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 propose a new account of the exploration-exploitation dilemma, by defining independent mathematical objectives for exploration and exploitation. The standard account has them share an objective, typically maximizing reward. Using an axiomatic approach we derive a new general metric for information value. Using this value and our reformulation of the problem we prove a simple, deterministic, and greedy algorithm can optimally maximize exploration and exploitation. The cost of this solution is a increase in the worst-case algorithmic run-time, that depends strictly on the size of the environment to be learned.

In an informal sense, information is clearly valuable. From the printing press and the internet to the very existence of language humans are fundamentally curious (?), consistently invent new ways to find and share information.

Information seeking is not a uniquely human trait. Most animals appear to be intrinsically curious about their environments (?), prone to exploration both when no rewards are present or expected (?), and in some cases even when exploration explicitly costs reward (1). Animals with more complex central nervous systems build mental maps of their environment and its structure, that are learned through exploration (?).

These examples make clear that learning for its own sake is a crucial function for survival. Despite this, in the learning and decision sciences there is no *general* metric for the value of information. (see however, (2? , 3)). Instead, information is modeled as an analog of tangible, external rewards (a so-called *intrinsic reward*). In this case, information value is typically framed as a means for maximizing the amount of tangible rewards accrued over time (? ?). This approximation of information value, however, means that the problem of optimal exploration becomes mathematically intractable (? ? ? ?). Here we show how defining information seeking as as its own objective–even when the overall goal is to maximize reward–offers a way around the dilemma.

To separate the value of information from rewards completely we look to the field of information theory. Shannon developed the information theory without any sense of what information “means”. He focused on the statistical problem of transmitting symbols. As a result, the theory does need to know what the symbols refer to which makes it extremely general (?).

Like Shannon we take an axiomatic approach: *listing key properties (axioms) that any measure of information value should have, and arrive a metric that satisfies these.* Also like Shannon we focus on symbols (or the more general idea of sense observations) and not what they refer to. Unlike Shannon, we don’t focus on transmitting symbols, but on remembering them. A learning and memory view of value. This view leads to a surprising result. Information value can be formally disconnected from probability theory, and therefore from information theory.

Our contribution is threefold. We first offer five axioms that serve as a general basis to estimate the value of any observation, given a memory. Next we prove the computer science method of dynamic programming (4, 5) provides an optimal way to maximize this kind of information value. Finally, we describe a simple greedy scheduling algorithm (6) that can maximize both our notion for information value and reward value.

Results

Information is not a reward. In theory there is no need for exploration and exploitation to share a common objective; exploration happens without explicit rewards (3, 7–10). Here we take this to an extreme. We conjecture that information is always valuable for its own sake, and that an optimal model for maximizing the value of information should make no reference to reward learning. They are separate problems.

To make the difference between our approach and the standard one clear, let’s imagine a honeybee foraging in meadow. Let’s alo imagine the bee has been here before, but only long enough to find a single nectar-filled flower. Now it could exploit the flower, or it can explore the rest of the meadow. In Figure fig:f1**A** we depict the choice as a traditional dilemma. In Figure fig:f1**B** we show our view of the problem. It’s only goal during exploration is to learn about the meadow, *in general.* Not to seek reward directly.

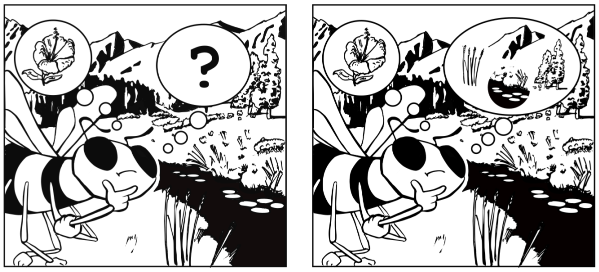

Fig. 1. Two views of exploration and exploitation. **A**. The classic exploration-exploitation dilemma depicted as a bee trying to maximize nectar (reward). **B.** Our approach, based on choice between maximizing reward *or* information. The bee in this panel is faced with a choice between reward (the nectar in the flower) and the information which would be gained by exploring the unfamiliar parts of the meadow. Notice how exploration here is based not on finding rewards (flowers), but on learning about the new environment *in general.*

The authors have no conflicts of interest to declare.

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

Our conjecture is backed up by conceptual analysis: Rewards and information are fundamentally distinct concepts. 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. 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 is very general, consisting of only an invertible encoder, a decoder, and the mathematical idea of a set.

For formalize memory we begin with some notation. We let time t denote an index $t = (1, 2, 3, \ldots)$ which tracks the order of observations $(s_t, s_{t+1}, \ldots)$. Observations s are taken from a finite state space $s \in S$, whose size is N (denoted S^N). We define a memory M as a finite set, whose maximum size is also N (M^N). Adding s to M happens by an invertible encoder f, $M_{t+1} = f(M_t, s)$ and $M_t = f^{-1}(M_{t+1}, s)$. Memories z about s are recovered from M by a decoder g, such that $z = g(M, s)$.

This definition for memory covers all modes of working or episodic memory (11, 12), probabilistic memory (13–15), and memories based on information compression or latent-states (16, 17).

Axiomatic Information Value (E)

Why axioms? Past attempts at valuing information have been based on information theory and often Bayesian probability (2, 9, 10, 13–15, 18, 19). The problem with this all these accounts require an animal do information theoretic calculations, which are complex. If an animal can't do complex information theory calculations, it can't value information. Likewise, if it turns out animals are not in fact general probabilistic or Bayesian reasoning systems, these assumptions become problematic.

Below we develop a simple set of axioms to value information. These require only a very general notion of memory (define above), along with the requirement value behave loosely like the mathematical notion of distance.

Why these axioms? Any set of axioms can be replaced arbitrarily with some other set. We don't argue ours are uniquely correct, only that they are as minimal as we can imagine while producing a theory which adds new insight into a classic problems (the dilemma), is consistent with existing Bayesian theory (with additional assumptions, that is), and is performant in practice.

Axiom 1 (Axiom of Memory). *$E(s)$ depends only on how the memory M changes when making an observation s (which we denote ΔM)*

Axiom 2 (Axiom of Novelty). *An observation s that doesn't change the memory M has no value. $E(s) = 0$ if and only if $\Delta M = 0$.*

Axiom 3 (Axiom of Scholarship). *All learning in M about s should be considered valuable. $E(s) \geq 0$.*

This should hold true even if the consequences of that learning later on turn out to be negative. Our theory of value is concerned only with the act of learning.

Axiom 4 (Axiom of Specificity). *More specific an observation s changes a memory M the more valuable that information is. We can use the mathematical idea of compactness to formalize specificity. $E(s)$ is monotonic with the compactness C of ΔM (Eq. 7).*

Axiom 5 (Axiom of Equilibrium). *An observation s must be learnable by M. $\frac{\Delta^2 M}{\Delta s^2} \leq 0$.*

By learnable we mean PAC learnable, in the sense of Valiant's definition (20).

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

In Theorem 1 we prove our definition of memory has one critical property, optimal substructure, which is needed for a dynamic programming solution (4, 6). The other two properties, $E \geq 0$ and the Markov property (4, 6), 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).

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

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

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

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

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 (9, 21, 22) and to adversarial curiosity (23). Axiom 5 also implies that observations which lead to a simpler (i.e. compressed) understanding of the environment, generate value consistent with Schmidhuber's theory (24)

[*] Assuming a stable environment

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).

$$V_R^{\pi_R}(s) = \mathbb{E}\Big[\sum_{k=0}^{\infty} R_{t+k+1}\Big| s = s_t\Big] \qquad [4]$$

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 (6). We restate this solution as a set of inequalities controlling the action policy, π_π given some state s (Eq. 5).

$$\pi_\pi(s) = \begin{cases} \pi_E^*(s) & : E_s - \eta > R_s \\ \pi_R(s) & : E_s - \eta \leq R_s \end{cases}$$

$$\text{subject to the constraints} \qquad [5]$$

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

$$p(R_s) < 1$$

We designed Eq. 5 to be myopic (25). 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 η, a boredom threshold for exploration. When $E_t < \eta$ we say the policy is "bored" with exploration. To be consistent with the value axioms, $\eta > 0$ and $E + \eta \geq 0$.

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.

Exploration without regret. The essential aspects of the exploration-exploitation dilemma are embodied by the classic multi-armed bandit task. Like the slot machines which inspired them, when selected each bandit returns a reward, according to a predetermined probability. But the agent can only chose one bandit at a time, and so must decide whether to explore and explore at each trial.

Table 1. Artificial agents.

Agent	Abbrv.	Description
Dual value	η	Our algorithm (Eq 5.)
E-greedy	ϵ	
Annealed e-greedy	Annealed-ϵ	Identical to E-greedy, but ϵ is exponential decayed at
Infomax	β	$R + \beta I$
Infomax (deterministic)	β (det)	A deterministic greedy version of Infomax
Random	σ	Each action is random (a control)

We study four prototypical bandits. The first (denoted as bandit **I**) has a single winning arm ($p(R) = 0.8$, Figure 2**A**) and is a simple challenge. We'd expect any learning agent to be able to consistently solve this task by learning the high value or "winning" arm by the end of any reasonable learning period. The second bandit (**II**) has two winning arms. One of these (arm 7, $p(R) = 0.8$) though has a higher payout than the other (arm 3, $p(R) = 0.6$) which can act as a "distraction", luring an agent with insufficient exploration into settling on a suboptimal choice. The third (**III**) has a single winning arm but the overall probability of receiving any reward is very low ($p(R) = 0.02$ for the winning arm, $p(R) = 0.01$ for all others). Sparse rewards problems are difficult for most current reinforcement learning algorithms, and is a common feature of both the real world, and artificial environments like go, chess, and class Atari video games (**?**). The fourth bandit (**IV**) has 121, and a complex randomly generated reward struture. Bandits of this type and size are probably at the limit of human performance (**?**).

The simple structure of bandits tasks require only a simple memory system–discrete and probabilistic, which we depict in Figure 2**B-C**. To estimate the value of some observation s we compared sequential changes in the probabilistic memory, M_{t+1} and M_t using relative entropy (i.e. the KL distance; Figure 2**B-G**). The KL distance is a standard way to measure the distance between two distributions and is, by design, consistent with the axioms.

We compared the performance of 6 artificial agents (Table. 1) All agents used the same temporal difference learning algorithm (TD(0); (5)); see *Supplementary Methods*). The only difference between the agents was their exploration mechanism. Ours is denoted η.

The E-greedy algorithm is a simple classic exploration mechanism (5**?**). It's annealed variant (see Table) is common in state-of-the-art reinforcement learning papers, like Mnih *et al* ((26)). Infomax algorithms are Bayesian exploration methods which, like our model, use active sampling and treat information as intrinsically value. In Infomax though information is treated as an intrinsic reward, and goal of all exploration is to maximize total reward (extrinsic and intrinsic) ().

In reinforcement learning a regret term G can measure the cost of exploration ((**?**); Eq. 6), where V^* represents the value of pure exploitation and V_x represents how the value changes using some exploratory action $A_x \subset A$.

$$G = V^* - V_x \qquad [6]$$

An optimal exploration should have zero total regret, while accumulating the maximum possible number of rewards. When comparing across classic and state-of-the-art methods for exploration, only our algorithm consistently meets both these criterion.

Algorithmic complexity. Computational complexity is an important concern for any learning algorithm (20). 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

We propose a fundamentally new theory of the exploration-exploitation dilemma, framed as a balance of *both* reward and information. Using an axiomatic approach we derive a new general metric of information value. Using this metric, agents can explore maximizing value coming from information and rewards. This approach allows us to discover an optimal (i.e. regret-free) policy to solve the exploration-exploitation dilemma. Given our solution is deterministic it also offers a new empirical route to exactly predict moment-by-moment exploratory behavior in range of both new, and existing, data sets.

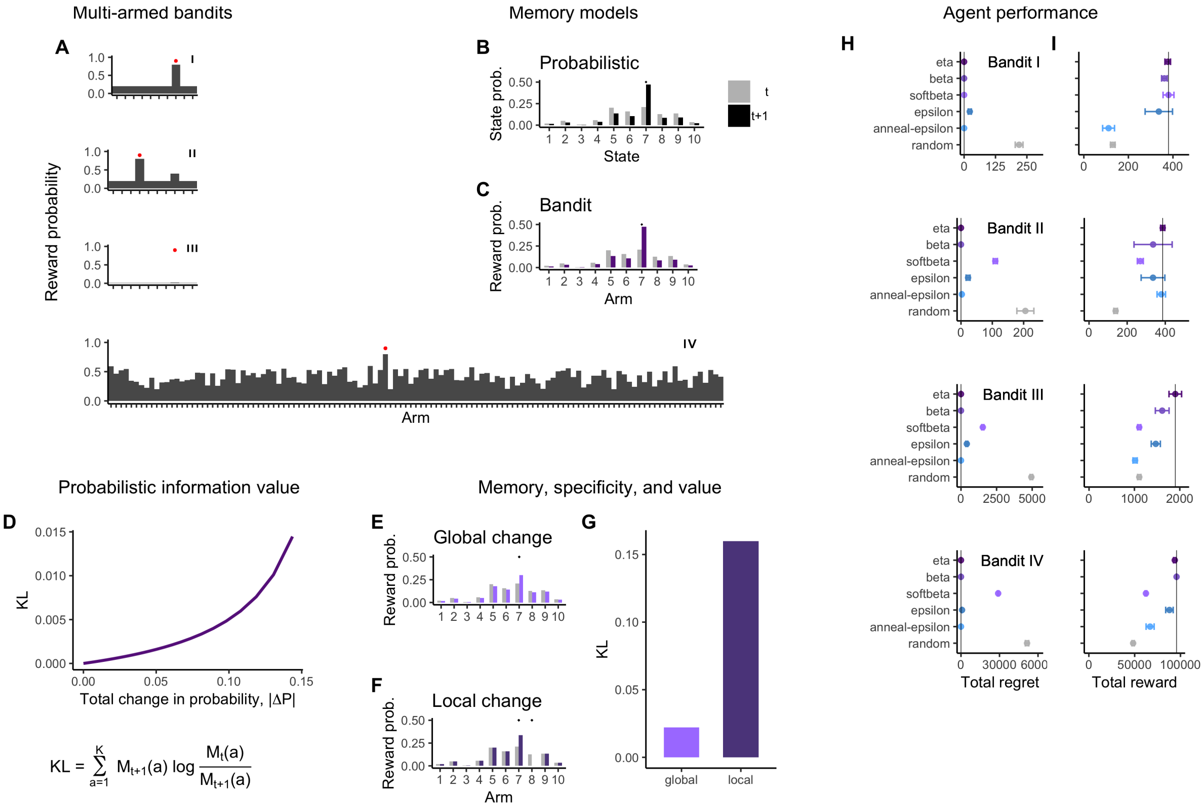

Fig. 2. Bandits, memory, and value. **A**. Bandit tasks I-IV. (See main text for a complete description). **B**. A discrete, state-based, probabilistic memory. **C**. The discrete memory adapted for the multi-armed bandit task. **D**. Information value (defined here by the KL divergence) increases with total probability flux $|\Delta P|$. **E-G**. Specificity and value. Panels E and F have the same flux in probability and the same increase in $p(R)$ for arm 7. In **E**, the decrease in probability needed to counter the increase is uniformly distributed between the remaining arms (i.e. global renormalization of the distribution). In **F**, the decrease in probability is comes predominantly and specifically from a decrease in probability at arm 8. In panel **G** we show the KL how differs in the local and global case, consistent with Axiom 4. Final performance of several agents on the four prototypical bandit tasks (see panel A, and main text). **H** Median total regret. **I** Median total reward. Error bars in all plots represent median absolute deviation. $N = 100$ experiments.

1. Wang MZ, Hayden BY (2019) Monkeys are curious about counterfactual outcomes. *Cognition* 189:1–10.
2. 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.
3. Berger-Tal O, Nathan J, Meron E, Saltz D (2014) The Exploration-Exploitation Dilemma: A Multidisciplinary Framework. *PLoS ONE* 9(4):e95693.
4. Bellmann R (1954) The theory of dynamic programming. *Bull. Amer. Math. Soc* 60(6):503–515.
5. Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction*, Adaptive Computation and Machine Learning Series. (The MIT Press, Cambridge, Massachusetts), Second edition edition.
6. Roughgarden T (2019) *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming.* Vol. 1.
7. Mehlhorn K, et al. (2015) Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision* 2(3):191–215.
8. Gupta AA, Smith K, Shalley C (2006) The Interplay between Exploration and Exploitation. *The Academy of Management Journal* 49(4):693–706.
9. Schwartenbeck P, et al. (2019) Computational mechanisms of curiosity and goal-directed exploration. *eLife* (e41703):45.
10. 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.
11. 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.
12. Tulving E (2002) Episodic Memory: From Mind to Brain. *Annual Review of Psychology* 53(1):1–25.
13. Park IM, Pillow JW (2017) Bayesian Efficient Coding, (Neuroscience), Preprint.
14. Itti L, Baldi P (2009) Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.
15. Friston K, et al. (2016) Active inference and learning. *Neuroscience & Biobehavioral Reviews* 68:862–879.
16. Kingma DP, Welling M (2013) Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*.
17. Ganguli S, Sompolinsky H (2010) Short-term memory in neuronal networks through dynamical compressed sensing. p. 9.
18. 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.
19. Kolchinsky A, Wolpert DH (2018) Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus* 8(6):20180041.
20. Valiant L (1984) A theory of the learnable. *Communications of the ACM* 27(11):1134–1142.
21. Shyam P, Jaśkowski W, Gomez F (2018) Model-Based Active Exploration. *arXiv:1810.12162 [cs, math, stat]*.
22. Pathak D, Gandhi D, Gupta A (2019) Self-Supervised Exploration via Disagreement. *Proceedings of the 36th International Conference on Machine Learning* p. 10.
23. Schmidhuber J (2019) Unsupervised Minimax: Adversarial Curiosity, Generative Adversarial Networks, and Predictability Minimization. *arXiv:1906.04493 [cs]*.
24. Schmidhuber J (2010) Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2(3):230–247.
25. Hocker D, Park IM (2019) Myopic control of neural dynamics. *bioRxiv*.
26. Mnih V, et al. (year?) Playing Atari with Deep Reinforcement Learning. p. 9.

Mathematical Appendix.

A. A dual value problem.

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. 7).

$$C = \frac{\left(2\sum_i^O d_i\right)^2}{\prod_i^O d_i} \qquad [7]$$

Information value as a dynamic programming problem. To use theorems from dynamic programming (5, 6) 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. 8).

$$\begin{aligned} V^*_{\pi_E}(M_0) &= \max_{a \in A}\left[\sum_{t=0}^{\infty} F(M_t, a_t)\right] \\ &= \max_{a \in A}\left[F(M_0, a_0) + \sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1})\right] \\ &= F(M_0, a_0) + \max_{a \in A}\left[\sum_{t=1}^{\infty} F(M_{t+1}, a_{t+1})\right] \\ &= F(M_0, a_0) + V^*_{\pi_E}(M_{t+1}) + V^*_{\pi_E}(M_{t+2}),\ \dots \end{aligned} \qquad [8]$$

««< HEAD

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. ======= »»»> overleaf-2019-07-26-1525

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. 9.

$$a \leq b \Leftrightarrow \exists\ c;\ \ b = a + c \qquad [9]$$

$$a > b \Leftrightarrow (a \neq b) \wedge (b \leq a) \qquad [10]$$

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. 9 $E_i - c = E_j$.

Therefore, again by Eq. 9, $\exists \int \delta E(s) \rightarrow -c$.

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, $\sum E < \eta$ 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.