A way around the exploration-exploitation dilemma.

Erik J Peterson[a,1] **and Timothy D Verstynen**[a,b]

[a]Department of Psychology; [b]Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh PA

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 and define independent mathematical objectives for exploration and exploitation, rather than having them share the typical objective of maximizing reward. To create an independent value for exploration we derive a new general set of axioms that let us measure the value of any information. Using these axioms and our reformulation we show how a simple, deterministic, and greedy algorithm can optimally maximize exploration and exploitation.

The exploration-exploitation dilemma is a fundamental but intractable problem in the learning and decision sciences (1–7). For example, if a bee goes a familiar direction to gather nectar it is exploiting past knowledge. If it goes in an unknown direction, it is exploring. It can't know if this exploration will lead to more nectar or less and so faces an exploration-exploitation dilemma. This general dilemma is faced routinely by agents of all kinds, including by foraging bees, humans, and computer algorithms.

In formal version of the dilemma, the actions taken by an agent are based on a set of learned values (3, 8). Exploitation is defined as choosing the most valuable action. Exploration is defined as simply making any other choice. When only one action can be taken a time there must be a trade-off between exploitation and exploration. This basic trade-off however is not a dilemma.

To become a dilemma, and a fundamental problem, it must satisfy two more conditions. The first is a shared objective. For example, in reinforcement learning exploration and exploitation both attempt to maximizes rewards, like nectar (3, 9). The second is partial observability. For example, the bee in our example can't know the if exploration will lead to finding more or less nectar. Making choices about a common objective but with limited knowledge is what defines the dilemma. It also makes a direct mathematical solution to the problem intractable (4–7).

But is this dilemma really fundamental? Do exploration and exploitation need to share an objective?

In the natural world exploration can have two different explanations. If there is no reason to expect a reward, exploration is described theoretically as a search for novel or maximum information (10–15). For example, when mouse is placed in a maze it has never visited before it will explore extensively even if no tangible rewards are present (10). On the other hand, if reward is expected exploration gets re-interpreted as a reinforcement learning problem (3).

Here we conjecture that all exploration needs only a single explanation based on maximum information. This simple conjecture lets us define independent mathematical objectives for exploration and exploitation. Unlike the dilemma, solving these independent objectives is quite tractable.

Our contribution is threefold. We first offer five axioms that serve as a completely general basis to estimate the value of any information. Next we prove the computer science method of dynamic programming (3, 16) provides an optimal way to maximize this information value. Finally, we describe a simple greedy scheduling algorithm (8) that can maximize both information value and reward value.

Results

Information is not a reward. In principle there is no need for exploration and exploitation to share a common objective; we know that exploration happens without reward to motivate it (1, 2, 10, 11, 15).

In practice reward and information contain at least two fundamentally distinct ideas. 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 less food for herself. Second, reward value is constant during learning while the value of information must decline*. For example, if a mouse learns where a bag of potato chips is she will generally keep going back and eating more. Whereas if a student learns the capital of the United States, there is no value to the student in being told again that the capital of the United States is Washington DC.

A definition of information value. To make our eventual solution to the exploration-exploitation trade-off general, we separate reward value from information axiomatically.

To form a basis for our axioms, we reasoned that the value of any observation s made by an agent depends entirely on what the agent learns by making that observation. In other words, how it changes the agent's memory. A memory in our definition consists of only an invertible encoder, a decoder, and the mathematical idea of a set.

We let time t denote an index $t = (1, 2, 3, \ldots, \infty)$. We let M be a finite set, whose maximum size is N (denoted M^N). This memory M learns by making observations s from a finite state space $s \in S^N$. Learning in M happens by an invertible encoder f, such that $M_{t+1} = f(M_t, s)$ and $M_t = f^{-1}(M_{t+1}, s)$. Memories z are recovered from M by a decoder g, such that $z = g(M, s)$. For simplicity the initial memory M_0 is an empty set, $M_0 = \emptyset$.

This definition of memory is extremely general, covering all modes of working or episodic memory (17, 18), probabilistic memory (19–21), and even memories based on compression or

*Assuming a stable environment

The authors have no conflicts of interest to declare.

[1]To whom correspondence should be addressed. E-mail: Erik.Exists@gmail.com

latent-states (22, 23). Of course, we could have used state prediction (11, 15, 24), information theory (25, 26), or Bayesian learning (19–21) to estimate the value of information. Each of these though requires strong assumptions that may not hold up. For example if it turns out that animals are not in fact general Bayesian reasoning systems.

Our goal is to describe behavior across the natural world, as well as to model the behavior of artificial agents. This lead us to work at a high level of mathematical abstraction, to develop our general notion of memory, and to work from simple axioms. As a result of these choices, the theory we develop applies to *all* of these more standard approaches.

Axiomatic information value (E)

Axiom 1 (Axiom of Now). *E is a distance ΔM between M_{t+1} and M_t.*

That is, the value of an observation s depends only on how much s changes M.

Axiom 2 (Axiom of Novelty). *$E = 0$ if and only if $\Delta M = 0$.*

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

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

In principle all new information is valuable, even if the consequences later on are negative.

Axiom 4 (Axiom of Specificity). *E is monotonic with the compactness C of ΔM (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). $\frac{\Delta^2 M}{\Delta t^2} < 0$

The environment must be learnable by M (in the sense of Valiant's definition (27)).

Exploration as a dynamic programming problem. A useful solution to maximize information value would be a dynamic programming solution. Dynamic programming guarantees total cumulative value is maximized by a simple, deterministic, and greedy algorithm.

In Theorem 1 we prove our definition of memory has one critical property, optimal substructure, that is needed for a dynamic programming solution (8, 16). The other two properties, $E \geq 0$ and the Markov property (8, 16), 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 π denote any policy function that maps a state s to an action a, $\pi : s \to a$, such that $\forall s_t \in S, \forall a_t \in A$. We use $a_t \sim \pi(s)$ as a shorthand to denote an action a_t being drawn from this policy. 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 we redefine E as $F(M_t, a_t)$, a "payoff function" in dynamic programming (Eq 1).

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

The value function for F is Eq. 2. The Bellman solution to recursively learn this function is found in Eq. 3.

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

$$V^*_{\pi_E}(M_t) = F(M_t, a_t) + \max_{a \in A} \left[F(M_{t+1}, a_t) \right] \quad [3]$$

Eq. 3 implies the optimal action policy π^*_E for E (and F) is a simple greedy policy. This greedy policy ensures exploration of any finite space S is exhaustive (Theorems 2 and 3).

Axiom 5 requires that learning in M converge. Axiom 4 requires information value increases with surprise, re-scaled by specificity. When combined with a greedy action policy like π^E, these axioms naturally lead to active learning (15, 28, 29) and to adversarial curiosity (30).

Scheduling a way around the dilemma. A common objective of reinforcement learning is to maximize reward value $V_R(s)$ using an action policy π_R (Eq. 4).

$$V^{\pi_R}_R(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} R_{t+k+1} \middle| s = s_t \right] \quad [4]$$

Where $\mathbb{E}[.]$ denotes the expected value of a random variable given that the agent follows policy π_R.

To find an algorithm that maximizes both information and reward value we imagine the policies for exploration and exploitation acting as two possible "jobs" competing for control of a fixed behavioral resource. We know (by definition) each of these "jobs" produces non-negative values which an optimal job scheduler could use: E for information or R for reward/reinforcement learning. We also know (by definition) each of the "jobs" takes a constant amount of time and each policy can only take one action a time. These two properties are sufficient to find a solution.

The solution to scheduling problems that produce non-negative values and have fixed run times is known to be a simple greedy algorithm (8). We restate this solution as a set of inequalities controlling the action policy, π_π given some state s (Eq. 5).

$$\begin{aligned} \pi_\pi(s) &= \begin{cases} \pi^*_E(s) & : E_s - \epsilon > R_s \\ \pi_R(s) & : E_s - \epsilon \leq R_s \end{cases} \\ &\text{subject to the constraints} \\ R_s &\in \{0, 1\} \\ p(R_s) &< 1 \end{aligned} \quad [5]$$

We designed Eq. 5 to be myopic (31). E_s and R_s represent the reward of information value for *only* the last time-point, making them distinct from the cumulative value terms V_R and

V_E (Eq. 4 and 2). This choice simplifies both the analysis and implementation of the algorithm.

By definition instantaneous (*i.e.*, myopic) values must be monotonic with total cumulative value. The inequalities in Eq. 5 therefore ensure V_R and V_E are maximized (Theorem 4).

In stochastic environments, M may show small continual fluctuations and never fully reach equilibrium. To allow Eq. 5 to achieve a stable solution we introduce ϵ, a boredom threshold for exploration. When $E_t < \epsilon$ we say the policy is "bored" with exploration. To be consistent with the value axioms, $\epsilon > 0$ and $E + \epsilon \geq 0$.

The initial value E_0 for π^*_E can also 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 algorithm's short-term dynamics and so might be quite important in practice. By definition a pure greedy policy, like π^*_E, cannot 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 affect the transient dynamics.

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

Algorithmic complexity. Computational complexity is an important concern for any learning algorithm (27). 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 $r_T = \sum_{T_R} R$, then the worst case training time for π_π is $T_E + T_R$. This is 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 that both exploration and exploitation must have the same objective, typically assumed to be maximizing rewards. By fully recognizing that exploration can be formalized on its own, we have found a way around the dilemma.

Exploration in our solution only tries to maximize information value. This greedy policy allows an animal to learn a general and reward-independent memory of the world. General learning like this is critical for long-term planning (3). An important "side effect" of learning a general memory is that reward or reinforcement learning performances must also improve, to optimality.

1. Gupta AA, Smith K, Shalley C (2006) The Interplay between Exploration and Exploitation. *The Academy of Management Journal* 49(4):693–706.
2. Berger-Tal O, Nathan J, Meron E, Saltz D (2014) The Exploration-Exploitation Dilemma: A Multidisciplinary Framework. *PLoS ONE* 9(4):e95693.
3. Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction*, Adaptive Computation and Machine Learning Series. (The MIT Press, Cambridge, Massachusetts), Second edition edition.
4. Thrun SB (1992) Eficient Exploration In Reinforcement Learning. *NIPS* p. 44.
5. Dayan P, Sejnowski TJ (1996) Exploration bonuses and dual control. *Machine Learning* 25(1):5–22.
6. Findling C, Skvortsova V, Dromnelle R, Palminteri S, Wyart V (2018) Computational noise in reward-guided learning drives behavioral variability in volatile environments, (Neuroscience), Preprint.
7. Gershman SJ (2018) Deconstructing the human algorithms for exploration. *Cognition* 173:34–42.
8. Roughgarden T (2019) *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming*. Vol. 1.
9. Woodgate JL, Makinson JC, Lim KS, Reynolds AM, Chittka L (2017) Continuous Radar Tracking Illustrates the Development of Multi-destination Routes of Bumblebees. *Scientific Reports* 7(1).
10. Mehlhorn K, et al. (2015) Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision* 2(3):191–215.
11. Pathak D, Agrawal P, Efros AA, Darrell T (2017) Curiosity-Driven Exploration by Self-Supervised Prediction in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (IEEE, Honolulu, HI, USA), pp. 488–489.
12. Wang MZ, Hayden BY (2019) Monkeys are curious about counterfactual outcomes. *Cognition* 189:1–10.
13. Calhoun AJ, et al. (2015) Neural Mechanisms for Evaluating Environmental Variability in Caenorhabditis elegans. *Neuron* 86(2):428–441.
14. Schmidhuber J (2008) 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]*.
15. Schwartenbeck P, et al. (2019) Computational mechanisms of curiosity and goal-directed exploration. *eLife* (e41703):45.
16. Bellmann R (1954) The theory of dynamic programming. *Bull. Amer. Math. Soc* 60(6):503–515.
17. Miller G (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review* 63:81–97.
18. Tulving E (2002) Episodic Memory: From Mind to Brain. *Annual Review of Psychology* 53(1):1–25.
19. Park IM, Pillow JW (2017) Bayesian Efficient Coding, (Neuroscience), Preprint.
20. Itti L, Baldi P (2009) Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.
21. Friston K, et al. (2016) Active inference and learning. *Neuroscience & Biobehavioral Reviews* 68:862–879.
22. Kingma DP, Welling M (2013) Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*.
23. Ganguli S, Sompolinsky H (2010) Short-term memory in neuronal networks through dynamical compressed sensing. p. 9.
24. Schmidhuber (1991) A possibility for implementing curiosity and boredom in model-building neural controllers. *Proc. of the international conference on simulation of adaptive behavior: From animals to animats* pp. 222–227.
25. Polani D, Martinetz T, Kim J (2001) An Information-Theoretic Approach for the Quantification of Relevance in *Advances in Artificial Life*, eds. Goos G, Hartmanis J, van Leeuwen J, Kelemen J, Sosík P. (Springer Berlin Heidelberg, Berlin, Heidelberg) Vol. 2159, pp. 704–713.
26. Kolchinsky A, Wolpert DH (2018) Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus* 8(6):20180041.
27. Valiant L (1984) A theory of the learnable. *Communications of the ACM* 27(11):1134–1142.
28. Shyam P, Jaśkowski W, Gomez F (2018) Model-Based Active Exploration. *arXiv:1810.12162 [cs, math, stat]*.
29. Pathak D, Gandhi D, Gupta A (2019) Self-Supervised Exploration via Disagreement. *Proceedings of the 36th International Conference on Machine Learning* p. 10.
30. Schmidhuber J (2019) Unsupervised Minimax: Adversarial Curiosity, Generative Adversarial Networks, and Predictability Minimization. *arXiv:1906.04493 [cs]*.
31. Hocker D, Park IM (2019) Myopic control of neural dynamics. *bioRxiv*.

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).

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

Information value as a dynamic programming problem. To use theorems from dynamic programming (3, 8) we must prove our memory M has optimal substructure. By optimal substructure we mean that M can be partitioned into 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 $\hat{\pi}_E$. □

Bellman solution. 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 (Eq 3) we arrive at an iterative optimal and greedy solution (Eq. 7).

$$\begin{aligned} V^*_{\pi_E}(M_0) &= \max_{a \in A}\left[\sum_{t=0}^{\infty} F(M_t, a_t)\right] \\ &= \max_{a \in A}\left[F(M_0, a_0) + \sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1})\right] \\ &= F(M_0, a_0) + \max_{a \in A}\left[\sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1})\right] \\ &= F(M_0, a_0) + V^*_{\pi_E}(M_{t+1}) + V^*_{\pi_E}(M_{t+2}),\ \dots \end{aligned} \quad [7]$$

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

Let Z be the 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. 8.

$$a \leq b \Leftrightarrow \exists\ c;\ b = a + c \quad [8]$$

$$a > b \Leftrightarrow (a \neq b) \wedge (b \leq a) \quad [9]$$

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. 8 $E_i - c = E_j$.

Therefore, again by Eq. 8, $\exists \int \delta E(s) \rightarrow -c$.

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 $\pi^* E$ 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 π_π can be seen by direct inspection. If $p(R = 1) < 1$ and we have an infinite horizon, then π_E will have a unbounded number of trials meaning the optimally of P^* holds. Likewise, $\sum E < \epsilon$ 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.