 16, 17], for examples and references).

Most of the literature on the "exploration vs. exploitation" problem (as it is often referred to) is relatively heuristic in nature (a review of these techniques is given in [12], Chapter 10). Standard techniques involve mixing exploration vs. exploitation in a fixed, pre-specified ratio, a declining ratio (epsilon-greedy optimization), and Boltzmann exploration (which explores decisions with a weight that is proportional to the attractiveness of a decision). Other techniques can be viewed as heuristic variations of Gittins indices, including interval exploration [18] and upper confidence bounding [19]. Bickel & Smith [20] illustrate optimal learning in the context of a binary choice model, motivated by an application of drilling oil wells.

2.2 Prior work on change-point detection

Learning with costs is more difficult when the relation between labels and the value of examining the cases changes. Precisely this situation arises when there is an emerging threat such as a changing political environment, etc. When monitoring message traffic for emerging threats, if an adversary is discussing a plan of attack, the traffic in related documents will increase. Similarly, the number of money transfers to and from rogue states may increase (as there are caps on the amounts that can be wired in one time, large amounts can be transferred only in numerous smaller chunks). When a hacker gains access to a computer, he quickly executes several typical commands (for example, "change directory" and "list") as he navigates to the password files. Sequential change-point detection algorithms originate with fault detection and isolation in industrial processes, target detection and identification in national defense, and in radar and sonar processing, speech and image analysis, and bio-surveillance; see, for example, [21,22], and [23] for an extensive overview.

2.3 Prior work on adaptive document filtering

Adaptive document filtering has been studied in information retrieval. Albert & Kraft [24] modeled the quitting by a searcher perusing a list of documents. Kantor [25] simplified the model using the conjugate Beta and binomial distributions. Related work on selective dissemination of information (SDI) focused on complex Boolean queries to describe the interests of specific clients. These systems evolved at a time when end-users designed queries and made an implicit decision about how much time should be spent refining the query. This assumes that the flow of incoming material (scientific literature, message traffic) does contain material of interest, and that some of it should be routed to the recipient.

In the mid 1990's, using the TREC (Text Retrieval Conference), the Intelligence Community sought to assess these assumptions. The first filtering work at TREC (TREC4,5) was essentially binary text classification. Adaptive filtering recognizes that the "filtering" problem must assign a value to every relevant item routed to the user, and must have a cost for every item the end-user examines [26]. This is precisely the problem we address.

In TREC5, metrics corresponded to values of the ratio v/c (1/2, 1, 3), where c is the cost of evaluating and v is the value of a relevant item. In TREC6 [27], the "Adaptive Filtering," task was formulated as a constant horizon task (H = 1000 documents) and the total score for any particular solution was computed *without discounting*, as

$$V(P)=\sum_{i=1}^{\min(S(R),1000)}[v_i-c].$$

In this equation, R labels the decision rule, and S(R) is the stopping point of the rule. This recognizes that an optimal policy may stop sending documents, either if there is not enough value in the stream, or it cannot figure out what the end-user wants. At first, even the best system could "barely justify its own existence"[27]. Most teams set thresholds to balance the cost of presenting an irrelevant document against the information gained by learning about its irrelevance. Most used ranked retrieval systems, thresholding the document score. Some converted score to probability using logistic regression. AT&T used terms, phrases (adjacent pairs) and nonadjacent pairs based on weights derived from the Rocchio expansion [28]. ANU used terms and phrases from training documents [27], and weighted terms based on contrasting the probability they occur in the relevant documents and non-relevant documents. City University ordered terms and word pairs on a Bayesian model, iterating to improve quality of the fit [29]. CLARITECH used a Bayesian model to select the terms, and a Rocchio algorithm in a second pass. Their CLROUTE used a similar method, while CLCOMM used two training sets, retaining only terms identified in **both** training sets [30].

These approaches, all based on linear classifiers or naive Bayes, moved beyond traditional word-based vectors, to include other features. Some reduced the resulting space using a variant of Singular Value Decomposition. All, via machine learning or statistical techniques, sought to improve both the linear classifier and the threshold setting. All these extend the basic stopping model, which simply asks whether a line of investigation will not pay off. Few explicitly considered either the value of learning, or the specified finite horizon. An exception is the work of Zhang et al.[31], Zhang & Callan [32] who considered a single step look-ahead. The eventual best performance, was by Chinese Academy of Sciences [33]. Examination of that work, using homotopic techniques [34, 35] found that their specific parameter settings were extremely close to the optimal values of those parameters for that data set, raising the possibility that the selection of the parameters had been inadvertently guided by exposure to the test data.

3. The binary decision problem

3.1 Analytical Models

We consider several variants of an underlying binary decision problem. Cases arrive sequentially. Each case has a *visible* label x_t, based on which we decide among: $\{$select | further evaluation | discard$\}$. A variant model problem presents several cases at once (batched arrivals) and we choose a subset to evaluate. We use this to begin the study of correlated features. Finally, we add transient or change-point behavior, such as will arise in the presence of emerging threats.

3.2 A model with unrelated labels

We must, here, learn which among a discrete set of labels provides sufficient expected value to be worth sending for further analysis. A label may be a vector of attributes $x = (x_1, x_2, \ldots, x_m)$. Let $p(x)$ be the probability that a case with the label x will turn out, on inspection, to be a positive one. If the value of finding a positive is v and the cost of the detailed examination is c then the condition for being immediately worth sending is, $E(x) = vp(x) - c > 0$. We will assume an infinite horizon formulation with discount γ, but finite horizon problems are also important.

3.2.1 Optimization for binary selection

Our most basic model posits a sequence of cases, each of which must either be discarded, or selected for further evaluation *at the time it is presented*. Let t index the cases, and let X_t be the random variable representing the label for the case with index t. We then make a decision Y_t ($Y_t = 1$ is evaluation and 0 otherwise). Finally, we let $Z_t = 1$ if we find that the case is positive , and 0 otherwise. As noted, we only observe Z_t if Y_t is 1. The overall process is represented by the sequence of random variables $(X_t, Y_t, Z_t)_{t=1}^{\infty}$. We further suppose examining a case has a cost c and, if the document is interesting, there is a reward $v = 1$ (that is, the value is taken as the numeraire). We suppose an infinite horizon with a discount factor γ. The infinite horizon discounted reward is related to the random variables by the equation:

$$Reward = \sum_{t=0}^{\infty} \gamma^t Y_t (Z_t - c).$$

The behavior of a ``solution" to this problem will depend on the ``underlying reality" of the stream of cases. We give a general probabilistic framework governing X_t and Z_t. Let α be the underlying rate at which interesting documents arrive. Specifically, we assume that the sequence $(Z_t)_{t=1}^{\infty}$ is a sequence of independent Bernoulli random variables with success probability α. The labels that we can see are governed by two (conditional) probability distributions on the space of labels: P_0 and P_1. Together they generate an unconditional distribution: $P_{01} := \alpha P_1 + (1-\alpha) P_0$ of the X_t. We let $\mathbb{P}$ be the set of the three distributions (P_0, P_1, P_{01}). Now we define a filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$ by letting $\mathfrak{F}_t$, $t \geq 1$, be the sigma-algebra generated by $X_1, Y_1, Y_1 Z_1, \ldots, X_t, Y_t, Y_t Z_t$. We will require that Y_{t+1} be measurable with respect to the sigma-algebra generated by $\mathfrak{F}_t$ and X_{t+1}. This sets a mathematical framework, except that we have not specified how any particular decision rule, whether algorithmic or heuristic, is to be represented. A *rule* π will be a rule for obtaining Y_{t+1} from $\mathfrak{F}_t$ and X_{t+1}. Mathematically our problem becomes one of finding a rule π that achieves the supremum

$$\sup_{\pi} \mathbb{E}^{\pi} \sum_{t=0}^{\infty} \gamma^t Y_t (Z_t - c).$$

If α, P_0, and P_1 were known perfectly, and hence P_{01} as well, we could choose Y_t by computing the conditional probability

$$\mathbb{P}\{Z_t = 1 \mid X_t = x\} = \mathbb{P}\{Z_t = 1\}\mathbb{P}\{X_t \in dx \mid Z_t = 1\} / \mathbb{P}\{X_t \in dx\}$$
$$= \alpha P_1(dx) / P_{01}(dx)$$

and then setting $Y_t = I_{\{\mathbb{P}\{Z_t = 1 | X_t\} > c\}}$. However, the distributions P_0, and P_1, and the value of α, are generally unknown. So the rule for choosing the action Y_t must reflect the need to learn them from data. To develop a theoretical framework we formalize the precise way in which α, P_0, and P_1 are unknown, using a Bayesian approach. Let us suppose α, P_0, and P_1 are themselves random, under some prior distribution, with α taking values in [0,1] and the P_0 and P_1 taking values in some family of probability measures. This family of probability measures may be a parametric family such as the family of normal distributions, or it may be an empirically parameterized family, admitting a much broader class of priors, perhaps at the cost of increased complexity.

3.2.2 Dynamic programming formulation

In principle, we may solve the general problem using dynamic programming. We formulate the solution abstractly, and then specialize to a specific case. Let $\mathbb{S}$ be the space of all possible joint posterior distributions on the random variables α, P_0, and P_1. Let S_0 be the prior distribution on (α, P_0, P_1). Our measurements give a sequence of posterior distributions $(S_n)_{n=0}^{\infty}$ which may be thought of as conditional distributions on (α, P_0, P_1) given $(\mathfrak{F}_n)_{n=0}^{\infty}$. These conditional distributions may be obtained recursively using Bayes rule. It can be shown that the supremum in (1) remains unchanged if we restrict the policy space to stationary policies of the form $\pi : \mathbb{S} \times \mathcal{X} \mapsto \{0,1\}$, where $Y_{t+1} = \pi(S^n, X^{n+1})$ under π. For each such stationary policy we define the value function for that policy $V^{\pi} : \mathbb{S} \mapsto \mathbb{R}$ as

$$V^{\pi}(S_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \pi(S_t, X_{t+1})(-c + Z_{t+1}) \right].$$

$V(s) = \sup_{\pi} V^{\pi}(s)$. Then the value function satisfies Bellman's recursion

$$V(S_t) = \sup_{\pi} \mathbb{E}_t \left[\gamma V(S_{t+1}) + \pi(S_t, X_{t+1})(-c + Z_{t+1}) \right] \quad (1)$$
$$= \mathbb{E}_t \left[\max\{Q(S_t, X_{t+1}, 0), Q(S_t, X_{t+1}, 1)\} \right],$$

where we define the Q-factor : $Q : \mathbb{S} \times \mathcal{X} \times \{0,1\} \mapsto \mathbb{R}$ by

$$Q(S_t, X_{t+1}, y) = \mathbb{E}_t \left[y(-c + Z_{t+1}) + \gamma V(S_{t+1}) \mid X_{t+1}, Y_{t+1} = y \right].$$

Given unlimited computational power, we could compute the value function V as the fixed point of the Bellman recursion using an algorithm such as value iteration. In practice, the size of the state space $\mathbb{S}$ prevents this, or at least makes it very difficult. If the value function can be computed, however, we can use it to find an optimal policy π^* according to the formula $\pi^*(s,x) = I_{\{Q(s,x,1) \geq Q(s,x,0)\}}$. Since the sigma-algebra $\mathfrak{F}_t \vee \sigma(X_{t+1}, Z_{t+1})$ resulting from choosing $Y_{t+1} = 1$ contains the sigma-algebra $\mathfrak{F}_t \vee \sigma(X_{t+1})$ resulting from choosing $Y_{t+1} = 0$, Jensen's inequality and the convexity of the value function imply that

$$\mathbb{E}_t\left[V(S_{t+1}) \mid X_{t+1}, Y_{t+1} = 1\right] - \mathbb{E}_t\left[V(S_{t+1}) \mid X_{t+1}, Y_{t+1} = 0\right] \geq 0.$$

Thus, an optimal policy may always pass the document along to the analyst if $\mathbb{P}_t\{Z_{t+1} = 1 \mid X_{t+1}\} \geq c$, that is, if the expected one-period reward is nonnegative (note that we have scaled the costs by assuming that the reward is equal to 1). More significantly, an *optimal* policy will sometimes choose to pass the document along even in situations in which the expected one-period reward is negative because the immediate one-period cost is offset by the term (2). This term may be thought of as a learning bonus, or as the value of the information gained from the analyst's feedback. The tradeoff between the learning bonus and a negative one-period reward is an example of the classic tradeoff between exploration and exploitation.

The problem can be illustrated for the case where the unconditional distribution of cases is known. Let us analyze the transition from S_t to S_{t+1}. If our policy chooses not to inspect the case, setting $Y_{t+1} = 0$, then we only observe X_{t+1} and not Z_{t+1}. Since we already know P_{01}, this limited information does not change the posterior, so $S_{t+1} = S_t$. Conversely, if our policy chooses $Y_{t+1} = 1$, it can be shown that if S_t is the product of independent beta distributions with parameters $(a_t, b_t) = (a_{t1}, \ldots, a_{td}, b_{t1}, \ldots, b_{td})$, then S_{t+1} is also the product of independent beta distributions, but with parameters (a_{t+1}, b_{t+1}), where

$$a_{t+1,x} = a_{t,x} + I_{\{x=X_{t+1}\}} Z_{t+1}$$

$$b_{t+1,x} = b_{t,x} + I_{\{x=X^{t+1}\}}(1 - Z_{t+1}) .$$

The posterior on p_x becomes progressively sharper as the sum $a_x + b_x$ increases. In the limit as $a_x + b_x \to \infty$, we have

$$p_x = \mathbb{E}p_x = (a_x + 1)/(a_x + b_x + 2) .$$

If we know p_x exactly and we see $X_t = x$ we know that the value of choosing $Y_t = 1$ is $\mathbb{E}[Z_{t+1} \mid X_{t+1} = x, p_x] = p_x$, with no learning bonus. We know then that the optimal decision is $Y_t = I_{\{p_x > c\}}$. In the limit as $\min_x(a_x + b_x) \to \infty$, we know every p_x exactly and the expected reward obtained at time t is $\mathbb{E}[(-c + p_{x_t})^+ \mid p] = \sum_{x \in \mathcal{X}} P_{01}(x)(-c + p_x)^+$. Then in the limit as $\min_x(a_x + b_x) \to \infty$, we set $p_x = \lim(a_x + 1) / (a_x + b_x + 2)$ and we have

$$\lim_{\substack{\min(a_x + b_x) \to \infty \\ x}} V(a,b) = \sum_{t=1}^{\infty} \sum_{x \in \mathcal{X}} P_{01}(x)(-c + p_x)^+$$

$$= \frac{1}{1-\gamma} \sum_{x \in \mathcal{X}} P_{01}(x)(-c + p_x)^+ .$$

3.2.3 A Value Iteration Algorithm

This *value iteration algorithm* approximates the value function, and serves to illustrate the principle, but will be extremely slow unless the size of $\mathcal{X}$ is fairly small. The algorithm requires that we choose a parameter N which plays the role of ∞, so larger N values will increase accuracy but increases the running time. We now approximate $V(a,b)$ by (3) with $p_x = (a+1)/(a+b+2)$ for all a, b with $\min_x(a_x + b_x) \geq N$. This approximation defines a new optimization problem with value function $\tilde{V}_N$ in which $\tilde{V}_N(a,b)$ is defined according to (3) for a, b such that $\min_x(a_x + b_x) \geq N$, and obeys Bellman's recursion for other a, b. We have the inequality $\tilde{V}_N > V$, since the approximation (3) is an upper bound on V. As N increases to infinity, $\tilde{V}_N$ decreases to V.

We can compute a sequence of approximations $\bar{v}_N^n$ decreasing to $\tilde{V}_N$. To do this we must store the values of $\bar{v}_N^n(a,b)$ with a_x ranging from 0 to N and b_x ranging from 0 to $N - a_x$. We do not actually need to store $\bar{v}_N^n$ for those a, b with every x satisfying $a_x + b_x = N$, but doing so simplifies implementation without harming correctness, and only marginally degrades performance. Begin with $\bar{v}_N^0(a,b)$ given by (3) for every a and b, not just those a and b with $\min_x(a_x + b_x) = N$. Then compute $\bar{v}_N^{n+1}$ from $\bar{v}_N^n$ according to the recursion

$$\bar{v}_N^{n+1}(a,b) = \sum_{x \in \mathcal{X}} P_{01}(x) \max\left\{ \gamma\bar{v}_N^n(a,b), -c + \frac{(a_x+1)\left(1+\gamma\bar{v}_N^n(a+e_x,b)\right) + (b_x+1)\gamma\bar{v}_N^n(a,b+e_x)}{a_x+b_x+2} \right\} . \tag{3}$$

At each iteration n, we compute this recursion for every a_x ranging from 1 to N and b_x ranging from 1 to $N - a_x$, assuming that $\bar{v}_N^n(a,b)$ is given by $\tilde{V}_N(a,b)$ for (a,b) outside this range. We then advance to the next iteration until we are satisfied that $\bar{v}$ is ``sufficiently close" to $\tilde{V}_N$. The error can be characterized in a rigorous way in terms of the sup-norm [36]. (3)

To store the function $\bar{v}_N^n$ for one value of n, the number of values we need to store is

$$\left|\{a, b \in \mathbb{N} : a + b \leq N\}\right|^{|\mathcal{X}|} = \left|\sum_{a=1}^{N} N - a\right|^{|\mathcal{X}|} = \left[N(N-1)/2\right]^{|\mathcal{X}|} .$$

A naive implementation would store $\bar{v}_N^n$ for two different values of n at a time, one for the previous iteration and one for the iteration being computed. A more sophisticated implementation could improve this by updating the $\bar{v}_N^n$ in place, but would still need to store at least $\left[N(N-1)/2\right]^{|\mathcal{X}|}$ values at each step. With N set to 12, this amounts to $66^{|\mathcal{X}|}$ values. Using 16-bit precision, and limiting ourselves to working entirely in RAM on a single computer with 3 GB available memory, this will limit us to

$$|\mathcal{X}| \leq \log\left(\frac{3 \times 10^9 \text{bytes}}{2\text{bytes}}\right) / \log(66) \approx 5.04 .$$

This illustrates the principle. In practice, we can use approximate dynamic programming and exploit the convexity of the value function (see [12], chapter 11).

3.2.4 Dynamic ranking/ selection

If we are in an on-line learning situation where many cases arrive at once, and we assume that the effects of the labels on the probability of "value" are independent, we have, at each step the classic multi-armed bandit problem to which Gittins indices are suited. Alternatively, we may formulate an off-line learning problem, where we budget to learn as much as we can from a set of labels, after which we apply our knowledge to solve a problem. If measurements are uncorrelated, we can use the recently-proposed knowledge-gradient algorithm ([8], see also [37]) which uses a one-step lookahead to estimate the value of a measurement. The approach of one-step lookahead has in fact been applied to the adaptive filtering problem in the TREC setting by Zhang and Callan [31] In this heuristic, the choice of label d to measure, indicated by $x_d = 1$, is determined using

$$X^{KG}(S^n) =$$
$$argmax_{\{x|\sum_d x_d=1\}} \mathbb{E}\left[V(S^M(S^n, x, W^{n+1})) - V(S^n) \mid S^n\right]$$

where S^n is the current state of knowledge about the relationship between the label and its value, $x_d = 1$ if we are choosing label d, W^{n+1} is the information gained from the next measurement, and $S^M(S^n, x, W^{n+1})$ is the updated state of knowledge (we propose to use Bayesian updating). $V(S^n)$ captures the value of the current state of knowledge. The incremental value of a measurement is called the knowledge-gradient index, and it is given by first computing

$$\zeta_d^n = -\left| \frac{\bar{\theta}_d^n - \max\limits_{d' \neq d} \bar{\theta}_{d'}^n}{\tilde{\sigma}_d^n} \right|.$$

where $\bar{\theta}_d^n$ is the current estimate of the value of label d after n observations, and $\tilde{\sigma}_d^n$ is the square root of the reduction in variance of belief, due to measuring label d (recursive formulas for $\bar{\theta}_d^n$ and $\tilde{\sigma}_d^n$ are easily derived from Bayesian concepts).

ζ_d^n is called the *normalized influence* of decision d. It measures the number of standard deviations from the current estimate of the value of decision d, given by $\bar{\theta}_d^n$, and the best alternative other than decision d. We then find

$$f(\zeta) = \zeta\Phi(\zeta) + \varphi(\zeta),$$

where $\Phi(\zeta)$ and $\varphi(\zeta)$ are, respectively, the cumulative standard normal distribution and the standard normal density. The knowledge gradient algorithm chooses the decision d with the largest value of $\nu_d^{KG,n}$ given by

$$\nu_d^{KG,n} = \tilde{\sigma}_d^n f(\zeta_d^n).$$

The knowledge gradient algorithm is easy to implement for independent measurements. Frazier [8] shows this method is asymptotically optimal, fully optimal for certain special cases and has an error bound. Experimentally it compares favorably to other heuristics, including sophisticated OCBA algorithms.

3.3 *Correlated* measurements

In contrast to Gittins indices, the knowledge gradient can be extended to handle the case of correlated beliefs. Let

$$f(S^n) = \sum_{i=1}^{M} a_i(\Phi(c_i) - \Phi(c_{i-1})) + b_i(\varphi(c_i) - \varphi(c_{i-1}))$$

where a_i, b_i and c_i are specific constants that are computed for the i^{th} label. Again, S^n is our ``state of knowledge," consisting of the vector of all the current means μ^n, and the covariance matrix relating the different types of labels, Σ^n. Now let $\Sigma^n(x)$ be the updated covariance matrix that will result if we make the measurement decision x (which in this setting indicates the particular type of label we are going to measure). The knowledge gradient policy for correlated rewards is then given by

$$X^{KG} = argmax_x f(\mu^n, \Sigma^n(x)).$$

Figure 1 shows the correlated knowledge gradient searching for the maximum of a continuous univariate function mapping labels to values. Knowledge of the continuity of the underlying function induces correlation in our belief, since similar (nearby) labels should have similar values, and the correlated knowledge gradient policy uses this correlation to achieve much greater efficiency than could an algorithm using an uncorrelated belief. In Figure 1, the policy already has a good estimate of the maximum of the underlying function after only 8 measurements, while an uncorrelated measurement policy would require a number of measurements at least as large as the number of labels to get a reasonable estimate.

The knowledge gradient policy is fairly computable for problems with tens of thousands of labels, but difficult to apply to populations of hundreds of thousands or millions of labels or features. For such applications, additional research on feature selection is needed. For example, the methods of Latent Semantic Indexing sharply reduce the dimensionality of the space of cases, when the cases are documents.

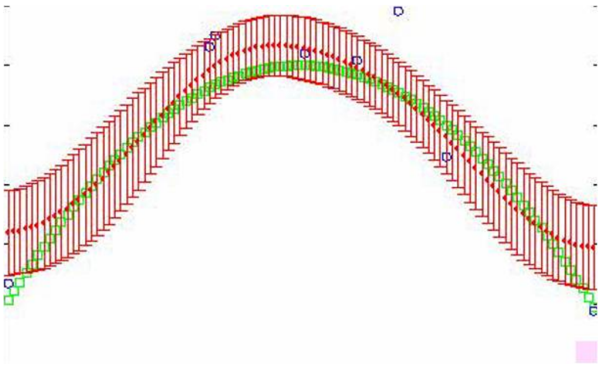

***Figure 1**. The correlated knowledge gradient policy searching for the maximum of a continuous function. The green line is the underlying but unknown true function; the blue dots are measurement results; and the red dots and error bars are the center and standard deviations of the posterior belief after measurement.*

3.4 Learning with costs: change environment

We address this problem in terms of Markov models, and only sketch the method here. Let $\{H_t, t \geq 0\}$ represent some hidden Markov process with state-space $\{0,1\}$ and a one-step transition probability matrix P, where $H_t = 1$ means that at time t a given relation between interesting cases (ground truth) and labels holds, and at other times it does not. For example, before a plot on Target X, the phrase target X is not interesting. Subsequently, it is. Given the process $H_t, t \geq 0$, the ground truth Z_t, $t \geq 0$ are conditionally independent Bernoulli random variables with arrival rates α_{H_t}, $t \geq 0$, respectively. We also assume that the label X_t representing case t has the conditional probability density f_0 (respectively, f_1) given that $Z_t = 0$ (respectively, $Z_t = 1$.) At each time t we make two decisions, Y_t and G_t, where $Y_t = 1$ if we choose the case for further evaluation (with 0 otherwise); and $G_t = 1$ if we believe that the source of interesting cases is active, but 0 otherwise. The immediate cost and reward of forwarding a document to an expert is c and 1, respectively. Additionally, we incur an immediate misclassification cost (c_{FN} for a false negative, and c_{FP} for a false positive) if at time t the status of the source is classified incorrectly. Now the expected total discounted reward becomes

$$\mathbb{E}\sum_{t=0}^{\infty}\gamma^t\left[Y_t(Z_t - c) - c_{FN}(H_t - G_t)^+ - c_{FP}(G_t - H_t)^+\right] \quad (4)$$

The goal is to find an admissible decision rule $(Y_t, G_t)_{t\geq 0}$ which maximizes, as before, the expected total discounted net reward. The solution of this partially observed Markov decision problem (POMDP) depends on the posterior-probability-distribution process

$$\Pi_t := \mathbb{P}\{H_t = 1 \mid X_0, X_1, \ldots, X_{t-1}, X_t,$$
$$Y_0, Y_1, \ldots, Y_{t-1}, Z_0 Y_0, Z_1 Y_1, \ldots, Z_{t-1} Y_{t-1}\}, \qquad t \geq 0,$$

which satisfies

$$\Pi_t \propto \left(1 - \Pi_{t-1}\right)[\alpha_0]^{Y_{t-1}Z_{t-1}}[1-\alpha_0]^{Y_{t-1}-Y_{t-1}Z_{t-1}} P_{01} f_1(X_t)$$
$$+ \Pi_{t-1}[\alpha_1]^{Y_{t-1}Z_{t-1}}[1-\alpha_1]^{Y_{t-1}-Y_{t-1}Z_{t-1}} P_{11} f_1(X_t), \qquad t \geq 0$$

(we omit the proportionality constant)

Note that the *sufficient statistic* for this problem, $\{\Pi_t, \mathfrak{F}; t \geq 0\}$ is a controlled Markov chain on $[0,1]$ adapted to the filtration

$$\mathfrak{F}_t = \sigma\{X_0, X_1, \ldots, X_{t-1}, X_t, Y_0, Y_1, \ldots$$
$$, Y_{t-1}, Z_0 Y_0, Z_1 Y_1, \ldots, Z_{t-1} Y_{t-1}\}, \qquad t \geq 0.$$

The expectation in Eq. (4) can be rewritten as

$$\sum_{t=0}^{\infty}\gamma^t\mathbb{E}\begin{bmatrix} Y_t(\mathbb{E}[Z_t \mid \mathfrak{F}_t] - c) - c_{FN}(1-G_t)\mathbb{P}\{H_t \\ = 1 \mid \mathfrak{F}_t\} - c_{FP} G_t \mathbb{P}\{H_t = 0 \mid \mathfrak{F}_t\} \end{bmatrix}$$
$$= \sum_{t=0}^{\infty}\gamma^t\mathbb{E}\begin{bmatrix} Y_t((1-\Pi_t)\alpha_0 + \Pi_t\alpha_1 - c) - \\ c_{FN}(1-G_t)\Pi_t - c_{FP}G_t(1-\Pi_t) \end{bmatrix}$$
$$= \sum_{t=0}^{\infty}\gamma^t\mathbb{E}\begin{bmatrix} Y_t(\Pi_t(\alpha_0 - \alpha_1) - c + \alpha_0) - c_{FN}\Pi_t - \\ (c_{FP} - (c_{FN} + c_{FP})\Pi_t)G_t \end{bmatrix},$$

This, in turn, implies that the maximum expected total discounted net reward is attained by

$$Y_t^* = \left\{\begin{array}{ll} 1, & \Pi_t \geq \dfrac{c - \alpha_0}{\alpha_1 - \alpha_0} \\ 0, & \text{otherwise} \end{array}\right\} \qquad (5)$$

and

$$G_t^* = \left\{\begin{array}{ll} 1, & \Pi_t \geq \dfrac{c_{FP}}{c_{FP} + c_{FN}} \\ 0, & \text{otherwise} \end{array}\right\}.$$

Since we have said that when the hidden Markov process H is in state 1 (rather than 0), then interesting cases will arrive at a higher rate (raised threat), it makes sense to assume $\alpha_0 \leq \alpha_1$. Now note that in (5) we have $Y_t^* \equiv 0$ if $\alpha_0 \leq \alpha_1 \leq c$; that is, no cases are forwarded for examination if the a priori odds that they are worth looking at is too low. Similarly, $Y_t^* \equiv 1$ if $c \leq \alpha_0 \leq \alpha_1$; that is, every case will be inspected if examination is very cheap. The interesting and realistic case is when $\alpha_0 < c < \alpha_1$, in which case the optimal strategy is given by (5).

To handle more realistic cases where arrival rates α_0 and α_1, transition probabilities $(P_{ij})_{i,j\in\{0,1\}}$, and densities f_0 and f_1 are all unknown, we treat all of these unknowns as random variables with suitable prior probability distributions. Then we can derive the dynamics of the corresponding posterior-probability-distribution process, rewrite the expected total discounted costs in terms of this process, and use (approximate) dynamic programming techniques to solve it.

This approach has worked for similar POMDPs. [38,39,40,41] studied stochastic systems which may undergo sudden changes at unknown and unobserved times and has provided determined Bayesian quickest change detection and identification rules in discrete time. Figure 2 illustrates a typical implementation. In continuous time, [42] have explicitly characterized the solution of adaptive Poisson disorder problem, while [43] have proposed nearly-optimal online algorithms to detect a sudden unknown unobservable change in the arrival rate and mark distribution of compound Poisson processes.

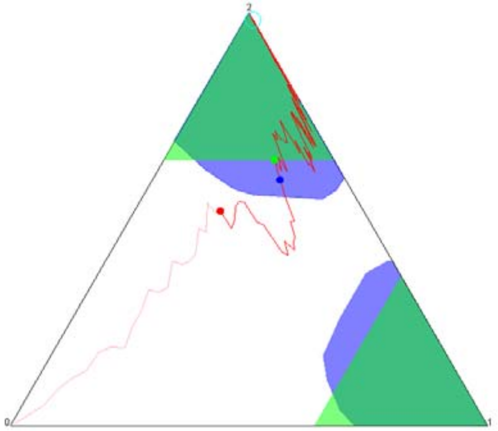

Figure 2. *Quickest detection and identification of a unknown and unobserved sudden change in the distribution of a random sequence. It is optimal to raise an alarm as soon as the posterior probability process (pink before change time/red afterwards) enters the shaded region (blue is optimal, green is approximate). Each corner is associated with the best possible change-diagnosis upon an alarm.*

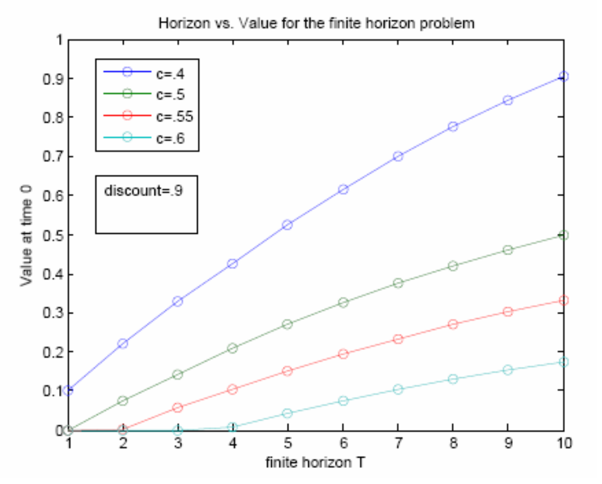

Figure 3*. The key feature is that without the Dynamic Programming solution, one cannot know that for higher values of the cost, and certain values of the discount and horizon, there is no chance to learn in time! With dynamic programming, one can do the calculation in advance, and not waste any funds on an exploration that will only produce negative net present value.*

3.5 The distribution of costs and benefits

Following the principles of utility theory, we have presented the entire analysis in terms of expected values. In real applications, the methods used to determine the rules, from the data, represent possible policies. Since the cases from which we learn arrive in random orders, the actual performance of any policy, and of the resulting rules, will itself be a random variable. We must understand the range of variability of this performance, to know what kinds of risks are associated with selecting a policy.

One important class of policies, for each label, says precisely: "continue sending items with this label for evaluation until an irrevocable decision is reached to either send all, or send no more". The net present value of this rule can be computed precisely, for any given value of the cost c and the probability p that cases with this label are, indeed, "positive cases". For such rules one may compute the probability that the rule will fire when exactly k items with this label have been examined. With this information, one may compute not only the expected value, but also the complete distribution of the expected value of this policy, as a function of the probability p, and the cost, discount and/or horizon parameters. An example of such a tractable policy class is: $D =$ "decide to stop when the number of positive items seen, g, minus the number of negative items seen b falls below a preset threshold". The probability that a specific rule D, resulting from this policy, fires at step k can be computed using random walks with an absorbing barrier. Other stopping rules, corresponding to different degrees of certainty about the prior estimate of the probability of good outcomes, may depend on a more complicated procedure, such as the rules used in Sequential Analysis.

The probability that this rule results in a particular net present value can be expressed in closed form. By the reflection principal, the probability that the state will reach any particular value of $b-g$ at precisely the kth step, is the difference of two binomial distributions, rescaled by a bias factor determined by the Radon-Nikodym derivative of the actual distribution with respect to the symmetric distribution. For many steps we can approximate by the normal distribution. A general sense of the behavior of a rule can be found by simulation process. The ability of a rule to correctly discriminate among the labels will depend on the discounting. We show the value achieved using a cutoff rule of this type, for the case of a small discounting of future gains Figure 4a and a large one Figure 4b. When little benefit is realized from long term gains it is possible to get sometimes only a negative benefit, even when processing a label for which the long term expected value is positive.

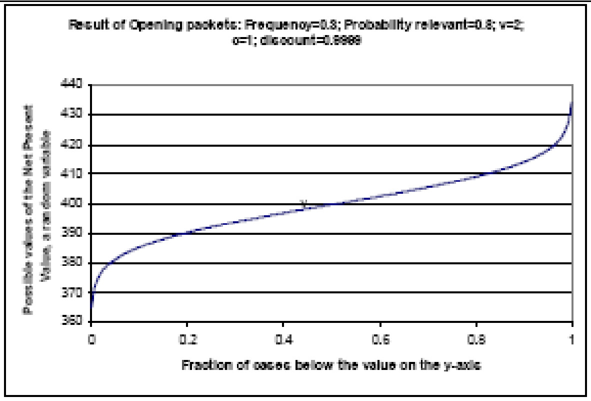

***Figure 4a**. The results of a simulation for specific values of the parameters of the problem, for the rule: “terminate cases that go negative in value”. The discount factor is very close to 1, so that long term benefits are realized. It would be very rare to terminate the service.*

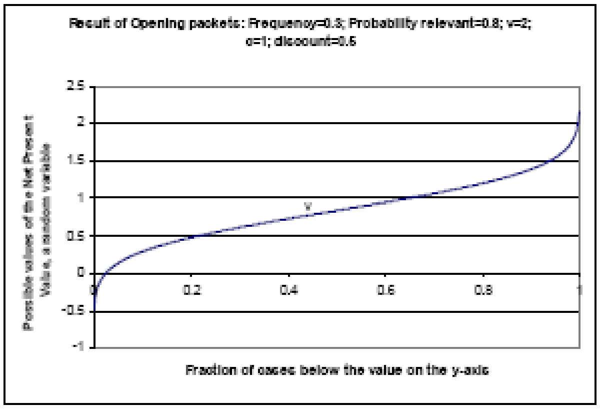

***Figure 4b**. With a sharper discount, it is possible that the net present value will be negative, although this is still a rare event. In either case, this “label" ought to be identified as one worth submitting.*

Applying the value-bounding methods discussed above, we can ask what happens as the cost is raised while the value is fixed at 1, and the horizon is changed. The curves of Figure 3 show that if the cost is low, it makes sense to submit cases for examination, even if the horizon is one case. When the cost is 0.5, and the prior distribution of the probability of relevance is uniform, then with no time to learn (Horizon=0) the expected value is 0. If the cost is higher, trying to learn with short horizon would have negative value, and is dominated by the rule "do nothing". As the lower curve shows, it is only with sufficient time to learn, that the rule can distinguish among the good and bad situations, and learn whether to continue submitting for evaluation.

Using these methods the probability of false negatives can be computed for any specific value of the rate at which cases of interest are detected and sent for examination.

4. Illustrations of the Concepts

We illustrate these concepts with a number of simulations. Fig. 1 shows the effect of correlated knowledge gradient. Fig 2. shows the expansion of the region of “adequate certainty” in change point detection. Extensive simulation of simple stopping rules is discount dependent (Fig. 4a,b). More complicated simulations (Fig. 3) show the effectiveness of the dynamic programming solution for the infinite horizon case, with various discounts. Note that with the DP solution the value **never becomes negative.** Thus it is clearly more effective than the application of a heuristic that might produce negative net present value.

5. References

[1] Bechhofer, R., Kiefer, J. & Sobel, M. (1968), Sequential Identification and Ranking Procedures, University of Chicago Press, Chicago.

[2] DeGroot, M. H. (1970), Optimal Statistical Decisions, John Wiley and Sons.

[3] Goldsman, D. & Nelson, B. (1994), Ranking, selection and multiple comparisons in computer simulation, in J. D. Tew, S. Manivannan, D. A. Sadowski & A. F. Seila, eds, ‘Proceedings of the 1994 Winter Simulation Conference’.

[4] Bechhofer, R., Santner, T. & Goldsman, D. (1995), Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons, J.Wiley & Sons, New York.

[5] D. A. Cohn and Z. Ghahramani and M. I. Jordan. Active learning with statistical models. J. of Artificial Intelligence, 4:129-145, 1996.

[6] Gittins, J. C. & Jones, D. M. (1974), A dynamic allocation index for the sequential design of experiments, in J. Gani, ed., 'Progress in Statistics', pp. 241–266.
[7] Gittins, J. (1989), Multi-Armed Bandit Allocation Indices, John Wiley and Sons, New York.
[8] P. Frazier, W.B. Powell, and S. Dayanik (2007), 'A Knowledge-Gradient Policy for Sequential Information Collection', SIAM Journal on Control & Optimization 47 (5). http://www.princeton.edu/%7Epfrazier/IndependentKG.pdf>. Submitted.(http://www.princeton.edu/~pfrazier/IndependentKG.pdf)
[9] Duff, M. & Barto, A. (1997), 'Local bandit approximation for optimal learning problems', Advances in Neural Information Processing Systems 9, 1019.
[10] Topaloglu, H. & Powell, W. B. (2006), 'Dynamic programming approximations for stochastic, timestaged integer multicommodity flow problems', Informs Journal on Computing 18(1), 31–42.
[11] Powell, W. B., Ruszczyński, A. & Topaloglu, H. (2004), 'Learning algorithms for separable approximations of stochastic optimization problems', Mathematics of Operations Research 29(4), 814–836.
[12] Powell, W. B. (2007), Approximate Dynamic Programming: Solving the curses of dimensionality, John Wiley and Sons, New York.
[13] Fu, M. (2002), 'Optimization for simulation: Theory vs. practice', INFORMS Journal on Computing14(3), 192–215.
[14] Swisher, J., Jacobson, S. & Yücesan, E. (2003), 'Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey', ACM Transactions on Modeling and Computer Simulation (TOMACS) 13(2), 134–154.
[15] Chen, C.H., Lin, J., Yücesan, E. & Chick, S.E. (2000), 'Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization', Discrete Event Dynamic Systems 10(3), 251–270.
[16] Thorsley, D. & Teneketzis, D. (2007), 'Active acquisition of information for diagnosis and supervisory control of discrete event systems', Discrete Event Dynamic Systems 17, 531–583.
[17] He, D., Chick, S.E. & Chen, C-H. Opportunity Cost and OCBA Selection Procedures in Ordinal Optimization for a Fixed Number of Alternative Systems. IEEE Transactions on Systems, Man and Cybernetics — Part C: Applications and Reviews, 37(5):951-961, 2007.
[18] Kaelbling, L. P. (1993), Learning in Embedded Systems, MIT Press, Cambridge, MA.
[19] Chang, H., Fu, M., Hu, J. & Marcus, S. (2007), Simulation-Based Algorithms for Markov Decision Processes, Springer, Berlin.
[20] J. E. Bickel and J. E. Smith. Optimal sequential exploration: A binary learning model. Decision Analysis, 3(1):16-32, 2006.
[21] Basseville, M. & Nikiforov, I. V. (1993), Detection of Abrupt Changes: Theory and Application,Prentice Hall Information and System Sciences Series, Prentice Hall Inc., Englewood Cliffs, NJ.
[22] Tartakovsky, A. G. & Veeravalli, V. V. (2004), 'General asymptotic Bayesian theory of quickest change detection', Teor. Veroyatn. Primen. 49(3), 538–582.
[23] Lai, T. L. (1995), 'Sequential changepoint detection in quality control and dynamical systems', J. Roy. Statist. Soc. Ser. B 57(4), 613–658. With discussion and a reply by the author.
[24] Albert, D. & Kraft, D. T. (1978), 'A dynamic search stopping rule for an information storage and retrieval system'.
[25] P. B. Kantor. A model for stopping behavior of the users of on-line systems, Journal of the American Society for Information Science V38(3)p 211-214, May 1987.
[26] David D. Lewis. The TREC-5 Filtering Track. In E. M. Voorhees and D. K. Harman, (Eds.), The Fifth Text REtrieval Conference (TREC 5). NIST 500-238: Washington D.C. 1997..
[27] David A. Hull. The TREC-6 Filtering Track: Description and Analysis. In E. M. Voorhees and D. K. Harman, (Eds.).
[28] A. Singhal. AT&T at TREC-6. *op. cit.*
[29] S. Walker, S. E. Robertso, M. Boughanem, G. J. F. Jones & K. Sparck Jones. Okapi at TREC-6Automatic Ad Hoc, VLC, Routing, Filtering and QSDR. *Op cit.*
[30] C. Zhai, P. Jansen, E. Stoica, N. Grot, & D. A. Evans. Threshold Calibration in CLARIT Adaptive Filtering.In E. M. Voorhees and D. K. Harman, (Eds.). The Seventh Text REtrieval Conference (TREC 7): NIST 500-242, 149-156, Washington, DC, 1999.
[31] Y. Zhang and W. Xu and J. Callan. Exploration and Exploitation in Adaptive Filtering Based on Bayesian Active Learning. In Proc. 20h (ICML), 2003.
[32] Y. Zhang and J. J. Callan. YFilter at TREC-9.In E. M. Voorhees and D. K. Harman, (Eds.). The Ninth Text REtrieval Conference (TREC 9): NIST 500-249, 135-140, Washington, DC, 2000.
[33] H. Xu and Z. Yang and B. Wang and B. Liu and J. Cheng and Y. Liu and Z. Yang and X. Cheng and S. Bai. TREC 11 Experiments at CAS-ICT: Filtering and Web.In E. M. Voorhees and L. P. Buckland, (Eds.). The Eleventh Text REtrieval Conference (TREC 11): NIST 500-251, Washington, DC, 2002.
[34] Willard I. Zangwill and C. B. Garcia. Pathways to solutions, fixed points, and equilibria. Prentice-Hall, Englewood Cliffs, N.J., 1981. W.I. Zangwill, C.B. Garcia.; Includes indexes.; Bibliography: p. 459-472.
[35] D. Fradkin and P. B. Kantor. A Design Space Approach to Analysis of Information Retrieval Adaptive Filtering Systems. 13th (CIKM), 2004.
[36] Ross, S. (1983), Introduction to Stochastic Dynamic Programming, Academic Press, New York.
[37] Gupta, S. & Miescke, K. (1994), 'Bayesian look ahead one stage sampling allocations for selecting the largest normal mean', Statistical Papers 35, 169–177.
[38] S. Dayanik, C. Goulding, and H. V. Poor (2007). Bayesian sequential change diagnosis. Mathematics of Operations Research. To appear. (http://www.princeton.edu/~sdayanik/papers/diagnosis.pdf)
[39] S. Dayanik and C. Goulding (2007). Detection and identification of an unobservable change in the distribution of a Markov-modulated random sequence. Submitted (http://www.princeton.edu/~sdayanik/papers/markov.pdf)
[40] Savas Dayanik and Christian Goulding and H. Vincent Poor. Joint Detection and Identification of an Unobservable Change in the Distribution of a Random Sequence. CISS, pages 68-73, 2007.
[41] S. Dayanik, W. Powell, K. Yamazaki (2007). Asymptotic theory of sequential change detection and identification. Working paper. Preprint, 2007.
[42] Bayraktar, Erhan and Dayanik, Savas and Karatzas, Ioannis. Adaptive Poisson disorder problem. Ann. Appl. Probab., 16(3):1190--1261, 2006.
[43] Dayanik, Savas and Sezer, Semih O. Compound Poisson disorder problem. Math. Oper. Res., 31(4):649-672, 2006.
[44] Frazier, P., Powell, W., Dayanik, S. (under review), 'The Knowledge-Gradient Policy for Correlated Normal Beliefs', INFORMS Journal on Computing. http://www.princeton.edu/~pfrazier/CorrelatedKG.pdf.