Abstract

Introduction

Explore/Exploit

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.

MAB

Many tasks requiring this kind of trade-off can be characterized 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 (Thompson, 1933) and Monte Carlo tree search (Coulom, 2007). Rather than relying on simulations, myopic strategies define a value function for approximating long-term reward. Common myopic strategies include upper confidence bound (UCB) (Agrawal, 1995) and epsilon-greedy (Sutton & Barto, 1998) algorithms.

CMAB/Function Learning

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 (Langford & Zhang, 2008). 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 Carroll (1963) considers polynomials up to degree 6, and Koh & Meyer (1991) 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. Busemeyer et al. (2005) 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. McDaniel & Busemeyer (2005) extend their connectionist model by including a layer of hidden nodes, each corresponding to a parameterization of a particular functional family.

More recently, Lucas et al. (2015) 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 (Snoek et al., 2012) 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 $z = \frac{m(a|h_{1:t-1}) - m(a*|h_{1:t-1})}{\sqrt{v(a|h_{1:t-1})}}$.

Basic CMAB Strategies

(eps greedy, mean/var greedy, ucb/pmi/mi, entropy search)

Once an agent has a mapping between features and reward, the policy $\tilde{\pi}$ must be selected that chooses the action a_t given the history of observed actions, features, and rewards $h_{1:t-1}$. Given that this policy does not take into account all possible trajectories, it must include a value function that approximates the long-term reward of any given action.

Alternative Goals

In addition to CMABs, the explore/exploit tradeoff can also be observed in other common context-dependent problems. In active learning (AL) Bramley et al. (2016), participants assume some degree of control over the contexts that they observe, rather than passively observing a predetermined set of examples. Optimization problems Rachlin et al. (1981) require finding the set of features x that maximizes some reward $f(x)$; for example, finding the number of hours to dedicate to work and to leisure respectively that maximizes satisfaction.

While these tasks might all share the same mapping between action and reward, their goal-specific reward, and thus the function used to approximate long-term reward, are distinct. For any particular action reward pair (a_t, r_t), there exists the goal-specific action reward pair (a_t, r'_t). Since the goal in the CMAB task is to maximize cumulative reward over time, the goal-specific reward is the same as the reward, that is:

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

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:

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

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:

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

References

Agrawal, R. (1995). Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, *27*(4), 1054-1078. Retrieved from http://www.jstor.org/stable/1427934

Bramley, N., Gerstenberg, T., & Tenenbaum, J. B. (2016). *Natural science: Active learning in dynamic physical microworlds.* 38th Annual Meeting of the Cognitive Science Society.

Busemeyer, J. R., Byun, E., & McDaniel, M. A. (2005). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks..

Carroll, J. D. (1963). Functional learning: The learning of continuous functional mappings relating stimulus and response continua. *ETS Research Bulletin Series*, *1963*(2), i–144. Retrieved from http://dx.doi.org/10.1002/j.2333-8504.1963.tb00958.x doi: 10.1002/j.2333-8504.1963.tb00958.x

Coulom, R. (2007). Efficient selectivity and backup operators in monte-carlo tree search. In H. J. van den Herik, P. Ciancarini, & H. H. L. M. J. Donkers (Eds.), *Computers and games: 5th international conference, cg 2006, turin, italy, may 29-31, 2006. revised papers* (pp. 72–83). Berlin, Heidelberg: Springer Berlin Heidelberg.

Koh, K., & Meyer, D. (1991, 10). Function learning: Induction of continuous stimulus-response relations. , *17*, 811-36.

Langford, J., & Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in neural information processing systems 20* (pp. 817–824). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/3178-the-epoch-greedy-algorithm-for-multi-armed-bandits-with-side-information.pd

Lucas, C., L Griffiths, T., Williams, J., & Kalish, M. (2015, 03). A rational model of function learning. , *22*.

McDaniel, M. A., & Busemeyer, J. R. (2005). The conceptual basis of function learning and extrapolation: comparison of rule-based and associative-based models. *Psychonomic bulletin & review*, *12*, 24-42.

Rachlin, H., Battalio, R., Kagel, J., & Green, L. (1981). Maximization theory in behavioral psychology. *Behavioral and Brain Sciences*, *4*(3), 371388. doi: 10.1017/S0140525X00009407

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th international conference on neural information processing systems - volume 2* (pp. 2951–2959). USA: Curran Associates Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2999325.2999464

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (1st ed.). Cambridge, MA, USA: MIT Press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3/4), 285-294. Retrieved from http://www.jstor.org/stable/2332286