A way around the exploration-exploitation dilemma

Erik J Peterson (erik.exists@gmail.com)
Carnegie Mellon, Dept. of Psychology
5000 Forbes Ave.
Pittsburgh, PA 15213

Timothy Verstynen (timothyv@andrew.cmu.edu)
Carnegie Mellon, Dept. of Psychology
5000 Forbes Ave.
Pittsburgh, PA 15213

Abstract

The exploration-exploitation dilemma is considered a fundamental but intractable problem in the learning and decision sciences. Here we challenge the basic formulation of the dilemma by defining independent mathematical objectives for exploration and exploitation, rather than having them share a single objective. This dual formulation greedy algorithm that provably and optimally maximizes both information and reward value.

Keywords: reinforcement learning; exploration; theory

Introduction

The exploration-exploitation dilemma is a fundamental but intractable problem in the learning and decision sciences (Gupta, Smith, & Shalley, 2006; Berger-Tal, Nathan, Meron, & Saltz, 2014; Sutton & Barto, 2018; Thrun, 1992; Dayan & Sejnowski, 1996; Findling, Skvortsova, Dromnelle, Palminteri, & Wyart, 2018; Gershman, 2018). The problem, as typically posed, is to model the actions taken by an agent (e.g., rodent, human, computer algorithm) as samples from a selection policy based on an set of values (Sutton, n.d.; Roughgarden, 2019). Exploitation then means choosing the most valuable action while exploration is simply making any other choice. When only one action can be taken at a time, there must be a trade-off between exploitation and exploration. This trade-off is not, however, what makes the problem a dilemma.

The dilemma needs two more conditions. The first is shared objective. For example, during reinforcement learning exploration and exploitation both attempt to maximize reward (e.g., food, water, drugs) (Sutton & Barto, 2018). The second is partial observability. For example, if a mouse is placed in maze with four arms it might know that down one arm is a small pellet of food, but then have no knowledge about the other arms. In this condition, choosing to explore might lead to more chow, or to less. This limited knowledge makes the dilemma. It also makes a direct mathematically solution to the problem intractable (Thrun, 1992; Dayan & Sejnowski, 1996; Findling et al., 2018; Gershman, 2018).

But is the dilemma really fundamental?

Results

In the natural world, exploration during behavior can have two different explanations. If there is no reason to expect a reward, exploration is theoretically modelled as a search for novel or maximum information (Mehlhorn et al., 2015; Pathak, Agrawal, Efros, & Darrell, 2017; Wang & Hayden, 2019; Calhoun et al., 2015; Schmidhuber, 2008; Schwartenbeck et al., 2019). For example, when a mouse is placed in a maze it has never visited before it will explore extensively even if no tangible rewards are present (Mehlhorn et al., 2015; Wang & Hayden, 2019). If however reward is expected, exploration gets re-interpreted and given a theoretical account as a reinforcement learning problem (Sutton & Barto, 2018). This interpretation is what can lead to a dilemma. An theoretical open problem is unifying both these explanations into one parsimonious account.

Here we conjecture *all* exploration only needs a single explanation, based on maximum information.

Our contribution is threefold. We first set out a series of axioms to serve as principled basis for information value. Next we prove the computer science method of dynamic programming (Sutton & Barto, 2018) provides an optimal way to maximize information value. Finally, we prove the optimal trade-off in terms of both information and reward value relies on a simple greedy scheduling algorithm (Roughgarden, 2019).

Information is not a reward

In principle there is no need for exploration and exploitation to share a common objective–exploration happens without reward to motivate it (Mehlhorn et al., 2015; Gupta et al., 2006; Berger-Tal et al., 2014; Schwartenbeck et al., 2019; Pathak et al., 2017). If we give exploration its own objective (as we do below) then the dilemma (which depends on having a common objective) fades, giving way to two independent problems: exploration (recast as a search for maximum information and exploitation (reward maximization). A ideal solution to these problems would optimally balance the trade-off between exploration and exploitation.

In practice reward and information are fundamentally distinct ideas, in two important ways. First, rewards are a conserved resource. Information is not. For example, if a mouse shares a potato chip with a cage-mate, she must break the chip up leaving it less food for herself. Second, reward value is constant during learning while the value of information must decline[1]. For example, if our mouse knows where a bag

[1] Assuming a stable environment

potato chips is it will generally keep going back and eating more chips. Whereas if a student knows the capital of the United States, there is no value to the student in being taught the capital of the United States is Washington DC.

A definition of information value

To make our eventual optimal algorithm general, we first needed to separate reward value from information value in a principled but general way. We do this axiomatically.

To form a basis for our axioms, we reasoned that the value of any observation s made by an agent (e.g., rodent, human, computer algorithm) depends entirely on what the agent learns. In other words, how it changes the agent's memory. A memory in our definition consists of only a (invertible) encoder, a decoder, and the mathematical idea of a set. That any encoded observation can be inverted is the same as requiring that observation that can be remembered can be also forgotten.

We let a memory M be finite set whose maximum size is N–defined over a finite state space $s \in S^N$–whose elements are added by an invertible encoder f, such that $M_{t+1} = f(M_t, s)$ and $M_t = f^{-1}(M_{t+1}, s)$ and whose elements z from M are recovered by a decoder g, such that $z = g(M, s)$. for simplicity we an initial memory M_0 is defined as the empty set, $M_0 = \emptyset$.

Axiomatic information value (E)

Axiom 1 (Axiom of Now). *E depends only on difference dM between M_{t+1} and M_t.*

That is, the value of an observation s depends only on how the current memory changes.

Axiom 2 (Axiom of Novelty). *$E = 0$ if and only if $dM = 0$.*

An observation that doesn't change the memory has no value.

Axiom 3 (Axiom of Scholarship). *$E \geq 0$.*

All new information is, in principle, valuable even the consequences are later found to be negative to the agent.

Axiom 4 (Axiom of Specificity). *E is monotonic with the compactness C of dM (Eq. 6) .*

More specific information is more valuable than less specific information. (We use the mathematical idea of compactness to formalize specificity.)

Axiom 5 (Axiom of Equilibrium). $\frac{dM^2}{d\theta^s} < 0$

We require the environment is learnable by M (in a probably approximately correct sense (Valiant, 1984)). We formalize this by introducing the idea that some parameters θ can make up M, and require their gradient makes continual progress toward equilibrium, $dM \to 0$ and $E \to 0$.

Our axioms *do not* depend on information theory nor on Bayesian reasoning, the common approaches to valuing information (Friston et al., 2016; Haarnoja et al., 2018; Itti & Baldi, 2009; Ly, Marsman, Verhagen, Grasman, & Wagenmakers, 2017). A Bayesian or probabilistic assumption would be a formidable constraint if it turns out that animals are not in fact general Bayesian reasoning systems.

Exploration as a dynamic programming problem

An ideal solution for learning to maximize information value, E, would be a dynamic programming solution. Such a would guarantee that total value (Eq. 2) is maximized by a simple, deterministic and locally greedy, algorithm.

In Theorem 1 we prove our definition of memory (above) has one critical property, optimal substructure, that is needed for a dynamic programming solution (Roughgarden, 2019). The other two, $E \geq 0$ and the Markov property (Sutton & Barto, 2018), are fulfilled by the Axioms 3 and 1 respectively.

To write down the Bellman solution for E as a dynamic programming problem we need some new notation. Let a be an action drawn from a finite action space A^K. Let time t denote an index $t = (1, 2, 3, \ldots, \infty)$. Let π denote any policy function that maps a state s to an action a, $\pi : s \to a$, such that $\forall s_t \in S, \forall a_t \in A$. Let δ be a transition function which maps (s_t, a_t) to a new state s_{t+1}, $\delta : (s_t, a_t) \to s_{t+1}$ where, once again, $\forall s_t \in S, \forall a_t \in A$.

For simplicity of notation we let E be redefined as $F(M_t, a_t)$, a "payoff function" in dynamic programming (Eq 1).

$$\begin{aligned} F(M_t, a_t) &= E(M_{t+1}, M_t) \\ &\text{subject to the constraints} \\ a_t &= \pi(s_t) \\ s_{t+1} &= \delta(s_t, a_t), \\ M_{t+1} &= f(M_t, s_t) \end{aligned} \tag{1}$$

The value function for E is Eq 2.

$$V_{\pi_E}(M_0) = \left[\max_{a \in A} \sum_{t=0}^{\infty} F(M_t, a_t) \;\middle|\; \pi_E,\, M \right] \tag{2}$$

To find the recursive Bellman solution we decompose Eq. 2 into an initial value F_0 and the remaining series in the summation. When this decomposition is applied recursively we eventually find an iterative optimal and greedy solution. The steps for our decomposition in terms of F and M are shown in Eq 4 and the result in Eq. 3.

$$\begin{aligned} V^*_{\pi_E}(M_0) &= \max_{a \in A} \Big[\sum_{t=0}^{\infty} F(M_t, a_t) \Big] \\ &= \max_{a \in A} \Big[F(M_0, a_0) + \sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1}) \Big] \\ &= F(M_0, a_0) + \max_{a \in A} \Big[\sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1}) \Big] \\ &= F(M_0, a_0) + V^*_{\pi_E}(M_{t+1}) + V^*_{\pi_E}(M_{t+2}),\ \ldots \end{aligned} \tag{3}$$

$$V^*_{\pi_E}(M_t) = F(M_t, a_t) + \max_{a \in A} \Big[F(M_{t+1}, a_t) \Big] \tag{4}$$

A note on exploration quality

Having a policy π_E^* that maximizes E also does not necessarily ensure that exploration is of good quality. Theorems 2 and 3 prove π_E^* also leads to a complete and exhaustive exploration of any *finite* space S.

Scheduling a way around the dilemma

To solve the dual problems that we have posed, i.e., information and reward maximization, we need an algorithm that can maximize the total value of both objectives. A simple solution to this exists in the computer science sub-field of optimal scheduling (Roughgarden, 2019).

Because there exists a range of optimal and useful reinforcement learning algorithms algorithm that operate on a discrete Markov decision spaces (Sutton & Barto, 2018), in principle, any of these are compatible with our approach. As a result we use π_R to denote any suitable reinforcement learning algorithm.

Agent behavior is generally viewed as a fixed resource, since agents can only take one action at a time. With this in mind, we imagine the policies for exploration and exploitation acting as two possible "jobs" competing for control of this fixed behavioral resource. By definition each of these "jobs" naturally produces non-negative values that an optimal job scheduler could use: E for information or R for reward/reinforcement learning. Each of the "jobs" takes a constant amount of time and each policy only results in a single action $a \sim A$.

The solution to scheduling problems that produce non-negative value and have fixed run times is known to be a simple greedy algorithm (Roughgarden, 2019). We restate this solution as set of inequalities controlling the action policy, π_π (Eq. 5).

$$\pi_\pi = \begin{cases} \pi_E^* & : E_t - \varepsilon > R_t \\ \pi_R & : E_t - \varepsilon \leq R_t \end{cases}$$

$$\text{subject to the constraints} \qquad (5)$$

$$R \in \{0,1\}$$

$$p(R) < 1$$

The fine print The inequalities in Eq 5 ensure total value summed over E_t and R_t is maximized (Roughgarden, 2019). It does not ensure that each policy also maintains its own optimality. Specifically, the search Theorems 2 and 3. To do this we introduce two mild constraints the reward distribution (Eq 5) which we prove are sufficient in Theorem 4).

In stochastic environments learning in M may show small continual fluctuations, mirroring these stochastic environmental dynamics. These will of course get reflected in E fluctuations. To allow Eq. 5 to achieve a stable solution we introduce ε, a boredom threshold for exploration. When $E_t < \varepsilon$ we say the policy is "bored" with exploration. To be consistent with the value axioms, $\varepsilon > 0$ and $E + \varepsilon \geq 0$.

To ensure the default policy is reward maximization, Eq. 5 breaks ties between R_t and E_t in favor of π_R.

The initial value E_0 for π_E^* can be arbitrary with the limit $E_0 > 0$. In theory E_0 does not change π_E^*'s long term behavior, but different values will change the algorithms short-term dynamics and so might be quite important in practice.

By definition a pure greedy policy, like π_E^*, can't handle ties. There is simply no mathematical way to rank equal values. Theorems 3 and 2 ensure that any tie breaking strategy is valid. However like the choice of E_0, tie breaking can strongly effect the transient dynamics.

Algorithmic complexity

Computational complexity is an important concern for any learning algorithm (Valiant, 1984). The worst case run time for π_π is linear and additive in its policies. That is, if in isolation it takes T_E steps to earn $E_T = \sum_{T_E} E$, and T_R steps to earn $r_T = \sum_{T_R} R$, the worst case training time for π_π is $T_E + T_R$. This in only true though if neither policy can learn from the other's actions. There is however no reason each policy can't observe the transitions (s_t, a_t, R, s_{t+1}) caused by the other. If this is allowed, worst case training time improves to $\max(T_E, T_R)$.

Summary

The exploration-exploitation dilemma is only a dilemma if we assume both exploration and exploitation must have the same objective. By recognizing that exploration can be formalized, quite generally, on its own, we've found a way around the dilemma. Our provably optimal solution is combination of two greedy algorithms, borrowed from different computer science sub-fields.

Exploration in our definition will maximizes information value. This allows an animal to learn a general, reward-independent, memory of the world. General learning of this kind is critical for long-term planning (Sutton & Barto, 2018). An important "side effect" of learning a general memory is that reward or reinforcement learning performances must also improve, to optimality.

References

Berger-Tal, O., Nathan, J., Meron, E., & Saltz, D. (2014, April). The Exploration-Exploitation Dilemma: A Multidisciplinary Framework. *PLoS ONE*, *9*(4), e95693. doi: 10.1371/journal.pone.0095693

Calhoun, A. J., Tong, A., Pokala, N., Fitzpatrick, J. A., Sharpee, T. O., & Chalasani, S. H. (2015, April). Neural Mechanisms for Evaluating Environmental Variability in Caenorhabditis elegans. *Neuron*, *86*(2), 428-441. doi: 10.1016/j.neuron.2015.03.026

Dayan, P., & Sejnowski, T. J. (1996, October). Exploration bonuses and dual control. *Machine Learning*, *25*(1), 5-22. doi: 10.1007/BF00115298

Findling, C., Skvortsova, V., Dromnelle, R., Palminteri, S., & Wyart, V. (2018, October). *Computational noise in reward-guided learning drives behavioral variability in volatile environments* (Preprint). Neuroscience. doi: 10.1101/439885

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., ODoherty, J., & Pezzulo, G. (2016, September). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, *68*, 862-879. doi: 10.1016/j.neubiorev.2016.06.022

Gershman, S. J. (2018, April). Deconstructing the human algorithms for exploration. *Cognition*, *173*, 34-42. doi: 10.1016/j.cognition.2017.12.014

Gupta, A. A., Smith, K., & Shalley, C. (2006). The Interplay between Exploration and Exploitation. *The Academy of Management Journal*, *49*(4), 693-706.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... Levine, S. (2018, December). Soft Actor-Critic Algorithms and Applications. *arXiv:1812.05905 [cs, stat]*.

Itti, L., & Baldi, P. (2009, June). Bayesian surprise attracts human attention. *Vision Research*, *49*(10), 1295-1306. doi: 10.1016/j.visres.2008.09.007

Ly, A., Marsman, M., Verhagen, J., Grasman, R., & Wagenmakers, E.-J. (2017, May). A Tutorial on Fisher Information. *arXiv:1705.01064 [math, stat]*.

Mehlhorn, K., Newell, B. R., Todd, P. M., Lee, M. D., Morgan, K., Braithwaite, V. A., ... Gonzalez, C. (2015, July). Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision*, *2*(3), 191-215. doi: 10.1037/dec0000033

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017, July). Curiosity-Driven Exploration by Self-Supervised Prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (p. 488-489). Honolulu, HI, USA: IEEE. doi: 10.1109/CVPRW.2017.70

Roughgarden, T. (2019). *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming* (Vol. 1) (No. 3).

Schmidhuber, J. (2008, December). Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes. *arXiv:0812.4360 [cs]*.

Schwartenbeck, P., Passecker, J., Hauser, T. U., FitzGerald, T. H., Kronbichler, M., & Friston, K. J. (2019). Computational mechanisms of curiosity and goal-directed exploration. *eLife*(e41703), 45.

Sutton, R. S. (n.d.). IntegraBteadseAd rocnhiAtepct-purroexsimfoartiLnegaDrnyinnga,mPiclaPnrnoingrga, mandinRgeacting. , 9.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second edition ed.). Cambridge, Massachusetts: The MIT Press.

Thrun, S. B. (1992). Eficient Exploration In Reinforcement Learning. *NIPS*, 44.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11), 1134-1142.

Wang, M. Z., & Hayden, B. Y. (2019, August). Monkeys are curious about counterfactual outcomes. *Cognition*, *189*, 1-10. doi: 10.1016/j.cognition.2019.03.009

Mathematical Appendix.

Compactness

The compactness C of a hyper-cube has a simple formula, $C = \frac{P^2}{A}$ where P is the cube's perimeter and A is its area. Therefore as M_t moves to M_{t+1}, the we can measure the distances $\{d_i, d_{i+1}, d_{i+2}, \dots d_N\}$ for all $O \leq N$ observed states Z, such that $Z \subseteq S$ and treat them as if they formed a O-dimensional hyper-cube. In measuring this imagined cube we arrive at a geometric estimate for the compactness of changes to memory (Eq. 6).

$$C = \frac{\left(2 \sum_i^O d_i\right)^2}{\prod_i^O d_i} \tag{6}$$

Information value as a dynamic programming problem

To use theorems from dynamic programming (Roughgarden, 2019; Sutton & Barto, 2018) we must prove our memory M has optimal substructure. By optimal substructure we mean that M can be partitioned into to a small number collection or series, each of which is an optimal dynamic programming solution. In general by proving we can decompose some optimization problem into a set of smaller problems whose optimal solution is easy to find or prove, it is trivial to prove that we can also grow the series optimally, which, in turn, allows for direct proofs by induction.

Theorem 1 (Optimal substructure)**.** *Assuming transition function δ is deterministic, if $V^*_{\pi_E}$ is the optimal information value given by π_E, a memory M_{t+1} has optimal substructure if the the last observation s_t can be removed from M_t, by $M_{t+1} = f^{-1}(M_{t+1}, s_t)$ where the resulting value $V^*_{t-1} = V^*_t - F(M_t, a_t)$ is also optimal.*

Proof. Given a known optimal value V^* given by π_E we assume for the sake of contradiction there also exists an alternative policy $\hat{\pi}_E \neq \pi_E$ that gives a memory $\hat{M}_{t-1} \neq M_{t-1}$ and for which $\hat{V}^*_{t-1} > V^*_{t-1}$.

To recover the known optimal memory M_t we lift $\hat{M}_{t-1}$ to $M_t = f(\hat{M}_{t-1}, s_t)$. This implies $\hat{V}^* > V^*$ which in turn contradicts the purported original optimality of V^* and therefore $\hat{\pi}_E$. □

A greedy policy explores exhaustively

Our proofs for exploration breadth are really sorting problems. If every state must be visited (or revisited) until the state of the world is learned, then under a greedy policy every state's value must, at one time or another, be the maximum value.

Let Z be set of all visited states, where Z_0 is the empty set $\{\}$ and Z is built iteratively over a path P, such that $Z = \{s | s \in P \text{ and } s \notin Z\}$.

Sorting requires ranking, so we define an algebraic notion of inequality for any three numbers $a, b, c \in \mathbb{R}$ are defined in Eq. 7.

$$a \leq b \Leftrightarrow \exists\, c;\; b = a + c \tag{7}$$

$$a > b \Leftrightarrow (a \neq b) \wedge (b \leq a) \tag{8}$$

Theorem 2 (State search: completeness and uniqueness)**.** *A greedy policy π is the only deterministic policy which ensures all states in S are visited, such that $Z = S$.*

Proof. Let $\mathbf{E} = (E_1, E_2, ...)$ be ranked series of E values for all states S, such that $(E_1 \geq E_2, \geq ...)$. To swap any pair of values $(E_i \geq E_j)$ so $(E_i \leq E_j)$ by Eq. 7 $E_i - c = E_j$.

Therefore, again by Eq. 7, $\exists \int \delta E(s) \rightarrow -c$.

Recall: $\nabla \mathcal{L}_M < 0$

However if we wished to instead swap $(E_i \leq E_j)$ so $(E_i \geq E_j)$ by definition $\nexists c; E_i + c = E_j$, as $\nexists \int \delta \rightarrow c$.

To complete the proof, assume that some policy $\hat{\pi}_E \neq \pi^*_E$. By definition policy $\hat{\pi}_E$ can be any action but the maximum, leaving $k-1$ options. Eventually as $t \rightarrow T$ the only possible swap is between the max option and the kth, but as we have already proven this is impossible as long as $\nabla \mathcal{L}_M < 0$. Therefore, the policy $\hat{\pi}_E$ will leave at least 1 option unexplored and $S \neq Z$. □

Theorem 3 (State search: convergence)**.** *Assuming a deterministic transition function Λ, a greedy policy π_E will resample S to convergence as $t \rightarrow T$, $E_t \rightarrow 0$.*

Proof. Recall: $\nabla \mathcal{L}_M < 0$.

Each time π^*_E visits a state s, so $M \rightarrow M'$, $F(M', a_{t+1}) < F(M, a_t)$

In Theorem 2 we proved only a deterministic greedy policy will visit each state in S over T trials.

By induction, if π^*E will visit all $s \in S$ in T trials, it will revisit them in $2T$, therefore as $T \rightarrow \infty$, $E \rightarrow 0$. □

Optimality of π_π

Theorem 4 (Optimality of π_π)**.** *Assuming an infinite time horizon, if π_E is optimal and π_R is optimal, then π_π is also optimal in the same sense as π_E and π_R.*

Proof. The optimality of $|\pi_\pi$ can be seen by direct inspection. If $p(R = 1) < 1$ and we have an infinite horizon, the π_E will have a unbounded number of trials meaning the optimally of P^* holds. Likewise, $\sum E < \varepsilon$ as $T \rightarrow \infty$, ensuring pi_R will dominate π_π therefore π_R will asymptotically converge to optimal behavior. □

In proving the total optimality of π_π we limit the probability of a positive reward to less than one, denoted by $p(R_t = 1) < 1$. Without this constraint the reward policy π_R would always dominate π_π when rewards are certain. While this might be useful in some circumstances, from the point of view π_E it is extremely suboptimal as the model would never explore. Limiting $p(R_t = 1) < 1$ is reasonable constraint, as rewards in the real world are rarely certain. A more naturalistic but complex way to handle this edge case might be to introduce reward satiety, and have reward value decay with repeated exposure.