Proceedings of the IEEE Conf. on Decision and Control, Maui, HI, 2012

Towards optimization of a human-inspired heuristic for solving explore-exploit problems

Paul Reverdy[1], Robert C. Wilson[2], Philip Holmes[1,3] and Naomi E. Leonard[1]

***Abstract*— Motivated by models of human decision making, we consider a heuristic solution for explore-exploit problems. In a numerical example we show that, with appropriate parameter values, the algorithm performs well. However, the parameters of the algorithm trade off exploration against exploitation in a complicated way so that finding the optimal parameter values is not obvious. We show that the optimal parameter values can be analytically computed in some cases and prove that suboptimal parameter tunings can provide robustness to modeling error. The analytic results suggest a feedback control law for dynamically optimizing parameters.**

I. INTRODUCTION

The problem of choosing among a set of actions with unknown and uncertain payoffs is pervasive in many fields of control, as well as everyday life. When exploring such a set, there is a tension between continuing to carry out the best of the known actions (termed exploitation) and trying new actions in the hopes of finding something better than one's current best option (termed exploration).

Given sufficient information about the structure of the space and the associated payoffs, an optimal solution to this explore-exploit problem can, in principle, be found using dynamic programming. Unfortunately, due to the well-known curse of dimensionality phenomenon, the fully optimal solution is intractable for all but very small problems. Because of this intractability, there is great interest in approximations to the optimal solution, for example using heuristics [1].

Understanding how humans solve the explore-exploit problem is an active area of research in neuroscience and cognitive psychology [2], [3]. Such understanding is of interest to the controls community for several reasons. Humans have been shown to perform optimally in simple decision-making tasks [4], and they perform reasonably well across a wide range of tasks. The degree of flexibility they provide in dealing with complex problems and environments is a desirable characteristic of an automated decision maker. Additionally, models based on this understanding would provide the possibility to integrate humans and automated decision makers using a common framework.

Recent work has investigated how humans solve the explore-exploit problem in a simple context, where it was shown that human behavior is well explained by an *ambiguity bonus* heuristic [5]. This heuristic makes decisions based on a value function Q, which assigns to each option i a value that trades off the expected payoff of that option, ΔR_i, with the information to be gained by testing it, ΔI_i:

$$Q_i = \beta \Delta R_i + (1-\beta)\Delta I_i, \ \beta \in \mathbb{R}. \tag{1}$$

The name of this heuristic derives from the influence of ΔI_i. Trying options that have greater ambiguity, i.e. less is known about their associated rewards, yields more information ΔI_i; for $\beta < 1$ these options are assigned greater values by the heuristic.

The parameter β trades off between explore and exploit strategies. In the case $\beta = 0$, the heuristic only weights information gain and therefore produces a pure explore strategy. By contrast, if $\beta = 1$, the heuristic only weights expected rewards and so produces a pure exploit strategy. If β is too high, the strategy may not explore enough to discover the best possible option, while if it is too low the strategy may find the best option but not value it sufficiently to profit from its discovery.

In this paper we consider an extension of the heuristic to a larger problem of resource collection in a scalar field, which can be modeled as a multi-dimensional explore-exploit problem. The problem is similar to that of optimal foraging in ecology [6]. It is also an example of a *multi-armed bandit* problem studied in the operations research literature; see the recent review [7] and references therein. We show that, with proper parameter tunings, the heuristic-based algorithm performs well but that finding the optimal parameter values is non-trivial. However, in some cases the optimal parameter values can be found analytically and suboptimal parameter values can provide robustness to modeling error. These results suggest a feedback control law for dynamically optimizing the parameters.

The remainder of the paper is organized as follows. In Section II we pose the resource collection problem, while in Section III we detail the algorithm based on the ambiguity bonus heuristic. In Section IV we present a motivating numerical example. In Section V we compute optimal parameter values for certain cases in which they can be found analytically and present the optimizing algorithm. In Section VI we suggest extensions and conclude.

This research has been supported in part by AFOSR grant FA9550-07-1-0-0528. This research was made with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

[1] Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08540, USA preverdy@princeton.edu, naomi@princeton.edu

[2] Department of Psychology and Neuroscience Institute, Princeton University, Princeton, NJ 08540, USA rcw2@princeton.edu

[3] Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08540, USA pholmes@math.princeton.edu

II. RESOURCE COLLECTION PROBLEM

Consider a d-dimensional discrete grid with N^d grid points. In the following, we consider the cases $d \in \{1, 2\}$, but the generalization to arbitrary dimensions is straightforward. Each of the N^d grid points has an associated reward m_i, which remains fixed for the duration of the problem. The vector $\mathbf{m} \in \mathbb{R}^{N^d}$ of the rewards is unknown to the agent but drawn from a distribution $\mathcal{D}$ with mean $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{N^d}$ and covariance $\overline{\Sigma}$.

The agent collects rewards by visiting one point at each time interval $t = 1, \dots, T$, receiving reward r_t which is the mean reward associated with the point plus Gaussian noise: $r_t \sim \mathcal{N}(m_i, \sigma_r)$. The agent's objective is to maximize cumulative rewards by choosing a sequence of points $\{\mathbf{x}_t\}$:

$$\max_{\{\mathbf{x}_t\}} V, \quad V = \sum_{t=1}^{T} r_t. \tag{2}$$

Note that due to the stationary nature of the sampled reward, in the long time horizon limit $T \gg N^d$ this problem reduces to the problem of finding the peak value among the m_i. We are particularly interested in the case of large spaces or short time horizons, in which case the explore-exploit tension is consequential. A similar situation arises in the long time horizon limit if the rewards are non-stationary.

III. THE AMBIGUITY BONUS HEURISTIC ALGORITHM

In order to solve the optimization problem, the agent needs to learn about the reward surface and make a decision based on their beliefs. With reasonable assumptions on the distribution of rewards $\mathbf{m}$, Bayesian inference provides a tractable optimal solution to the learning problem. The ambiguity bonus heuristic (1) then provides a tractable alternative to solving the full dynamic programming problem.

A. Inference algorithm

We begin by assuming that the agent's prior distribution of $\mathbf{m}$ (i.e. the agent's initial beliefs about the mean reward values $\bar{\boldsymbol{\mu}}$ and their covariance $\overline{\Sigma}$) is multivariate Gaussian with mean $\boldsymbol{\mu}_{\mathbf{0}}$ and covariance Σ_0:

$$\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{0}}, \Sigma_0),$$

where $\boldsymbol{\mu}_{\mathbf{0}} \in \mathbb{R}^{N^d}$ and $\Sigma_0 \in \mathbb{R}^{N^d \times N^d}$ is a positive-definite matrix. Note that this does not assume that the field is truly described by these statistics, simply that these are the agent's initial beliefs, informed perhaps by previous measurements of the mean value and covariance.

With this prior, the posterior distribution is also Gaussian, so the optimal inference algorithm is linear and can be written down as follows. At each time t, the agent, located at $\mathbf{x}_t \in (\mathbb{Z}_N)^d$, observes a reward r_t. Define $\boldsymbol{\phi}_t \in \mathbb{R}^{N^d}$ to be the indicator vector corresponding to $\mathbf{x}_t$ where $(\boldsymbol{\phi}_t)_k = 1$ if $k = (\mathbf{x}_t)$ is the location in a vector representation of the grid, and zero otherwise. Then the belief state $(\boldsymbol{\mu}_t, \Sigma_t)$ updates as follows:

$$\mathbf{q} = r_t \boldsymbol{\phi}_t + \Lambda_{t-1} \boldsymbol{\mu}_{t-1} \tag{3}$$

$$\Lambda_t = \frac{\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T}{\sigma_r^2} + \Lambda_{t-1}, \ \Sigma_t = \Lambda_t^{-1} \tag{4}$$

$$\boldsymbol{\mu}_t = \Sigma_t \mathbf{q}, \tag{5}$$

where $\Lambda_t = \Sigma_t^{-1}$ is the *precision* matrix. This assumes that the sampling noise σ_r is known, e.g. from previous observations or known sensor characteristics.

This gives us the first component of the decision heuristic: at time t, the expected payoff $\Delta R_{i,t}$ of option i is $\mu_{i,t}$, the i^{th} component of $\boldsymbol{\mu}_t$.

We now turn to the information value component $\Delta I_{i,t}$.

B. Information value

We use entropic information as our information metric. Since the posterior distribution is Gaussian, its entropy at time t is

$$H_t = \frac{N^d \log 2\pi + \log \det \Sigma_t}{2} = \frac{N^d \log 2\pi - \log \det \Lambda_t}{2},$$

where the second equality comes from the definition of Λ_t. This form of the expression for the entropy is convenient because the Λ_t update rule (4) is linear and $\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T$ is sparse. For $(\mathbf{x}_t) = k$, $\left(\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T\right)_{kk} = 1$ is the only non-zero element.

Because of this sparsity, we can calculate the change in the determinant over one time step analytically:

$$\det \Lambda_t = \det \left(\frac{\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T}{\sigma_r^2} + \Lambda_{t-1} \right) = \det \Lambda_{t-1} + \frac{1}{\sigma_r^2} M_{kk}, \tag{6}$$

where M_{kk} is the (k, k) minor of Λ_{t-1}.

Then the change in entropy is

$$\begin{aligned} H_t - H_{t-1} &= -\frac{1}{2} \left(\log \det \Lambda_t - \log \det \Lambda_{t-1} \right) \\ &= \frac{1}{2} \left(\log \det \Lambda_{t-1} - \log \left(\det \Lambda_{t-1} + \frac{1}{\sigma_r^2} M_{kk} \right) \right) \\ &\approx -\frac{1}{2} \frac{M_{kk}}{\sigma_r^2 \det \Lambda_{t-1}}, \end{aligned}$$

where the last approximation becomes increasingly valid as t increases and the information gain gets smaller and smaller.

Motivated by this approximation we define the information value of location i at time t to be

$$\Delta I_{i,t} = \frac{M_{ii}}{\sigma_r^2 \det \Lambda_{t-1}}, \tag{7}$$

where M_{ii} is the (i, i) minor of Λ_{t-1} as above.

See also the Backward Selection for Gaussian method of Choi and How [8],[9], who examine other, more general cases of information-based search.

C. Decision heuristic

An important aspect of human decision making is that it is *noisy*, so that humans do not necessarily deterministically optimize a value function. For example, when faced with a completely unknown situation, a good model is that human subjects will pick randomly among their options.

We choose to incorporate decision noise in our model by adding i.i.d. (over i and t) random noise to the heuristic value function. Putting all the terms together the value function $Q_{i,t}$ becomes

$$Q_{i,t} = \beta\mu_{i,t} + (1-\beta)\Delta I_{i,t} + \sigma_D\varepsilon_{i,t},\ \varepsilon_{i,t} \sim \mathcal{N}(0,1). \quad (8)$$

The decision given by the heuristic at time t is

$$\arg\max_i Q_{i,t}.$$

For purposes of numerical implementation we scale both $\mu_{i,t}$ and $\Delta I_{i,t}$ by their maximum values at each time step:

$$\frac{\mu_{i,t}}{\max_j \mu_{j,t}},\ \frac{\Delta I_{i,t}}{\max_j \Delta I_{j,t}}.$$

With this normalization, both deterministic elements of the value function are scaled to lie in $[0,1]$. The magnitude of the decision noise, σ_D, also naturally lies in $[0,1]$, since for cases $\sigma_D \geq 1$ the noise term dominates the deterministic terms in Q and decisions will be made at random.

The introduction of decision noise results in another tradeoff in addition to the explore-exploit tradeoff, this time between two different types of exploration: directed exploration driven by the ΔI term which seeks information about the rewards, and random exploration driven by the $\sigma_D\varepsilon$ term. The following numerical example shows that these two terms can trade off in an interesting way.

IV. MOTIVATING NUMERICAL EXAMPLE

In this section, we motivate the role of parameters β and σ_D in the explore-exploit tradeoff. We study a numerical example using a reward structure previously used in human experiments, as discussed in [10] and Chapter 4 of [11]. This reward structure is designed such that an agent that carries out insufficient exploration is likely to get caught at a local maximum. If β is too high, the agent will pay excessive attention to immediate rewards $\mu_{i,t}$ and not seek enough information $\Delta I_{i,t}$; however, it may be able to compensate by adding decision noise $\sigma_D\varepsilon_{i,t}$.

Consider a two-dimensional ($d = 2$) example with grid size $N = 10$. The reward surface is as shown in Figure 1: it has the characteristic that there is no gradient along the y direction, both ends along the x direction are local maxima, but the line $x = 10$ is the unique global maximum.

This reward surface intuitively requires exploratory behavior because of the two local maxima: if started on the left side of the domain, a simple gradient following algorithm will get stuck at the suboptimal local maximum. We choose $T = 90$ time steps so that the agent can sample at most 90% of the space. The variance of the sampling noise is $\sigma_r^2 = 1/1200$ while the mean surface value is 0.25 so that the average signal-to-noise ratio is $0.25/\sigma_r \approx 8.66$.

The algorithm requires values of the priors $\boldsymbol{\mu}_0$ and Σ_0. For the means it is reasonable to set the uniform prior $\boldsymbol{\mu}_0 = \mathbf{0}$. The appropriate prior on covariance is less obvious. Following [12], we choose a prior that is exponential with a spatial length scale:

$$\Sigma_0(i,j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/\lambda)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the 1-norm of the distance between points i and j. For the present example, we set $\lambda = 3$.

In order to understand the tradeoff between directed exploration and noise-based exploration, we computed via simulation the expected total rewards accumulated by the algorithm for $(\beta, \sigma_D) \in [0,1] \times [10^{-5}, 10^{0.25}]$. The resolution of the set of simulations was 30 linearly-spaced points in β and 20 log-spaced points in σ_D, and for each pair of values (β, σ_D), the expected value was computed by simulating 200 runs of the problem. For each simulation, the initial location of the agent was drawn from a uniform distribution.

Expected reward per time step as a function of the two parameters (β, σ_D) for this experiment is shown in Figure 2. As expected, some exploration was required to perform well in the task: in the deterministic decision limit $\sigma_D \to 0$, maximum rewards are achieved for a value of β of about 0.5. Comparison between Figures 1 and 2 shows that at the optimal tunings of the parameters, the expected rewards per time step of about 0.5 are near the value at the global optimum, so the algorithm is achieving near-optimal performance.

Furthermore, Figure 2 shows a tradeoff between weighting on directed exploration and random exploration. As σ_D increases, making action selection more random, one can maintain high performance by increasing β, thereby paying more attention to immediate rewards and reducing the weight on directed exploration.

We can develop a better understanding of the role of exploration by measuring it. The agent's trajectory $\mathbf{x}_t = (x(t), y(t))$ forms a curve on the grid. We define a measure of exploration e_T over the T time steps by taking the variance of the time series representing this trajectory:

$$e_T = \frac{1}{T}\sum_{t=1}^{T}\left((x(t)-\bar{x})^2 + (y(t)-\bar{y})^2\right),$$

where $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x(t)$ and $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y(t)$ are the average values of x and y. This measure has the physical interpretation of being the moment of inertia of the trajectory curve. It is bounded below by zero (representing an agent that does not move at all), and larger values of e_T correspond to more time being spent away from the average position.

Figure 3 plots e_T for the same set of parameters as in Figure 2. Again there is a tradeoff between β and σ_D: as random exploration is increased by increasing σ_D, a constant level of total exploration (as measured by e_T) can be maintained by increasing β, thereby paying more attention to immediate rewards and reducing the weight on directed exploration. The monotonic nature of the tradeoff is intuitive, although its specific shape is not trivial to explain.

Furthermore, the plots show that the level sets of e_T and expected reward have essentially the same structure. This strongly suggests that tuning β and σ_D has an effect by altering the overall level of exploration, and it is this overall level of exploration that governs performance.

The effects of β in the $\sigma_D \to 0$ deterministic decision case are also interesting. Although it is difficult to develop intuition for the effect of β in this case because of the

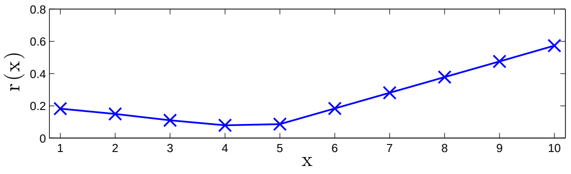

Fig. 1. Profile of the mean reward surface for the numerical example. The grid points are at $x = 1, 2, \ldots, 10$. There is no gradient in the y direction, while in the x direction there is a local maximum at $x = 1$, a local minimum at $x = 4$, and a global maximum at $x = 10$.

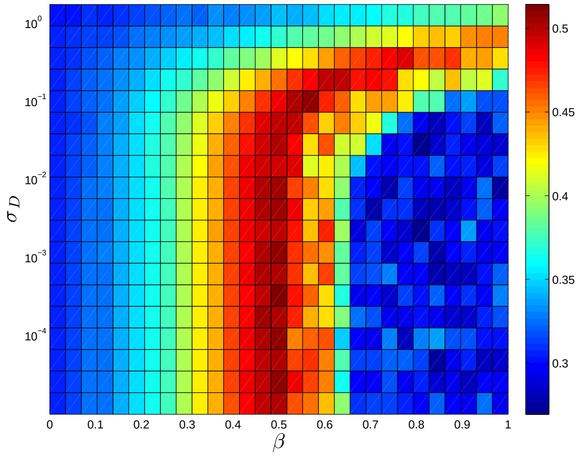

Fig. 2. Expected reward per time step for various parameter values. Note the tradeoff between weighting on immediate reward β and decision noise σ_D. For small decision noise, expected rewards are highest for $\beta \approx 0.5$, but as noise increases one can maintain performance by increasing β.

large values of N and T, in the following section we derive analytical results for more tractable cases.

V. OPTIMIZED HEURISTIC AND THE ROLE OF β

As the previous example shows, the two parameters β and σ_D in the heuristic interact in a complex way to affect performance of the algorithm. In this section we derive analytical results in the $\sigma_D = 0$ limit. The analysis provides insight into the role of β, and we can compute optimal tunings in the cases addressed. In Section V.A we analyze a low-dimensional case that yields key insights. In Section V.B we discuss generalizations to higher dimensions, other true distributions of rewards, and the $\sigma_D \neq 0$ case.

A. Analytical optimization of a low-dimensional case

To start, consider the $d = 1$ dimensional problem where $N = 2$, i.e. a grid with two options. Furthermore, let $\sigma_r = 0$ so there is no sampling noise and $T = 2$ so the objective is simply $\max(r_1 + r_2)$. Let the true reward values $\mathbf{m}$ be jointly Gaussian distributed as

$$\mathbf{m} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \bar{\mu}_2 \end{bmatrix}, \begin{bmatrix} 1 & \sigma\rho \\ \sigma\rho & \sigma^2 \end{bmatrix}\right).$$

Similarly, let the agent's prior over those values be the joint Gaussian distribution

$$\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0), \text{where } \boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_0 = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Also, assume that $\rho \geq 0$ for convenience.

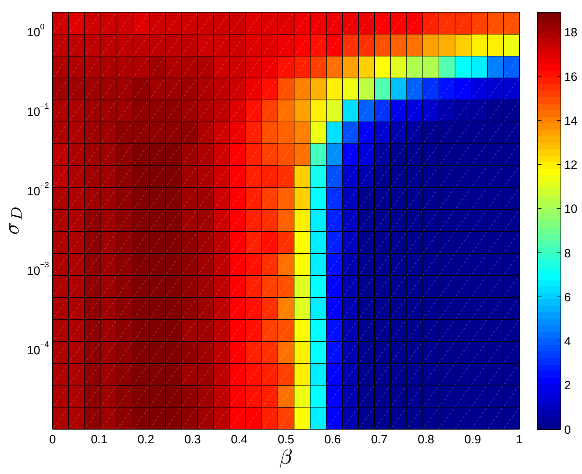

Fig. 3. Exploration measure e_T for the same parameter values as in Figure 2. Here the tradeoff between the two types of exploration is made clear: level sets of e_T represent sets of constant total exploration. As one increases random exploration through σ_D, one can maintain a constant level of total exploration by increasing β to decrease directed exploration. The level sets of e_T look very similar to the level sets of expected rewards, suggesting that it is the overall level of exploration that drives performance.

Note that in the case $\bar{\mu}_2 = 0$ and $\sigma = 1$ the prior is identical to the actual distribution of the rewards, but in any other case they are distinct. The difference could be due to, e.g., measurement error in calibrating the prior or a change in the true statistics since the last time the agent was confronted with the problem.

We are interested in choosing the value of β for our heuristic that maximizes expected total rewards over all possible reward values and initial locations.

Since the agent can begin in either of the two locations with equal probability, $\mathbb{E}[r_1] = \bar{\mu}_2/2$ independent of β. Therefore the optimization problem reduces to

$$\beta^* = \arg\max_{\beta} \max_{x_2} \mathbb{E}[r_2|r_1]. \tag{9}$$

That is, given r_1, the algorithm has to decide whether to stay in its current location or switch to the alternative location. This is a well-studied problem in signal detection theory (see, e.g., [13] or Example II.B.2 of [14]). The optimal β maximizes the expected payoffs of the decision made by the algorithm.

The detection theory solution consists of setting a threshold m^* on the observed reward r_1 and switching if $r_1 < m^*$. The optimal threshold is a function of the prior beliefs about m and the costs associated with each decision. If the agent is equally likely to be in either initial location, the optimal threshold is

$$m^* = \frac{1}{2}(0 + \bar{\mu}_2) = \bar{\mu}_2/2. \tag{10}$$

We show that the optimal tuning of our algorithm reduces to the optimal solution of the detection problem.

We begin by computing the expected value of the algorithm for a given value of β. At time $t = 1$, the agent observes either m_1 or m_2, each with probability 1/2. In either case the observed reward m_i is now known with certainty, so its inferred value is $\mu_{i,1} = m_i$ and the inferred value of the unobserved reward m_j is $\mu_{j,1} = \rho m_i$. Similarly, $\Lambda_0 = \Sigma_0^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$. The two minors M_{kk} are both equal to $1/(1-\rho^2)$, so $\Delta I_{i,1} = 0$ for the observed location and $\Delta I_{j,1} = M_{jj} = \frac{1}{1-\rho^2}$ for the unobserved location. For a given value of β, the expected value in the optimization problem (9) is the average of the expected values $\mathbb{E}[r_2|r_1]$ for the two cases $\mathbf{x}_1 = 1, 2$.

We proceed by computing the expected value of the algorithm for the case of starting in location 1, so $\mathbf{x}_1 = 1$. In this case the function $Q_{i,t}$ (8) takes the following values:

$$Q_{1,1} = \beta m_1,\ Q_{2,1} = \beta\mu_{2,1} + \frac{1-\beta}{1-\rho^2} = \beta\rho m_1 + \frac{1-\beta}{1-\rho^2}.$$

The algorithm picks the maximum of $\{Q_{1,1}, Q_{2,1}\}$ and switches if $Q_{2,1} > Q_{1,1}$, or equivalently if $\beta\rho m_1 + \frac{1-\beta}{1-\rho^2} > \beta m_1$. This is equivalent to setting a threshold value m^* and switching if

$$r_1 = m_1 < m^* = \frac{1-\beta}{\beta}\frac{1}{(1-\rho)(1-\rho^2)}, \tag{11}$$

which sets a threshold m^* as a function of β and ρ.

If the algorithm decides to switch locations, the agent will then obtain the reward $r_2 = m_2$. Otherwise it stays in the original location and receives $r_2 = m_1$. The expected value of r_2 given the algorithm's decision is then given by

$$\mathbb{E}[r_2|\mathbf{x}_1 = 1] = \mathbb{E}[m_1|m_1 \geq m^*] + \mathbb{E}[m_2|m_1 < m^*].$$

Since m_1 and m_2 are jointly Gaussian, this expectation is analytically tractable and is equal to

$$\phi(m^*) + \mu_2\Phi(m^*) - \rho\sigma\phi(m^*) = \mu_2\Phi(m^*) + (1-\rho\sigma)\phi(m^*),$$

where $\phi(z)$ and $\Phi(z)$ are the the pdf and cdf, respectively, of the standard normal distribution.

In the case where the agent's initial location is $\mathbf{x}_1 = 2$, the agent observes $r_1 = m_2$. The function $Q_{i,t}$ takes the following values:

$$Q_{1,1} = \beta\mu_{1,1} + \frac{1-\beta}{1-\rho^2} = \beta\rho m_2 + \frac{1-\beta}{1-\rho^2}, Q_{2,1} = \beta m_2.$$

This is symmetric to the case $\mathbf{x}_1 = 1$ under interchange of $i = 1$ and $i = 2$ because of the symmetry of the prior. Again, the algorithm switches to the alternate location if $Q_{1,1} > Q_{2,1}$, or

$$r_1 = m_2 < m^* = \frac{1-\beta}{\beta}\frac{1}{(1-\rho)(1-\rho^2)},$$

where the threshold m^* is the same as above, again due to the symmetry of the prior. The expected value of r_2 given the algorithm's decision is

$$\mathbb{E}[r_2|\mathbf{x}_1 = 2] = \mathbb{E}[m_2|m_2 \geq m^*] + \mathbb{E}[m_1|m_2 < m^*].$$

This expectation can again be expressed in closed form, and takes the value

$$\bar{\mu}_2\left(1 - \Phi\left(\frac{m^* - \bar{\mu}_2}{\sigma}\right)\right) + \sigma(1-\rho\sigma)\phi\left(\frac{m^* - \bar{\mu}_2}{\sigma}\right).$$

Since $x_1 = 1$ or 2 with equal probability, for a given threshold m^*, the expected value in the optimization problem (9) is the simple average of the expected rewards for each initial position $\mathbb{E}[r_2|\mathbf{x}_1 = 1]$ and $\mathbb{E}[r_2|\mathbf{x}_1 = 2]$:

$$\begin{aligned}\mathbb{E}[r_2|r_1] = \\ &\frac{1}{2}\left[\bar{\mu}_2\left(1 + \Phi(m^*) - \Phi\left(\frac{m^* - \bar{\mu}_2}{\sigma}\right)\right)\right. \\ &\left. + (1-\rho\sigma)\phi(m^*) + \sigma(1-\rho\sigma)\phi\left(\frac{m^* - \bar{\mu}_2}{\sigma}\right)\right].\end{aligned}$$

The parameter ρ is fixed, so the optimization (9) reduces to picking the value $\beta = \beta^*$ that results in the threshold m^* that maximizes $\mathbb{E}[r_2|r_1]$. The expression for $\mathbb{E}[r_2|r_1]$ is somewhat unwieldy, but several special cases are informative.

First, consider the case $\bar{\mu}_2 = 0, \sigma = 1$, which is the case where the prior is equal to the actual distribution. In this case the expectation reduces to

$$\mathbb{E}[r_2|r_1] = (1-\rho)\phi(m^*).$$

We want to pick the value of β that maximizes this expectation, which means maximizing $\phi(m^*)$ since $1-\rho$ is fixed. If $\rho = 1$ the expected rewards are zero independent of m^*, so consider cases $\rho < 1$. The function $\phi(z)$ takes its unique maximum at $z = 0$, so we set the threshold $m^* = 0$. Equation (11) then implies that the optimal value of β is $\beta^* = 1$, so the optimal tuning of the heuristic is

$$Q_{i,t} = \mu_{i,t}.$$

In this case the optimal tuning of the algorithm is pure exploit and no explore. The heuristic ignores the information gain component ΔI and only weights inferred rewards $\boldsymbol{\mu}$. The threshold is set equal to 0, cf. Equation (10) where $\bar{\mu}_2 = 0$. This is identical to the standard optimal detection theory result [14], and the heuristic only weights $\mu_{i,t}$ because in this case the linear inference model is optimal. In this case the heuristic is not particularly beneficial, and setting β to anything less than one is suboptimal. However, we show next that the heuristic provides robustness in cases where the field statistics are not known perfectly.

Consider the case above with $\sigma = 1$ but $\bar{\mu}_2 \neq 0$, so the prior is exact except for the mean value $\bar{\mu}_2$. In this case the inference is no longer optimal, so neither is weighting only the inferred reward. The expected reward $\mathbb{E}[r_2|r_1]$ is

$$\begin{aligned}&\frac{1}{2}\left[\bar{\mu}_2\left(1 + \Phi(m^*) - \Phi\left(m^* - \bar{\mu}_2\right)\right)\right. \\ &\left. + (1-\rho)\left(\phi(m^*) + \phi(m^* - \bar{\mu}_2)\right)\right].\end{aligned}$$

For any given $\bar{\mu}_2$ and ρ, the expectation can be maximized with respect to the threshold m^*, and in general the optimal threshold is non-zero. For example, if $\bar{\mu}_2 = 1, \rho = 0.5$, the maximum occurs at $m^* = 0.5$, or $\beta^* = 16/19 \approx 0.84$. If, instead, $\bar{\mu}_2 = -1, \rho = 0.5$, the maximum occurs at $m^* = -0.5$, or $\beta^* = 16/13 \approx 1.23$. Again, the optimal threshold in both cases is $m^* = \bar{\mu}_2/2$, as in the detection theory solution. This shows how setting $\beta \neq 1$ provides robustness by helping the algorithm recover the optimal threshold in the face of suboptimal inference.

B. Discussion and generalizations

The results in the previous section make intuitive sense because in the case where the true distribution $\mathcal{D}$ is Gaussian and the prior statistics are correctly calibrated, the inference model is optimal. In that case the inferred value term $\mu_{i,t}$ is the optimal expected value of the option i at time t, and the optimal action at each time t is simply to pick the maximum of the $\mu_{i,t}$, so the optimal β reflects that and is equal to one.

If, however, the true distribution $\mathcal{D}$ is not Gaussian or the prior statistics are incorrect, the inference model will be suboptimal. If the world is "better" than expected by the prior, as in the case where $\bar{\mu}_2 = 1$, setting $\beta < 1$ provides robustness by encouraging exploration, whereas if it is "worse", as in the case where $\bar{\mu}_2 = -1$, setting $\beta > 1$ provides robustness by weighting expected rewards more highly and discouraging exploration.

This suggests the form of a simple feedback control law for β: at each time step, if the world appears "better" than implied by the prior, decrease β to encourage guided exploration. If, instead, the world appears "worse", increase β to discourage it. At time t, an estimate p_t of the degree to which the world is "better" or "worse" could be made, e.g., by the mean difference between the inferred rewards at the current and previous time steps:

$$p_t = \frac{1}{N^d} \sum_{i=1}^{N^d} \left(\mu_{i,t} - \mu_{i,t-1}\right).$$

Then if the inferred values $\mu_{i,t}$ are increasing, the world appears to be "better" than expected and $p_t > 0$. Furthermore, since the field is stationary, the inference is getting monotonically more accurate in time, so $p_t \to 0$ as $t \to \infty$. Then, setting $K > 0$, the proportional control law $\beta_t = \beta_{t-1} - K p_t$ biases β in the desired direction.

In the case that the true distribution $\mathcal{D}$ is Gaussian and the prior statistics are correctly calibrated, we make the claim that the optimal value of β is $\beta^* = 1$ in the general case of $N > 2, d > 1$. The proof of this claim, as well as a more rigorous claim about when the optimal value of β lies above or below 1 will be the subject of future work.

VI. Conclusion

In this paper we have presented a heuristic that was developed to describe human behavior in a simple explore-exploit task. The heuristic includes two forms of exploratory behavior: directed exploration, guided by seeking information about rewards, and random exploration, provided by random noise. We use this heuristic to construct an algorithm to solve explore-exploit problems in spatially distributed scalar fields. The algorithm uses an optimal Bayesian inference algorithm for building beliefs about the field, and then applies the heuristic to solve the decision problem of which location to visit next.

Using a numerical example, we show that the two types of exploratory behavior trade off in an interesting way, but that both influence an overall level of exploration which, when measured, is shown to strongly correlate with task performance. In particular, in the case where there is no random exploration, we show that there is a level of directed exploration that produces optimal performance in the task.

To gain intuition for the role of the level of directed exploration in the case without random exploration, we consider an example problem where the field is distributed over two points, and show that in the case where the inference is optimal, the optimal tuning of the heuristic is to put full weight on expected rewards at the expense of all directed exploration; in this case the heuristic reduces to an optimal Bayesian detector. However, in the general case where the inference is not optimal, for example if it was given incorrect field statistics, including some directed exploration provides robustness to modeling errors.

We make the claim that in the general case of $N \geq 2, d \geq 1$, if the inference is optimal, then the optimal value of β is still $\beta^* = 1$. We also provide the intuition that if the prior is "pessimistic" compared to the true distribution of rewards, β^* should be less than 1, whereas if it is "optimistic", the opposite should be the case. This intuition suggests the form of a feedback control law for β. The relationship between the true distribution $\mathcal{D}$, the prior reward statistics $\boldsymbol{\mu}_0$ and Σ_0, and the optimal value of β will be the subject of future work.

References

[1] W.B. Powell, *Approximate Dynamic Programming*, 2nd Ed. Hoboken, NJ: Wiley, 2011.

[2] G. Aston-Jones and J.D. Cohen. "An integrative theory of locus coeruleus-norepinephrine function: Adaptive gain and optimal performance," *Annu. Rev. Neurosci.*, Vol. 28, No. 1, pp. 403-450, 2005.

[3] J.D. Cohen, S.M. McClure and A.J. Yu. "Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration," *Phil. Trans. R. Soc. B*, Vol. 362, pp. 933-942, 2007.

[4] F. Balci, P. Simen, R. Niyogi, A. Saxe, P. Holmes and J.D. Cohen. "Acquisition of decision making criteria: Reward rate ultimately beats accuracy," *Attention, Perception & Psychophysics* 73 (2), 640-657, 2011.

[5] R.C. Wilson, A. Geana, J.M. White, E.A. Ludvig, and J.D. Cohen. "To boldly go: Ambiguity-seeking in human exploratory decision making." In preparation.

[6] D.W. Stephens and J.R. Krebs, *Foraging Theory*. Princeton, NJ: Princeton University Press, 1987.

[7] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," arXiv:1204.5721, 2012.

[8] H.-L. Choi and J.P. How. "A multi-UAV targeting algorithm for ensemble forecast improvement," in *Proc. AIAA Guidance, Navigation And Control Conference and Exhibit*, Hilton Head, SC, 2007.

[9] H.-L. Choi. "Adaptive sampling and forecasting with mobile sensor networks," PhD thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, 2008.

[10] D. Tomlin, A. Nedic, M.T. Todd, R. C. Wilson, D. A. Prentice, P. Holmes, and J. D. Cohen. "Group foraging task reveals separable influences of individual experience and social information," presented at *Neuroscience* 2011, Washington, DC.

[11] A. Nedic. "Models for individual decision-making with social feedback," PhD thesis, Dept. of Electrical Engineering, Princeton Univ., Princeton, NJ, 2011.

[12] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, Vol. 95, No. 1, pp. 48-74, Jan. 2007.

[13] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice Hall PTR, 1993, Ch. 12.

[14] H. Poor, *An Introduction to Signal Detection and Estimation*, 2nd Ed. New York: Springer-Verlag, 1994.