

2022-01-06

# Active hedge: hedge meets active learning

---

V. Saligrama, B. Kumar, J. Abernathy. 2022. "Active Hedge: Hedge meets Active Learning"  
International Conference on Machine Learning.

<https://hdl.handle.net/2144/47051>

*Downloaded from OpenBU. Boston University's institutional repository.*

---

# ActiveHedge: Hedge meets Active Learning

---

Bhuvash Kumar<sup>1</sup> Jacob Abernethy<sup>1</sup> Venkatesh Saligrama<sup>2</sup>

## Abstract

We consider the classical problem of multiclass prediction with expert advice, but with an active learning twist. In this new setting the learner will only query the labels of a small number of examples, but still aims to minimize regret to the best expert as usual; the learner is also allowed a very short *burn-in* phase where it can fast-forward and query certain highly-informative examples. We design an algorithm that utilizes Hedge (aka Exponential Weights) as a subroutine, and we show that under a very particular combinatorial constraint on the matrix of expert predictions we can obtain a very strong regret guarantee while querying very few labels. This constraint, which we refer to as  $\zeta$ -compactness, or just compactness, can be viewed as a non-stochastic variant of the disagreement coefficient, another popular parameter used to reason about the sample complexity of active learning in the IID setting. We also give a polynomial time algorithm to calculate the  $\zeta$ -compactness of a matrix up to an approximation factor of 3.

## 1. Introduction

The problem of multiclass prediction with expert advice has emerged as a simple yet powerful framework for reasoning about sequential decision tasks. We imagine we have a set of  $N$  experts, at each round there are  $K$  possible outcomes, and where each expert  $j$  makes a prediction  $X_{t,j} \in [K]$  at time  $t$  about an unknown label  $y_t \in [K]$ . Our learning task is to emit our own estimate  $\hat{y}_t \in \Delta_K$  of  $y_t$ , that takes into account the advice of each expert along with their historical performance up until this time point. The simple goal is: can we predict well, in the long run, relative to the expert who performs optimally over the full sequence of predictions, despite that we do not know in advance which expert is

best? Moreover, what can we guarantee even when some of these experts may be predicting in an arbitrary or perhaps adversarial fashion? These questions have received a great deal of attention over the past two decades.

The classical algorithm for this problem is commonly known as Hedge (Freund & Schapire, 1995), although variants are often referred to as *exponential weights* or *weighted majority*. While we give a precise description in Algorithm 1, Hedge is quite simple to explain in words: the algorithm combines the predictions of all the experts on a given round by taking their weighted average, where the weight of an expert exponentially decays according to the number of previous mistakes. Important details must be addressed, such as the exponential decay factor and what to do with fractional predictions, but a great deal of research has made one point very clear: Hedge is essentially the minimax optimal algorithm for the problem of prediction with expert advice.

One of the downsides of Hedge, as with many online learning algorithms, is that it is not *label efficient*: the learning process requires that we observe the target  $y^t$  on each round. Obtaining individual labels can, quite often, be very expensive to the learner; indeed this is central to why we design prediction algorithms in the first place. *Active learning*, which refers broadly to a family of frameworks in which the learning algorithm can make selective label queries, are designed precisely with the goal of minimizing the number of needed labels while achieving a suitable learning performance. The key idea is that we do not necessarily need to have a batch of labelled examples prior to training, in many natural scenarios the algorithm may be able to actively engage with the labelling process to query labels on a set of unlabelled examples. The classical Binary Search algorithm is, in some sense, an active learning algorithm to find an element in a sorted list.

It would be hard to argue against the wealth of empirical results showing the benefits of active learning (Settles, 2011; Nguyen & Smeulders, 2004; Wang & Hua, 2011; Kapoor et al., 2007; Li & Guo, 2013). At the same time, while our theoretical understanding of the label-efficiency gains achieved using this new learning model has been studied in a range of scenarios (Hanneke, 2007; Zhang, 2018; Hanneke & Yang, 2012; Hanneke, 2011; Kulkarni et al., 1993; Koltchinskii, 2006; Freund et al., 1997; Dasgupta et al.,

---

<sup>1</sup>Georgia Institute of Technology <sup>2</sup>Department of Electrical and Computer Engineering, Boston University. Correspondence to: Bhuvash Kumar <bhuvash@gatech.edu>.

2008), our progress towards a full-fledged concrete mathematical foundation of active learning has been relatively slow. A persistent challenge is that precisely identifying scenarios in which active label querying can provide provable benefits, versus those where it necessarily can not, has proven quite difficult (Zhang, 2018; Hanneke, 2011). The one notable exception is *disagreement-based* active learning (Hanneke, 2014): it has been shown that, as long as the binary hypothesis class possesses a particular property with respect to the underlying probability distribution, known as the *disagreement coefficient*, a recursive algorithm can “zoom in” to the optimal hypothesis and achieve faster learning with lower label complexity. While the disagreement coefficient is somewhat difficult to define, the theoretical work associated to this framework has been perhaps the crowning achievement of the area.

In the following section we give longer outline of the existing work in this area. But it is worth noting up front that nearly all work on active learning has imagined a “batch” setting, where the algorithm is evaluated only at the end of the learning process, in expectation, on new samples. This is surprising, in particular, given that active learning methods are by their nature online, as they seek to iteratively refine their learning process and selection of samples. But thus far there has been no work on putting active learning algorithms to the test in a no-regret setting of prediction with expert advice, where the algorithm’s decision is evaluated at each round of the sequence, and where the expert’s predictions as well as the labels can be non-stochastic and potentially chosen by an adversary.

In the present paper we aim to remedy this gap, and show that there is a natural framework for active learning in the no-regret setting of prediction with expert advice with strong learning guarantees as well as bounded label complexity. First, we define a notion of complexity of the experts’ predictions, somewhat akin to the disagreement coefficient, that provides a key tool in obtaining a provable guarantee; we refer to this as *compactness* for a parameter  $\zeta \geq 1$ . Quite notably, this quantity can be efficiently estimated up to a constant factor!

**Theorem 1.1** (Informal). *There is a polynomial time algorithm to calculate the compactness  $\zeta$  of a matrix up to an approximation factor of 3.*

Second, we define “no-regret active learning” by laying out what we believe is the appropriate analogue to the batch setting. To put it briefly, we imagine a scenario in which the learner must still make sequential predictions on an  $M$ -length list of examples, but with the following modifications: (a) the learner is given the sequence of all experts’ predictions in advance, (b) the learner can only query the true label  $y_t$  on a small number of examples, and (c) the learner is given a very short *burn-in period* where it can “fast-forward”

to future rounds in order to query particularly-informative examples. It is this last feature that makes our setting truly *active*, as this term is used in the batch setting, since the learner can recursively seek out useful datapoints. After the short burn-in, however, the learner must play the remainder of the sequence in its original order while querying only a small fraction of the labels.

Third, we propose an online learning algorithm for this setting, ActiveHedge, that leans heavily on Hedge as a subroutine yet uses dramatically fewer label queries. We are able to show the following:

**Theorem 1.2** (Informal). *Assume we must predict a sequence of labels in  $[K]$ , we have  $N$  experts who have provided predictions (in  $[K]$ ) on all  $M$  examples, and the prediction matrix  $\mathbf{X} \in [K]^{M \times N}$  is  $\zeta$ -compact for some  $\zeta \geq 1$ . If some expert makes only  $\epsilon M$  mistakes, for some  $\epsilon > 0$ , then with probability  $\geq 1 - \rho$  algorithm ActiveHedge guarantees that*

1. *with burn-in period of only  $O(\zeta \log N \log \frac{1}{\epsilon})$  rounds,*
2. *no more than  $O\left(\zeta \epsilon M \text{polylog}\left(\frac{N}{\epsilon \zeta \rho}\right)\right)$  label queries,*
3. *can achieve regret  $O\left(\sqrt{\epsilon M \ln N} + \ln N\right)$ .*

Assuming the prediction matrix  $\mathbf{X}$  is  $\zeta$ -compact for a reasonably-sized constant  $\zeta$ , this theorem states that the regret of ActiveHedge is indeed *no worse* than Hedge, yet requires a dramatically lower label complexity: roughly  $\tilde{O}(\zeta \epsilon M)$  queries are needed. The only extra power we give the learner is a very brief burn-in period, roughly  $\tilde{O}(\zeta)$  rounds, where it can do active exploration of future examples. We now give an illustrative example to view this setting in comparison with more classical batch active learning.

**Batch vs Online Active Learning** Before we dive into the related work and our results, let us lay out an intriguing scenario. Imagine that a worldwide viral pandemic has recently emerged, and a drug company has been working furiously for months to develop a vaccine to provide immunity to the novel virus. The company has been able to design two candidate vaccines,  $A$  and  $B$ , has proven to federal regulators that both drugs are safe enough to study in humans, but there’s a challenge: some people have a mild allergic reaction to vaccine  $A$  but not  $B$ , and everyone else has a similar allergic reaction to vaccine  $B$  but not  $A$ , but this only occurs months after exposure. The company knows that the allergic reaction is based on one of thousands of possible genetic variants, yet must determine quickly which is the relevant gene. Unfortunately there are only two ways to determine if the allergic reaction will occur: (a) wait months to inquire with the patient, or (b) run an expensive test after administering the vaccine that determines immediately whether the allergic reaction will occur.

In this scenario, the “experts” (hypotheses) correspond to candidate genes, a recipient of the vaccine is an example, the true label is their sensitivity to  $A$  or  $B$ , and the label query cost is incurred by the expensive test needed to detect a future allergic reaction. We introduce this challenge because it helps to highlight the distinction between the two modes of active learning, the classical batch framework and our online setting.

1. If the company decides to take a *batch* active learning approach, they would begin by asking random members of the population to submit their genetic profile and sign up for a vaccine study, but with only a small chance to be selected. The company would then adaptively filter applicants, zero in on particularly-suitable individuals with the relevant genetic information, administer one of the two vaccines, and then immediately give the expensive test to detect for future allergic reactions. A population-wide vaccine administration protocol can then be developed once the key gene in question is determined.
2. The *online* approach is more aggressive: the company announces that anyone who would like to be vaccinated will have the opportunity, but they must submit a certified genetic profile in advance, arrive at the local mall on a Saturday by 11am, and then wait in a line. All are promised to receive one of the two vaccines, with the goal of minimizing potential allergic reaction; some recipients will be given the expensive test to quickly determine this. Also, all participants are told that *a small number may be brought to the front of the line* so that more medically-informative candidates are treated first; this is the “burn-in” phase which we’ll discuss more in Section 2.

The typical way that medical procedures are tested and refined is using the first protocol, but we would argue<sup>1</sup> that the second is superior in how it accounts for and manages the costs and benefits of both vaccine recipients and developers. The batch active learning framework has generally been focused on simply minimizing the number of label queries (expensive tests) in order to achieve  $\epsilon$  accuracy on future examples, but prediction errors that occur in the study phase are not accounted for in the loss objective. The online active learning framework, on the other hand, does not distinguish between study participants and regular vaccine recipients – the goal is simply to induce the least number of allergic reactions at the smallest possible testing cost over the long term.

<sup>1</sup>We want to emphasize that we are **not** proposing to change the drug design and trial framework, as this involves a host of ethical and legal issues not considered here. Rather, drug development provides a useful hypothetical to consider the relative costs of testing and accuracy in an adaptive experimentation problem.

It is important to note that batch active learning methods, including disagreement-based learning we describe below, can not immediately be applied in the online setting. Batch active learning only considers *label query costs* in the training phase and *prediction error costs* in the testing phase. Another relevant distinction is that our results do not rely on any IID assumption – indeed since the algorithm is allowed to move certain examples ahead in the queue adaptively, new examples are almost certain to be non-independent.

**Related Work** We briefly survey prior work in the general area of active learning. We will describe salient aspects of these works, and outline how our paper differs from these existing approaches in terms of framework, method, and theory. At a fundamental level, active learning deals with label efficient learning, namely, identifying a good predictor,  $h_*$ , from within a hypothesis class,  $\mathcal{H}$ , based on selectively choosing examples to query for labels. Within this context, a number of methods under a variety of scenarios and assumptions have been studied.

There has been a great deal of work in this area, yet we limit our survey here to a few important themes, in order to draw contrasts and parallels to our setting. Label efficient learning has been considered in pool-based (Settles, 2012; Hanneke, 2014), streaming (Cohn et al., 1994; Balcan et al., 2006; Beygelzimer et al., 2008) and online scenarios (Cesa-Bianchi et al., 2006; 2009; Dekel et al., 2012). Pool and stream-based scenarios have been considered largely within the setting of IID examples and/or labels, whereas online methods have been considered under probabilistic (Dekel et al., 2012) as well as adversarial (Cesa-Bianchi et al., 2006) label noise assumptions. A number of approaches including disagreement-based (Beygelzimer et al., 2008; 2010; Hanneke, 2007; Dasgupta et al., 2008; Hanneke, 2009; Hanneke & Yang, 2012), margin-based (Dasgupta et al., 2005; Balcan et al., 2007; Balcan & Long, 2013; Awasthi et al., 2014; 2015; Zhang, 2018), importance-sampling-based (Beygelzimer et al., 2008; Cortes et al., 2019), and multiplicative-weight update-based (Cesa-Bianchi et al., 2006) and other online (Yang, 2011; Dekel et al., 2012) based methods.

In much of the pool and streaming based methods, the underlying assumption is that the examples and labels, are or can be, drawn IID from some fixed unknown distribution, with labels hidden from the learner. The learner after making a number of label requests, not exceeding, say  $U$ , outputs a predictor  $\hat{h}$ . In this line of work, the active-learning protocol is based on comparing  $\hat{h}$  against the Bayes optimal predictor on an independent labeled sequence. While there is a rich history of methods, which have been explored under a variety of label noise assumptions, the setting of our work is quite different, in that we make no probabilistic assumptions on the data generation process or label noise; and our active learning protocol, in contrast to these works, does not

require independence between training and test scenarios. In particular, our protocol follows the online regret setting, and the incorrect predictions are penalized on the dataset available to the learner during the training process. On the other hand, our proposed method and theoretical results are fundamentally related to the so called disagreement based methods, and leverages key insights of Hanneke’s disagreement coefficient (Hanneke, 2014). In particular, we develop the notion of  $\zeta$ -compactness, which can be interpreted, in some sense, as a deterministic and combinatorial version of disagreement coefficient. Nevertheless, since we make no probabilistic assumptions all previous disagreement-based methods, we cannot leverage classical empirical risk minimization bounds in our context. For this reason, we draw upon insights from the Hedge algorithm and its associated regret bounds, which are agnostic to such probabilistic assumptions.

Our work is also closely related to the label efficient online learning methods, which have been analyzed both under unbiased probabilistic noise as well as adversarial noise assumptions. (Cesa-Bianchi et al., 2005) describes a selective sampling method within the framework of online regret minimization for bounded loss functions. The learner plays  $M$  rounds and at time  $t$  gets an input  $x_t$ , and can decide to seek a label, while being aware of the overall label budget  $U$ . Within this setting, leveraging a variant of the Hedge algorithm, and with no additional assumptions on data process, (Cesa-Bianchi et al., 2005) provides regret guarantees, which scale as  $M\sqrt{\frac{\log(N)}{U}}$  for  $N$  experts (number of hypothesis). A number of online variants to this selective sampling approach have been proposed. (Cesa-Bianchi et al., 2009; Dekel et al., 2012) introduce probabilistic noise assumptions, and in particular assume that the regression function is linear, and the label noise is unbiased and independent of other examples or queries. The linearity of the regression function together with independent label noise allows them to leverage recursive least-squares techniques. Similar to these works, we also consider a regret-minimization techniques. Different from (Cesa-Bianchi et al., 2009; Dekel et al., 2012) we make no probabilistic assumptions on label noise. (Zhao et al., 2013; Hao et al., 2018) consider the same setting as that of selective sampling where the learner can request the label after making the predictions in each round but don’t give any theoretical guarantees on the label complexity. In contrast to (Cesa-Bianchi et al., 2005) we assume data from all the  $N$  rounds are available to the learner a priori. In addition, we impose the notion of  $\zeta$ -compactness on the dataset of experts’ predictions via a concept closely related to disagreement coefficient, which allows for dramatic improvements in label efficiency. As a matter of comparison, say the optimal expert makes  $\epsilon M$  errors, then the existing selective sampling results with budget  $U = O(\epsilon M)$ , would lead to a regret equal to  $\sqrt{\frac{M \log(N)}{\epsilon}}$  in

comparison to our result suggesting  $\sqrt{\epsilon M \log(N)}$ . Nevertheless, improvement in our result can be attributed to the additional imposition of  $\zeta$ -compactness.

## 2. Notation, Setting, and Background

For the remainder of the paper, we will consider a matrix  $\mathbf{X} \in [K]^{M \times N}$  that represent the predictions of a set of  $N$  experts on a sequence of  $M$  rounds. We will use the notation  $X_t$  to refer to the  $t$ th row of  $\mathbf{X}$ , although we will often index rows using the letter  $i$  or  $I$ . We write  $X_{i,j}$  to denote the  $(i, j)$ th entry of  $\mathbf{X}$ . Alongside this matrix will be an (unknown) sequence of labels  $y_1, \dots, y_M \in [K]$ . We require a loss function  $\ell : \Delta_K \times [K] \rightarrow \mathbb{R}$ , and for simplicity we restrict our attention to the absolute loss  $\ell(\hat{y}, y) := \frac{1}{2} \|\hat{y} - \delta_y\|_1$ . Here  $\delta_y \in \{0, 1\}^K$  is the indicator vector, with all zeros except a 1 in the  $y$ -th coordinate.

### 2.1. Basics: Prediction with Expert Advice, and Hedge

In the classical setting of prediction with expert advice, the learner receives prediction vector  $X_t$  at round  $t$ , makes a prediction  $\hat{y}_t \in \Delta_K$ , observes the true label  $y_t$ , and suffers the loss  $\ell(\hat{y}_t, y_t)$ . Each expert  $j$  suffers a loss as well,  $\ell(X_{t,j}, y_t)$ , and note that this loss is conveniently the 0-1 loss as well,  $\mathbb{1}_{[X_{t,j} \neq y_t]}$ . The algorithm wants to choose the predictions  $\hat{y}_1, \dots, \hat{y}_M$  in order to minimize the *regret*:

$$\text{REG}_{\text{alg}} := \sum_{t=1}^M \ell(\hat{y}_t, y_t) - \min_{j \in [N]} \sum_{t=1}^M \ell(X_{t,j}, y_t).$$

At times it will be convenient to refer to the cumulative loss of expert  $j$  as  $L_j^M = \sum_{i=1}^M \ell(X_{i,j}, y_i)$ . Similarly, the loss of the algorithm is  $L_{\text{Hedge}}^M = \sum_{t=1}^M \ell(\hat{y}_t, y_t)$ .

We have already discussed Hedge, the most well-known algorithm for the problem of prediction with expert advice. We lay this out in full detail in Algorithm 1, with two important subroutines, HedgeUpdate and HedgePredict, that will be needed later.

**Theorem 2.1.** Assume we know a quantity  $L^*$  such that  $\min_{j=1, \dots, N} L_j^M \leq L^*$ . Then, choosing  $\eta = \log \left( 1 + \sqrt{\frac{2 \ln N}{L^*}} \right)$  Algorithm 1 guarantees

$$L_{\text{Hedge}}^M - \min_{j=1, \dots, N} L_j^M \leq \sqrt{2L^* \ln N} + \ln N. \quad (1)$$

This is, in many respects, a fundamental bound. We know, for example, that this can not be made any tighter, even up to constants (Cesa-Bianchi & Lugosi, 2006).

### 2.2. Prediction Matrix Compactness

In the typical adversarial learning setting we assume that the experts’ predictions and labels are chosen in some arbitrary



**Algorithm 1:** Hedge

---

```

1 Input:  $\eta > 0$  /* learning rate parameter */
2 Init:  $\vec{w}^0 = [1, \dots, 1]$  /*  $N$  initial weights */
3 for  $t = 1, \dots, M$  do
4    $X_t \leftarrow \text{Preds}(t)$  /* Receive expert predictions */
5    $\hat{y}_t \leftarrow \text{HedgePredict}(X_t, \vec{w})$ 
6    $y_t \leftarrow \text{QueryLabel}(t)$ 
7    $\vec{w} \leftarrow \text{HedgeUpdate}(\vec{w}, X_t, y_t, \eta)$ 
8 end
1 Procedure HedgePredict( $\vec{x}, \vec{w}$ )
2    $\vec{p} \leftarrow \left[ \frac{w_1}{\sum_{i=1}^N w_i}, \dots, \frac{w_N}{\sum_{i=1}^N w_i} \right]$  /*  $\vec{p} \in \Delta_N$  */
3    $\hat{y} \leftarrow \vec{p} \cdot \text{ONEHOT}(\vec{x})$  /* Weighted multiclass pred */
   /* ONEHOT converts multiclass preds  $\vec{x} \in [K]^N$  to one-hot
   matrix encoding  $\in (\Delta_K)^N$  */
4   return  $\hat{y}$  /*  $\hat{y}$  is a probability vec in  $\Delta_K$  */
1 Procedure HedgeUpdate( $\vec{w}, \vec{x}, y, \eta$ )
   /* Decrease weight of incorrect experts */
2   for  $j = 1, \dots, N$  do
3      $w_j^+ \leftarrow w_j \exp(-\eta \mathbb{1}_{[x_j \neq y]})$ 
4   end
5   return  $\vec{w}^+$ 

```

---

fashion. On the other hand, it is well understood that to obtain any reasonable learning result in an active label-efficient mode one requires stronger assumptions on the input data. In our framework of prediction with expert advice this will mean we must constrain the matrix  $\mathbf{X}$  in an appropriate fashion. Let us now describe a particular condition on  $\mathbf{X}$ , which we call compactness, that measures a purely combinatorial property of the space of predictions.

**Definition 2.2.** Given  $\mathbf{X} \in [K]^{M \times N}$ , and for any subset  $V \subseteq [N]$  of experts, the *points of contention* of  $V$  is the set

$$\text{PoC}_{\mathbf{X}}(V) := \{i \in [M] \mid \exists j, j' \in V : X_{i,j} \neq X_{i,j'}\}$$

For any set of experts, the points of contention are the collection of examples where at least two of the experts in the set disagree.

**Definition 2.3** ( $\zeta$ -Compactness). For some  $\zeta \geq 1$ , we say that an expert prediction matrix  $\mathbf{X}$  is  $\zeta$ -compact if it satisfies

$$\frac{|\text{PoC}_{\mathbf{X}}(V)|}{\max_{j,j' \in V} |\text{PoC}_{\mathbf{X}}(\{j, j'\})|} \leq \zeta \quad (2)$$

for each  $V \subset [N]$  with  $|V| \geq 2$ . We refer to the *compactness* of  $\mathbf{X}$  as the smallest  $\zeta$  for which inequality (2) holds.

Given a prediction matrix  $\mathbf{X}$ , the compactness of  $\mathbf{X}$  controls the divergence between two key quantities of a group of experts  $V$ : the total number of points of contention of all of  $V$  versus the largest number of points of contention over

any pair in the group. In one sentence, the matrix  $\mathbf{X}$  is  $\zeta$ -compact if the size of the contentious set for any subset of experts is never  $\zeta$  larger than that of the most contentious pair of experts in it. Here are two illuminating examples that illustrate matrix compactness:

1. Let  $K = 2$ ,  $M = N$  and let  $\mathbf{X}$  be the identity matrix, with all 0 entries except 1s on the diagonal. The compactness of this matrix is  $\frac{M}{2}$ , unfortunately, which is very large. That's because if you take  $V = [N]$  we see that  $\text{PoC}_{\mathbf{X}}(V) = [M]$  the whole set of examples. But for any pair  $j, j'$  we have  $\text{PoC}_{\mathbf{X}}(\{j, j'\}) = \{j, j'\}$ . In other words, any group of experts has as many points of contention as members in the group, but any pair of experts will disagree on only two points. This is indeed a very hard case for active learning, as individual examples are not very informative.
2. Continue to let  $M = N$  and now let  $\mathbf{X}$  be the upper triangular matrix with all 1s on and above the diagonal, and 0s below. This is a very compact matrix, with  $\zeta = 1$ ! That's because for any subset  $V$  we have  $\text{PoC}_{\mathbf{X}}(V) = \text{PoC}_{\mathbf{X}}(\{\min(V), \max(V)\})$ , i.e. the points of contention in  $V$  is identically the points of contention for the largest-index and smallest-index experts in the set.

Following point 1 above, we can give a simple bound on the compactness of any expert prediction matrix  $\mathbf{X}$ , whose proof is in Appendix C. But this bound is mostly useless from the perspective of our main results, as we need  $\zeta \ll M$  for a non-trivial guarantee on label complexity.

**Theorem 2.4.** For any matrix  $\mathbf{X} \in [K]^{M \times N}$ , for  $M \geq 2$ , the compactness of  $\mathbf{X}$  is less than or equal to  $\min\{M, N\}$

**Comparison to the Disagreement Coefficient.** As we mentioned early in the paper, one of the major theoretical accomplishments in the literature on label-efficient statistical learning is the work on disagreement-based active learning, first introduced by (Hanneke, 2007) with several followup works (Hanneke, 2009; 2011; Balcan et al., 2006; Hanneke, 2014; Hanneke & Yang, 2015). The key quantity of interest in this work is known as the *disagreement coefficient*, a scalar that measures the difficulty of active learning with respect to a particular hypothesis class and data distribution. What was shown all the way back to (Hanneke, 2007) was that this coefficient controls the label complexity of learning on the given task, and they show several examples where the disagreement coefficient is of reasonable size.

While we developed our notion of compactness independently, and with a different model in mind, we later realized that in the case of binary classification our definition can in some sense be viewed as a “derandomization” of Hanneke’s disagreement coefficient; we make this more precise in the proposition below. The compactness  $\zeta$  of a prediction

matrix  $\mathbf{X}$  does not depend on any notion of IID sampling from an underlying data distribution, as  $\zeta$  is purely a combinatorial property of the experts' predictions which could have been adversarially chosen. And, while there is some resemblance between the *burn-in* procedure in Phase I of ActiveHedge and the  $A^2$  algorithm of (Hanneke, 2007), our results are not at all comparable: the goal of our work was to produce an algorithm that suffers low regret, as it is forced to make a prediction and suffer loss on each example, and be robust against non-stochastic sequences of data.

**Proposition 2.5.** *Consider a binary expert prediction matrix  $\mathbf{X}$  with compactness  $\zeta$ . Construct a data distribution  $D$  which generates an  $x, y$  pair by uniformly sampling  $x$  as a row of  $\mathbf{X}$  and let  $y$  be the corresponding label. We can consider the set of experts as an  $N$ -sized hypothesis class  $\mathcal{H}$ . Then the disagreement coefficient of  $(D, \mathcal{H})$ , as defined by (Hanneke, 2007), is  $2\zeta$  where  $\zeta$  is the compactness of  $\mathbf{X}$ .*

### 2.3. Online active learning with experts

Let us now specify the details of our framework for active learning with expert advice. It can be described in terms of the vanilla Hedge setting, but with three key modifications:

1. The sequence of expert predictions, specified by  $\mathbf{X}$ , can be precomputed and is given to the learner in advance of the prediction task.
2. The learner aims to make only a small number of label queries, limiting the number of times  $y_t$  is observed.
3. We allow a very brief *burn-in* period, which we call Phase I, where the learner can “fast-forward” to act on particular examples, and query their labels, out of turn. In Phase II the learner then plays the remaining points, which are the vast majority, in the order they are given, with the occasional label query if needed.

Modification 1 above is not unusual and arises naturally in settings where the experts are a set of pre-selected deterministic hypotheses, the rounds/examples are given by a queue of contexts/input vectors, and we can pre-evaluate each hypothesis on each context (the vaccine development scenario given in the introduction is another such example). Modification 2 captures the underlying goal that we want to skip the potentially-expensive step of obtaining the correct multiclass label in all but a small fraction of rounds; adding this modification alone is often referred to as *label efficient online learning*, e.g. (Sculley, 2007).

Modification 3 is perhaps the most unusual in the context of adversarial online learning, where one assumes that the learner the sequence of examples and labels is chosen in an adversarial fashion. But we would argue that this is actually necessary to achieve any kind of non-trivial guarantee: without a small number of fast-forward rounds, the adversary can simply postpone all informative examples to the end of the sequence, at which point querying their labels would

provide no benefit to the learner. Indeed we show that the burn-in period can be extremely short, no more than roughly  $O(\zeta \log N \log \frac{1}{\epsilon})$  where  $\zeta$  is the compactness of  $\mathbf{X}$ , in order to obtain *the same regret* as Hedge with vastly fewer label queries (roughly  $\tilde{O}(\zeta \epsilon M)$ ).

Note that if we don't allow a burn in phase, the lower bounds of Cesa-Bianchi et al. (2005, Theorem 13) apply to the online active learning setting as well. This implies that if we don't allow a burn-in phase, then to guarantee the same  $\sqrt{2\epsilon M \ln N}$  regret as Hedge, any algorithm would require at least  $\frac{C \cdot M}{\epsilon}$  labels for some constant  $C$ . Since  $\epsilon \leq 1$ ,  $\frac{C \cdot M}{\epsilon} = \Omega(M)$ . Thus, without a *burn-in* period, any algorithm would require  $\Omega(M)$  labels to get the same regret guarantee as Hedge. Since Hedge also request  $O(M)$  labels, there would be no advantage in using anything other than Hedge.

## 3. Algorithm And Performance Guarantee

Henceforth we will let  $b$  denote the index of the best expert, i.e.  $b = \operatorname{argmin}_{j \in [N]} L_j^M$ , and that the number of mistakes satisfies  $L_b^M \leq \epsilon M$ .

### 3.1. An Overview of ActiveHedge

We present a multiplicative style algorithm ActiveHedge, described precisely in Algorithm 2. First let us give a high-level intuitive description of the procedure. ActiveHedge is divided into two phases.

1. **Phase I.** This is the so-called burn-in period, where the algorithm can fast-forward to future examples out of turn. On each such example, the algorithm must still make a prediction, and can then query the label. This phase, while short, is done in small epochs of length  $k = O(\zeta \log(N/\rho))$ , with a total of  $T = O(\log(1/\epsilon))$  epochs. In a given epoch  $\tau$  the algorithm has a set of “candidate experts”  $V^\tau$  who have predicted reasonably well thus far. To reduce the number of candidate experts, the algorithm samples future rounds from the points of contention of  $V^\tau$ , makes a Hedge prediction on each, and then queries the label. At the end of the epoch the algorithm discards any experts in  $V^\tau$  whose average error was above a given threshold. On the next epoch we shrink the threshold and consider the new set of candidate experts  $V^{\tau+1}$ , and sample examples from the new set  $\text{PoC}_{\mathbf{X}}(V^{\tau+1})$ , etc.
2. **Phase II.** At the start of this phase the algorithm has a relatively small set of candidate best experts,  $V^T$ , that were selected in Phase I, and with high probability  $b$  remains in  $V^T$  and also every expert in  $V^T$  agrees with  $b$  on all but  $O(\epsilon M)$  examples. With the burn-in segment over the algorithm now plays the remaining

examples, which make up the vast majority, in their original (adversarial) order; rounds played in Phase I are skipped. Uses a very simple prediction strategy:

- (a) if the example  $i$  is in  $\text{PoC}_{\mathbf{X}}(V^T)$ , we use Hedge to make a prediction on this example, we query the label  $y_i$ , and we do a Hedge update on the weights;
- (b) if  $i \notin \text{PoC}_{\mathbf{X}}(V^T)$ , we simply use an *arbitrary* expert  $j^* \in V^T$  and use  $X_{i,j^*}$  as our prediction.

---

**Algorithm 2:** ActiveHedge
 

---

```

Parameters      :  $\epsilon, \eta, k, T, \zeta$ 
Input           :  $\mathbf{X} \in [K]^{M \times N}$ 
Initialize      :  $V^0 \leftarrow [N], t \leftarrow 0, \text{DONE} \leftarrow \emptyset$ 
/* /// PHASE I /// Recursively shrink candidate experts */
1 for  $\tau = 0, \dots, T-1$  do
2    $Z_j^\tau \leftarrow 0 \ (\forall j \in [N])$  /* #errs expert  $j$  at epoch  $\tau$  */
3   for  $c = 0, \dots, k-1$  do
4      $I \sim \text{PoC}_{\mathbf{X}}(V^\tau)$  /* Sample w/ replacement */
5     if  $I \notin \text{DONE}$  then
6        $\hat{y}_I \leftarrow \text{HedgePredict}(X_I, \vec{w}^t)$ 
7        $y_I \leftarrow \text{QueryLabel}(I)$ 
8        $\vec{w}^{t+1} \leftarrow \text{HedgeUpdate}(\vec{w}^t, X_I, y_I, \eta)$ 
9        $t \leftarrow t+1$  /* increment hedge update count */
10       $\text{DONE} \leftarrow \text{DONE} \cup \{I\}$ 
11    end
12     $Z_j^\tau \leftarrow Z_j^\tau + \mathbb{1}_{[X_{I,j} \neq y_I]} \ \forall j \in V^\tau$ 
13  end
14   $\delta^\tau \leftarrow \frac{M}{2|\text{PoC}_{\mathbf{X}}(V^\tau)|} \left( \frac{1}{2^{\tau+1}\zeta} - \epsilon \right)$  /* Update thresh */
15   $V^{\tau+1} \leftarrow \{j \in V^\tau : Z_j^\tau/k \leq \delta^\tau\}$  /* Shrink  $V$  */
16 end
/* /// PHASE II /// Play all remaining rounds */
17 Select  $j^* \in V^T$  arbitrarily
18 for  $i = 1, \dots, M$  do
19   if  $i \in \text{DONE}$  then
20     continue /* skip if example already done */
21   else if  $i \in \text{PoC}_{\mathbf{X}}(V^T)$  then
22      $\hat{y}_i \leftarrow \text{HedgePredict}(X_i, \vec{w}^t)$ 
23      $y_i \leftarrow \text{QueryLabel}(i)$ 
24      $\vec{w}^{t+1} \leftarrow \text{HedgeUpdate}(\vec{w}^t, X_i, y_i, \eta)$ 
25      $t \leftarrow t+1$  /* increment hedge update count */
26   else
27      $\hat{y}_i \leftarrow \text{ONEHOT}(X_{i,j^*})$  /* use default expert  $j^*$  */
28   end
29 end
    
```

---

The choice in condition (b) might seem unusual, but recall that *all experts in  $V^T$  agree on examples  $i \notin$*

$\text{PoC}_{\mathbf{X}}(V^T)$ . As long as we did not accidentally evict  $b$  from our candidate experts in Phase I, the prediction  $X_{i,j^*}$  will match that of  $X_{i,b}$ . Therefore on these rounds we should suffer no regret.

### 3.2. Regret and Label Guarantees

We now present the regret and label complexity guarantee for ActiveHedge (Algorithm 2)

**Theorem 3.1.** *Assume we have  $\epsilon, \rho > 0$ ,  $\vec{y}$ , and  $\zeta$ -compact matrix  $\mathbf{X}$  such that  $10\epsilon\zeta \leq 1$  and for some  $b \in [N]$  we have  $\sum_{i \in [M]} \mathbb{1}_{[X_{i,b} \neq y_i]} \leq \epsilon M$ . We set the ActiveHedge params*

$$k := \left\lceil 192\zeta \log \left( \frac{N}{\rho} \log \frac{1}{10\epsilon\zeta} \right) \right\rceil, \quad T := \left\lceil \log \frac{1}{10\epsilon\zeta} \right\rceil \text{ and}$$

$$\eta := \log \left( 1 + \sqrt{\frac{2 \ln N}{\epsilon M}} \right). \quad (3)$$

Then with probability at least  $1 - \rho$ :

1. the number of calls to QueryLabel is no more than

$$O \left( \zeta \log \left( \frac{N}{\rho} \log \frac{1}{10\epsilon\zeta} \right) \log \frac{1}{10\epsilon\zeta} + \epsilon \zeta M \right)$$

2. the length of Phase I is no more than  $Tk$  which, up to logarithmic terms, is  $\tilde{O}(\zeta)$  rounds;
3. and finally we have that

$$\text{REG}_{\text{ActiveHedge}} \leq \sqrt{2\epsilon M \ln N} + \ln N.$$

**Corollary 3.2.** *If the burn-in phase in ActiveHedge is limited to only  $B$  rounds, then we can achieve the same regret as Hedge with label complexity  $\tilde{O}(B + \frac{M}{2^{B/7\zeta}})$ .*

Theorem 3.1 states that ActiveHedge achieves the same regret guarantee as Hedge with high probability while using considerably less labels. Hedge requires a label complexity of  $M$ , where as for a small  $\epsilon$  and  $\zeta$ , the label complexity of ActiveHedge is closer to  $\tilde{O}(\zeta \epsilon M)$ .

The proof of Theorem 3.1 can be found in the Appendix A. The basic proof sketch is that we divide the regret analysis and the label complexity analysis into the regret and label complexity of the two phases.

In Phase I, using induction, we show that with high probability, the size of the candidate experts set  $V^\tau$  shrinks in every round and the best expert is always present in  $V^\tau$ . After the end of the Phase I, we have narrowed down to the set of candidate experts  $V^T$  so that with high probability  $|\text{PoC}_{\mathbf{X}}(V^T)| = O(\zeta \epsilon M)$ , using compactness, yet still  $b \in V^T$ . In Phase II we only request the labels for the examples that are in  $\text{PoC}_{\mathbf{X}}(V^T)$ , thus the label complexity of Phase II is bounded by  $O(\epsilon \zeta M)$ .

Bounding the regret of ActiveHedge is surprisingly easy, since for all examples played in Phase I as well as for those played in Phase II from  $\text{PoC}_{\mathbf{X}}(V^T)$ , we appeal directly to



Hedge where we have an optimal bound. In many examples in Phase II, where  $i \notin \text{PoC}_{\mathbf{X}}(V^T)$ , we make a prediction that (with high probability) agrees with expert  $b$  and thus we suffer no regret on these rounds.

It should be noted that even though the guarantees in Theorem 3.1 are dependent on the knowledge of  $\epsilon$  and  $\zeta$  for initializing the parameters  $K$  and  $T$  of Algorithm 2, for our proofs to follow through, we just an upper bound on the error rate  $\epsilon$  of the best expert, and similarly for the compactness  $\zeta$ . In Theorem 4.1, we give a polynomial time algorithm to approximate  $\zeta$ ; this can be used to initialize Algorithm 2. Using  $\epsilon' > \epsilon$  in Theorem 3.1, we still get the same regret guarantee of  $\sqrt{2\epsilon M \ln N} + \ln N$  that still depends on  $\epsilon$ , but the label complexity will now be  $O\left(\zeta \log\left(\frac{N}{\rho} \log \frac{1}{10\epsilon'\zeta}\right) \log \frac{1}{10\epsilon'\zeta} + \epsilon'\zeta M\right)$ .

If no estimate for  $\epsilon$  is available, a standard halving trick can be applied to obtain similar regret and sample complexity guarantees. See Appendix B for more details.

## 4. Calculating compactness

### Algorithm 3: Calculate compactness

```

1 Input:  $\mathbf{X} \in [K]^{M \times N}$  /* Expert prediction matrix */
2 Init:  $\tilde{\zeta} \leftarrow 0$ 
3 for all pairs  $j, j' \in [N]$  do
4    $V_{j,j'} \leftarrow \{j, j'\}$  /* Initialize  $V_{j,j'}$  */
   /* Add experts with distance from  $j \leq \text{dist}(j, j')$  */
5    $V_{j,j'} \leftarrow V_{j,j'} \cup \{h | \text{dist}(h, j) \leq \text{dist}(j, j')\}$ 
   /* Add experts with distance from  $j' \leq \text{dist}(j, j')$  */
6    $V_{j,j'} \leftarrow V_{j,j'} \cup \{h | \text{dist}(h, j') \leq \text{dist}(j, j')\}$ 
7    $\zeta_{j,j'} \leftarrow \frac{|\text{PoC}_{\mathbf{X}}(V_{j,j'})|}{\text{DIAM}(V_{j,j'})}$ 
   /* Update  $\tilde{\zeta}$  if a bigger ratio is found */
8   if  $\zeta_{j,j'} > \tilde{\zeta}$  then
9      $\tilde{\zeta} \leftarrow \zeta_{j,j'}$ 
10  end
11 end
12 Return:  $\tilde{\zeta}$ 
    
```

The compactness of an expert prediction matrix is a combinatorial quantity which is easy to compute for some concept classes, but in the worst case it might be hard to compute exactly as we have a supremum over all subsets of experts. We present an algorithm that gives a 3-approximation of the compactness in polynomial time.

For the remainder of this section and the appendix, for any  $V \subset [N]$  let  $\text{DIAM}(V) := \max_{j,j' \in V} |\text{PoC}_{\mathbf{X}}(\{j, j'\})|$  and for any experts  $j, j'$ , let  $\text{dist}(j, j') = |\text{PoC}_{\mathbf{X}}(\{j, j'\})|$ .

**Theorem 4.1.** *If the input matrix  $\mathbf{X}$  to Algorithm 3 is  $\zeta$ -compact, then Algorithm 3 returns  $\tilde{\zeta}$  such that  $\frac{\zeta}{3} \leq \tilde{\zeta} \leq \zeta$*

in runtime  $O(N^4 M)$

As stated earlier, for initializing Algorithm 2 for the results in Theorem 3.1, we just need an upper bound on the  $\zeta$ -compactness. Using Algorithm 3, we can obtain an estimate  $\hat{\zeta} = 3\tilde{\zeta}$  such that  $\zeta \leq \hat{\zeta} \leq 3\zeta$ .

## 5. Experiments

We provide preliminary experiments to compare ActiveHedge (Algorithm 2), with standard Hedge (Algorithm 1) and the label efficient algorithm given by Cesa-Bianchi et al. (2005).

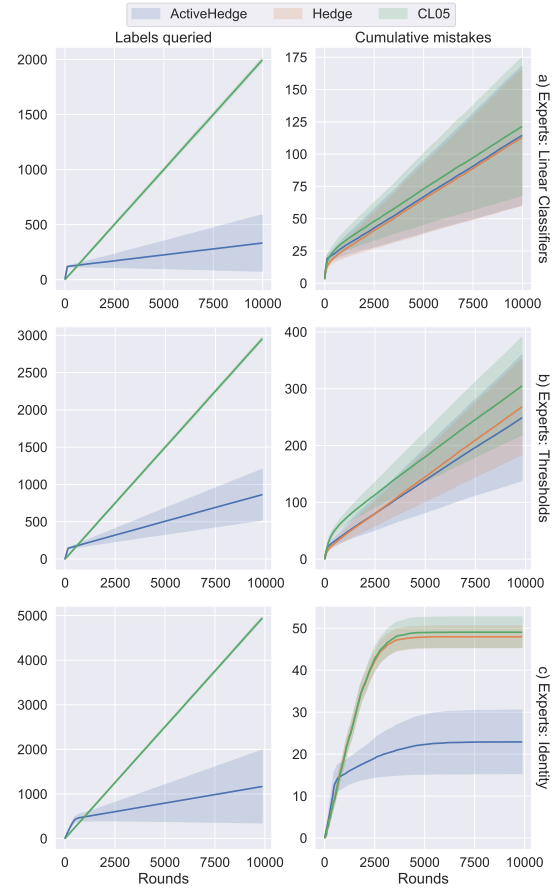


Figure 1. Labels queried and the cumulative mistakes of ActiveHedge, Hedge, and Cesa-Bianchi et al. (2005)(CL05) in 3 different settings. Hedge queries label in every round and is not shown in Labels queried plots to maintain readability.

We consider three different classes of experts for our experiments. In Figure 1: a) we consider linear classifiers passing through the origin as experts. We uniformly  $N$  sample linear classifiers from a unit sphere centred at origin. We then sample  $M$  points from a unit sphere and classify each point using the  $N$  experts to create the expert prediction matrix  $\mathbf{X}$ .

Similarly, in Figure 1: b), we consider multi-dimensional thresholds as experts where a point  $x \in \mathbb{R}^d$  is labeled 1 by an expert  $h \in \mathbb{R}^d$  if  $x_i \geq h_i \forall i \in [d]$ . The experts are sampled by sampling thresholds uniformly between 0 and 1. In both the cases, ActiveHedge is able to achieve similar accuracy to Hedge and achieves better performance than Cesa-Bianchi et al. (2005) in terms of both regret and label complexity.

We also consider the more adversarial case in Figure 1: c), where the expert prediction matrix has an identity matrix like structure with  $\zeta = O(N)$ . Here the expert prediction matrix is designed such that only one of the experts labels each point as 1, and every expert classifies approximately equal number of points as 1. Even in this adversarial case where the  $\zeta$  compactness is very high, ActiveHedge outperforms the competition. Thus happens because even though the  $\zeta$  compactness is high, it also implies that by removing an expert from consideration, we also remove a significant fraction of points we are confused on. This allows us to quickly converge to the optimal expert. All experiments are repeated 100 times, with  $M = 10000$  and  $N = 100$  and  $d = 10$ . We use upper bounds for  $\zeta$  and  $\epsilon$  and other parameters are set optimally. For all these experiments, ActiveHedge required less than 10% of the labels with the burn-in phase being less than 2% of the points.

## Acknowledgements

This research was supported by the Army Research Office Grant W911NF2110246, the National Science Foundation grants CCF-2007350 and CCF-1955981, and the Hariri Data Science Faculty Fellowship Grants, and a gift from the ARM corporation.

## References

- Awasthi, P., Balcan, M. F., and Long, P. M. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 449–458. ACM, 2014. 1
- Awasthi, P., Balcan, M.-F., Haghtalab, N., and Urner, R. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pp. 167–190, 2015. 1
- Balcan, M.-F. and Long, P. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pp. 288–316, 2013. 1
- Balcan, M.-F., Beygelzimer, A., and Langford, J. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 65–72. ACM, 2006. 1, 2.2, D.1
- Balcan, M.-F., Broder, A., and Zhang, T. Margin based active learning. In *International Conference on Computational Learning Theory*, pp. 35–50, 2007. 1
- Beygelzimer, A., Dasgupta, S., and Langford, J. Importance weighted active learning. *ArXiv*, abs/0812.4952, 2008. 1
- Beygelzimer, A., Hsu, D. J., Langford, J., and Zhang, T. Agnostic active learning without constraints. *Advances in Neural Information Processing Systems*, pp. 199–207, 2010. 1
- Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006. 5
- Cesa-Bianchi, N., Lugosi, G., and Stoltz, G. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, 2005. 1, 2.3, 5, 1, 5
- Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 7:1205–1230, Jul 2006. 1
- Cesa-Bianchi, N., Gentile, C., and Orabona, F. Robust bounds for classification via selective sampling. In *Proceedings of the 26th annual international conference on machine learning*, pp. 121–128. ACM, 2009. 1
- Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine learning*, 15(2): 201–221, 1994. 1
- Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Zhang, N. Region-based active learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2801–2809, 2019. 1
- Dasgupta, S., Kalai, A. T., and Monteleoni, C. Analysis of perceptron-based active learning. In *International Conference on Computational Learning Theory*, pp. 249–263, 2005. 1
- Dasgupta, S., Hsu, D. J., and Monteleoni, C. A general agnostic active learning algorithm. In *Advances in neural information processing systems*, pp. 353–360, 2008. 1, 1
- Dekel, O., Gentile, C., and Sridharan, K. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13:2655–2697, Sep 2012. 1
- Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pp. 23–37. Springer, 1995. 1

- Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997. [1](#)
- Hanneke, S. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 353–360. ACM, 2007. [1](#), [1](#), [2.2](#), [2.5](#)
- Hanneke, S. *Adaptive rates of convergence in active learning*. In *COLT*. Citeseer, 2009. [1](#), [2.2](#)
- Hanneke, S. Rates of convergence in active learning. *The Annals of Statistics*, pp. 333–361, 2011. [1](#), [2.2](#)
- Hanneke, S. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3): 131–309, 2014. [1](#), [1](#), [2.2](#)
- Hanneke, S. and Yang, L. Surrogate losses in passive and active learning. *arXiv preprint*, 2012. [1](#), [1](#)
- Hanneke, S. and Yang, L. Minimax analysis of active learning. *The Journal of Machine Learning Research*, 16(1): 3487–3602, 2015. [2.2](#)
- Hao, S., Hu, P., Zhao, P., Hoi, S. C., and Miao, C. Online active learning with expert advice. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):1–22, 2018. [1](#)
- Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE, 2007. [1](#)
- Koltchinskii, V. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006. [1](#)
- Kulkarni, S. R., Mitter, S. K., and Tsitsiklis, J. N. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–35, 1993. [1](#)
- Li, X. and Guo, Y. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 859–866, 2013. [1](#)
- Mitzenmacher, M. and Upfal, E. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017. [E](#)
- Nguyen, H. T. and Smeulders, A. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 79, 2004. [1](#)
- Sculley, D. Online active learning methods for fast label-efficient spam filtering. In *CEAS*, volume 7, pp. 143, 2007. [2.3](#)
- Settles, B. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pp. 1–18, 2011. [1](#)
- Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012. [1](#)
- Wang, M. and Hua, X.-S. Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(2):1–21, 2011. [1](#)
- Yang, L. Active learning with a drifting distribution. In *Advances in Neural Information Processing Systems*, pp. 2079–2087, 2011. [1](#)
- Zhang, C. Efficient active learning of sparse halfspaces. *arXiv preprint*, 2018. [1](#), [1](#)
- Zhao, P., Hoi, S., and Zhuang, J. Active learning with expert advice. *arXiv preprint arXiv:1309.6875*, 2013. [1](#)

## A. Proof of Theorem 3.1

To prove Theorem 3.1, we need a few preliminary lemmas.

**Lemma A.1.** *If a set of experts  $H_1$  is a subset of another set of experts  $H_2$ , then  $\text{POC}_{\mathbf{X}}(H_1) \subseteq \text{POC}_{\mathbf{X}}(H_2)$*

*Proof.* If  $i \in \text{POC}_{\mathbf{X}}(H_1)$ , then there exist two experts  $j, j' \in H_1$ , such that  $X_{i,j} \neq X_{i,j'}$ . Since  $H_1 \subseteq H_2$ ,  $j, j' \in H_2$ , hence  $i \in \text{POC}_{\mathbf{X}}(H_2)$ .  $\square$

In each epoch  $\tau$  of Phase I, we maintain a set of candidate experts  $V^\tau$  and a set of candidate points  $\text{POC}_{\mathbf{X}}(V^\tau)$  we might query the labels for. For ease of notation, let  $S^\tau = \text{POC}_{\mathbf{X}}(V^\tau)$ ,  $\text{DIAM}(V) := \max_{j,j' \in V} |\text{POC}_{\mathbf{X}}(\{j, j'\})|$ , and for any experts  $j, j'$ , let  $\text{dist}(j, j') = |\text{POC}_{\mathbf{X}}(\{j, j'\})|$ .

For the purpose of analysis, we partition the set  $V^\tau$  into two sets. Let

$$B^\tau = \left\{ j \in V^\tau \mid \text{dist}(b, j) > \frac{M}{2^{\tau+1}\zeta} \right\}$$

and also  $\overline{B}^\tau = V^\tau \setminus B^\tau$ .

Intuitively,  $B^\tau$  are the experts which are far from the best expert and thus they make more mistakes and we want to remove them. Using an inductive analysis, we will show that in each epoch, with high probability, we can shrink the set of candidate experts, i.e for all  $\tau$ ,  $V^{\tau+1} \subseteq \overline{B}^\tau$  and that we never remove the best expert  $b$ , i.e  $b \in V^{\tau+1}$ . For the rest of the section, we set  $k = \lceil 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}) \rceil$ ,  $T = \lceil \log \frac{1}{10\epsilon\zeta} \rceil$  and  $\eta = \log(1 + \sqrt{\frac{2\ln N}{\epsilon M}})$

In the following lemma, we show that the size of the set of candidate points sampled from in epoch  $\tau$  is bounded.

**Lemma A.2.** *If  $V^\tau \subseteq \overline{B}^{\tau-1}$ , then  $|S^\tau| \leq \frac{M}{2^{\tau-1}}$*

*Proof.* By definition,  $S^\tau = \text{POC}_{\mathbf{X}}(V^\tau)$ . Since  $V^\tau \subseteq \overline{B}^{\tau-1}$ , using Lemma A.1,  $S^\tau \subseteq \text{POC}_{\mathbf{X}}(\overline{B}^{\tau-1})$ . By definition, of  $\overline{B}^{\tau-1}$ , these experts are at a distance of at most  $\frac{M}{2^\tau\zeta}$  from the best expert, the diameter of this set is at most  $\frac{M}{2^{\tau-1}\zeta}$ . Using definition of  $\zeta$ -compactness,  $|\text{POC}_{\mathbf{X}}(\overline{B}^{\tau-1})| \leq \zeta \cdot \frac{M}{2^{\tau-1}\zeta} = \frac{M}{2^{\tau-1}}$ . Hence  $|S^\tau| \leq \frac{M}{2^{\tau-1}}$ .  $\square$

Now we show that in expectation, any expert in  $B^\tau$  makes a large number of mistakes in epoch  $\tau$  which we will use to obtain a high probability bound.

**Lemma A.3.** *If  $b \in V^\tau$  then for any  $j$  in  $B^\tau$ , if  $Z_j^\tau$  is the number of mistakes made in epoch  $\tau$ , then  $\mathbf{E}[Z_j^\tau] \geq \frac{k}{|S^\tau|} (\frac{M}{2^{\tau+1}\zeta} - \epsilon M)$*

*Proof.* Since  $j \in V^\tau$  and  $b \in V^\tau$ , By definition of  $S^\tau = \text{POC}_{\mathbf{X}}(V^\tau)$ , if for some  $i$ ,  $X_{i,j} \neq X_{i,b}$ , then  $i \in S^\tau$ .  $b$  makes at-most  $\epsilon M$  mistakes, so in the worst case,  $j$  can disagree with  $b$  on these points and be correct, but it has to be wrong on at least  $\frac{M}{2^{\tau+1}\zeta} - \epsilon M$  points in  $S^\tau$  as it disagrees with  $b$  on  $\frac{M}{2^{\tau+1}\zeta}$  points in  $S^\tau$ .

We samples  $k$  points from  $S^\tau$ . Let the examples samples in epoch  $\tau$  be  $(I^1, \dots, I^k)$ , then  $Z_j^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}$ ,  $\implies \mathbf{E}[Z_j^\tau] = \sum_{c=1}^k \mathbf{E}[\mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}] = \sum_{c=1}^k \mathbf{P}[X_{I^c,j} \neq y_{I^c}] \geq \sum_{c=1}^k \frac{1}{|S^\tau|} (\frac{M}{2^{\tau+1}\zeta} - \epsilon M) = \frac{k}{|S^\tau|} (\frac{M}{2^{\tau+1}\zeta} - \epsilon M)$   $\square$

**Lemma A.4.** *If  $b \in V^\tau$  and  $V^\tau \subseteq \overline{B}^{\tau-1}$  then with probability at least  $1 - \frac{\rho|B^\tau|}{N \log \frac{1}{10\epsilon\zeta}}$ ,  $V^{\tau+1} \subseteq \overline{B}^\tau$*

*Proof.* For a fixed  $j \in B^\tau$ , by definition the number of mistakes,  $Z_j^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}$ . The probability that we keep  $j$



in  $V^{\tau+1}$  is

$$\begin{aligned}
 & \mathbb{P}\left[\frac{Z_j^\tau}{k} \leq \frac{1}{2|S^\tau|}\left(\frac{M}{2^{\tau+1}\zeta} - \epsilon M\right)\right] \\
 &= \mathbb{P}\left[\frac{Z_j^\tau}{k} - \frac{1}{|S^\tau|}\left(\frac{M}{2^{\tau+1}\zeta} - \epsilon M\right) \leq -\frac{1}{2|S^\tau|}\left(\frac{M}{2^{\tau+1}\zeta} - \epsilon M\right)\right] \\
 &\leq \mathbb{P}\left[\frac{Z_j^\tau}{k} - \mathbf{E}\left[\frac{Z_j^\tau}{k}\right] \leq -\frac{1}{2|S^\tau|}\left(\frac{M}{2^{\tau+1}\zeta} - \epsilon M\right)\right] \\
 &\leq \exp\left(-\frac{k}{12}\left(\frac{\frac{M}{2^{\tau+1}\zeta} - \epsilon M}{2|S^\tau|}\right)\right) \quad (\text{Chernoff Lower tail}) \\
 &\leq \exp\left(-\frac{k}{12}\left(\frac{1 - 2^{\tau+1}\zeta\epsilon}{8\zeta}\right)\right) \quad (\text{as } |S^\tau| \leq \frac{M}{2^{\tau-1}}) \\
 &\leq \exp\left(-\frac{k}{12}\left(\frac{1}{16\zeta}\right)\right) \quad (\text{as } \tau < \log_2 \frac{1}{10\epsilon\zeta}) \\
 &= \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}))
 \end{aligned}$$

Thus, with probability at least  $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$ ,  $Z_j^\tau > \delta^\tau$ , thus  $j \notin V^{\tau+1}$ . A union bound over  $j \in B^\tau$  gives the proof.  $\square$

So far in the inductive process we have shown that we shrink  $V^\tau$  to only keep experts from  $\overline{B^\tau}$ . Now we show that with high probability, we never remove the best expert  $b$ .

**Lemma A.5.** *If  $Z_b^\tau$  is the number of mistakes made in epoch  $\tau$  by the best expert  $b$ , then  $\mathbf{E}[Z_b^\tau] \leq \frac{k\epsilon M}{S^\tau}$*

*Proof.* Since the best expert makes at-most  $\epsilon M$  mistakes, in the worst case all of these  $\epsilon M$  examples are present in  $S^\tau$ . Since we sample  $k$  points from  $S^t$ ,  $Z_b^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c, b} \neq y_{I^c}]} \implies \mathbf{E}[Z_b^\tau] = \sum_{c=1}^k \mathbf{E}[\mathbb{1}_{[X_{I^c, b} \neq y_{I^c}]}] = \sum_{c=1}^k \mathbb{P}[X_{I^c, b} \neq y_{I^c}] \leq \sum_{c=1}^k \frac{\epsilon M}{S^\tau} = \frac{k\epsilon M}{S^\tau}$   $\square$

**Lemma A.6.** *If  $b \in V^\tau$  and  $V^\tau \subseteq \overline{B^{\tau-1}}$  then with probability at least  $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$ ,  $b \in V^{\tau+1}$*

*Proof.* The probability that  $b$  is not present in  $V^{\tau+1}$  is

$$\begin{aligned}
 & \mathbb{P}\left[\frac{Z_b^\tau}{k} \geq \frac{1}{2|S^\tau|}\left(\frac{M}{2^{\tau+1}\zeta} - \epsilon M\right)\right] \\
 &= \mathbb{P}\left[\frac{Z_b^\tau}{k} \geq \frac{\epsilon M}{2|S^\tau|}\left(\frac{1}{2^{\tau+1}\epsilon\zeta} - 1\right)\right] \\
 &\leq \mathbb{P}\left[\frac{Z_b^\tau}{k} \geq \mathbf{E}\left[\frac{Z_b^\tau}{k}\right] \frac{1}{2}\left(\frac{1}{2^{\tau+1}\epsilon\zeta} - 1\right)\right] \\
 &\leq \exp\left(-\frac{k\epsilon M}{6|S^\tau|} \frac{1}{2}\left(\frac{1}{2^{\tau+1}\epsilon\zeta} - 3\right)\right) \quad (\text{Chernoff upper tail}) \\
 &\leq \exp\left(-\frac{k}{3}\left(\frac{\frac{M}{2^{\tau+1}\zeta} - 3\epsilon M}{2|S^\tau|}\right)\right) \\
 &\leq \exp\left(-\frac{k}{3}\left(\frac{1 - 2^{\tau+1}3\zeta\epsilon}{8\zeta}\right)\right) \quad (\text{as } |S^\tau| \leq \frac{M}{2^{\tau-1}}) \\
 &\leq \exp\left(-\frac{k}{3}\left(\frac{1}{16\zeta}\right)\right) \quad (\text{as } \tau < \log_2 \frac{1}{10\epsilon\zeta}) \\
 &= \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}))
 \end{aligned}$$

$\square$

Combining the two results, we can prove the inductive step.

**Lemma A.7.** *If  $b \in V^\tau$  and  $V^\tau \subseteq \overline{B^{\tau-1}}$ , then with probability at least  $1 - \frac{\rho}{\log \frac{1}{10\epsilon\zeta}}$ ,  $b \in V^{\tau+1}$  and  $V^{\tau+1} \subseteq \overline{B^\tau}$*

*Proof.* Union bound over Lemma A.4 and A.6.  $\square$

We consider the base case and show that even in the first round, we shrink  $V^0$  to get  $V^1$  and that we don't remove  $b$ .

**Lemma A.8.** *With prob.  $\geq 1 - \frac{\rho}{\log \frac{1}{10\epsilon\zeta}}$ ,  $V^1 \subseteq \overline{B^0}$  and  $b \in V^1$*

*Proof.*  $\delta^0 = \frac{k}{2}(\frac{1}{2\zeta} - \epsilon)$ . For any fixed  $j \in B^0$ ,  $\mathbf{E}[Z_j^0] \geq k(\frac{1}{2\zeta} - \epsilon)$  (A.3). Probability that  $j \in V^1$  is

$$\begin{aligned} & \mathbf{P}\left[\frac{Z_j^0}{k} \leq \frac{1}{2}\left(\frac{1}{2\zeta} - \epsilon\right)\right] \\ & \leq \mathbf{P}\left[\frac{Z_j^0}{k} - \mathbf{E}\left[\frac{Z_j^0}{k}\right] \leq -\frac{1}{2}\left(\frac{1}{2\zeta} - \epsilon\right)\right] \\ & \leq \exp\left(-\frac{k}{12}\left(\frac{1 - 2\zeta\epsilon}{4\zeta}\right)\right) \quad (\text{Chernoff lower tail}) \\ & \leq \exp\left(-\frac{k}{12}\left(\frac{1}{8\zeta}\right)\right) \quad (\text{as } 1 - 2\zeta\epsilon > 1/2) \\ & \leq \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})) \end{aligned}$$

Thus with probability at least  $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$ ,  $j \notin V^1$

For  $b$ ,  $\mathbf{E}[Z_b^0] \leq \frac{k}{\epsilon}$ . Probability that  $b \notin V^1$

$$\begin{aligned} & \mathbf{P}\left[\frac{Z_b^0}{k} \geq \frac{1}{2}\left(\frac{1}{2\zeta} - \epsilon\right)\right] \\ & \leq \mathbf{P}\left[\frac{Z_b^0}{k} - \mathbf{E}\left[\frac{Z_b^0}{k}\right] \geq \frac{1}{2}\left(\frac{1}{2\zeta} - 3\epsilon\right)\right] \\ & \leq \exp\left(-\frac{k}{3}\left(\frac{1 - 6\zeta\epsilon}{4\zeta}\right)\right) \quad (\text{Chernoff lower tail}) \\ & \leq \exp\left(-\frac{k}{3}\left(\frac{1}{8\zeta}\right)\right) \quad (\text{as } 1 - 6\zeta\epsilon > 1/2) \\ & \leq \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})) \end{aligned}$$

Thus with probability at least  $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$ ,  $b \in V^1$

Union bound over  $j \in B^0$  and over  $b$  proves the statement of the lemma.  $\square$

Now that we have proved the inductive step and the base case, we can use these results to state the result for Phase I.

**Lemma A.9.** *In ActiveHedge (algorithm 2), when Phase I ends after  $T = \frac{1}{10\epsilon\zeta}$  epochs, with probability at least  $1 - \rho$ ,  $b \in V^T$  and for all  $j \in V^T$ ,  $\text{dist}(b, j) \leq 10\epsilon M$*

*Proof.* Using induction and union bound over  $\tau = 1, \dots, T$  for Lemmas A.8 and A.7, we get that with probability at least  $1 - \rho$ ,  $b \in V^T$ , and  $V^T \in \overline{B^{T-1}} \subseteq \left\{j \in [M] \mid \text{dist}(b, j) \leq \frac{M}{2^T\zeta}\right\}$ ,  $\frac{M}{2^T\zeta} = \frac{M}{2^{\log(\frac{1}{10\epsilon\zeta})}\zeta} = 10\epsilon M$   $\square$

Now that we have shown that at the end of Phase I, i.e the *burn-in* period, we have considerably shrunk down our set of candidate experts and thus confusing points. We can prove Theorem 3.1.

Since, ActiveHedge (Algorithm 2) is divided into two phases, a portion of the regret is incurred in each phase. The examples we predict and request labels for in Phase I are denoted by the set  $\text{DONE}$  at the end of Phase I. So the portion of regret incurred in Phase I be  $R^I = \sum_{i \in \text{DONE}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$ . For Phase II, the points are either in  $S^T = \text{POC}_{\mathbf{X}}(V^T)$  where we make hedge updates and request for labels, or they are not in  $\text{POC}_{\mathbf{X}}(V^T)$ , and we use an arbitrary expert  $j^* \in V^T$  to make predictions. Let the regret on the points in  $\text{POC}_{\mathbf{X}}(V^T)$ , i.e. the points of contention for  $V^T$  in phase II

be  $R^{\text{con}} = \sum_{i \in ([M] \setminus \text{DONE}) \cap S^T} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$  and the total regret for the points in Phase II not in  $\text{POC}_{\mathbf{X}}(V^T)$  be  $R^{\text{agree}} = \sum_{i \in ([M] \setminus \text{DONE}) \setminus S^T} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$

*Proof of Theorem 3.1.* First, let's show the regret bound,

Regret Bound:

Since  $\text{REG}_{\text{ActiveHedge}} = R^I + R^{\text{con}} + R^{\text{agree}}$ , let's consider the terms individually.

- $R^I$  and  $R^{\text{con}}$ : We are using Hedge (Algorithm 1) to make predictions and make updates. If we re-sample a point for which we have already made a prediction, we do not incur loss on it again. We know that  $L_b^M \leq \epsilon M$ , hence  $L^* = \epsilon M$  is an upper bound on the loss of the best expert in  $R^I + R^{\text{con}}$  as well. Setting  $\eta = \log \left( 1 + \sqrt{\frac{2 \ln N}{\epsilon M}} \right)$ , we can directly use the regret bound of Theorem 2.1, to show that

$$\begin{aligned} R^I + R^{\text{con}} &= \sum_{i \in \text{DONE} \cup S^T} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i)) \\ &\leq \sum_{i \in \text{DONE} \cup S^T} \ell(\hat{y}_i, y_i) - \min_{j \in [N]} \sum_{i \in \text{DONE} \cup S^T} \ell(X_{i,j}, y_i) \\ &\leq \sqrt{2\epsilon M \ln N} + \ln N \end{aligned}$$

- $R^{\text{agree}}$ : Using Lemma A.9, with probability at least  $1 - \rho$ , the best expert  $b \in V^T$ . Since  $S^T = \text{POC}_{\mathbf{X}}(V^T)$ , all the experts present in  $V^T$  agree on  $[M] \setminus S^T$ . Since  $([M] \setminus \text{DONE}) \setminus S^T \subseteq M \setminus S^T$  all the experts in  $V^T$  agree on all examples in  $([M] \setminus \text{DONE}) \setminus S^T$ . Thus for all  $i \in ([M] \setminus \text{DONE}) \setminus S^T$ , for any  $j \in V^T$ ,  $X_{i,j} = X_{i,b}$ . This is also true for  $j^*$  selected before the start of Phase II, We get

$$\begin{aligned} R^{\text{agree}} &= \sum_{i \in ([M] \setminus \text{DONE}) \setminus S^T} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i)) \\ &= \sum_{i \in ([M] \setminus \text{DONE}) \setminus S^T} \ell(X_{i,j^*}, y_i) - \ell(X_{i,b}, y_i) \\ &= \sum_{i \in ([M] \setminus \text{DONE}) \setminus S^T} \ell(X_{i,b}, y_i) - \ell(X_{i,b}, y_i) = 0 \end{aligned}$$

Thus with probability at least  $1 - \rho$ ,

$$\text{REG}_{\text{ActiveHedge}} \leq \sqrt{2\epsilon M \ln N} + \ln N$$

Label complexity:

Let's consider the number of labels requested in each phase.

- Phase I:

Since number of epochs  $T = \log \frac{1}{10\epsilon\zeta}$  and in each epoch we request the label for  $k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})$  examples, the number of labels requested in Phase I is at most  $192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}) \log \frac{1}{10\epsilon\zeta}$ . This is also the size of the *burn-in* period.

- Phase II:

Using Lemma A.9, with probability at least  $1 - \rho$ , for every  $j \in V^T$ ,  $\text{dist}(b, j) \leq 10\epsilon M$ , thus  $\text{DIAM}(V^T) \leq 20\epsilon M$ . Using the definition of  $\zeta$ -compactness,  $|S^T| = |\text{POC}_{\mathbf{X}}(V^T)| \leq \zeta \text{DIAM}(V^T) \leq 20\epsilon\zeta M$ . Since we only request labels for the examples in  $\text{POC}_{\mathbf{X}}(V^T)$ , the number of labels requested in Phase II is bounded by  $|\text{POC}_{\mathbf{X}}(V^T)|$ , which is less than or equal to  $20\epsilon\zeta M$

Hence with probability at least  $1 - \rho$ , the number of labels requested in Phase II is at most  $20\epsilon\zeta M$

Combining the label complexity for each of the phase, with probability at least  $1 - \rho$ , the number of labels requested by Algorithm 2 is at most

$$O \left( \zeta \log \left( \frac{N}{\rho} \log \frac{1}{10\epsilon\zeta} \right) \log \frac{1}{10\epsilon\zeta} + \epsilon\zeta M \right)$$

Note that the regret bound and the label complexity result hold simultaneously with probability at least  $1 - \rho$ .  $\square$

## B. Halving trick for unknown $\epsilon$

As the algorithm already works by zooming in on tighter error levels, a standard halving technique can be used to easily adapt to an unknown  $\epsilon$  as well. Note that  $k$ 's dependence on  $\epsilon$  is only  $\log \log \frac{1}{\epsilon}$  coming from a union bound which can be upper bounded by  $\log \log M$ . Now to run the algorithm adaptively, instead of fixing  $T$ , we keep the  $\tau$  loop running till error rate of some expert is less than 0.5, i.e.  $\min_j Z_j^\tau / k < 0.5$  and setting  $\delta^\tau \leftarrow \frac{M}{2^{\tau+2}\zeta |\text{POC}_{\mathbf{X}}(V^\tau)|}$ . Using similar steps as Theorem 3.1, this ensures that with high probability, the best expert  $b$  is never removed from  $V^\tau$ . When  $\tau < \log \frac{1}{4\epsilon\theta}$ , we can show that with high probability, the best expert  $b$  has  $Z_b^\tau / k \leq 0.5$  and for  $\tau > \log \frac{1}{2\epsilon\theta}$ ,  $\min_j Z_j^\tau / k > 0.5$ . Thus, without knowing  $\epsilon$ , we stop at the correct,  $\tau$ , leading to the optimal regret of  $\sqrt{2\epsilon M \ln N} + \ln N$  with  $O\left(\zeta \log\left(\frac{N}{\rho} \log M\right) \log \frac{1}{10\epsilon\zeta} + \epsilon\zeta M\right)$ .

## C. Proof of theorem 2.4

*Proof of Theorem 2.4.* If for a set of experts  $V$ , if  $|V| \leq 2$  then  $|\text{POC}_{\mathbf{X}}(V)| = \text{DIAM}(V)$ . Assume  $V$  has all unique experts. For any set  $V \in [N]$ ,  $|\text{POC}_{\mathbf{X}}(V)| \leq M$ , thus  $\zeta \leq M$ .

For any  $V$ , we show that  $|\text{POC}_{\mathbf{X}}(V)| \leq |V| \text{DIAM}(V)$ . Let show this by induction over the size of  $V$ . For  $|V| \leq 2$ , the base cases are direct. Assume that it is true for some  $V$ , i.e.  $|\text{POC}_{\mathbf{X}}(V)| \leq |V| \text{DIAM}(V)$ . If we add one more expert  $h$  to this set, then two cases are possible, a)  $\text{DIAM}(V + h) = \text{DIAM}(V)$  or b)  $\text{DIAM}(V + h) > \text{DIAM}(V)$ .

### a) $\text{DIAM}(V + h) = \text{DIAM}(V)$

We can show that  $|\text{POC}_{\mathbf{X}}(V + h)| \leq |\text{POC}_{\mathbf{X}}(V)| + \text{DIAM}(V)$ . If this is not true, i.e. if  $|\text{POC}_{\mathbf{X}}(V + h)| > |\text{POC}_{\mathbf{X}}(V)| + \text{DIAM}(V)$  then  $h$  disagrees with all  $j \in V$  on at least  $\text{DIAM}(V) + 1$  points which are not in  $\text{POC}_{\mathbf{X}}(V)$ . Thus  $|\text{POC}_{\mathbf{X}}(h, j)| \geq \text{DIAM}(V) + 1 > \text{DIAM}(V)$  which would imply  $\text{DIAM}(V + h) > \text{DIAM}(V)$  which is a contradiction. Thus  $|\text{POC}_{\mathbf{X}}(V + h)| \leq |V + h| \text{DIAM}(V + h)$

### b) $\text{DIAM}(V + h) > \text{DIAM}(V)$

The extra points added in  $\text{POC}_{\mathbf{X}}(V)$  by adding  $h$  is bounded by  $\text{DIAM}(V + h)$ . We get

$$\begin{aligned} |\text{POC}_{\mathbf{X}}(V + h)| &\leq |\text{POC}_{\mathbf{X}}(V)| + \text{DIAM}(V + h) \\ &\leq |V| \text{DIAM}(V) + \text{DIAM}(V + h) \\ &\leq |V + h| \text{DIAM}(V + h) \end{aligned}$$

This implies for any  $V$ ,  $|\text{POC}_{\mathbf{X}}(V)| \leq \text{DIAM}(V)|V|$ . Since  $|V| \leq N$ ,  $\zeta \leq N$ .  $\square$

## D. Proof of Theorem 4.1

*Proof.* Consider the subset

$$V^* = \underset{V, \text{DIAM}(V) > 0}{\text{argmax}} \frac{|\text{POC}_{\mathbf{X}}(V)|}{\text{DIAM}(V)}$$

Let  $h_1, h_2 \in V^*$  be the experts such that  $\text{dist}(h_1, h_2) = \text{DIAM}(V^*)$ . For any  $h' \in V^*$ ,  $\text{dist}(h', h_1) \leq \text{DIAM}(V)$  and  $\text{dist}(h', h_2) \leq \text{DIAM}(V)$ , hence  $h' \in V_{h_1, h_2}$ , i.e.  $V^* \subseteq V_{h_1, h_2}$  in Algorithm 3. This gives us that  $|\text{POC}_{\mathbf{X}}(V_{h_1, h_2})| \geq |\text{POC}_{\mathbf{X}}(V^*)|$

Since we include all experts that are at a distance of at most  $\text{dist}(h_1, h_2)$  from  $h_1$  or  $h_2$ , the diameter  $\text{DIAM}(V_{h_1, h_2}) \leq 3\text{dist}(h_1, h_2) = 3\text{DIAM}(V^*)$

Using these two facts, we get  $\frac{|\text{POC}_{\mathbf{X}}(V_{h_1, h_2})|}{\text{DIAM}(V_{h_1, h_2})} \geq \frac{|\text{POC}_{\mathbf{X}}(V^*)|}{3\text{DIAM}(V^*)} = \frac{\zeta}{3}$

We consider all pairs of experts in Algorithm 3, hence the  $\tilde{\zeta}$  returned satisfies

$$\tilde{\zeta} \geq \frac{|\text{POC}_{\mathbf{X}}(V_{h_1, h_2})|}{\text{DIAM}(V_{h_1, h_2})} \geq \frac{\zeta}{3}$$



For the upper bound, since the  $\tilde{\zeta}$  returned is  $\frac{|\text{POC}_{\mathbf{x}}(V_{j,j'})|}{\text{DIAM}(V_{j,j'})}$  for some  $j, j'$ , it is obvious that

$$\tilde{\zeta} \leq \max_{V, \text{DIAM}(V) > 0} \frac{|\text{POC}_{\mathbf{x}}(V)|}{\text{DIAM}(V)} = \zeta$$

The run time comes from the fact that we consider all  $O(N^2)$  pairs of experts and for any subset  $V \subseteq [N]$ ,  $|\text{POC}_{\mathbf{x}}(V)|$  can be computed in  $O(|V|M)$  and  $\text{DIAM}(V)$  can be computed in  $O(|V|^2M)$   $\square$

### D.1. Proof of Corollary 3.2

*Proof.* In ActiveHedge (Algorithm 2), in the results of Theorem 3.1, the learner is allowed to set the length of the burn-in period itself, i.e. it can decide how many examples that we actually need to actively select and move ahead in the queue. The burn-in phase in Theorem 3.1 is set in such a way that it minimizes the overall label complexity of the the algorithm required to get the same regret bound as Hedge.

If instead of giving the learner the freedom to set its own length of Phase I, if the learner is only given a budget  $B$  of number of examples it can move ahead in the queue, then by setting  $k = \tilde{O}(\zeta)$  and  $T = B/k$ , the size of the burn-in phase becomes  $B$ . At the end of Phase I, in this case, the size of the set of points of contentions, that is  $|\text{POC}_{\mathbf{x}}(V^T)|$  is  $\tilde{O}(\frac{M}{2^{B/\zeta}})$  (Lemma A.2). Thus, the total samples queried would be  $\tilde{O}(B + \frac{M}{2^{B/\zeta}})$ .

Similar to Theorem 3.1, since we don't make any mistakes on the points outside  $\text{POC}_{\mathbf{x}}(V^T)$  in Phase II, the number of mistakes is bounded by the mistakes made by Hedge, resulting in the same regret guarantee.  $\square$

If we were to ignore the mistakes in the learning part, then using an off-the-shelf active learning algorithm (eg (Balcan et al., 2006)) to solve this problem, i) We would need bounded VC dimension  $d$ , and ii) we would require a  $\tilde{O}(\zeta d \log \frac{1}{\epsilon} + \epsilon \zeta M d)$ -long burn-in to ensure an excess error rate of the same order as the  $O(\sqrt{\epsilon M})$  regret on the remainder of the examples.

This brings out a key benefit of our formulation: in pool-based batch active learning, there is no way to separate the number of *targeted queries* (i.e. burn-in) and the *label complexity*; in our online setting the former can be dramatically smaller than the latter.

## E. Auxiliary lemmas

**Lemma E.1** (Chernoff Bounds). *Let  $X_1, \dots, X_n$  be independent random variables, and  $X_i$  lies in the interval  $[0, 1]$ . Define  $X = \sum_{i=1}^n X_i$  and denote  $E[X] = \mu$ . For any  $\delta \in [0, 1]$ , we have **Chernoff lower tail**:*

$$\Pr\{X < (1 - \delta)\mu\} \leq \exp(-\frac{\mu\delta^2}{3})$$

and we have **Chernoff upper tail**:

$$\Pr\{X > (1 + \delta)\mu\} \leq \begin{cases} \exp(-\frac{\mu\delta}{3}) & \text{for } \delta > 1 \\ \exp(-\frac{\mu\delta^2}{3}) & \text{for } \delta \in [0, 1] \end{cases}$$

The proofs for the inequalities in Lemma E.1 can be found in Theorem 4.4 and Theorem 4.5 of (Mitzenmacher & Upfal, 2017)