Abstract

Introduction

Many decisions that people are faced with require finding a balance between exploiting known information for short-term gain and exploring new sources of information that may lead to future reward. For example, a person might choose to go to a restaurant that they know well and have had multiple satisfying meals at, or choose to try a new restaurant that could result in a better or worse experience. This is known as the explore-exploit trade-off.

Tasks requiring this kind of trade-off can be often be interpreted as multi-arm bandit (MAB) problems. A MAB problem consists of A possible actions, where the a-th action yields the random reward r_t drawn from an unknown distribution. Over T trials, the agent must choose $a_{t:T}$ such that $\sum r_{t:T}$ is maximized. The policy $\tilde{\pi}$ maps the observed history of actions and rewards $h_{1:t-1} = [(a_1, r_1), ..., (a_{t-1}, r_{t-1})]$ to the next action a_t for all possible histories $h_{1:t-1} \in \mathcal{H}$. The goal in a MAB problem is to find the optimal policy such that the expected reward over T trials, $\mathbf{E}[r_{1:T}|\tilde{\pi}(h_{1:t-1})]$, is maximized. This is the solution to the Bellman equation

$$\mathbf{E}[r_{1:T}|\tilde{\pi}(h_{1:t-1})] = \mathbf{E}[r_t|\tilde{\pi}(h_{1:t-1})] + \mathbf{E}[r_{t+1:T}|\tilde{\pi}(h_{1:t})] \quad (1)$$

where the first term yields the expected reward on trial t and the second term recursively defines the expected reward on all subsequent trials up to T. This second term can be evaluated for trial $t+1$ by weighing the expected reward given all possible histories $\mathcal{H}_{1:t} = [(a_1, r_1), ..., (a_{t-1}, r_{t-1}), (a_t, r_t)]$ for $r_t \in \mathcal{R}$ by the probability of that history occurring:

$$\mathbf{E}[r_{t+1:T}|\tilde{\pi}(h_{1:t})] = \sum_{r_t \in \mathcal{R}} p(r_t|a_t, h_{1:t-1})\mathbf{E}[r_{1:T}|\tilde{\pi}(h_{1:t})] \quad (2)$$

If the reward r observed after choosing action a is drawn from the distribution $H(\theta_a)$, an agent must employ a strategy that balances choosing the action that maximizes the reward on trial t, $a_t^* = \text{argmax}_{a_t \in A}\mathbf{E}[H(\theta_{a_t})]$, and learning more about θ_A by choosing novel actions to maximize reward on future trials. As the problem of evaluating all possible histories is intractable, one of two general classes of approximations can be used. The first, simulation-based methods, involves using a subset of possible histories to evaluate the long-term reward of an action, and includes Thompson sampling (**?**) and Monte Carlo tree search (**?**). Rather than relying on simulations, myopic strategies define a value function for approximating long-term reward. Common myopic strategies include upper confidence bound (UCB) (**?**) and epsilon-greedy (**?**) algorithms.

Contextual multi-armed bandit (CMAB) problems introduce additional information into the standard MAB problem by way of a set of features associated with the set of possible examples (**?**). For example, a standard MAB formulation of the problem of choosing which restaurant to eat at assumes that the reward yielded by any two restaurants will be uncorrelated. However, it might be the case that these restaurants share a set of features (e.g. size, location, menu items) such that choosing similar restaurants can be assumed to yield similar rewards. Rather than having to execute an action to be able to evaluate its expected reward, considering shared features allows one to learn a function, $R : x_t \rightarrow r_t$ that maps features of the action a_t, x_{a_t} to that action's expected reward, r_t.

With the inclusion of context, CMAB problems require the additional step of learning the function R between updating the history, $h_{1:t-1} = [(a_1, x_1, r_1), ..., (a_{t-1}, x_{t-1}, r_{t-1})]$, and choosing $a_t = \tilde{\pi}(h_{t-1})$. Formally, function learning describes how people predict a continuous-valued output given an input, and can be thought of as a continuous extension of category learning. Theories of function learning typically follow either a rule-based or similarity-based approach. Rule-based approaches posit that people learn this mapping by assuming that the unknown function belongs to a particular parametric family, then inferring the most likely parameterization after observing input/output pairs. For example **?** considers polynomials up to degree 6, and **?** consider power-law functions. While this approach attributes rich representations to learners, it not clear how these representations are acquired. Similarity-based theories suggest instead that learning is the result of forming associations between input/output pairs and generalizing these associations to similar inputs. **?** implement a connectionist network where inputs activate a set of input nodes according to a Gaussian similarity function and each output node is activated according to learned weights between the input and output nodes. This approach does not require any assumptions about functional form and allows for flexible interpolation, but does not support generalization to inputs that are distant from past examples. While neither approach seems to fully capture human function learning, hybrid models have been introduced that take advantages from both rule and similarity-based theories. **?** extend their connectionist model by including a layer of hidden nodes, each corresponding to a parameterization of a particular functional family.

More recently, **?** proposed Gaussian process regression (GRP) as a unified approach to function learning. GPR solves the problem of learning to map inputs to outputs by assuming that the outputs $\mathbf{y}_N$ are drawn from the N dimensional distribution $\mathcal{N}(m(\mathbf{x}_N), k(\mathbf{x}_N, \mathbf{x}_N))$, where m defines a mean function that broadly captures the function in the absence of data, and k defines how the inputs relate to each other. A common class of kernels, radial basis functions, e.g. $k(x_i, x_j) = \sigma^2 \exp(-\frac{(x_i - x_j)^2}{2l^2})$, follows the assumption made by similarity-based models that similar inputs will map to similar outputs, with the parameter l determining how

quickly the correlation between two inputs decreases as the distance between them increases. Similarly, many rule-based models can be expressed in terms of the polynomial kernels $k(x_i,x_j) = \sigma^2(x_i x_j + c)^d$ of degree d. For the unobserved input x^*, the output y^* is drawn from a normal distribution with the posterior mean and variance functions

$$\begin{aligned} m(x^*|\mathbf{x}_N,\mathbf{y}_N) &= m(x^*) + \mathbf{k}(x^*)^T(\mathbf{K}+\sigma^2\mathbf{I})(\mathbf{y}_N - m(\mathbf{x}_N)) \\ v(x^*|\mathbf{x}_N,\mathbf{y}_N) &= k(x^*,x^*) - \mathbf{k}(x^*)^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(x^*) \end{aligned} \tag{3}$$

where $\mathbf{k}(x^*) = [k(x_1,x^*),...,k(x_N,x^*)]$ and $\mathbf{K} = k(\mathbf{x}_N,\mathbf{x}_N)$.

Bayesian optimization (**?**) provides a framework for solving CMAB problems when the reward function is assumed to be drawn from a Gaussian process. On each trial, the acquisition function $\text{acq}(a) = \text{acq}(m(a|h_{1:t-1}), v(a|h_{1:t-1}))$ is used to approximate the expected long-term reward of each action, and the next action is chosen according to the policy $\tilde{\pi}(h_{1:t-1}) = \text{argmax}_{a\in A}\text{acq}(a)$. Once the action/reward pair has been observed, the mean and variance functions at each action are updated. Of the commonly used acquisition functions, most follow the heuristic of favoring actions where both the mean and variance of the associated distribution over rewards are high; that is, where there is opportunity to either exploit a known reward or explore an unknown area of the reward function. The Gaussian process UCB acquisition function directly weighs exploration and exploitation by taking a weighted sum of the mean and variance of the reward function at a

$$\text{acq}_{UCB}(a) = m(a|h_{1:t-1}) + \beta\sqrt{v(a|h_{1:t-1})} \tag{4}$$

where β determines the preference for exploration. Another strategy is to maximize the probability of improving the current best reward:

$$\text{acq}_{PoI}(a) = \Phi\left(\frac{m(a|h_{1:t-1}) - m(a*|h_{1:t-1})}{\sqrt{v(a|h_{1:t-1})}}\right) \tag{5}$$

where a^* is the action that is believed to maximize reward and Φ is the cumulative distribution function of the standard normal. A similar strategy is to maximize the expected improvement:

$$\text{acq}_{EI}(a) = (m(a|h_{1:t-1}) - m(a^*|h_{1:t-1}))\Phi(z) + \sqrt{v(a|h_{1:t-1})} \tag{6}$$

where $z = \frac{m(a|h_{1:t-1}) - m(a^*|h_{1:t-1})}{\sqrt{v(a|h_{1:t-1})}}$.

An alternative approach to maximizing this heuristic is to work directly with the distribution that describes the maximum of the function. Entropy search (**?**) achieves this by maximizing the expected change in entropy about $p(x^*)$, where $x^* = \text{argmax}_{x\in X}R(x)$, yielded after choosing the action a:

$$\text{acq}_{ES}(a) = H(p(x^*|h_{1:t-1})) - \mathbf{E}[H(p(x^*|h_{1:t}))] \tag{7}$$

and predictive entropy search (**?**) maximizes an equivalent value:

$$\text{acq}_{PES}(a) = H(p(y|h_{1:t-1},x)) - \mathbf{E}[H(p(y|h_{1:t-1},x,x^*))] \tag{8}$$

Max-value entropy search **?** takes a slightly different approach, instead working to minimize entropy about $p(y^*)$:

$$\text{acq}_{MES}(a) = H(p(y|h_{1:t-1},x)) - \mathbf{E}[H(p(y|h_{1:t-1},x,y^*))] \tag{9}$$

While each of these strategies has the advantage of working with a global distribution over features and their associated rewards rather than a local heuristic, none of the required entropies can be computed analytically. However, a number of approximations have been proposed.

Human learners' approach to the explore/exploit trade-off has typically been studied in tasks where the goal is to maximize cumulative reward (e.g. **???**). However, the need for this trade-off can be observed in a number of other tasks. For instance, an agent might be instead interested in finding the maximum possible reward after T trials. This problem is known as *optimization* (e.g.). In these cases, earning a high reward on any particular task t is not important, as long as the actions lead to finding the global maximum. Another example is when an agent might instead be interested in learning the the reward function across all actions. These cases are described by *active learning* (AL) (e.g. **??**). In typical learning paradigms learners are asked to make judgments based on evidence that has been preselected for them. In contrast, AL describes tasks where the learner plays a role in selecting the evidence to observe.

While reinforcement learning, optimization, and active learning are often studied in isolation, there are cases where different tasks might rely on the same mapping between features and rewards. *Multi-task learning* (e.g.) involves exploiting common structure among distinct tasks in order to learn each task more effectively.

Modeling the Explore/Exploit Trade-Off

We consider models of the three types of task requiring an explore/exploit trade-off that were described above. While the mapping from features to rewards, $R : x_N \rightarrow \mathcal{R}$, can be described in similar terms for all three tasks, their goal-specific reward, and thus the function used to approximate long-term reward, are distinct.

Since the goal in RL tasks is to maximize cumulative reward over time, the goal-specific reward is the same as the reward, that is:

$$\mathbf{E}_{rl}[r'_t|a_t,h_{1:t-1}] = \mathbf{E}[r_t|a_t,h_{1:t-1}] \tag{10}$$

In contrast, the goal of the optimization problem is simply to find the maximum possible reward withing T steps. As such, the goal specific reward is defined as:

$$\mathbf{E}_{opt}[r'_t|a_t,h_{1:t-1}] = max(\mathbf{E}[r_t|a_t,h_{1:t-1}] - \sum_{i=1}^{t-1} r'_i, 0) \tag{11}$$

That is, on trial if trial t yields an increase in reward over the previous maximum reward, r'_t is the difference. If not, r'_t is 0. Since active learning is concerned with learning the reward function rather than the magnitude of the rewards themselves,

its goal-specific reward can be described as the sum of the decrease in variance across all possible actions:

$$\mathbf{E}_{al}[r'_t|a_t, h_{1:t-1}] = \sum_{a \in A} \mathbf{Var}[r|a, h_{1:t-1}] - \mathbf{Var}[r|a, h_{1:t}] \quad (12)$$

We hope to address three questions through the comparison of a number of models and there abilities to capture human performance on these tasks. First, we ask whether human learners act rationally with respect to information about the underlying reward function; that is, are people explicitly modeling the underlying reward function as an intermediate step in their decision making process, or relying on a less costly heuristic? To answer this question, we compare the class of Gaussian process-based strategies (e.g. GP UCB, probability of improvement, expected improvement, entropy search) with a model based on gradient descent. Second, we ask whether human learners act rationally with respect to their task-specific goal. In other words, do people use different strategies depending on their current goal, or rely on a more general measure of utility (e.g. choose actions with high expected reward and high uncertainty). We compare three task specific strategies with the general strategies of GP UCB, probability of improvement, expected improvement, and epsilon greedy. Third, we ask whether people adopt strategies that take into account higher-order information about the underlying reward function across multiple distinct tasks. We compare both strategies that include information about R in their estimation of long-term reward across tasks and with those that consider only the long-term reward of the current task.

Experiments

In the following experiments, we consider sets of tasks where each relies on a similar mapping from features to rewards. For each task, a set of 80 vertical bars appear on the screen. At the beginning of the task, each bar appears uniformly gray and 500 pixels tall. On each trial participants were invited to click on of the bars, after which its height is revealed to be between 0 and 500 pixels according to an unknown reward function. In the RL and optimization tasks the participant's score after their most recent action is revealed. In the RL task scores increased by the height of the bar that was selected on the most recent trial. In the optimization task scores where determined by the height of the highest bar of all those chosen on previous trials. In the AL task participants are not scored on the initial task. Instead, they are given a second task during which they are asked to indicate the correct height of a subset of the bars whose heights were not revealed during the preceding trials. On each of these trials scores were increased by the maximum height (500) minus the mean absolute error of the participants judgment (0 to 500).

Experiment 1A

Are people rational with respect to their goal (goal specific strategies) or do they rely on a general heuristic (high mean high variance)?

Experiment 1B

Are people rational with respect to information they are given about the underlying reward function (GP strategies) or do they rely on model-free strategies (gradient descent)?

Experiment 2

Are people rational with respect to shared structure across different tasks? Do people act to improve higher-order knowledge across multiple tasks with different goals or act to maximize reward on only the current task?