Low-regret selective sampling

SIDDHARTHA BANERJEE, JOSEPH Y. HALPERN, SPENCER PETERS

Settings like content moderation and law differ from other common online decision-making settings such as recommendation systems in that there is often no “direct” feedback. In these settings, decision-makers rely on experts (e.g., experienced colleagues, managers, or higher courts) to clarify difficult decisions. This pattern of interaction is also characteristic of humans supervising algorithms to mitigate bias and error. However, human supervisors and experts in general have limited time, so decision-makers must interact with them efficiently for the overall system to be workable. We draw from the learning theory literature to show that, in certain model settings, the simple strategy of always asking experts for advice when unsure is nearly optimal. In the models we study, this strategy is easy to apply, computationally efficient, and requires no knowledge of the time horizon or the precise parameters of the data distribution. Moreover, it always makes consistent decisions, thus ruling out double standards. This property is especially desirable in the context of legal systems, where a consistent body of case law is crucial to reasoning by precedent.

Additional Key Words and Phrases: Online algorithms, binary classification, selective sampling, regret minimization.

1 INTRODUCTION

In many applications, agents must make a sequence of binary classification decisions without the benefit of direct feedback. Consider the following examples.

Vignette 1. *Carol is a content moderator working for a social media platform. She evaluates thousands of posts a day to see if they violate the platform's policies (e.g., Facebook's Community Standards). Although some posts are marked as violations by users, most are not, and what user feedback exists is often out of line with policy. In particularly unclear cases, Carol can ask her boss, Dolores. Dolores is an expert on the platform's policies, but her time is valuable, so Carol is careful not to waste it.*

Vignette 2. *Rob is an AI sysadmin for an online crowdsourced encyclopedia. Rob gets a stream of "speedy deletion" requests for new pages that violate the encyclopedia's policies, and has the authority to act on these immediately. Rob can also submit them to the formal deletion process, which involves discussion and consensus building among all the core contributors to the encyclopedia, as well as possibly modifying the encyclopedia's policies. This takes time, which may cause problems if the page in question has false or protected information.*

Vignette 3. *Darius is a dictator trying to design a good legal system. He realizes that consistent legal judgments are critical, but his regime lacks the resources to carefully examine each case. So he designs a two-tier system comprised of many district courts and one Supreme Court. He instructs judges to first look at the Supreme Court's past judgments for guidance, and to send cases to the Supreme Court only if they are sufficiently unsure of the right decision. Of course, in real life, judges don't send cases to higher courts. Instead, a judge rules, and the litigants appeal the ruling to a higher court. However, from the perspective of the system as a whole, the net effect is similar; controversial cases are sent to higher courts.*

In each of these vignettes, decision-makers (Carol, Rob, and Darius's judges) are initially unsure of what to do, but must make a sequence of decisions. They face a tradeoff between possibly making the wrong decision, and asking for advice, which consumes the time and effort of more experienced agents,

Arguably, the situations described by the vignettes are very different. Carol is performing a task with a relatively clear-cut objective—separate the unacceptable content from the acceptable content. Most of the time it is clear which is which, but certain exceptional cases require her boss's expertise. Rob and the judges, on the other hand, face much more open-ended scenarios. Rob is trying to predict the consensus opinion of the platform contributors, which might change from one day to the next, or depend on a context that is invisible to Rob, or simply be inconsistent. For the judges, things are even more complicated; they must produce, case by case, a consistent body of *precedents* that flesh out and interpret the law of the land. However, all the vignettes share a common pattern of interaction between lower-level decision makers and higher-up figures. The decision makers can be content moderators, AI assistants, or judges, and the higher-ups can be policy experts, encyclopedia contributors, or higher courts, but in all cases, the decision-makers learn mainly via feedback from the higher-ups. Henceforth, we will use *moderators* and *experts* as shorthand for the lower-level decision makers and the higher-ups, respectively.

This paradigm of interaction is one way of mitigating algorithmic bias. Recently, Alkhatib and Bernstein [2019] have developed a theory of *street-level* algorithms. which, by analogy to street-level bureaucrats [21] (police officers, censors, judges), are those that directly interact with the public. Alkhatib and Bernstein argue convincingly that street-level algorithms are less flexible than street-level bureaucrats, because street-level algorithms cannot judge an exceptional case directly on its merits. Instead, they make their decision solely based on previous labeled examples; moreover, they get feedback on the new case (if at all) only after the fact. This inflexibility results in real-world pain for the people the street-level algorithms interact with. For example, according to [2], workers on Mechanical Turk are often denied payment for good-faith work by quality-control algorithms, because these algorithms are unable to recognize alternate interpretations of the task at hand. Discussing this problem, Alkhatib and Bernstein mention a missing link, the *foreman*. In factories, foremen directed workers and "provided feedback on the output of the work" [2]. Alkhatib and Bernstein write that "The foreman's job was important because even the most standardized work sometimes surfaces unique circumstances". By way of contrast, many algorithms deployed today seem to operate under the assumption that the work never surfaces unique circumstances; that it is pinned down exactly by the training data. When this assumption is violated, workers on Mechanical Turk are denied pay, transgender YouTubers are demonetized, and systematic inequality in the justice system is reinforced [2]. Reincorporating the foreman role into these systems in the form of a human supervisor, and designing algorithms that can interact with the foreman efficiently (in part, by asking for advice at appropriate times) has potential for ameliorating bias and error.

Designing practical decision-making algorithms for the settings outlined above is a complex task, involving considerations from sociology, human-computer interaction, legal reasoning and more. In this work, we examine the problem from a learning-theoretic standpoint. We find that, perhaps counterintuitively, the simple approach of asking for nadvice whenever previous advice leaves room for uncertainty is nearly optimal (in certain standard settings). This approach has multiple practical advantages; it can be described independently of the specifics of the problem, and it leads to consistent decisions (we elaborate on this in the next paragraph). The learning-theoretic setting we operate in is called *selective sampling*, and our simple approach is sometimes termed the *CAL algorithm*, after Cohn, Atlas and Ladner [8]; however, our view of the problem differs from traditional perspectives on machine learning classification systems (such as recommendation systems) in two ways. First, the feedback does not come from large labeled datasets, or even (as in active learning) large datasets of unlabeled points for which labels can be collected. Instead, it comes from experts who can be asked about specific instances that come up. Second, we focus on achieving consistent decisions, rather than simply making few misclassifications and asking for few labels (although it will turn out that our approach does ask for few labels).

In all of the settings we've discussed so far, *consistency* is an important desideratum. Indeed, the requirement of consistency is arguably the most important common feature of the vignettes above. A set of decisions is consistent when, intuitively, there is a single explanation that explains all the decisions. Besides the natural human desire for consistency, inconsistency often leads to uncertainty about the basis for the decisions, and concerns about unequal treatment. While inconsistency does not necessarily imply a double standard, it is often cited as evidence of unequal treatment. For example, a recent New York Times article [18] reporting on a Senate hearing on social media and misinformation summed up the discussions on content moderation as follows: "What Republicans and Democrats agreed on was that Facebook and Twitter have enforced their policies inconsistently, and often without elucidating why they had taken the steps that they did." This perception of inconsistency coincided with concerns that conservative postings were moderated more often [18]. Moreover, in a legal setting, where (in the US, UK, and other common law jurisdictions) previous decisions serve as precedent for future cases, inconsistency seems to introduce fundamental problems. Reasoning from inconsistent precedents, legal actors can deduce arbitrary conclusions, which makes the body of case law even more inconsistent. According to [23], the US Supreme Court has overruled previous decisions specifically because these decisions were inconsistent with other decisions.

Thus, our aim is to study decision-making problems under the constraint that the decisions be consistent (alternatively, error-free). In particular, in the setting where there is a known set of reasonable explanations (hypotheses) that the experts might be using to make decisions, we take an *error-free* moderator to be one who asks for advice whenever there are two reasonable explanations for the decisions already made that disagree on the decision at hand. That approach is often referred to as the *CAL algorithm* in the learning theory literature [8]. We show that the error-free moderator is surprisingly effective; in the simplified cases that we study, it achieves optimal or nearly optimal results (Theorems 1 and 3).

1.1 Our model

The most generic formulation of our model is as follows. In each round $t \in [T]$,[1] the moderator A receives an *instance* X_t from some set $\mathcal{X}$. X_t represents features of the current decision, for example, the number of profane words in a social media post. Upon receiving X_t, A takes one of three actions. It either *classifies* X_t *as negative* ($\hat{Y}_t = -1$), *classifies* X_t *as positive* ($\hat{Y}_t = 1$), or *asks for advice* ($\hat{Y}_t = 0$). If A asks for advice, then the expert selects $Y_t \in \{-1, 1\}$ and reveals Y_t to A.

This model has been called *selective sampling* in the learning theory literature, since the algorithm (moderator) selects which X_t to ask for advice on [11, 14]. In selective sampling, typically Y_t is defined for all rounds t, and the objectives are to bound the number of classification mistakes $0 \neq \hat{Y}_t \neq Y_t$ (or the related generalization error) while simultaneously bounding the number of requests for advice; see Section 3 for a more in-depth discussion.

The interpretation of Y_t depends on the application, but it typically represents the classification that the moderator is trying to match. In Vignette 1, Y_t represents Dolores's best guess for whether the post is acceptable or unacceptable according to the platform's policies. In Vignette 2, rather than a "best guess", Y_t directly represents the preferences that Rob is trying to conform to. In Vignette 3, Y_t represents the opinion of the Supreme Court. Here the Y_t are viewed as good judgments because the Supreme Court, unlike the district courts, has enough resources and legal talent to reliably make fair and consistent decisions.

Likewise, there are many different reasonable objectives the moderator could have, depending on the situation. The moderator typically wants to make the classification that the expert would have suggested, had they been asked. Under assumptions that allow us to determine this classification Y_t for all rounds, not just those rounds on which the moderator asked for advice, it makes sense to penalize classification mistakes; that is, rounds t where $\hat{Y}_t \neq 0$ and $\hat{Y}_t \neq Y_t$. For penalizing classification mistakes, a standard and fairly general approach is to assign a numerical cost $c_{mistake} \geq 0$ to each such mistake. The moderator could also quantify the resource use associated with asking for advice using a numerical cost $c_{advice} \geq 0$. In general, it may make sense for these costs to depend on the instance X_t, whether the misclassification was negative or positive, and even the time t. So long as there are bounds c_{low}, c_{up} such that, regardless of context, $c_{low} \leq c_{mistake}, c_{advice} \leq c_{up}$, our asymptotic results apply; more fine-grained analysis is outside the scope of this paper. In what follows, for simplicity, we will take $c_{mistake} = c_{advice} = 1$. Hence the total cost C_T up to time T is simply the total number of mistakes plus the total number of requests for advice: $C_T = \sum_{i=1}^{T} c_i$, where $c_i = 1$ if $\hat{Y}_i \neq Y_i$ and $c_i = 0$ if $\hat{Y}_i = Y_i$.

This generic formulation does not prescribe how X_t and Y_t are chosen. This flexibility is desirable, since we expect that specific applications will differ widely. The X_t might be chosen at random, adversarially, or by some other process; and what counts as a correct label may vary over time. However, in the rest of this paper, we mostly focus on the special case where there is a known hypothesis class $\mathcal{H}$ consisting of functions $h : \mathcal{X} \to \{-1, 1\}$, and there is some *target hypothesis* $h^* \in \mathcal{H}$ such that $Y_t = h^*(X_t)$ for all t; that is, we can view the decisions as having been made according to h^*. In contrast to the typical machine learning viewpoint, h^* need not be interpreted as assigning "ground truth" or "correct" labels to instances for our results to make sense. Our

[1] In this paper, for $n > 0$, $[n]$ denotes the set $\{1, \ldots, n\}$.

results show that if consistent decisions with respect to experts are desired, they can be obtained efficiently, whether or not the experts' decisions are in any sense "correct". We mainly consider the case where the X_t are drawn i.i.d. from some distribution $\mathcal{D}$, but we also study the case where the X_t are chosen adversarially.

2 RESULTS

The central theme of this paper is that consistent decisions can be made cheaply. That is, a moderator that never makes mistakes (but potentially asks for advice frequently) performs surprisingly well. A moderator who is guaranteed not to make any mistakes must ask for advice whenever there are two hypotheses that agree with previous advice but disagree on the decision at hand. The error-free moderator A_{CAL} is the one who asks for advice iff this condition is met. To define A_{CAL} formally, we introduce some notation.

Given $n > 0$ and *labeled instances* $\mathcal{S} \in (\mathcal{X} \times \{0,1\})^n$, let $\mathcal{H}_{\mathcal{S}} := \{h \in \mathcal{H} \mid \forall (x,y) \in \mathcal{S}, h(x) = y\}$ be the set of hypotheses consistent with $\mathcal{S}$. Thus, $\mathcal{H}_{\mathcal{S}}$ consists of all the hypotheses that are consistent with the the labels on the instances in $\mathcal{S}$. A set of hypotheses $\mathcal{H}'$ *determines* x, written $\mathcal{H}' \twoheadrightarrow x$, if there is some $y \in \{0,1\}$ such that for all $h \in \mathcal{H}'$, $h(x) = y$. Intuitively, $\mathcal{H}'$ determines x if all the hypotheses in $\mathcal{H}'$ agree that the label of x should be y. The subset of $x \in \mathcal{X}$ such that $\mathcal{H}'$ does not determine x is often called the *region of uncertainty*. In this case, we call y the *inferred label* of x. Next, for an instance x, $\mathcal{S}$ *determines* x, written $\mathcal{S} \twoheadrightarrow x$, if $\mathcal{H}_{\mathcal{S}} \twoheadrightarrow x$. Finally, X_t *is determined* if $\mathcal{S}_t$ determines X_t, where $\mathcal{S}_t := \{(X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})\}$. Now we can define A_{CAL}: if X_t is determined, A_{CAL} classifies X_t according to the inferred label; otherwise A_{CAL} asks for advice. This is allowed because at time t, A_{CAL} knows the labels of all X_s with $s < t$; this is because either A_{CAL} asked for advice at time s, or X_s is determined. In either case, A_{CAL} knows the label Y_s. Notice that A_{CAL}

(1) makes no mistakes (and is therefore consistent), as discussed earlier,
(2) is independent of the parameters of the data distribution,
(3) and is independent of the time horizon T over which the cost is to be minimized.

We also demonstrate, both theoretically and experimentally, that it is computationally efficient in the simple cases that we study. In particular, the update in round t will turn out to be independent of t, even though the set $\mathcal{S}_t$ is growing with t; see Section 5.1 for details.

Given these desirable properties, it is natural to consider how A_{CAL} compares to other decision algorithms in our selective sampling model in terms of the cost C_T. In the next subsection we consider two simple, standard settings and show that A_{CAL} performs near-optimally.

2.1 Bounds on Cost

We begin by studying the case where the features X_t are bounded scalar values, and the hypotheses are thresholds. This case is particularly simple and admits strong results; it also serves to build intuition. Our main result for this case is as follows.

Theorem 1. Given a set S, let $\mathbb{1}_S$ be the indicator function of S. If $\mathcal{X} = [0,1]$, the instances X_t for $t \in [T]$ are drawn i.i.d. from a continuous distribution $\mathcal{D}$ over $\mathcal{X}$, and $\mathcal{H} = \{\mathbb{1}_{\{x \geq k\}} \mid k \in [0,1]\}$, then for all hypotheses $h \in \mathcal{H}$, the following claims hold:

(1) A_{CAL} has expected cost $O(\log T)$.
(2) For all $\delta > 0$, with probability $1 - \delta$, A_{CAL} incurs cost $O(\log T + \log(1/\delta))$.

Proof. Suppose that the target hypothesis corresponds to the threshold $k \in [0,1]$. Notice that at time t, the region of uncertainty is the interval (A_t^k, B_t^k) between the largest instance A_t^k among $X_1, \ldots, X_{t-1}$ with $A < k$, and the smallest instance B_t^k with $B_t^k \geq k$. (If $X_j \geq k$ for all j, define

$A_t^k = 0$; if $X_j < k$ for all j, define $B_t^k = 1$.) Instances smaller than A_t^k are known to have label -1 if k is the correct threshold; those larger than B_t^k are known to have label 1.

Since (A_t^k, B_t^k) is the interval between two adjacent instances, and there are $t-1$ instances total, we expect the probability mass $Pr_{X \sim D}(X \in (A_t^k, B_t^k))$ of (A_t^k, B_t^k) to be of order $1/t$. Here is a proof: The event that X_t falls between A_t^k and B_t^k is the same as the event that either $X_t = A_{t+1}^k$ or $X_t = B_{t+1}^k$. But since the X_i, $i \in [t]$, are i.i.d., each X_i has equal probability of being A_{t+1}^k or B_{t+1}^k. Furthermore, there is some $i \leq t$ for which either $X_i = A_{t+1}^k$ or $X_i = B_{t+1}^k$. Hence, $1/t \leq Pr(X_t = A_{t+1}^k \text{ or } X_t = B_{t+1}^k) \leq 2/t$. It follows $1/t \leq Pr(X_t \in (A_t^k, B_t^k)) \leq 2/t$. However, the cost c_t at round t is 1 if A_{CAL} queries and 0 otherwise; and A_{CAL} queries iff $X_t \in (A_t^k, B_t^k)$. So we have

$$\mathbb{E}\, C_T = \sum_{t=1}^{T} \mathbb{E}\, c_t = \sum_{t=1}^{T} \mathbb{E}\, \mathbb{1}_{X_t \in (A_t^k, B_t^k)} = \sum_{t=1}^{T} Pr(X_t \in (A_t^k, B_t^k)).$$

It follows $\sum_{t=1}^{T} 1/t \leq \mathbb{E}\, C_T \leq \sum_{t=1}^{T} 2/t$. Standard bounds for these harmonic sums then give $\log T \leq \mathbb{E}\, C_T \leq 2 \log T + 1$, completing the proof. A proof of (2) can be found in Appendix B in the supplementary material. □

Theorem 1 is tight, as the following theorem shows.

Theorem 2. In the setting of Theorem 1, if $\mathcal{D}$ is continuous, then for all selective sampling algorithms *ALG*, there exists a target hypothesis such that *ALG* has expected cost $\Omega(\log T)$.

A proof of this theorem can be found in Appendix B in the supplementary material. To get a sense of why this statement is true, note that after dropping t instances in $[0, 1]$, even if all the labels are observed, the gap between the largest negatively labeled instance and smallest positively labeled instance is of order $1/t$. Any new instance that falls in, say, the middle half of this gap has a reasonable chance of being misclassified if the moderator does not ask for advice. Hence the expected cost cannot be better than $\sum_{t=1}^{T}(1/t) < \log T + 1$.

Combining these results, we see that so long as the X_t are drawn from a fixed continuous distribution on $[0, 1]$, then with high probability, the error-free moderator achieves optimal cost. It is, however, easy to see that if the X_t are chosen arbitrarily, no moderator can beat the trivial $O(T)$ cost bound in the worst case (see Proposition 7).

Our analysis of the case $\mathcal{X} = [0, 1]$ can be extended to provide lower and upper bounds on cost in other cases; we later apply it to our main example. We can get a more general lower bound by applying PAC (Probably Approximately Correct) learning bounds [27]. This bound seems almost trivial (and may well be known by experts in the field, although we could not find a reference), but it is tight up to a factor polynomial in $\log \log 1/\epsilon$ in the cases we consider. To state this lower bound, we first recall the definition of *PAC learning sample complexity* [22, 26].[2]

Definition 1. Given a set $\mathcal{X}$, a hypothesis class $\mathcal{H}$ consisting of hypotheses $h : \mathcal{X} \longrightarrow \{-1, 1\}$, a distribution D over $\mathcal{X}$, and parameters $\epsilon > 0, \delta > 0$, the *PAC learning sample complexity* $m_{\mathcal{H},D}(\epsilon, \delta)$ is the minimum number m such that there exists an algorithm *ALG* such that for all $h \in \mathcal{H}$, given m labeled samples $(X_1, h(X_1)), \ldots, (X_m, h(X_m))$, *ALG* returns a function $f : \mathcal{X} \longrightarrow \{-1, 1\}$, not necessarily in $\mathcal{H}$, such that with probability $1 - \delta$, $P_{x \sim D}(f(X) \neq h(X)) \leq \epsilon$.

Our general lower bound is given in the following proposition.

[2] There is a small difference between the definitions of PAC learning sample complexity given in [26] and [22]; see Remark 3.2 of [26].

Proposition 1. *If $m_{\mathcal{H},D}(\epsilon,\delta) \in \Omega(f(\epsilon,\delta))$, then there exist $\Delta > 0$ and $c > 0$ such that, defining $f'(\epsilon) = f(\epsilon,\Delta)$ and f'^{-1} to be the inverse of f', all selective sampling algorithms have*

$$\mathbb{E}\, C_T \in \Omega\left(\sum_{t=1}^{T} f'^{-1}(ct)\right).$$

Proof. Consider a modified setting where the algorithm always observes Y_t, whether it asks for the label or not. Clearly, any algorithm for the selective sampling setting can be viewed as an algorithm for the modified setting, by simply ignoring the labels Y_t on rounds t where it does not ask for the label. Thus, it suffices to show an $\Omega(\log T)$ bound on expected cost in the modified setting. Fix an algorithm *ALG*. The bound $m_{\mathcal{H},D}(\epsilon,\delta) \in \Omega(f(\epsilon,\delta))$ implies the existence of $\mathcal{E} > 0$, $\Delta > 0$, and $c > 0$ such that for all $\epsilon \le \mathcal{E}$ and all $\delta \le \Delta$, there exists a target hypothesis h^* such that if *ALG* is given fewer than $cf(\epsilon,\delta)$ labeled samples, with probability at least δ, it will output a hypothesis h such that $P_{x\sim\mathcal{D}}(h(x) \neq h^*(x)) \ge \epsilon$. As in the theorem statement, let $f'(\epsilon) = f(\epsilon,\Delta)$ and f'^{-1} be the inverse of f. Then the above claim implies that if *ALG* is given fewer than t samples, with probability at least Δ, it will output a hypothesis h such that $P_{x\sim\mathcal{D}}(h(x) \neq h^*(x)) \ge f^{-1}(ct)$. Hence the expected cost incurred by *ALG* in round $t-1$ is at least $\Delta f^{-1}(ct)$. It follows that

$$\mathbb{E}\, C_t = \sum_{i=1}^{T} \mathbb{E}\, c_t = \sum_{i=1}^{T} \Delta f^{-1}(ct) = \Omega(\sum_{i=1}^{T} f^{-1}(ct)),$$

as claimed. □

Intuitively, this bound holds because in the PAC setting, labels are free, so PAC is easier than selective sampling. Using the so-called “fundamental theorem of statistical learning” and the fact that the VC-dimension of the class of threshold hypotheses is 1, (see [26], Section 6), this implies a weaker version of Theorem 2; namely, that the $\Omega(\log T)$ bound holds in the worst case over distributions.

On the other hand, PAC upper bounds do not in general imply any upper bound for the label complexity of the CAL algorithm. In particular, there are combinations of hypothesis classes and data distributions that can be PAC-learned efficiently, but for which CAL must always query *all* labels. Consider learning (arbitrary) linear separators in $\mathbb{R}^2$ against the following distribution for X_t: Z_t is drawn uniformly from $[-5, 5]$, and $X_t = (Z_t, 1/Z_t)$. The true hypothesis is that instances with $x < 0$ are negative, and those with $x \ge 0$ are positive. It is easy to see that no X_t is ever determined, so the CAL algorithm will ask for every label; see Figure 1 for an illustration. However, the VC-dimension of arbitrary linear separators in $\mathbb{R}^2$ is 3, so the fundamental theorem of statistical learning [26, Theorem 6.8] implies a PAC sample complexity of $O(\frac{d\log(1/\epsilon)+\log(1/\delta)}{\epsilon})$.

We can generalize our analysis of the upper bound as follows. Define

$$\mathcal{B}_t = \{X_i | i \in [t], (\{(X_1,Y_1),\ldots,(X_t,Y_t)\} \setminus (X_i,Y_i)) \not\twoheadrightarrow X_i\}.$$

The points $X_t \in \mathcal{B}_t$ are called *undetermined observations*; they are the instances observed by the algorithm whose labels cannot be deduced from the other labeled instances that the algorithm has seen up until time t. In the proof of Theorem 1(1) above, $\mathcal{B}_t \subseteq \{A_t^k, B_t^k\}$.[3]

Proposition 2. *The cost incurred by A_{CAL} is $C_T = \sum_{i=1}^{T} Pr(X_t \in \mathcal{B}_t)$.*

Proof. The proof is a straightforward generalization of the proof of Theorem 1, Claim 1.

First let us formalize the notion of *region of uncertainty.*

[3] While it is typically the case that $\mathcal{B}_t = \{A_t^k, B_t^k\}$, if $X_i \ge k$ (resp. $X_i < k$) for all $i < t$, then $\mathcal{B}_t = \{B_t^k\}$ (resp. $\mathcal{B}_t = \{A_t^k\}$).

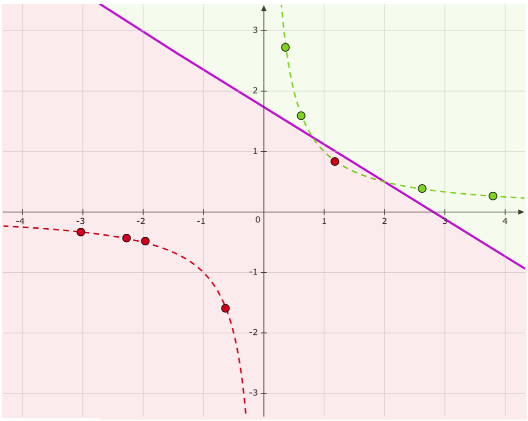


Fig. 1. Separating X_t from other instances with the same label. The red (resp., green) dashed line is the support of the distribution of the negative (resp. positive) instances. The dots represent instances X_t. All instances on the green dashed line have positive labels, however, for any such instance, a linear separator can be found (the purple line) which classifies it as negative, but is consistent with the labels of all other instances.

Definition 2. For $t \geq 1$, the *region of uncertainty* $\mathcal{R}_t$ is the set $\{x \in \mathcal{X} \mid \mathcal{S}_t \not\twoheadrightarrow x\}$.

In the proof of Theorem 1(1), $\mathcal{R}_t = (A_t^k, B_t^k)$. In that proof, it was shown $C_T = \sum_{t=1}^{T} Pr(X_t \in (A_t^k, B_t^k))$. An identical argument shows that in general,

$$C_T = \sum_{t=1}^{T} Pr(X_t \in \mathcal{R}_t). \tag{1}$$

Also in the proof of Theorem 1(1), $\mathcal{B}_t$ were the instances X_i, $i \in [t]$ such that either $X_i = A_{t+1}^k$ or $X_i = B_{t+1}^k$. It was shown that $Pr(X_t \in (A_t^k, B_t^k)) = Pr(X_t = A_{t+1}^k \text{ or } X_t = B_{t+1})$. Similarly, it holds that $Pr(X_t \in \mathcal{R}_t) = Pr(X_t \in \mathcal{B}_t)$. This is by definition of $\mathcal{R}_t$ and $\mathcal{B}_t$; X_t falls in the region of uncertainty iff X_t is not determined by the previous labeled points; that is, if X_t is an undetermined observation. This completes the proof. □

It follows that

Corollary 1. *The expected cost incurred by A_{CAL} is $\mathbb{E}\, C_T = \sum_{i=1}^{T} \mathbb{E}\, |\mathcal{B}_t|/t$, where $|\mathcal{B}_t|$ denotes the cardinality of $\mathcal{B}_t$.*

Proof. As in the proof of Theorem 1, since the X_t are i.i.d., each X_i, $i \in [t]$, is equally likely to be in $\mathcal{B}_t$. Hence $Pr(X_t \in \mathcal{B}_t) = |\mathcal{R}_t|/t$. Taking expectations of both sides of Equation 1 completes the proof. □

These expressions relating expected cost and the number of undetermined observations are straightforward, and not entirely new (a similar approach can be found in the proof of Theorem 21 in [12]). The difficult task is to find the probabilities $Pr(X_t \in \mathcal{B}_t)$ or the expectations $\mathbb{E}\, |\mathcal{B}_t|$. However, in the simple cases we study, the expectations $\mathbb{E}\, |\mathcal{B}_t|$ can be found. The proof sketch of Theorem 1 provides an example; for thresholds, $|\mathcal{B}_t| \leq 2$ for all t, so $\mathbb{E}\, C_T \sim 2 \log T$.

Of course, agents often have access to more than a single numerical score. If the X_t each comprise d numerical features, we can view them as points in $\mathcal{X} \subseteq \mathbb{R}^d$. One simple and natural hypothesis class for such features is the class of *linear separators* $\mathcal{H} = \{\mathbb{1}_{x \cdot w \geq b} \mid w \in \mathbb{R}^d, b \in \mathbb{R}\}$. For $a \in \mathbb{R}^d$

and $i \in [d]$, let $a[i]$ denote the i^{th} coordinate of a; then hypotheses in this class correspond to giving each feature $x[i]$ a weight $w[i]$ and applying a threshold b. Their decision boundaries are hyperplanes. In the following, we restrict attention to the case where the X_t are drawn *i.i.d.* from a multivariate Gaussian and where $b = 0$ (*homogenous* linear separators). These assumptions imply a balanced sample $P(Y_t = -1) = P(Y_t = 1) = 1/2$, which doesn't seem too unreasonable in the context of law, or content selected for moderation. While it would certainly be interesting to consider a wider class of distributions, as well as arbitrary linear separators, our proof techniques and those of the work that we build on rely on the symmetry of the distribution of the X_t with respect to the decision boundaries. Furthermore, strong negative results are known for slightly more complex settings; even if we keep the distributional assumption on the X_t, but allow $b \neq 0$, selective sampling algorithms must ask for *all* labels to ensure a low misclassification rate [9].

Note that for hypotheses $h = \mathbb{1}_{x \cdot w \geq b}$ with $b = 0$, the label $h(x)$ does not depend on the magnitude of x. Hence, normalizing the instances does not affect the performance of A_{CAL}. The instances of a multivariate Gaussian whose covariance matrix is the identity, after normalization, are uniformly distributed on the unit hypersphere. Thus, any bound for the uniform distribution on the unit hypersphere also applies to this multivariate Gaussian, and vice versa. We exploit this connection in the lower bound that follows.

For the case of linear separators and the uniform distribution on the unit hypersphere, there is a well-known tight lower bound on PAC learning sample complexity; specifically [22, Theorem 1] shows that $m_{\mathcal{H},\mathsf{Unif}\,S^d}(\epsilon, \delta) = \Omega(\frac{d}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta})$.

Thus, we can take $f(\epsilon, \delta) = \frac{d}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta}$ in our general lower bound (Proposition 1). Doing so yields the following lower bound.

Proposition 3. *If $\mathcal{X} = \mathbb{R}^d$ and the X_t are drawn i.i.d. from the uniform distribution on S^d, then all algorithms for selective sampling incur expected cost $\Omega(d \log T)$.*

Proof. Taking $f(\epsilon, \delta) = \frac{d}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta}$ in Proposition 1, we have that $f'(\epsilon) = \frac{d + \log 1/\Delta}{\epsilon}$. Hence $f'^{-1}(x) = \frac{d + \log 1/\Delta}{x}$, and Proposition 1 says that $\mathbb{E}\, C_T \in \Omega(\sum_{t=1}^{T} \frac{d + \log 1/\Delta}{ct})$. Simplifying,

$$\mathbb{E}\, C_T \in \Omega\left(\sum_{t=1}^{T} \frac{d + \log 1/\Delta}{ct}\right) = \Omega\left(\frac{d + \log 1/\Delta}{c} \log T\right) = \Omega(d \log T).$$

□

Our main result shows that A_{CAL} nearly matches this lower bound.

Theorem 3. If $\mathcal{X} = \mathbb{R}^d$ and the X_t are drawn i.i.d. from a (possibly unknown) multivariate Gaussian $N(0, \Sigma)$ with mean zero, then for all hypotheses $h \in \mathcal{H}$, A_{CAL} incurs expected cost $\Theta(d^{3/2} \log T)$.

This theorem shows that the expected cost incurred by A_{CAL} has the optimal $\log T$ dependence on T. To the best of our knowledge, our analysis is the first to show a strict $O(\log T)$ dependence, as opposed to $O(\log T \log \log T)$. We note that, when viewed as a selective sampling algorithm, A_{CAL} achieves an expected generalization error of ϵ using $O(d^{3/2}/\epsilon)$ samples and $O(d^{3/2} \log(1/\epsilon))$ labels. Again, to the best of our knowledge, our analysis shows the first $O(\log(1/\epsilon))$ label complexity, as opposed to $O(\log(1/\epsilon) \log \log(1/\epsilon))$ label complexity. However, the expected cost incurred by A_{CAL} does not have optimal dependence on d. There are several known algorithms [4, 5, 11] that attain optimal linear dependence or $O(d \log d)$ dependence, at the cost of the additional $\log \log T$ factor in the bounds mentioned above. See Section 3 for more discussion of these algorithms.

3 RELATED WORK

Our model is one of many models of sequential decision making or learning problems without full feedback. The simple algorithm A_{CAL} that we study is often called the CAL algorithm, after Cohn, Atlas, and Ladner, who first described it [8]. It operates in the setting of selective sampling that we describe, where the algorithm (moderator) receives instances one at a time, and for each one, must decide whether to ask for its label or not. Note that the term "selective sampling" was coined by Cohn, Atlas, and Ladner specifically to refer to the CAL algorithm, but most later papers use the term as we do here (see, e.g., [5, 11, 14]). In selective sampling, examples (X_t, Y_t) are drawn i.i.d. from some fixed distribution. The goal is typically to learn a hypothesis h from some hypothesis class $\mathcal{H}$ that has low generalization error, while using few labels, and making few classification mistakes (even on unlabeled points). These three objectives are related, in the sense that if the algorithm has high generalization error, it cannot simultaneously make few label requests and few mistakes. The case we study in this paper corresponds to the "realizable" case, where $Y_t = h^*(X_t)$ for some function h^*, and h^* is in the hypothesis class. This realizability assumption is implicit in the PAC learning framework, and is made in [5, 11, 14]. Query By Committee (QBC), the first selective sampling algorithm to achieve a given error with exponentially fewer labels than its supervised counterpart, was analyzed in [14] for essentially the same hypothesis class and instance distribution that we consider in this paper, namely, linear separators through the origin and the uniform distribution on the unit hypersphere. However, the computational complexity of QBC's t^{th} update step scales (polynomially) with t.

Several more recent algorithms improve on QBC in the sense that the computational complexity of these algorithms' t^{th} update step is independent of t. The Active Modified Perceptron (AMP) algorithm [11] is a variation on the classical Perceptron algorithm [25]. It maintains a hypothesis V_t and, if a label is inconsistent with V_t, updates V_t in the direction of $Y_t X_t$. It maintains and adaptively shrinks a margin threshold S_t and requests a label when $|X_t \cdot V_t| \leq S_t$. Another set of "margin-based" algorithms were presented in [4]. Like the AMP algorithm, these algorithms request labels when a margin condition $|X \cdot V_k| < B_k$ is met. However, the thresholds B_k in each phase k are fixed up front. All of these algorithms have label complexity bounds that depend on T as $\log T \log\log T$; see Appendix A in the supplementary material for details. These bounds typically have optimal linear or log-linear dependence on d. In contrast, we show that A_{CAL} has expected label complexity that depends optimally on T, specifically, $O(d^{3/2} \log T)$. If the dimension is a constant, our results show that A_{CAL} will eventually have the best performance. A_{CAL} has other advantages as well; most importantly, it is extremely simple to describe, and its description is independent of the details of the instance space, hypothesis class, and distribution. All the other algorithms discussed rely on the notion of "margin", which depends on the instance space. This does not mean that A_{CAL} will have similar guarantees in other settings; only that it is clear how to generalize to these settings.

It is worth noting that the margin-based algorithms in [4] have also been shown to apply to learning linear separators through the origin for a more general class of distributions, namely isotropic log-concave distributions. It would be interesting to see if we could extend our analysis of CAL to these distributions.

Removing the assumption that $Y_t = h^*(X_t)$ for some $h \in \mathcal{H}$ leads to the more general "agnostic" case. This case is also well studied; see [3, 10, 15]. The strategy A^2 described in [3] is conceptually very similar to A_{CAL}. It labels an instance whenever two surviving hypotheses disagree on that instance. However, rather than discarding a hypothesis whenever a label inconsistent with it is observed (which could result in discarding the best hypotheses), A^2 discards a hypothesis when it is confident that it is worse than some other hypothesis. This is a nice alternative to A_{CAL} if moderators have to contend with noisy advice. In principle, our techniques could be used to analyze A^2 and other algorithms for the agnostic case, although this seems much harder than in the realizable case.

Other related models include *apple tasting* and *selective classification.* In apple tasting [16], the algorithm makes classifications one at a time, but only learns the true label if it makes a positive classification. The goal is to minimize the number of mistaken classifications. Under the assumption that mistakes and labels have the same cost, all of our cost bounds immediately apply to apple tasting, via the following simple reduction: Whenever the algorithm queries, instead make a positive classification. Thus, the algorithm obtains the same information (the label) and incurs no more cost (a query always incurs cost 1, but the positive classification incurs cost at most 1).

In *selective classification* [7], in each round t, the algorithm receives X_t and decides whether to *abstain* or output a classification $\hat{Y}_t$. In either case, the algorithm then receives the true label Y_t. The goal is to attain high coverage (abstain on few rounds) while simultaneously making few classification mistakes. While this model is quite different from selective sampling in general, the case of *perfect selective classification* [12], where the goal is to make zero classification mistakes, is not so round t, he is nearly certain of Y_t, so the fact that in the selective sampling framework he does not receive Y_t is different. Intuitively, if an algorithm makes no classification mistakes with high probability, then if it outputs a classification on round t, the algorithm is is nearly certain of Y_t, so the fact that in the selective sampling framework it does not receive Y_t is immaterial. The CAL algorithm is a perfect selective classification algorithm. A reduction from perfect selective classification to selective sampling was given by El-Yaniv and Wiener [12, Theorem 7]. El-Yaniv and Wiener [12, Theorem 21] also use the "sample boundary" to give a lower bound on the label complexity of CAL learning (arbitrary) linear separators against a standard normal distribution. The proof of this bound seems to be the closest to our approach, although we prove tight upper and lower bounds for all hypotheses, rather than the existence of a hypothesis with high label complexity.

4 DISCUSSION

To briefly summarize our work: we have studied the design of a hierarchical two-level decision-making system. Moderators classifying incoming instances must either ask for advice from experts (which consumes their time and attention) or make decisions without guidance (and potentially make mistakes). In this context, our main contribution is in showing that a simple heuristic A_{CAL}, wherein moderators ask for advice whenever there are competing consistent hypotheses, is *near optimal* in terms of rate of queries in a wide variety of learning settings. Moreover, it makes no mistakes, which means that the decisions it makes are always consistent.

Our main technical contribution is to show that A_{CAL} asks for advice only $O(d^{3/2} \log T)$ times in expectation over T instances, when applied to learning homogenous linear separators for instances drawn from a given multivariate Gaussian distribution. This logarithmic dependence on T is new (we believe) and it is provably optimal. We also show that A_{CAL} can be implemented in $poly(d)$ time per round, independent of the number of samples T. These facts suggest that A_{CAL} is a valuable addition to the online decision-maker's toolkit, particularly in fields like content moderation. That said, our model clearly simplifies many features of real systems. In this section, we motivate how our insights can extend to cover some of these omisions, and speculate about some more nuanced questions that our framework can help study in future.

There are four immediate issues that deserve further attention:

(1) the unrealistic simplicity of A_{CAL} and the fact that it appears different from methods used in the wild;
(2) our use of a static hypothesis class,

(3) our particular choice of cost function, and
(4) the potential for strategic behavior in these settings.

First, despite our analysis, the heuristic of asking for advice whenever there is any uncertainty does not seem entirely realistic. Intuitively, one expects that in real decision-making, there is nearly always some uncertainty about the right decision, and deferring to experts at the first hint of uncertainty might result in prohibitively many label requests. Why is this not the case?

One thought could be that despite its simplicity, implementing A_{CAL} is actually much more expensive in practice than in theory. Thus, perhaps our intuition that real-world settings require more complex algorithms comes from computational limitations. Even if in principle we could use previous decisions to arrive at a conclusion about the decision at hand with certainty, the memory and computational burdens involved may be too severe. For example, in principle, a judge must consider *all* past cases which could have any bearing, which may be prohibitive in high-dimensional settings. [4] On the other hand, our experiments in Subsection 5.1 show that, if our theoretical assumptions about the instance distribution and hypothesis class hold, A_{CAL} can be implemented to run in a reasonable amount of time. Even for $d = 50$, each iteration averaged 0.32 seconds on a 2015 MacBook Pro laptop; for reference, human content-moderators at Facebook examine roughly one image every 3 seconds [1]. Thus, if the theoretical assumptions about the hypotheses and data distribution hold, it seems that computational performance is not an obstacle to using A_{CAL} in practice.

An alternative explanation is that the hypothesis classes we consider are of much lower complexity compared to those used in practice. While our assumptions of multivariate Gaussian instances and linear separators does not seem unreasonable (indeed, linear separators are a workhorse in machine learning, and extend to much more complicated hypothesis classes via the "kernel trick" [26]); our assumptions might still be too restrictive in practice. It is our understanding that, for example, at real social-media platforms, content-moderation algorithms are typically implemented via deep learning, which induces a much richer hypothesis class. That said, we believe much of our analysis can extend to *local linear models*, which may be appropriate for many settings; for example, a linear model on features like "monetary value" might be appropriate for judging larceny cases, while a separate linear model on features like "self-defense" and "strength of evidence linking murder weapon to suspect" may be more appropriate for murder cases.

Another consideration is that in real-world settings, both the target hypothesis class (for example, the platform's policies in content moderation) and the set of features used will change over time. To give an example, in December 2008, Facebook came under scrutiny for its decision to remove photos of women breastfeeding, because the existing nudity policy prohibited bare breasts (independent of context). In response, Facebook updated its policy to allow certain photos involving breastfeeding [1, 17]. In the linear model, this could be modeled as saying the dimension d increases over time; indeed, Section 14 of Facebook's Community Standards explains that "Our nudity policies have become more nuanced over time", and goes on to explain specific criteria that delineate acceptable nudity from unacceptable nudity [13]. A potential way to bootstrap our results is to argue that if the dimension of the linear separators used grows slowly with time as $d(t)$, the rate of new instances that must be examined by experts when running A_{CAL} is roughly $d(t)/t$. This can be used as a crude approximation for estimating, for example, the rate at which new experts should be hired, or conversely, the highest rate at which new features can be added without overwhelming the system. Using these ideas to study a more realistic model of evolving norms is a topic for future research.

[4] Indeed, one can view the process of legal arguments as a means of passing this computational burden on the lawyers, who need to find and exhibit the relevant precedents.

Next, our assumption that the cost of asking for advice is constant seems unlikely to hold in practice. Experts are typically few in number, and busy; thus, while they can accommodate a few requests for advice, the cost is likely convex in the number of requests. As a consequence, moderators cannot make too many requests for advice in a given interval without incurring prohibitive cost. This implies that, if the dimension of the hypothesis class is fixed, the effective rate at which the system can classify new instances is capped, but it will increase with time: initially, almost all instances will require attention from experts, while in later rounds t, only a fraction of instances (proportional to $1/t$) will require such attention, so the system as a whole will be capable of processing more instances. Studying this rate and how it depends on the cost function is another topic for future work.

Finally, an important topic, which is out of scope of the present paper, is strategic behavior on the part of decision-makers who would like to nudge the system as a whole towards a particular hypothesis. In this paper, we have mostly assumed that the expert has a particular hypothesis in mind, and the goal of the moderator is to learn this hypothesis. However, it is arguably more realistic to assume that the experts do not have the full hypothesis in mind up front; rather, it is constructed instance by instance. Under this assumption, the particular instances that are presented to experts might substantially impact the hypothesis that is eventually adopted. Given a particular model of how the experts resolve uncertainty, decision-makers can choose to ask for labels for only those cases which are likely to resolve the uncertainty in the eventual hypothesis in a favorable direction. This is especially important if the way the experts label cases varies over time. A good example of this is the US Supreme Court; in some years the Court leans liberal, and in others it leans conservative. To the extent to which litigants and lower courts can affect which cases the Supreme court hears, they are likely to send more cases to the court in years where the Court is biased towards their individual viewpoints. Of course, in the real-world legal setting, this perspective is complicated by the fact that what we call the "moderator" comprises many individual actors, including judges, jury members, the persons litigating the case, and even outside advocates. Our model is clearly a considerable oversimplification for this setting, but nevertheless, we feel that one of the most useful aspects of our work is in contributing a formal model for studying such questions. We leave further exploration of strategic behavior to future work.

5 COMPUTATIONAL COMPLEXITY AND PROOFS

5.1 Computational complexity

It is evident that A_{CAL} runs in $O(1)$ time per instance X_t, independent of t, in the setting of Theorem 1. A_{CAL} simply maintains A_t^k and B_t^k as in the proof of Theorem 1, and asks for a label if $X_t \in (A_t^k, B_t^k)$.

By contrast, it is not at all obvious that A_{CAL} runs in expected per-instance time independent of t in the setting of Theorem 3. Nevertheless, A_{CAL} can be implemented to run in amortized expected per-instance time $O(poly(d))$ independent of t, as Proposition 5 below shows. Our implementation is given in Algorithm 2 (CAL). CAL is a bona fide *implementation of* A_{CAL}; that is, if X_t is determined, then CAL outputs the inferred label; otherwise CAL asks for the label. It depends on checks of the form $x \in \text{cone}(S)$, where $x \in \mathbb{R}^d$, S is a finite set of points in $\mathbb{R}^d$, and cone(S) denotes the set of all nonnegative linear combinations of points in S, or alternatively, the cone spanned by S. These checks can be implemented efficiently using linear programming. Specifically, the linear program P given by

$$\begin{aligned} &\min c^T y \text{ s.t.} \\ &Ay = x, \\ &y \geq 0, \end{aligned} \tag{2}$$

where c is the zero vector and A is the matrix whose columns are the points in S, is feasible if and only if $x \in \text{cone}(S)$. Whether P is feasible can be checked in time polynomial in d and $|S|$.

To explain CAL, we need to define the notion of *equivalent positive sample.* For $i \in [t]$, define $Z_i = Y_i X_i$. It is easy to see that, for all $h \in \mathcal{H}$ and $i \in [t]$, $h(X_i) = Y_i$ iff $h(Z_i) = 1$. Thus, if we define the *equivalent positive sample* $\mathcal{Z}_t = \{(Z_1, 1), \dots, (Z_{t-1}, 1)\}$, we have $\mathcal{H}_{\mathcal{S}_t} = \mathcal{H}_{\mathcal{Z}_t}$; that is, $\mathcal{S}_t$ and $\mathcal{Z}_t$ are compatible with the same set of hypotheses. It follows that

Claim 1. *$\mathcal{S}_t \twoheadrightarrow X_t$ iff $\mathcal{Z}_t \twoheadrightarrow Z_t$.*

This fact permits us to think of $\mathcal{Z}_t$ as equivalent to the original labeled data $\mathcal{S}_t$; hence the name. The point of introducing the equivalent positive sample is that there is a simple characterization of when $\mathcal{Z}_t \twoheadrightarrow X_t$.

Lemma 4. *$\mathcal{Z}_t \twoheadrightarrow X_t$ iff either (1) $X_t \in \text{cone}(\{Z_1, \dots, Z_{t-1}\})$ or (2) $-X_t \in \text{cone}(\{Z_1, \dots, Z_{t-1}\})$. Moreover, if (1) holds, then the inferred label is $Y_t = 1$; if (2) holds, then it is $Y_t = -1$.*

Proof. Suppose $\mathcal{Z}_t \twoheadrightarrow X_t$. Since either $Z_t = X_t$ or $Z_t = -X_t$, by Lemma 8, $\mathcal{Z}_t \twoheadrightarrow Z_t$. Hence $Z_t \in \text{cone}(\{Z_1, \dots, Z_{t-1}\})$ (with inferred label 1). If $X_t = Z_t$, then X_t is also in the cone, and its inferred label is 1. If $-X_t = Z_t$, then $-X_t$ is in the cone, and its inferred label is 1; so the inferred label of X_t is -1. Conversely, suppose $X_t \in \text{cone}(\{Z_1, \dots, Z_{t-1}\})$. (The case $-X_t \in \text{cone}(\{Z_1, \dots, Z_{t-1}\})$ is similar.) Then $\mathcal{Z}_t \twoheadrightarrow X_t$, and the inferred label of X_t is 1, by the argument for the "only if" direction of Lemma 8. □

CAL works by maintaining a set $\mathcal{B}$ such that $\text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \dots, Z_{t-1}\})$, as the following lemma shows.

Lemma 5. *At the beginning of each iteration t of CAL, $\text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \dots, Z_{t-1}\})$.*

Proof. We proceed by induction on t. Clearly the result holds for $t = 1$. Suppose it is true for $t = k$. Let $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$ be the values of $\mathcal{B}$ at the beginning of iterations k and $k+1$ respectively. There are three cases corresponding to the three conditional branches. For the "if" and "else if" cases, if $X_k \in \text{cone}(\mathcal{B}_k)$ or $-X_k \in \text{cone}(\mathcal{B}_k)$, then $Z_k \in \text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \dots, Z_{k-1}\})$. Since $\mathcal{B}$ is not changed in the "if" or "else if" cases, we have $\mathcal{B}_{k+1} = \mathcal{B}_k = \text{cone}(\{Z_1, \dots, Z_{k-1}\}) = \text{cone}(\{Z_1, \dots, Z_k\})$. For the "else" case, let R be the set of vectors removed by RRP. Since R only contains vectors that are in the cone spanned by the remaining vectors, we have $\text{cone}(\mathcal{B}_k \setminus R) = \text{cone}(\mathcal{B}_k)$. Then we have $\mathcal{B}_{k+1} = \mathcal{B}_k \setminus R \cup \{Z_k\}$, so

$$\text{cone}(\mathcal{B}_{k+1}) = \text{cone}(\text{cone}((\mathcal{B}_k \setminus R) \cup Z_k)) = \text{cone}(\text{cone}(\{Z_1, \dots, Z_{k-1}\} \cup Z_k))) = \text{cone}(\{Z_1, \dots, Z_k\}).$$

□

Since $\text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \dots, Z_{t-1}\})$, it is immediate from Lemma 4 that the action taken by CAL in each iteration corresponds to the action chosen by A_{CAL}, so CAL is an implementation of A_{CAL}. This is formalized in the following proposition.

Proposition 4. *CAL is an implementation of A_{CAL}.*

Moreover, CAL runs in amortized expected time per instance independent of t, as the following proposition shows. This is a consequence of Lemma 7, which implies that the expected size of $\mathcal{B}$ is bounded by a constant independent of t; see Subsection 5.2 for details.

Proposition 5. *CAL runs in amortized expected time $O(d^{3.6})$ per instance X_t.*

Proof. First we need a lemma concerning the subroutine RRP.

Lemma 6. *If RRP is called in iteration t of CAL, then immediately after RRP returns, $\mathcal{B} = \{Z_i \mid i \in [t], \mathcal{Z}_i \not\twoheadrightarrow Z_i\}$.*

Proof. Lemma 5 states that at the start of iteration t, $\text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \ldots, Z_{t-1}\})$. It is easy to see that just before RRP is called, since Z_t has been added if it is not in the cone, $\text{cone}(\mathcal{B}) = \text{cone}(\{Z_1, \ldots, Z_t\})$. Each iteration of RRP preserves this property. Hence after RRP, $\mathcal{B} \supseteq \{Z_i \mid i \in [t], \mathcal{Z}_i \not\twoheadrightarrow Z_i\}$; that is, $\mathcal{B}$ is a superset of the edges of the cone. However, after RRP, all points Z_i that are determined by the other points (that is, $\mathcal{Z}_i \twoheadrightarrow Z_i$) have been removed. Hence $\mathcal{B} = \{Z_i \mid i \in [t], \mathcal{Z}_i \not\twoheadrightarrow Z_i\}$, as claimed. □

As in the proof of Theorem 3, without loss of generality, suppose that the correct hypothesis is $h^* = [1, 0, \ldots, 0]$. Hence, as in that theorem, the Z_i are i.i.d. with distribution $\mathsf{Unif}\, S^d_+$.

Let $T_1, T_2, \ldots$ be the iterations of CAL where RRP is called. By Lemma 6, at the end of each iteration T_i, we have $\mathcal{B} = \{Z_i \mid i \in [T_i], \mathcal{Z}_i \not\twoheadrightarrow i\}$. Furthermore, RRP is run every time $\mathcal{B}$ doubles in size. So for all iterations $T_i < t \leq T_{i+1}$, letting $\mathcal{B}_t$ be the value of $\mathcal{B}$ at the beginning of iteration t, we have $|\mathcal{B}_t| \leq 2|\{Z_i \mid i \in [T_i], \mathcal{Z}_i \not\twoheadrightarrow i\}|$. Now $\{Z_i \mid i \in [T_i], \mathcal{Z}_i \not\twoheadrightarrow i\}$ are the points corresponding to edges of $\text{cone}(\{Z_1, \ldots, Z_{T_i}\})$. So Lemma 7 (see Subsection 5.2) implies that $\lim_{d\to\infty} \lim_{t\to\infty} \mathbb{E}\,|\mathcal{B}_t| = 2kd^{3/2}$ for an absolute constant k. Thus for sufficiently large d, for all $T_i < t \leq T_{i+1}$, $\mathbb{E}\,|\mathcal{B}_t| = \Theta(d^{3/2})$.

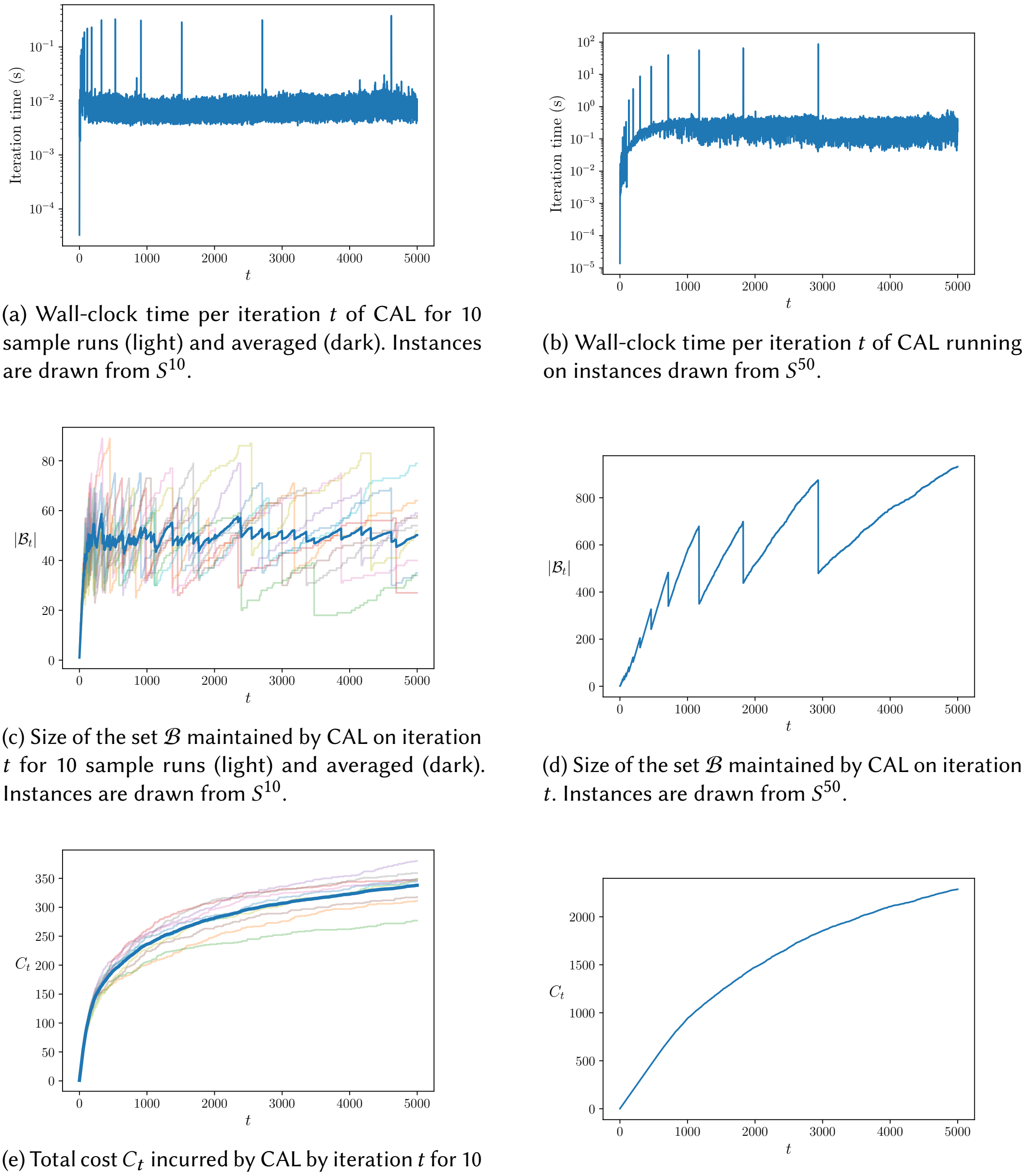
Each iteration t, $T_i < t < T_{i+1}$, consists of two checks of the form $x \in \text{cone}(\mathcal{B})$, plus some operations that are obviously $O(1)$. The iteration T_{i+1} consists of these two checks, along with a call to RRP, which consists of $|\mathcal{B}_{T_i}|$ checks, plus some $O(1)$ operations. Suppose that $T_{i+1} - T_i = m$. Then the expected running time $\mathbb{E}\,\tau$ of the m iterations $t = T_i + 1, \ldots, T_{i+1}$ is $O(m) + 2mc + |\mathcal{B}_{T_i}|c$, where c is the maximum expected running time of any of the checks. Since $\mathcal{B}$ doubles in size from T_i to T_{i+1}, but increases by at most 1 in any iteration, $|\mathcal{B}_{t_i}| < 2m$. Hence $\mathbb{E}\,\tau \leq O(m) + 4mc$.

The checks $x \in \text{cone}(\mathcal{B})$ are implemented as checking feasibility of the linear program (2). This linear program has d constraints and $|\mathcal{B}_t|$ variables. It is known that linear programs of the form 2 with no redundant constraints can be solved in time $O(n^\omega \log(n))$, where n is the number of variables, and $\omega < 2.39$ is the exponent of matrix multiplication [28]. The constraints of the linear program (2) are almost surely not redundant, because each constraint is a random equality constraint. Hence (2) can be solved (and its feasibility checked) in $O((d^{3/2})^{2.39}) \subset O(d^{3.6})$ time. This means $c \in O(d^{3.6})$. Hence $\mathbb{E}\,\tau = mO(d^{3.6})$. Dividing both sides by m, this implies that the amortized expected running time per iteration $\mathbb{E}\,\tau/m$ is $O(d^{3.6})$, as claimed. □

To demonstrate that CAL has not only good asymptotic runtime and cost bounds, but also good runtime and cost performance for realistic time horizons and values of d, we implemented CAL and tested it on simulated instances. See the supplementary material for the code and collected data. Figure 2 shows data from running CAL on instances drawn from $\mathsf{Unif}\, S^{10}$ (left) and $\mathsf{Unif}\, S^{50}$ (right). The top column shows the time per instance in seconds. For $d = 10$, the average time per instance is 0.0082 seconds; for $d = 50$, it is 0.32 seconds. Importantly, the time per instance is roughly constant in t for $t > 100$ ($d = 10$) and $t > 1000$ ($d = 50$). The “spikes” are calls to RRP. The middle column shows the size of $\mathcal{B}_t$, which levels off in lockstep with the time per instance, and does not vary too much between runs. The last column shows the cost C_t, which is just the total number of label requests.

5.2 Asking for Advice in d-Dimensions

To prove Theorem 3, we need the following lemma, which is essentially implicit in [19]. Let S^d_+ be the upper half-sphere $S^d_+ = \{x \in S^d \mid x[1] > 0\}$.



(a) Wall-clock time per iteration t of CAL for 10 sample runs (light) and averaged (dark). Instances are drawn from S^{10}.

(b) Wall-clock time per iteration t of CAL running on instances drawn from S^{50}.

(c) Size of the set $\mathcal{B}$ maintained by CAL on iteration t for 10 sample runs (light) and averaged (dark). Instances are drawn from S^{10}.

(d) Size of the set $\mathcal{B}$ maintained by CAL on iteration t. Instances are drawn from S^{50}.

(e) Total cost C_t incurred by CAL by iteration t for 10 sample runs (light) and averaged (dark). Instances are drawn from S^{10}.

(f) Total cost C_t incurred by CAL by iteration t. Instances are drawn from S^{50}.

Fig. 2. Data from runs of CAL on instances from Unif S^d.

Lemma 7. *Let $Z_1, \ldots, Z_t$ be drawn i.i.d. from the uniform distribution on S^d_+. Let $C_t = \text{cone}(\{Z_1, \ldots, Z_t\})$. Let N_t be the number of edges of C_t. Then*

$$\lim_{d\to\infty} \lim_{t\to\infty} \mathbb{E}\, N_t / d^{3/2} = \sqrt{\frac{2\pi}{3}}.$$

Algorithm 1 RRP

```
Input: $\mathcal{B}$, a finite set of points in $\mathbb{R}^d$.
for $Z \in \mathcal{B}$ do
    if $Z \in \text{cone}(\mathcal{B} - \{Z\})$ then
        $\mathcal{B} \leftarrow \mathcal{B} \setminus \{Z\}$
return $\mathcal{B}$
```

Algorithm 2 CAL

```
$\mathcal{B} \leftarrow \emptyset$
$N = 0$                                        ▷ $|\mathcal{B}|$ after last call to RRP.
for $t = 1, 2, \ldots$ do
    if $X_t \in \text{cone}(\mathcal{B})$ then
        Classify $X_t$ as positive.
    else if $-X_t \in \text{cone}(\mathcal{B})$ then
        Classify $X_t$ as negative.
    else
        Ask for the label $Y_t$.
        Compute $Z_t = Y_t X_t$.
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{Z_t\}$
        if $|\mathcal{B}| \geq 2N$ then
            $\mathcal{B} \leftarrow \text{RRP}(\mathcal{B})$
            $N \leftarrow |\mathcal{B}|$.
```

Proof. Equivalently, N_t is the number of vertices of the random *spherical polytope* $C_t \cap S^d_+$ (see Section 2 of [19]). Several recent papers have investigated the properties of this polytope [6, 19, 20]. In particular, the following theorem was shown regarding $f_k(C_t \cap S^d_+)$, the number of k-faces of $C_t \cap S^d_+$.

Before stating the theorem, we need some notation. Define $A[n, m]$ as follows: Given a polynomial $P(x)$ (a formal power series in positive and negative powers of x), let $[x^k]P(x)$ be the coefficient of x^k in $P(x)$. Let

$$Q_n(x) = \Pi_{j \in \{1,\ldots,n-1\}, j \not\equiv n (\text{mod } 2)}(1 + j^2x^2).$$

Then

$$A[n, m] = \begin{cases} [x^m]Q_n(x) & \text{if m is even,} \\ [x^m](\tanh(\frac{\pi}{2x}) \cdot Q_n(x)), & \text{if m is odd and n is even,} \\ [x^m](\coth(\frac{\pi}{2x}) \cdot Q_n(x)), & \text{if m is odd and n is odd.} \end{cases}$$

We remark that although the appearance of tanh and coth in these formulas is a little surprising, the proof does not use any special properties of these functions. Rather, the second and the third cases follow from the first via the Dehn-Sommerville equations relating the numbers of faces of different dimensions of a simplicial polytope, and the tanh and coth factors conveniently summarize the coefficients that arise when these equations are solved. See [19] for more details.

Proposition 6 (Theorem 2.1 of [19]).

$$\lim_{t \to \infty} \mathbb{E}\, f_k(C_t \cap S^d_+) = \frac{\pi^{k+1}}{(k+1)!} A[d, k+1].$$

We are interested in the special case of Proposition 6 where $k = 0$, corresponding to vertices of $C_t \cap S^d_+$. Recall that $\mathbb{E}\, N_t = \mathbb{E}\, f_0(C^t \cap S^d_+)$. By Proposition 6 with $k = 0$, we have

$$\lim_{t\to\infty} N_t = \lim_{t\to\infty} \mathbb{E}\, f_0(C^t \cap S^d_+) = \pi A[d,1]. \tag{3}$$

Zakhar Kabluchko has proved the following claim about the asymptotic behavior of the $A[n,m]$ (private correspondence, Jan 31, 2021).

Claim 2. *For all $m \geq 1$,*

$$\lim_{n\to\infty} \frac{A[n,m]}{n^{3m/2}} = \frac{1}{6^{m/2}\Gamma((m/2)+1)},$$

where $\Gamma(x) = \int_0^\infty y^{x-1}e^{-y}dy$ denotes the gamma function. In the special case of $m = 1$, Claim 2 becomes

$$\lim_{n\to\infty} \frac{A[n,1]}{n^{3/2}} = \frac{1}{\sqrt{6}\sqrt{\pi}/2} = \sqrt{\frac{2}{3\pi}}. \tag{4}$$

Hence dividing both sides of Equation (3) by $d^{3/2}$ and taking the limit as $d \to \infty$ yields

$$\lim_{d\to\infty}\lim_{t\to\infty} \mathbb{E}\, N_t/d^{3/2} = \pi\sqrt{\frac{2}{3\pi}} = \sqrt{\frac{2\pi}{3}},$$

as claimed. □

Theorem 3. If $\mathcal{X} = \mathbb{R}^d$ and the X_t are drawn i.i.d. from a (possibly unknown) multivariate Gaussian $N(0,\Sigma)$ with mean zero, then for all hypotheses $h \in \mathcal{H}$, A_{CAL} incurs expected cost $\Theta(d^{3/2}\log T)$.

Proof. We first prove the claim for a different distribution, namely $\mathsf{Unif}\, S^d$. Since $\mathsf{Unif}\, S^d$ is spherically symmetric, the expected cost is the same for all hypotheses. Without loss of generality, suppose the correct hypothesis is $h^* = [1,0,\ldots,0]$. That is, $Y_t = 1$ if $X_t[1] > 0$ and $Y_t = 0$ if $X_t[1] < 0$. (We will neglect the zero-probability case $X_t[1] = 0$.)

Since A_{CAL} guesses only when X_t is determined, it does not make any wrong classifications. Thus, the cost C_T it incurs is simply the number of times it queries:

$$\mathbb{E}\, C_T = \sum_{i=1}^{T} \mathbb{E}\, c_t = \sum_{i=1}^{T} \mathbb{E}\, \mathbb{1}_{\{\hat{Y}_t=0\}} = \sum_{i=1}^{T} P(\hat{Y}_t = 0).$$

Thus, the goal is to bound $P(\hat{Y}_t = 0)$, the probability that the naive algorithm queries X_t. Recall that the naive algorithm queries X_t exactly when X_t is not determined, that is, when $\mathcal{S}_t := \{(X_1,Y_1),\ldots,(X_{t-1},Y_{t-1})\} \not\twoheadrightarrow X_t$, and that $\mathcal{S}_t \twoheadrightarrow X_t$ iff $\mathcal{Z}_t \twoheadrightarrow Z_t$. The following lemma characterizes when $\mathcal{Z}_t \twoheadrightarrow Z_t$.

Lemma 8. *$\mathcal{Z}_t \twoheadrightarrow Z_t$ iff $Z_t \in \mathsf{cone}(\{Z_1,\ldots,Z_{t-1}\})$.*

Proof. The "only if" direction is straightforward. Indeed, suppose there are nonnegative numbers a_j such that $Z_i = a_1Z_1 + \cdots + a_{t-1}Z_{t-1}$. It suffices to show that for all $h = \mathbb{1}_{\{x\cdot w\geq 0\}} \in \mathcal{H}_{\mathcal{Z}_i}$, we have $h(Z_i) = 1$, that is, $Z_i \cdot w \geq 0$. We have $Z_i \cdot w = a_1(Z_1\dot{w}) + \cdots + a_{t-1}(Z_{t-1}\dot{w})$. But since $h \in \mathcal{H}_{\mathcal{Z}_i}$, $Z_i\dot{w} \geq 0$ for all $i \in [t-1]$. Hence $Z_t \cdot w \geq 0$. The "if" direction follows from the well-known Farkas Lemma (see, for example, [24, Lemma 6.1]), which states that given a $m \times n$ real matrix A and $c \in R^n$, exactly one of the following two systems has a solution:

(1) $Ad \leq 0$ and $c^Td > 0$
(2) $A^Ty = c$ and $y \geq 0$.

Let A be the $d \times (t-1)$ matrix whose columns are $Z_1, \dots, Z_{t-1}$, and let $c = Z_i$. The Farkas Lemma then implies that if Z_t is not a nonnegative linear combination of $Z_1, \dots, Z_{t-1}$ (that is, the system 2 has no solutions) then there is a vector d such that $Ad \leq 0$ and $c^T d > 0$. But then the hypothesis $h = \mathbb{1}_{\{x \cdot -d \geq 0\}}$ satisfies $h(Z_1) = \cdots = h(Z_{t-1}) = 1$, but $h(Z_t) = 0$. Since $h^*(Z_1) = \cdots = h^*(Z_t) = 1$, $\mathcal{Z}_i$ does not determine Z_i. □

Since the Z_i for $i \in [t]$ are i.i.d., this lemma implies the following claim.

Claim 3. *For all $i \in [t]$ $\mathcal{Z}_i \twoheadrightarrow Z_i$ iff $\mathcal{Z}_i \in \operatorname{cone}(\{Z_1, \dots, Z_{i-1}, Z_{i+1}, \dots, Z_t\})$.*

It follows from our general upper bound (Corollary 1) that

$$\begin{aligned}\mathbb{E}\, C_T &= \sum_{t=1}^{T} \mathbb{E}\, |\mathcal{B}_t|/t \\ &= \sum_{t=1}^{T} \mathbb{E}\, |\{S_i \mid i \in [t], \mathcal{S}_i \not\twoheadrightarrow S_i\}| \\ &= \sum_{t=1}^{T} \mathbb{E}\, |\{Z_i \mid i \in [t], \mathcal{Z}_i \not\twoheadrightarrow Z_i\}|,\end{aligned}$$

where the last equality follows from Claim 1.

It is easy to see that the Z_i are i.i.d. with distribution $\mathsf{Unif}\, S^d_+$. Let $C_t = \operatorname{cone}(\{Z_1, \dots, Z_t\})$. By Claim 3, the random variable $N_t := |\{Z_i \mid i \in [t], \mathcal{Z}_i \not\twoheadrightarrow Z_i\}|$ is the number of edges of C_t (each edge corresponds to an instance Z_i not contained in the cone of the other instances, and vice versa). By Lemma 7, we have

$$\lim_{d \to \infty} \lim_{t \to \infty} \mathbb{E}\, N_t / d^{3/2} = \pi \sqrt{\frac{2}{3\pi}} = \sqrt{\frac{2\pi}{3}}.$$

That is, for sufficiently large d, we have $\mathbb{E}\, N_T \in \Theta(d^{3/2})$. It follows that

$$\mathbb{E}\, C_T = \sum_{t=1}^{T} \mathbb{E}\, N_t / T \in \Theta(d^{3/2} \log T),$$

as claimed.

Finally, we generalize to the case of multivariate Gaussians. Suppose $\mathcal{D}$ is a multivariate Gaussian with mean 0 and covariance Σ. For simplicity, consider the case where Σ is invertible; if Σ is not invertible (so that $\mathcal{D}$ is supported on only a subspace of $\mathbb{R}^d$), the argument is similar, except that the definition of $T(x)$ must be modified to map the X_t to the unit hypersphere in the subspace corresponding to the support of $\mathcal{D}$.

Let $T(x) = \frac{\Sigma^{-1} X_t}{\|\Sigma^{-1} X_t\|}$. Take $X'_t = T(X_t)$. Further take the target hypothesis to be $h' = T(h)$. It is easy to see that for all $x \in X$ and all $g \in \mathcal{H}$, $T(g)(x) = g(T(x))$. Hence the following are equivalent:

- $\{(X_1, h(X_1)), \dots, (X_{t-1}, h(X_{t-1}))\} \twoheadrightarrow X_t$,
- $\{(X'_1, h'(X_1)), \dots, (X'_{t-1}, h'(X'_{t-1}))\} \twoheadrightarrow X'_t$.

Thus, A_{CAL} asks for the label of X_t given target hypothesis h iff A_{CAL} asks for the label of X'_t given target hypothesis h'. Since the cost incurred by A_{CAL} by time t is simply the number of labels requested on or before round t, the cost C_t that A_{CAL} incurs on hypothesis h and arrivals X_t is the same as the cost that A_{CAL} incurs on hypothesis h' and arrivals X'_t. But it is easy to see that the X'_t are i.i.d. with distribution $\mathsf{Unif}\, S^d$. Thus the previous argument applies and $EC_t = \Theta(d^{3/2} \log T)$. □

REFERENCES

[1] S. Adler. 2018. Post No Evil. Radiolab podcast; https://www.wnycstudios.org/podcasts/radiolab/articles/post-no-evil.

[2] A. Alkhatib and M. Bernstein. 2019. Street-level algorithms: A theory at the gaps between policy and decisions. In *Proc 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[3] M.-F. Balcan, A. Beygelzimer, and J. Langford. 2009. Agnostic active learning. *J. Comput. System Sci.* 75, 1 (2009), 78–89.

[4] M.-F. Balcan, A. Broder, and T. Zhang. 2007. Margin based active learning. In *20th Annual Conference on Learning Theory (COLT 2007)*. 35–50.

[5] M.-F. Balcan and P. Long. 2013. Active and passive learning of linear separators under log-concave distributions. In *26 Annual Conference on Learning Theory (COLT 2013)*. 288–316.

[6] I. Bárány, D. Hug, M. Reitzner, and R. Schneider. 2017. Random points in halfspheres. *Random Structures & Algorithms* 50, 1 (2017), 3–22.

[7] C.-K. Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers* EC-6, 4 (1957), 247–254.

[8] D. Cohn, L. Atlas, and R. Ladner. 1994. Improving generalization with active learning. *Machine Learning* 15, 2 (1994), 201–221.

[9] S. Dasgupta. 2005. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18 (NIPS 2005)*. 235–242.

[10] S. Dasgupta, D. J. Hsu, and C. Monteleoni. 2007. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. 353–360.

[11] S. Dasgupta, A. T. Kalai, and C. Monteleoni. 2005. Analysis of perceptron-based active learning. In *International Conference on Computational Learning Theory*. 249–263.

[12] R. El-Yaniv and Y. Wiener. 2012. Active Learning via Perfect Selective Classification. *Journal of Machine Learning Research* 13, 9 (2012).

[13] Facebook. 2021. Community Standards. https://www.facebook.com/communitystandards/. Accessed Feb. 9, 2021.

[14] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine Learning* 28, 2 (1997), 133–168.

[15] S. Hanneke. 2007. A bound on the label complexity of agnostic active learning. In *Proc. 24th International Conference on Machine learning*. 353–360.

[16] D. P. Helmbold, N. Littlestone, and P. M. Long. 2000. Apple tasting. *Information and Computation* 161, 2 (2000), 85–139.

[17] Y. Ibrahim. 2010. The breastfeeding controversy and Facebook: Politicization of image, privacy and protest. *International Journal of E-Politics (IJEP)* 1, 2 (2010), 16–28.

[18] M. Isaac and K. Browning. Nov. 17, 2020. Lawmakers drill down on how Facebook and Twitter moderate content. *The New York Times* (Nov. 17, 2020).

[19] Zakhar Kabluchko. 2020. Expected f-vector of the Poisson zero polytope and random convex hulls in the half-sphere. *Mathematika* 66, 4 (2020), 1028–1053.

[20] Z. Kabluchko, A. Marynych, D. Temesvari, and C. Thäle. 2019. Cones generated by random points on half-spheres and convex hulls of Poisson point processes. *Probability Theory and Related Fields* 175, 3 (2019), 1021–1061.

[21] Michael Lipsky. 2010. *Street-Level Bureaucracy: Dilemmas of the Individual in Public Service.* Russell Sage Foundation, New York.

[22] P. M Long. 1995. On the sample complexity of PAC learning half-spaces against the uniform distribution. *IEEE Transactions on Neural Networks* 6, 6 (1995), 1556–1559.

[23] B. J. Murrill. 2018. *The Supreme Court's Overruling of Constitutional Precedent.* Technical Report R45319. U.S. Congressional Research Service.

[24] D. J. Rader. 2010. *Deterministic operations research: models and methods in linear optimization.* John Wiley & Sons.

[25] F. Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (1958), 386–408.

[26] S. Shalev-Shwartz and S. Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, New York, NY.

[27] L. G. Valiant. 1984. A theory of the learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.

[28] Jan van den Brand. 2020. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 259–278.

A DETAILS OF LABEL COMPLEXITIES OF PREVIOUSLY PROPOSED ALGORITHMS

In [11, Theorem 3] it was proved that if the AMP algorithm is presented with

$$O(\frac{d}{\epsilon}\log(\frac{1}{\epsilon\delta})(\log d/\delta+\log\log\frac{1}{\epsilon}))$$

samples, with probability $1-\delta$, it will misclassify at most an ϵ fraction of the samples, and the same for labels requested. That is, the number of misclassifications and the number labels requested are both bounded by

$$O(d\log(\frac{1}{\epsilon\delta})(\log d/\delta+\log\log\frac{1}{\epsilon})).$$

It is easy to see that this implies that the cost C_T incurred by the AMP algorithm is $O(d\log T\log\frac{1}{\delta}(\log\frac{d}{\delta}+\log\log T))$. This is almost the optimal $O(d\log T)$ cost, but is higher by multiplicative $\log\log T$ and $\log d$ factors. It is the $\log\log T$ factor that we show A_{CAL} improves on.

To compare things another way, after processing $T=O(d^{3/2}/\epsilon)$ instances, A_{CAL} makes zero mistakes, makes $\Theta(d^{3/2}\log\frac{d^{3/2}}{\epsilon})$ label requests in expectation, and incurs expected cost ϵ.

Moving to the "margin-based" algorithms presented in [4], one of these algorithms (Margin-Based Active Learning; Procedure 2 of [4] with parameters as in Theorem 2 of [4]) makes just as few label queries, up to factors of $\log\log 1/\epsilon$. More precisely, the label query bound given in Theorem 2 simplifies to $O(\log(1/\epsilon)\sqrt{\log\log(1/\epsilon))}(d\log\log\log 1/\epsilon+\log\log\epsilon-\log\delta)$. This algorithm does make mistakes, and no mistake bound is given, so it is not possible to make a direct comparison with the AMP algorithm. Another algorithm of [4] (Margin-Based Active Learning with parameters as in Theorem 1 of [4]) makes no mistakes, but its label query bound depends on dimension as $d^{3/2}$ (that is, it has a $d^{3/2}$ factor). The label query bound given in Theorem 1 of [4] simplifies to $O(\log(1/\epsilon)d^{1/2}(d\log d+\log(\log 1/\epsilon)-\log(1/\delta)))$. This is consistent with the analysis of this paper, where the algorithm which is *guaranteed* to make no mistakes with probability 1 has a label complexity depending on $\Theta(d^{3/2})$. The question of whether every algorithm which makes no mistakes with high probability in this setting has label complexity depending on d as $\Omega(d^{3/2})$ is still open.

Indeed, for X_t drawn from any isotropic log-concave distribution D, there are parameters (see Theorem 5 of [5]) such that Margin-based Active Learning has label complexity $O((d+\log(1/\delta)+\log\log 1/\epsilon)\log(1/\epsilon))$. Again, the expected label complexity of A_{CAL} improves on this by a $\log\log 1/\epsilon$ factor, although its dependence on d is worse.

B ADDITIONAL PROOFS

We now give the proof of part (2) of Theorem 1. For the reader's convenience, we repeat the statement of the theorem.

Theorem 1. Given a set S, let $\mathbb{1}_S$ be the indicator function of S. If $\mathcal{X}=[0,1]$, the instances X_t for $t\in[T]$ are drawn i.i.d. from a continuous distribution $\mathcal{D}$ over $\mathcal{X}$, and $\mathcal{H}=\{\mathbb{1}_{\{x\geq k\}}\mid k\in[0,1]\}$, then for all hypotheses $h\in\mathcal{H}$, the following claims hold:

(1) A_{CAL} has expected cost $O(\log T)$.
(2) For all $\delta>0$, with probability $1-\delta$, A_{CAL} incurs cost $O(\log T+\log(1/\delta))$.

Proof of (2). Fix a threshold $k\in[0,1]$. Consider running A_{CAL} for infinitely many rounds. For $i\geq 0$, let Q_i be the time of the i^{th} label query. Let $Q_0=0$. The total cost incurred by A_{CAL} by time T is simply the total number of queries made on or before time T, that is,

$$C_T=\max_{Q_j\leq T} j.$$

Clearly, for all $j \geq 0$, we have $P(C_T \leq j) = P(Q_j \geq T)$.

Fix $\delta > 0$. It suffices to show that there is some c such that for $j = c\log(T/\delta)$, we have $P(Q_j \geq T) \leq \delta$.

Define A_t^k, B_t^k as in the proof of Theorem 1(1). Given a set $\{X_1, \dots, X_t\}$ of instances, whether A_{CAL} queries depends only on the relative order of these instances and the threshold k. We can thus, without loss of generality, assume that the distribution $\mathcal{D}$ is uniform; if it is not, we make it uniform by taking $X_t \mapsto \mathcal{F}(X_t)$ and setting the new threshold to $F(k)$, where F is the cumulative distribution function of $\mathcal{D}$. This leaves the relative ordering of the instances unchanged. Recall that A_{CAL} queries on round t iff $X_t \in (A_t^k, B_t^k)$. This occurs with probability $M_t := B_t^k - A_t^k$. If A_{CAL} does not query on round t, then $M_{t+1} = M_t$.

Therefore, $Q_j - Q_{j-1}$, the number of rounds between query j and query $j-1$, is distributed as $Geo(M_{Q_{j-1}+1})$, where $Geo(p)$ is the geometric distribution with parameter p (the number of independent coin flips required to get a heads if heads has probability p). Let $M = M_{Q_{j-1}+1}$. Thus we have

$$Pr(C_T \leq j) = Pr(Q_j \geq T) \geq Pr(Q_j - Q_{j-1} \geq T) = Pr(Geo(M) \geq T) = (1-M)^{T-1}. \tag{5}$$

Next we control M. We have $M_1 = 1$. For $i \geq 1$, let N_i be the random variable that is 1 if $X_{Q_i} \in (A_{Q_i}^k + (A_{Q_i}^k + B_{Q_i}^k)/3, A_{Q_i}^k + 2(A_{Q_i}^k + B_{Q_i}^k)/3)$, and 0 otherwise. That is, $N_i = 1$ if query i falls in the middle third of the region of uncertainty. If $N_i = 1$, then query i reduces the length of the region of uncertainty by at least a factor of $2/3$. Thus, defining $N = \sum_{i=1}^{j} N_i$, we have $M \leq (2/3)^N$. Combining this with (5), we have

$$Pr(Q_j \geq T) \geq (1-(2/3)^N)^{T-1} \geq 1-(T-1)(2/3)^N. \tag{6}$$

Since the instances X_t are uniformly distributed on $[0,1]$, conditioned on A_{CAL} querying in round t, X_t is uniformly distributed on (A_t^k, B_t^k). Hence, the distribution of X_{Q_i} is uniform on $(A_{Q_i}^k, B_{Q_i}^k)$. It follows that $Pr(N_i = 1) = 1/3$. Furthermore, since the X_i are independent, the events $N_i = 1$ are also independent. So the N_i are i.i.d. Bernouilli random variables with parameter $1/3$, and a standard Chernoff bound yields

$$Pr(N \leq (1-\epsilon)j/3) \leq \exp\big(\frac{-\epsilon^2 j}{9}\big). \tag{7}$$

Using the law of total probability and combining (6) and (7), we get

$$\begin{aligned} Pr(Q_j \geq T) &\geq Pr(N \geq (1-\epsilon)j/3)Pr(Q_j \geq T \mid N \geq (1-\epsilon)j/3) \\ &\geq \left(1-\exp\big(\frac{-\epsilon^2 j}{9}\big)\right)\left(1-(T-1)(2/3)^{(1-\epsilon)j/3}\right) \\ &\geq 1-\exp\big(\frac{\epsilon^2 j}{9}\big)-(T-1)(2/3)^{(1-\epsilon)j/3}. \end{aligned}$$

Choosing $\epsilon = 1/2$ and $j = 36\ln(T/\delta)$, we obtain

$$\begin{aligned} Pr(Q_j \geq T) &\geq 1-\exp(-\ln(T/\delta))-(T-1)(\exp(\ln(2/3)))^{36\ln(T/\delta)/6} \\ &\geq 1-\exp(-\ln(T/\delta))-(T-1)(\exp(6\ln(2/3)\ln(T/\delta)) \\ &\geq 1-\delta/T-(T-1)(T/\delta)^{6\ln(2/3)} \\ &= 1-\delta/T-(T-1)(\delta/T)^{6\ln(3/2)} \\ &\geq 1-\delta/T-(T-1)(\delta/T)^{2.4} \\ &\geq 1-\delta \end{aligned}$$

for all $T \geq 1$. □

Theorem 2. In the setting of Theorem 1, if $\mathcal{D}$ is continuous, then for all selective sampling algorithms *ALG*, there exists a target hypothesis such that *ALG* has expected cost $\Omega(\log T)$.

Proof. It suffices to show an $\Omega(\log T)$ bound on expected cost in the modified setting of Proposition 1. It is convenient to prove a slightly stronger statement; rather than proving that all selective sampling algorithms have $\Omega(\log T)$ expected cost in the worst case over hypotheses, we prove that this is true if the hypothesis is drawn from the distribution $\mathcal{D}$. This is a stronger statement, because if the expectation over hypotheses is $\Omega(\log T)$, there must be some hypothesis that achieves $\Omega(\log T)$. Since the hypothesis and the instances are all drawn from $\mathcal{D}$, and the cost only depends on the relative ordering of instances versus the hypothesis, we can assume without loss of generality that $\mathcal{D} = \mathsf{Unif}\,[0,1]$.

In the modified setting, the optimal algorithm is easy to characterize. In the modified setting, no optimal algorithm ever asks for a label, since this is guaranteed to incur cost 1, and the algorithm will see the label anyway. Define A_t^k, B_t^k as in the proof of Theorem 1(1). It is easy to check that after seeing $X_1, \ldots, X_{t-1}$, the posterior over hypotheses h is uniform on (A_t^k, B_t^k). Hence, the optimal algorithm classifies X_t as negative if $X_t \leq \frac{A_t^k + B_t^k}{2}$ and classifies X_t as positive otherwise. It is also easy to check that the expected cost incurred by this strategy at time t is

$$\begin{aligned}
2\int_{A_t^k}^{(A_t^k+B_t^k)/2} Pr(x > h)dx &= 2\int_{A_t^k}^{(A_t^k+B_t^k)/2} (x - A_t^k)/(B_t^k - A_t^k)dx \\
&= \frac{2}{(B_t^k - A_t^k)} \frac{((B_t^k - A_t^k)/2)^2}{2} \\
&= (B_t^k - A_t^k)/4.
\end{aligned}$$

Define $M_t = B_t^k - A_t^k$. It remains to lower bound $\mathbb{E}\, M_t$. By the law of iterated expectations, $\mathbb{E}\, M_t \geq \inf_{x_1, x_2, \ldots, x_{t-1}} E[M_t \mid X_1 = x_1, \ldots, X_t = x_{t-1}]$. Fix arbitrary values $x_1, \ldots, x_{t-1} \in [0,1]$. Let $x_{-1} = 0$ and $x_0 = 1$. Let $x^{(-1)} \geq x^{(0)} \geq \cdots \geq x^{(t-1)}$ be the points $x_{-1}, x_0, \ldots, x_{t-1}$ sorted in nondecreasing order. We have

$$\begin{aligned}
&\mathbb{E}\,[M_t \mid X_1 = x_1, \ldots, X_{t-1} = x_{t-1}] \\
&= \sum_{-1 \leq i \leq t-1} (x^{(i+1)} - x^{(i)}) P(x^{(i-1)} > B > x^{(i)}) \\
&= \sum_{-1 \leq i \leq t-1} (x^{(i-1)} - x^{(i)})^2 \\
&\geq (t+1)(1/t)^2 \\
&\geq 1/t,
\end{aligned}$$

where the last inequality follows because $\sum_{-1 \leq i \leq t-1} (x^{(i+1)} - x^{(i)}) = 1$ and the function x^2 is convex, so $\sum_{-1 \leq i \leq t-1} (x^{(i+1)} - x^{(i)})^2$ is minimized when $x^{(i+1)} - x^{(i)} = 1/t$ for all $-1 \leq i \leq t-1$. It follows that

$$\mathbb{E}\, C_T = \sum_{i=1}^{T} \mathbb{E}\, c_t = \sum_{i=1}^{T} \mathbb{E}\, \frac{M_t}{4} \geq \sum_{i=1}^{T} \frac{1}{4t} \geq \frac{1}{4} \log T.$$

So $\mathbb{E}\, C_T \in \Omega(\log T)$ as desired. □

Proposition 7. *If $\mathcal{X} = [a,b]$, the arrivals X_t are arbitrary, and $\mathcal{H} = \{\mathbb{1}_{x>t} \mid t \in [a,b]\}$, all algorithms incur worst-case cost $C_T = T$.*

Proof. Fix an algorithm *ALG*. Consider the modified setting in the proof of Proposition 1. For $t \geq 1$, define A_t^k, B_t^k as in the proof of Theorem 1, Claim 1. Recall that (A_t^k, B_t^k) is the region of uncertainty; at time 1, since (intuitively) no labels have been observed, we have $A_1 = 0$, $B_1 = 1$. For each round t, let $X_t = \frac{A_t^k + B_t^k}{2}$. Clearly, (A_t^k, B_t^k) is nonempty for all $t \geq 1$, and $B_t^k - A_t^k = 1/2^{t-1}$. If *ALG* classifies X_t as positive, label X_t as negative, and vice versa. If *ALG* queries, label X_t as positive. Clearly, if this labeling is valid, that is, if all these labels are consistent with some hypothesis $h \in [0, 1]$, then for this hypothesis, $C_T = T$ for all $T \geq 1$. Let h be the number with binary representation $0.\omega_1\omega_2 \ldots$, where $\omega_t = 0$ if *ALG* classifies X_t as negative, and $\omega_t = 1$ if *ALG* classifies X_t as positive or queries. It is clear that h is contained in all the intervals (A_t^k, B_t^k), hence consistent with all labels. □