AI-Assisted Scientific Data Collection with Iterative Human Feedback

Travis Mandel, James Boyd, Sebastian J. Carter, Randall H. Tanaka, Taishi Nammoto
University of Hawai'i at Hilo, Hilo, HI
{tmandel, boyd4, sjc7, dh404, nammoto}@hawaii.edu

Abstract

Although artificial intelligence has revolutionized data analysis, significantly less work has focused on using AI to improve scientific data collection. Past work in AI for data collection has typically assumed the objective function is well-defined by humans before starting an experiment; however, this is a poor fit for scientific domains where new discoveries and insights are made as data is being collected. In this paper we present a new framework to allow AI systems to work together with humans (e.g. scientists) to collect data more effectively in simple scientific domains. We present a novel algorithm, TESA, which seeks to achieve good performance by learning from past human behavior how to direct data to places that are likely to become scientifically interesting in the future. We analyze the problem theoretically, defining a novel notion of regret in this setting and showing that TESA is zero regret. Next, we show that TESA outperforms other related algorithms in simulations using real data drawn from three diverse domains (economics, mental health, and cognitive psychology). Finally, we run experiments with human subjects across these scientific domains to compare our iterative human-in-the-loop process to a (more standard) workflow in which information is communicated to the AI a priori.

Introduction

Scientists in domains such as psychology, behavioral economics, and ecology spend a large amount of time and effort designing experiments and collecting large datasets. Although artificial intelligence (AI) has dramatically impacted the process of scientific data analysis (Ardila et al. 2019; Weinstein et al. 2019), it has not yet had a similar impact on scientific *data collection*. Indeed, in many scientific domains the typical process for collecting data is simply to sample the space uniformly as possible; for instance, by allocating an equal number of participants to each condition. While this approach makes intuitive sense, the truth is that not all data are created equal - some datapoints are inherently more scientifically interesting. For example, in general reaction time tends to increase with the number of stimuli (Hick 1952), but say a psychologist is running an experiment varying the number of stimuli (N) between 1 and 20. After collecting some initial data, they find that in general reaction time does increase as expected, except for N=10, where reaction time is very low. In this case, they would want to confirm the anomaly by gathering more data at N=10. So a datapoint from a trial with N=10 would be much more scientifically interesting than one with N=5 or N=15. We wish to direct more data towards the interesting conditions, but this is difficult to do manually, especially if data is collected at a rapid pace. Therefore, this is an ideal fit for AI systems, which can process large amounts of data in real-time.

Past work in using AI to guide data collection, such as active learning (Settles 2009), reinforcement learning (Sutton and Barto 1998), and online learning (Erraqabi et al. 2017) assume a known and fully-defined (estimation) objective, which is extremely difficult to define in settings where the goal is to uncover new and unexpected phenomena. Previous work focusing on using AI to achieve scientific objectives (Liu 2015) requires humans to instead specify expected experimental results; however, specifying such a prediction accurately for a novel experiment is difficult.

Specifying precise scientific objectives a priori is particularly difficult because the most impactful scientific discoveries are often unrelated to the main purpose of the experiment. For example, Alexander Fleming was running experiments to study Staphylococcus and noticed that one of his samples was unexpectedly dying off due to an unknown contaminant - that contaminant turned out to be penicillin. Such completely unexpected discoveries cannot be captured by pre-programmed objectives, but instead must come from keeping humans "in the loop" of scientific data collection, leveraging their background knowledge, experience, and intuition to help the AI system collect the most useful data.

In this paper, we investigate a new paradigm for scientific data collection that allows humans (e.g. scientists) to periodically provide feedback to the AI system to help it more efficiently collect data[1] . Specifically, as an AI collects data, humans will iteratively add points (called "keypoints") indicating areas which appear to be scientifically interesting based on the data collected so far. This complicates the role of the AI system, which must tradeoff sampling existing keypoints more to increase confidence in their values, while

[1]We consider both cases where an AI system collects data directly (e.g. an online experiment) and cases where it collects data indirectly (e.g. by directing scientific divers).

sampling elsewhere to find new keypoints. To address these challenges, we develop a new algorithm, Threshold Estimating Sampling Algorithm (TESA), which we show (asymptotically) achieves zero regret in this setting. We examine the empirical performance of TESA, finding that it is superior to previously-proposed approaches in simulations based on real-world datasets, as well as in a human subject experiment.

Related Work

Optimal Experimental Design (OED) (Pukelsheim 1993) and active machine learning (Settles 2009) both seek to optimally direct data collection, but the objective of these frameworks is to most quickly learn an accurate model. This requires the prespecified model parameters to perfectly capture what a scientist finds interesting, which seems unlikely when seeking novel and unexpected discoveries.

Value of Information (VoI) (Howard 1966), and more generally, reinforcement learning (Sutton and Barto 1998) allow users to collect data in order to optimize arbitrary objective functions. However, these areas require users to prespecify an accurate objective function, which is difficult (Thomaz and Breazeal 2006), especially given the fact that scientists tend to make unexpected discoveries when analyzing data.

Given that objective functions are difficult to prespecify, much interest in AI has focused on automatically learning these objective functions (Noothigattu et al. 2018; Hadfield-Menell et al. 2016). Common approaches involve learning objective functions from positive or negative feedback on behaviors (Griffith et al. 2013), trajectory preferences (Akrour, Schoenauer, and Sebag 2011; Christiano et al. 2017), or near-optimal demonstrations (Zhifei and Meng Joo 2012). Unfortunately, when the goal is to uncover new and unexpected scientific insights in a novel experiment, these methods of feedback are not feasible.

The most closely related work is Liu (2015). Liu proposes Uncertainty Weighted Posterior Sampling (UWPS) to direct scientific data collection towards regions that are scientifically interesting. Liu requires humans to specify beforehand what the expected results for the experiment will be, and then data is directed towards points that are maximally different from this expectation. Although reasonable, specifying accurate predictions can be extremely difficult and time-consuming in real-world settings. Additionally, a difference from a prior prediction does not capture the fact that interesting points may change as scientists learn more about the domain over the course of the experiment (and some of these changes may be more intersting than others). This motivates our human-in-the-loop framework, where humans provide iterative feedback to AI systems over the course of the experiment. We compare to UWPS and Liu's framework in our experiments.

Problem Setup

We consider problems in which a human $\mathcal{H}$ and an AI $\mathcal{A}$ work together to gather the most scientifically useful data from an environment $\mathcal{E}$ (which maps x values to (noisy) y values). At each time step $t \geq 0$, $\mathcal{A}$ selects an x value $j_t \in \mathcal{X}$, where $\mathcal{X}$ is represented as the set of integer values $\{1, \ldots, N\}$, receives a sampled y value $y_t \in [0, 1]$ from $\mathcal{E}$, and adds it to a set of samples S_{j_t}. $\mathcal{H}$ is asked to provide feedback every $\mathcal{I}$ time steps. Specifically, when $t \mod \mathcal{I} = 0$, $\mathcal{A}$ will generate a visualization tuple $(f_h, f_l, \hat{f})$, where $f_h : \mathcal{X} \to [0, 1]$ is a higher bound function, $f_l : \mathcal{X} \to [0, 1]$ is a lower bound function and $\hat{f} : \mathcal{X} \to [0, 1]$ is an estimated function, which are estimated using the sample sets S_i (for all $i \in \mathcal{X}$).[2] After observing the visualized tuple $(f_h, f_l, \hat{f})$, $\mathcal{H}$ sends $\mathcal{A}$ a set of keypoints $\hat{K}_t \subseteq \mathcal{X}$ (these are intended to represent scientifically interesting x-values of the curve, and the set will ideally grow as more information is uncovered about the true curve).

Let K be the true set of keypoints, which is unknown to both $\mathcal{H}$ and $\mathcal{A}$ initially. We make the following assumption:

Assumption 1. *There exists some unknown n_i such that if $|S_{i,t}| \geq n_i$ then $i \in \hat{K}_t$ if and only if $i \in K$.*

Note that Assumption 1 is not particularly strong, as it only requires $\mathcal{H}$ to make correct decisions in a long-term sense (i.e. if $|S_{i,t}| \geq n_i$). With smaller numbers of samples, $\mathcal{H}$ could make mistakes, for instance marking something a keypoint when it is not or deleting a true keypoint.

The goal of $\mathcal{A}$ is to optimize the following score (zeta):

$$\zeta\left(K, \mathbf{S_t}\right) = \frac{|K|}{\sum_{i \in K} C\left(|S_{i,t}|\right)} - \sum_{i \notin K} |S_{i,t}| \qquad (1)$$

where $\mathbf{S_t}$ is the vector of sample set sizes for all x values at timestep t: $\langle |S_{1,t}|, \ldots, |S_{N,t}| \rangle$, and

$$C\left(x\right) = \sqrt{\frac{ln\left(\frac{2}{\delta}\right)}{2x + \epsilon}} \qquad (2)$$

represents the confidence interval, $\epsilon > 0$ represents a (generally small) smoothing term, and $1 - \delta$ represents the confidence level given $\delta \in (0, 1]$. This is similar to the Chernoff-Hoeffding bounds, except for the addition of ϵ in the denominator to prevent divide-by-zero issues.[3]

The motivation for equation (1) is that we want more identified keypoints (hence the $|K|$ numerator), but we also want to be certain of the value of each keypoint (hence the sum of keypoint confidence interval sizes in the denominator). Also, we want to minimize the number of samples devoted to places that are not scientifically interesting, hence the term which subtracts the number of non-keypoint samples.

Defining Regret

Since our problem involves running AI systems online in an unknown environment, theoretical guarantees are particularly important. **Here we provide proof sketches for these results. The full proofs can be found in the appendix.**[4]

[2] In our experiments $\hat{f}(i) = \frac{\sum_{y \in S_i} y}{|S_i|}$, $f_l(i) = \hat{f}(i) - C(|S_i|)$, and $f_h(i) = \hat{f}(i) + C(|S_i|)$, where C is as defined in equation (2).

[3] We let $\delta = 0.1$ and $\epsilon = 0.02$ in our experiments. Also, if $K = \emptyset$, we simply let $\zeta\left(K, \mathbf{S_t}\right) = 0$ to avoid divide-by-zero issues.

[4] The appendix can be found at https://datadrivengame.science/aaai21/.

Properties of the Score Function ζ

Here we show some properties of the score function ζ that will be useful for later results. First, we define a function d:

$$d\left(\mathbf{S_t}\right) = \sum_{i \in K} C\left(|S_{i,t}|\right) \tag{3}$$

In other words, d is the denominator of the first term of ζ.

Lemma 1. *d is monotonically decreasing with t, and $\zeta(K, \mathbf{S_t})$ is monotonically increasing on t such that $j_t \in K$.*

Proof Sketch This follows directly from analyzing the partial derivative of d.

Lemma 2. *$C(x)$ is monotonically decreasing and convex for all $x \geq 0$.*

Proof Sketch This follows directly from analyzing the first and second derivatives of C.

In the next lemma we show that there are diminishing returns. That is, if we choose a keypoint, the denominator shrinks less if there are more samples:

Lemma 3. *A sample $w \in [0, 1]$, was collected at timestep t_w at x-value $i_w \in K$, and similarly, a sample $q \in [0, 1]$, was collected at timestep t_q at x-value $i_q \in K$. Then for any two (possibly equal) algorithms A and B it must be that, if $|S_{t_w,i_w,A}| \leq |S_{t_q,i_q,B}|$, then*

$$d(\mathbf{S_{t_w,A}}) - d(\mathbf{S_{t_w-1,A}}) \leq d(\mathbf{S_{t_q,B}}) - d(\mathbf{S_{t_q-1,B}})$$

Proof Sketch From Lemma 2 we know C is monotonically decreasing and convex, which implies C shrinks less for i_q than for i_w (since i_q has more samples). Since d is the same at t_w except for the change to i_w's confidence intervals (and same for i_q), d likewise shrinks less for i_q than for i_w.

Comparing Regret Definitions

Analyzing performance based on ζ itself is challenging, since the maximum achievable score depends on $\mathcal{E}$ and $\mathcal{H}$. Therefore, we analyze regret compared to an optimal algorithm. In this setting the optimal algorithm is not immediately obvious, so we introduce Theorem 1.

Theorem 1. *Let A be an algorithm which selects the x value* $\text{argmin}_{i \in K} |S_i|$. *$A$ is optimal according to the ζ score.*

Proof Sketch For a contradiction, assume there is some algorithm B and timestep T such that $\zeta\left(K, S_{B,T}\right) > \zeta\left(K, S_{A,T}\right)$. Now, B must have more samples than A on certain x-values (those samples are denoted by the set η) and less samples on other x-values (the extra samples collected by A on those x-values are denoted by the set Ω). Since both algorithms have the same number of total samples, one cannot add a sample to one location without removing a sample from another location, and so $|\eta| = |\Omega|$. This allows us to define a bijection $m : \Omega \rightarrow \eta$. Next we argue that every sample in Ω (from A) reduces d more than (or equal to) the corresponding sample in η (from B). If the sample in η is not a keypoint, then d will not decrease for B but will for A (since A always samples keypoints), so this is trivial. If on the other hand, the sample in η is a keypoint, then the corresponding sample in Ω must have been a keypoint with fewer than (or equal) samples, since A samples keypoints as evenly as possible. So by Lemma 3, d must shrink less in this case as well. Since d shrinks less (or equal) for B than for each corresponding sample collected by A, and A never samples any non-keypoints, ζ must be less (or equal) for B compared to A.

Given Theorem 1, we wish to define regret of some algorithm D relative to A. Perhaps the most natural definition of regret would be $\zeta\left(K, \mathbf{S_{T,A}}\right) - \zeta\left(K, \mathbf{S_{T,D}}\right)$. However, since the ζ metric incorporates information from all timesteps (and not just the current one), this does not quantify how much regret we incur on each timestep t. This can be achieved by computing the change in score from $t-1$ to t for both algorithms:

$$\begin{aligned} r'_D\left(t\right) = &\left[\zeta\left(K, \mathbf{S_{t,A}}\right) - \zeta\left(K, \mathbf{S_{t-1,A}}\right)\right] \\ &- \left[\zeta\left(K, \mathbf{S_{t,D}}\right) - \zeta\left(K, \mathbf{S_{t-1,D}}\right)\right] \end{aligned} \tag{4}$$

Note that, since our problem setup allows some quantities to be chosen adversarially (for instance, the addition and removal of keypoints), equation (4) measures regret with respect to the best possible policy in the face of these adversarial decisions, in other words, it is very similar to policy regret (Arora, Dekel, and Tewari 2012). An alternative is to use a notion closer the external regret (Auer et al. 2002):

$$\begin{aligned} r_D\left(t\right) = &\left[\zeta\left(K, \mathbf{S_{t-1,D}} \cup A\left(\mathbf{S_{t-1,D}}\right)\right) - \zeta\left(K, \mathbf{S_{t-1,D}}\right)\right] \\ &- \left[\zeta\left(K, \mathbf{S_{t,D}}\right) - \zeta\left(K, \mathbf{S_{t-1,D}}\right)\right] \end{aligned} \tag{5}$$

$$r_D\left(t\right) = \left[\zeta\left(K, \mathbf{S_{t-1,D}} \cup A\left(\mathbf{S_{t-1,D}}\right)\right) - \zeta\left(K, \mathbf{S_{t,D}}\right)\right] \tag{6}$$

where $A\left(\mathbf{S_{t-1,D}}\right)$ denotes the sample algorithm A would have received if it was initialized at $\mathbf{S_{t-1,D}}$.[5]

In other settings, policy regret (i.e. Equation (4)) is thought to be strictly stronger than external regret (i.e. Equation (6)) (Arora, Dekel, and Tewari 2012). Therefore, in most adversarial settings, regret guarantees with respect to the policy regret would be preferred to guarantees related to the external regret. But in this situation, policy regret is actually weaker in some sense than external regret. This unusual property arises from the fact that, as long as an algorithm (asymptotically) samples only keypoints, the increase in the ζ score will go to zero due to the fact that the confidence intervals shrink towards zero. So an algorithm could conceivably be zero-regret (in a policy regret sense) if it simply found (and endlessly sampled) one keypoint while ignoring all others. In contrast, this algorithm would not be zero-regret under an external regret definition (as we show formally in Theorem 3 and Corollary 2) since A would choose one of the low-sampled keypoints, and thus it would always be able to increase ζ by much more than the single-keypoint algorithm. Therefore, we adopt the external regret definition of Equation (6) in the rest of this paper.

We define a zero-regret algorithm as follows:

*An algorithm B is **zero-regret** if and only if* $\lim_{T \rightarrow \infty} E\left[r_B\left(T\right)\right] = 0$

[5] ζ does not depend on y-values, so one can look at the x-value A would select given $\mathbf{S_{t-1,D}}$ and increment the size of that set.

This problem setup is unusual: The typical bandit-style assumption is that the algorithm observes the reward/loss it receives after making its decision, and simply needs to keep that reward high enough relative to the optimal reward. In our case, however, the algorithm does not know the true keypoints, and yet regret is evaluated with respect to the true set of keypoints K. Therefore the true reward for each action is unobserved, making the problem more difficult.

Properties of Zero-Regret Algorithms

To further illuminate this problem, we lay out some necessary properties of zero-regret algorithms. First, in Theorem 2 we show that an algorithm is zero-regret only if the proportion of samples assigned to non-keypoints converges to zero.

Theorem 2. *If $|K| > 0$, $\lim_{T\to\infty} E\left[r_B\left(T\right)\right] = 0$ only if:*

$$\lim_{T\to\infty} E\left[\frac{\sum_{i\notin K}|S_{i,T,B}|}{\sum_{i\in\mathcal{X}}|S_{i,T,B}|}\right] = 0$$

Proof Sketch Since A selects only keypoints, ζ monotonically increases under A (Lemma 1). However, when a non-keypoint is sampled, ζ decreases by 1. Therefore the regret on these timesteps is at least 1, so the fraction of time this occurs must go to zero in order to be zero-regret.

Corollary 1. *If $|K| > 0$, an algorithm which samples keypoints with probability $(1-\epsilon_g)$ and non-keypoints (if there exist any) with probability ϵ_g cannot be zero regret.*

Proof. This follows immediately from Theorem 2, as if we assign probability ϵ_g to non-keypoints, $\lim_{T\to\infty} E\left[\frac{\sum_{i\notin K}|S_{i,T,B}|}{\sum_{i\in\mathcal{X}}|S_{i,T,B}|}\right] = \epsilon_g$. □

Theorem 3. *An algorithm B is zero-regret only if, for any number ℓ, for all $i \in K$, $lim_{T\to\infty} P(|S_{i,T,B}| \geq \ell) = 1$.*

Proof Sketch If there is some constant probability that some set of x-values $K_n \subseteq K$ are all sampled less than ℓ times, then in those cases there must be some corresponding set of x-values K_p that are sampled an infinite number of times. If K_p contains non-keypoints, Theorem 2 applies, so $K_p \subset K$. Now, for each arm in K_p, the confidence function C shrinks to zero as we sample it more, so the increase in ζ on those timesteps also shrinks towards zero. However, for sufficiently large T, the optimal algorithm A would have preferred to sample keypoints in K_n since they have fewer samples. And since we do not sample K_n more than ℓ times in this case, the increase in ζ on these timesteps is constant. On an infinite number of timesteps, the x-value chosen by A would result in a constant positive increase in ζ, but the x-value chosen by B leads to an increase in ζ that vanishes towards zero. So, B cannot be zero-regret.

Corollary 2. *An algorithm which samples round-robin until the first keypoint is found, then samples that keypoint forever (perhaps reverting to round-robin if that keypoint is deleted), is not zero regret if $|K| > 1$.*

Proof. This follows directly from Theorem 3: if we have $\mathcal{H}$ such that the first true keypoint i_c is revealed after ℓ_c samples, and not removed thereafter, and for all $j \in K$ such that $j \neq i_c$, more than $\ell_c + 1$ samples are required to identify it as a keypoint, then
$P(|S_{j,t}| > \ell_c + 1) = 0$ for $j \in K, j \neq i_c$ and all t. □

A Zero-Regret Bayesian Algorithm

In developing an AI algorithm to solve this problem, we face an explore-exploit dilemma: Should we exploit the existing keypoints by sampling them more, or should we explore to find new keypoints? Since the set of x-values are discrete, at first this seems like a good fit for a multi-armed bandit (MAB) (Robbins 1952) algorithm such as epsilon-greedy or Thompson sampling (Thompson 1933). However, the decision to place a keypoint at a given x-value is unlikely to be a purely stochastic choice at each timestep, making these methods a poor fit. An alternative is to try to predict where scientists will place keypoints, but this is an extremely challenging problem, as the space of phenomena scientists could find interesting is vast and scientist interaction data is sparse.

We develop Threshold Estimating Sampling Algorithm (TESA), which takes a different approach. Instead of predicting *where* key points will be placed, it predicts the upper limit of *how much data* the human needs to determine whether a point is scientifically interesting, specifically $n_{max} = \max_{i\in K} n_i$. Then TESA is able to exploit this information to determine how to sample, since it is pointless to sample non-keypoints with more than n_{max} samples.

To estimate n_{max}, TESA takes a Bayesian approach. It models the unknown n_i for each x-value as being drawn from a $Uniform(0,\theta)$ distribution, with θ corresponding to the unknown n_{max}. The conjugate prior to a $Uniform(0,\theta)$ is a $Pareto$ distribution. Given a prior shape hyperparamter $a > 0$ and a prior scale hyperparameter $b_0 > 0$, after observing a set of keypoints $\hat{K}$ the posterior distribution is Pareto $\left(\max\left(b_0, \max_{i\in\hat{K}} n_i\right), a + |\hat{K}|\right)$. Since TESA does not know the true n_i, it substitutes b_i, which is the number of samples that belonged to x-value i at the time it was (most recently) made a keypoint. One complication in calculating b_i is that we do not ask the user for keypoints every timestep, rather we ask the human to update these periodically (every $\mathcal{I}$ steps). Since if a keypoint is acquired on time t, the true number of samples to identify it could be anywhere between $|S_{i,t}|$ and $|S_{i,t-\mathcal{I}}|$, therefore we use the midpoint of this quantity as our estimate. Similar to other Bayesian exploration approaches (Thompson sampling, UWPS, etc.), TESA draws a sample from the Pareto posterior distribution at each step to produce a threshold h which estimates n_{max}.

Given the threshold h, TESA will not select any non-keypoints with more than h samples. However, it still needs to determine how to tradeoff sampling keypoints versus sampling non-keypoints with fewer than h samples. Empirically, it is better to primarily exploit existing keypoints, but exploration is also important to uncover new keypoints. To achieve good performance early on, TESA defines an ϵ_p such that we select the least sampled non-keypoint with probability ϵ_p and select the least sampled keypoint otherwise. The pseudocode for TESA is shown in Algorithm 1.

Although some parts of TESA are similar to epsilon-

Algorithm 1 Threshold Estimating Sampling Algorithm (TESA)

Input: Tradeoff parameter $\epsilon_p \in [0, 1)$, prior scale $b_0 > 0$, prior shape $a > 0$
$\hat{K} \leftarrow \emptyset; \forall_{i \in \mathcal{X}} S_{i,0} \leftarrow \emptyset, S_{i,1} \leftarrow \emptyset$
for t = 1 to T **do**
  if $t \mod \mathcal{I} = 0$ **then**
    Send $(f_h, f_l, \hat{f})$ to human $\mathcal{H}$, get keypoints $\hat{K}_t$
    for all $i \in \hat{K}_t - \hat{K}$ **do**
      $b_i \leftarrow \frac{|S_{i,t}| + |S_{i,t-\mathcal{I}}|}{2}$
    $\hat{K} \leftarrow \hat{K}_t$
  $x_m \leftarrow \max\left(b_0, \max_{i \in \hat{K}} b_i\right)$
  $h \sim \text{Pareto}\left(x_m, a + |\hat{K}|\right)$
  $i_k \leftarrow \text{argmin}_{i \in \hat{K}} |S_{i,t}|$
  $i_h \leftarrow \text{argmin}_{i \in \mathcal{X}} |S_{i,t}|$
  $r \sim \text{Uniform}(0, 1)$
  if $|\hat{K}| > 0$ **and** $(r \geq \epsilon_p$ **or** $|S_{i_h,t}| \geq h)$ **then**
    $i_c \leftarrow i_k$
  else
    $i_c \leftarrow i_h$
  Choose x value i_c; Receive new sample y_t
  $S_{i_c,t+1} \leftarrow S_{i_c,t} \cup \{y_t\}$
  $\forall_{i \in \mathcal{X} \text{ s.t. } i \neq i_c} S_{i,t+1} \leftarrow S_{i,t}$

greedy, the Bayesian threshold-learning procedure[6] improves performance. Specifically, although epsilon-greedy is not zero-regret (Corollary 1), TESA is a zero-regret algorithm.

Theorem 4. *TESA (Algorithm 1) is zero regret.*

Proof Sketch First we show that TESA samples all x-values an infinite number of times. To show this we argue that if $X_n \subset \mathcal{X}$ is the set of x-values that TESA does not sample an infinite number of times, then there must be an only finite number of timesteps t where $|X_n \cap \hat{K}_t| > 0$, since TESA samples keypoints with probability at least $(1 - \epsilon_p)$. If on the other hand $|X_n \cap \hat{K}_t| = 0$ but $|X_n| > 0$, then every time TESA samples a non-keypoint it will prefer to sample $i_h \in X_n$, since the number of samples on x-values in $\mathcal{X} - X_n$ go to infinity. The probability TESA samples a non-keypoint is at least $\epsilon_p P(|S_{i_h,t}| < h)$. Since h is drawn from a Pareto distribution, the only way for this probability to shrink to zero (without sampling i_h infinitely often) is for the Pareto posterior parameters to update in such a way that the tail of the distribution becomes increasingly unlikely. Yet, since TESA updates the Pareto hyperparameters only when an x-value becomes a keypoint, there are only a finite number of x-values all with the bounded number of samples required to reveal whether or not they are keypoints (by Assumption 1). Therefore, the Pareto tail cannot shrink indefinitely, meaning TESA will select $j_t \in X_n$ infinitely often, contradicting the definition of X_n. Therefore, the only case that can occur is $|X_n| = 0$, meaning TESA samples all x-values an unbounded number of times. Since the number of samples on all x-values goes to infinity, $\hat{K}_t$ will eventually converge to K by Assumption 1. Further, since all non-keypoints are sampled infinitely often, $|S_{i_h,t}| \to \infty$ and thus $P(|S_{i_h,t}| < h) \to 0$ by the properties of the Pareto complementary CDF (recall h is drawn from a Pareto). So for TESA, $P(j_t \in K) \to 1$, and also $\hat{K}_t \to K$. On steps where $\hat{K}_t = K$ and $j_t \in K$, TESA selects an identical x-value to the optimal algorithm A, thus incurring zero regret. So the expected regret of TESA goes to 0 as $t \to \infty$.

[6]Note that TESA does not use the sampled y-value, only looking at the number of samples and where the keypoints are located. However, this approach has practical benefits, such as being agnostic to the criteria used to select keypoints, and also being somewhat robust to delayed y-values, which occur in many real-world scientific domains.

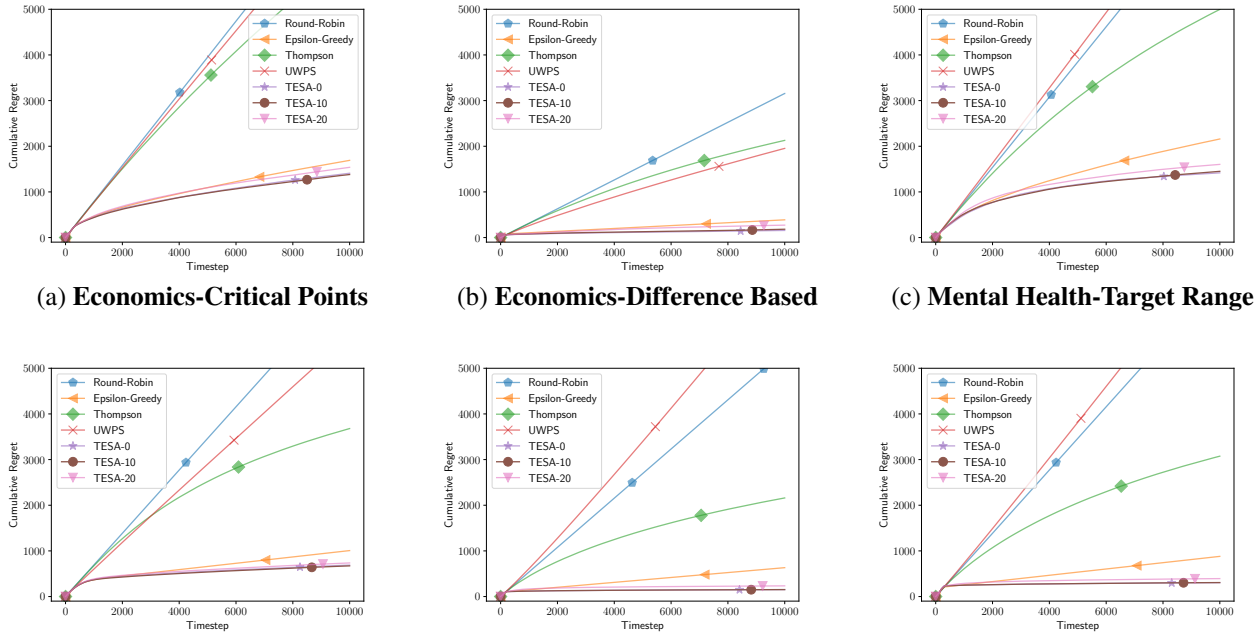
Simulated Experiments

Scientific Domains To evaluate empricial performance, we simulate data collection using public datasets from various domains. Specifically, we extract x and y values, and store the raw samples (y-values) for each x-value in sets. When the AI selects an x-value i, data is "replayed" by sampling a y-value (with replacement) from the set associated with i. This helps ensure that properties of the original data (variance, etc.) are preserved, while allowing us to simulate different AI algorithms.

Economics We replayed data collected by the Hass Avocado Board, relating price to weekly organic Avocado sales from 2015-2018 (Kiggins 2018). The independent variable was (discretized) price of organic avocados on a given week, while the dependent variable was demand (as measured by the number of avocados sold on that week). Although such an analysis is imperfect because it does not hold other factors constant (for instance, the avocado supply), it does give a sense of how well (or poorly) avocados historically sell at different price points, making for a (simulated) scientific experiment which has the potential to reveal interesting phenomena that differ from the commonly accepted view (that as price increases, demand decreases).

Mental Health We replayed data from the Open Sourcing Mental Illness 2016 Mental Health in Tech Survey (OSMI 2016). For our dependent variable we used responses of tech workers to the question of whether or not they had been diagnosed with a mood disorder (depression, bipolar disorder, etc.). For the independent variable, we examined (discretized) worker age, since a survey can be targeted towards different ages (e.g. student interns vs veteran employees). Each y-value represented one binary response of an individual that fell into the designated range. Although the highest rates of mood disorders are thought to occur among young adults (NIMH 2017), it is interesting to investigate whether this pattern holds across a population of tech workers.

Cognitive Psychology We replayed data from cognitive psychology experiments by Petzold et al. (2004). Participants were shown squares of various sizes and asked to make judgments about the square size. The reaction time of participants' responses forms the dependent variable in our analysis. For our independent variable, we use the (signed) difference between the size of the current square and the size



(a) **Economics-Critical Points** (b) **Economics-Difference Based** (c) **Mental Health-Target Range**

(d) **Mental Health-Difference Based** (e) **Cog. Psychology-Critical Points** **Cog. Psychology-Target Range**

Figure 1: Results comparing various AI data collection algorithms with various combinations of scientific domains (Economics, Mental Health, or Cognitive Psychology) and user simulators (Critical Points, Difference Based, and Target Range).

of the previous square, which the original experiment was designed to examine (Petzold and Haubensak 2001).

AI Algorithms

Round Robin This simply selects $j_t = (t \mod |\mathcal{X}|) + 1$ to ensure that all x-values are sampled evenly.

Epsilon-Greedy Keypoint Sampler This selects a (uniformly) random $j_t \in \hat{K}_t$ with probability $1 - \epsilon_g$ and a random $j_t \in \mathcal{X}$ otherwise. We used $\epsilon_g = 0.1$ in all of our experiments (this was not tuned).

Thompson Sampling This algorithm (Thompson 1933) performs well on bandit problems (Chapelle and Li 2011), and can outperform adversarial bandit algorithms such as EXP3 (Auer et al. 1995), even in certain adversarial settings (Lykouris, Mirrokni, and Leme 2020). Thompson sampling receivies immediate rewards in the range [0,1] and keeps track of a posterior distribution for each arm. When selecting an arm, it samples each posterior distribution and then selects the arm with the maximum sample. In our case, "arms" correspond to x-values, and for "rewards", we use the difference in ζ-score after processing that sample. Since these rewards are continuous, we use a variant with Normal posteriors (Agrawal and Goyal 2013).[7]

UWPS Uncertainty Weighted Posterior Sampling (Liu 2015) is an algorithm that asks $\mathcal{H}$ to provide a prior theory function τ which estimates the true response function, and uses that to determine what is scientifically interesting. Specifically, it maintains Bayesian posteriors over the value of each x-value i, draws a sample μ_i from each posterior at the current timestep, and and estimates $D_i = \tau(i) - \mu_i$. Then it calculates U_i as the variance of the posterior and picks the i that maximizes $D_i U_i$, thereby directing samples to points that are uncertain and different from the prediction. Similar to Thompson Sampling, UWPS uses Normal posteriors.

TESA-b_0 We used TESA with $\epsilon_p = 0.1$ (to match epsilon-greedy) and $a = 1$. We try various values of b_0, which equates to an initial guess of the threshold n_{max}.

Simulated Users To simulate specifying a prior function estimate (for UWPS) based on an incomplete understanding of the domain, 10 samples from the environment (at various x values) are interpolated. We then examine the following simulated keypoint selection methods:

Target Range This approach marks as keypoints x-values whose mean samples fall between a minimum y value y_{low} and a maximum y value y_{high}, and where difference between the high and low confidence intervals is less than o_{conf}. In our simulations we rescale the y-values to [0,1] and use $o_{conf} = 0.7$. For the cognitive psychology domain we set $y_{low} = 0.1$ and $y_{high} = 0.5$, and for the mental health domain we set $y_{low} = 0$ and $y_{high} = 0.33$.

Difference Based This method marks keypoints that are less than some threshold o_{dist} away from a prior function[8]. We set $o_{dist} = 100000$ in the Economics domain and $o_{dist} = 50$ in the Mental illness domain (due to the different

[7]To translate the rewards into [0,1] we use a reward lower bound of -1 and a reward upper bound of $\frac{|\mathcal{X}|}{C(1)} - \frac{|\mathcal{X}|}{C(0)}$.

[8]The prior used by the Difference Based user is identical to the one used by UWPS, making experiments with this simulated user essentially a "best case" case for UWPS since the keypoints are placed at the maximum distance from this prior function.

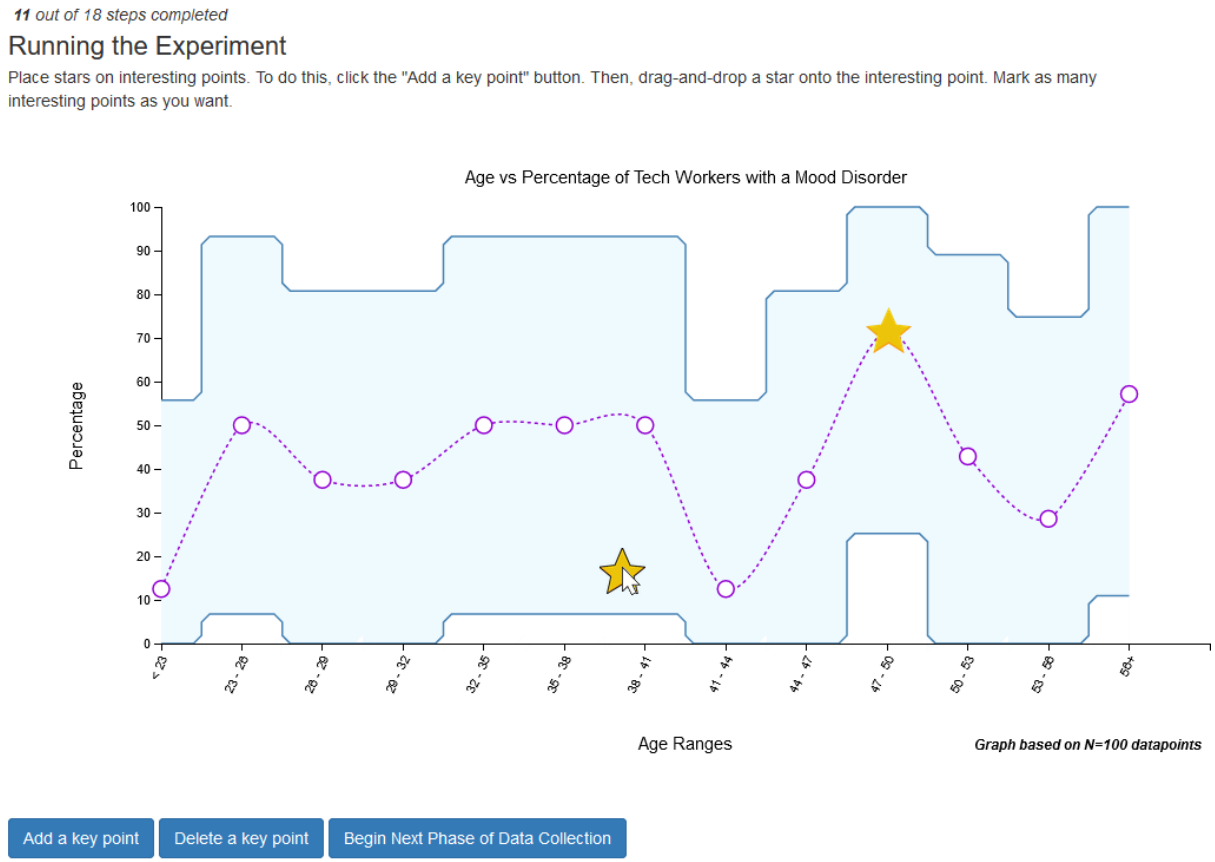


Figure 2: Our keypoint selection interface. We visualize the current estimated function (purple dotted line) and confidence region (blue area). Users can add and delete keypoints as well as drag them to interesting regions of the function.

y-axis scales).

Critical Points This approach looks for points that are either higher (by more than a threshold o_{diff}) than both neighboring points, or lower (by at least o_{diff}) than both neighboring points, and where the size of the confidence intervals of all three involved points are less than o_{conf}.We scaled the samples between [0,1] and set $o_{diff} = 0.05$ and $o_{conf} = 0.7$ in all domains.

Results

Our results (shown in Figure 1) are averaged over 500 runs, with a fixed seed.[9] Normal confidence intervals are not shown as they are too small; the maximum size of TESA's confidence intervals are roughly 3% of the graph height. TESA shows lower cumulative regret than other related algorithms across a variety of scientific domains and methods of placing keypoints. Using $b_0 = 10$ works well, but TESA is robust to variations of that parameter. In contrast, UWPS tends to perform relatively poorly, even in simulations with the Difference Based user. Part of the reason for this is that uncertainty term tends to dominate the calculation (a fact acknowledged by Liu (2015)), limiting its ability to direct samples effectively. Multi-armed bandit algorithms such as Thompson sampling also show mediocre performance, highlighting the differences between this domain and more standard explore-exploit problems. [10]

Human Subject Experiment

Experiment Setup We constructed a web-based task designed to compare TESA and our method of keypoint selection to previously proposed methods (prior function estimation and UWPS). Users are randomly assigned to one of our three replayed scientific domains (economics, mental health, or cognitive psychology) described above.

First, participants are asked to draw a graph (for UWPS) representing the expected results. Next, users are shown a visualized function (as well as upper and lower confidence intervals, Figure 2) and asked to drag stars (representing keypoints) to the scientifically interesting locations. After selecting keypoints, TESA is invoked to allocate samples for the next phase of data collection and process the received samples into a user-friendly visualization. This keypoint selection process was repeated three times. Users with

[9] Source code used to generate the graphs is available at https://datadrivengame.science/aaai21/.

[10] The epsilon-greedy keypoint sampler outperforms Thompson sampling in part because it switches to new keypoints immediately, while Thompson sampling slowly updates its posteriors.

zero keypoints after the second round were given an extra prompt to confirm this was intentional. In the final phase of the task, users were presented with two randomly-placed unlabelled graphs: one generated by TESA (which used their keypoints but not their prior curve) and one generated by UWPS (which used their prior curve but not their keypoints). The user's latest keypoints were overlaid over each graph, and underneath was a count of how many datapoints each method had collected at non-keypoints. Users were asked: *"Do you think the left or right experiment did a better job of collecting datapoints in place(s) that were scientifically interesting?"*.

Although the end goal is to use this interface with human scientists, recruiting sufficient subjects of this expert population can be difficult. The availability of non-experts is much higher, but the major limitation is that they do not have the same background knowledge or skills as an expert.

To mitigate this issue, we framed the task as one where laypeople would play the role of a scientist running an experiment. We included instructions on the particular domain, as well a short (two-question) multiple choice quiz to confirm retention. These were reviewed by domain experts to ensure they were correct and complete: The questions and instructions for the economics domain were reviewed by an associate professor of economics, the mental health domain was reviewed by an associate professor of psychology (specializing in counseling psychology), and the cognitive psychology domain was reviewed by a full professor of psychology (specializing in cognitive psychology). We also included instructions (and a two-question quiz) to help users understand the visualizations and the task of marking keypoints. Users were instructed to place keypoints on places they found scientifically interesting as long as they had enough data, but were never told exactly what "enough" was, nor told a precise definition for "scientifically interesting". If users got any questions wrong in either quiz, their results were excluded.[11]

We recruited participants from Amazon Mechanical Turk. In a small pilot study, we found that the task took 14.4 minutes on average, therefore workers were paid $2.34 to ensure a reasonable hourly wage. Each worker was not allowed to do more than one task to ensure independent results. We simulated the collection of 100 datapoints per round per the economics and mental health domains, and 30 datapoints per round for the cognitive psychology domain (since this required brining subjects into a lab, there would naturally be fewer participants).

Results Out of 101 total subjects, 21 managed to correctly answer all quiz questions. The high exclusion rate was likely due to the well-known difficulty of getting laypeople to reason accurately about uncertainty (Kahneman et al. 1982). Nevertheless, the 21 users also tended to behave reasonably in the rest of the task; for instance, they placed more keypoints as more data was gathered, from an average of 1.9 keypoints initially to 3.4 keypoints by the final round.

Our main result is that 20 out of 21 subjects (95%) selected TESA when asked which graph did a better job of collecting datapoints in places that were scientifically interesting. This preference was significant, as judged by the two-tailed Binomial test ($p < 0.001$). It also tended to match the ζ score, UWPS (average $\zeta = -243$) never achieved a better ζ score than TESA (average $\zeta = -113$). This result indicates that our human-in-the-loop paradigm appears to be a better method of directing data collection than specifying a prior function estimate, and that TESA effectively directs data so as to maximize the satisfaction of real human users.

[11]The full text of the quizzes and screenshots of all three parts of the task can be found in the appendix at https://datadrivengame.science/aaai21/.

Discussion and Conclusion

In this paper, we have explored how humans and AI systems can work together to more effectively collect data in scientific spaces. We are one of the first to consider incorporating human feedback about what is scientifically interesting iteratively throughout the process, eliminating the difficult task of having to provide the AI with a complete set of objectives a priori. We developed TESA, an zero-regret algorithm which carefully balances the task of gathering more data about places which are known to be interesting while searching for places that might become interesting in the future. We show TESA performs well using real-world scientific data, both in simulation and in a human study.

A limitation of our human study is that the subjects were laypeople instead of scientists. Although we used quizzes (reviewed by domain experts) to mitigate this, we acknowledge that it is unlikely that the marked keypoints exactly matched scientists' preferences. Even so, our experiments demonstrate the flexibility of our framework to direct data towards a broader set of human preferences.

While we show TESA is zero-regret, we leave regret bounds for future work. We feel that the zero-regret guarantee combined with comprehensive experiments should be sufficient to cause practitioners to consider adopting TESA.

Also, here we have primarily considered one dimensional response functions with a discretized set of x-values. Although we feel TESA is immediately useful in many real-world scientific problems, there are also many scientific domains for which this representation is too simple. For instance, domains such as aerial surveillance have unique challenges, such as needing to consider the cost and feasibility of movement in physical space (Lipor et al. 2017), or needing more complex methods to visualize these high-dimensional spaces. An interesting future direction is how to extend TESA to these more challenging scenarios.

Although there is work to be done, this paper takes the first steps towards developing AI systems that can work as teammates with human scientists, thereby helping greatly accelerate the process of scientific discovery and innovation, especially in cases where data is scarce or expensive.

Acknowledgments

We are grateful for the assistance of Ian Herman, Kayla Schlechtinger, Leon James, Steve Herman, and Keisuke Nakao. Funding was provided by NSF CAREER Award #HCC-1942229 and NSF EPSCoR Award #OIA-1557349.

Ethics Statement

The chief ethical concern of the framework proposed in this paper is its potential to introduce bias. Indeed, the main idea of algorithms such as TESA is to bias scientific data collection towards places that are scientifically interesting. Although this can be extremely valuable to maximize the efficiency of the collected data, it should also be used with caution when trying to make broad claims that cut across conditions (e.g., the framework would not be the best fit for trying to determine which x-value results in the *highest* response as that requires understanding the mean responses of all x-values, but it would be a good fit if the objective was simply to identify some x-value with an *abnormally high* response). Time is also a factor here: allocating samples based on keypoints devotes samples differently based on time, and since experiments (especially with human subjects), may not be identically distributed across time, this is a significant concern compared to A/B testing type experiments in which change across time affects all conditions evenly. In most cases, once the scientifically interesting phenomena are found using our method, these (and other related) problems can be resolved by confirming the findings with a more traditional follow-up experiment (usually involving only a subset of the x-values).

Another potential issue is that some settings of the independent variable, while perhaps scientifically interesting, may cause harm. For instance, while it would be interesting to see how many people would buy avocados if they were priced at $300, one must also consider the negative effect that price point would have on the general public (and on their perception of the avocado industry). We feel that it is the responsibility of the experiment designer to consider these issues carefully and ensure that the experimental setup is in some sense sound with respect to harm, prior to handing control to an AI system.

We also wish to note that our human subject experiment was approved by our Institutional Review Board (IRB).

References

Agrawal, S.; and Goyal, N. 2013. Further Optimal Regret Bounds for Thompson Sampling. In *AISTATS*, 99–107.

Akrour, R.; Schoenauer, M.; and Sebag, M. 2011. Preference-based policy learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 12–27. Springer.

Ardila, D.; Kiraly, A. P.; Bharadwaj, S.; Choi, B.; Reicher, J. J.; Peng, L.; Tse, D.; Etemadi, M.; Ye, W.; Corrado, G.; Naidich, D. P.; and Shetty, S. 2019. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine* 1.

Arora, R.; Dekel, O.; and Tewari, A. 2012. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning*, 1747–1754.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, 322–331. IEEE.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1): 48–77.

Chapelle, O.; and Li, L. 2011. An Empirical Evaluation of Thompson Sampling. In *NIPS*, 2249–2257.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 4299–4307.

Erraqabi, A.; Lazaric, A.; Valko, M.; Brunskill, E.; and Liu, Y.-E. 2017. Trading off rewards and errors in multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*.

Griffith, S.; Subramanian, K.; Scholz, J.; Isbell, C.; and Thomaz, A. L. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, 2625–2633.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, 3909–3917.

Hick, W. E. 1952. On the rate of gain of information. *Quarterly Journal of experimental psychology* 4(1): 11–26.

Howard, R. A. 1966. Information value theory. *IEEE Transactions on systems science and cybernetics* 2(1): 22–26.

Kahneman, D.; Slovic, S. P.; Slovic, P.; and Tversky, A. 1982. *Judgment under uncertainty: Heuristics and biases*. Cambridge university press.

Kiggins, J. 2018. Avocado Prices (Kaggle dataset). https://www.kaggle.com/neuromusic/avocado-prices. [Accessed June 10th, 2020].

Lipor, J.; Wong, B. P.; Scavia, D.; Kerkez, B.; and Balzano, L. 2017. Distance-penalized active learning using quantile search. *IEEE Transactions on Signal Processing* 65(20): 5453–5465.

Liu, Y.-E. 2015. *Building behavioral experimentation engines*. Ph.D. thesis, University of Washington.

Lykouris, T.; Mirrokni, V.; and Leme, R. P. 2020. Bandits with adversarial scaling. In *ICML*.

NIMH 2017. 2017. Prevalence of Any Mood Disorder Among Adults. https://www.nimh.nih.gov/health/statistics/any-mood-disorder.shtml. [Accessed July 31st, 2020].

Noothigattu, R.; Gaikwad, S. S.; Awad, E.; Dsouza, S.; Rahwan, I.; Ravikumar, P.; and Procaccia, A. D. 2018. A voting-based system for ethical decision making. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

OSMI 2016. 2016. OSMI Mental Health in Tech Survey 2016. https://www.kaggle.com/osmi/mental-health-in-tech-2016. [Accessed June 10th, 2020].

Petzold, P.; and Haubensak, G. 2001. Higher order sequential effects in psychophysical judgments. *Perception & Psychophysics* 63(6): 969–978.

Petzold, P.; and Haubensak, G. 2004. A comparison of magnitude estimations and category judgments. Primary data. https://doi.org/10.5160/psychdata.pdpr99ve20. doi: 10.5160/psychdata.pdpr99ve20.

Pukelsheim, F. 1993. *Optimal design of experiments*, volume 50. siam.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* 58(5): 527–535.

Settles, B. 2009. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison Department of Computer Sciences.

Sutton, R. S.; and Barto, A. G. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Thomaz, A. L.; and Breazeal, C. 2006. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*, volume 6, 1000–1005.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 285–294.

Weinstein, B. G.; Marconi, S.; Bohlman, S.; Zare, A.; and White, E. 2019. Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. *Remote Sensing* 11(11): 1309.

Zhifei, S.; and Meng Joo, E. 2012. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics* 5(3): 293–311.