Learning from partial correction

Sanjoy Dasgupta
UC San Diego

Michael Luby
Qualcomm Technologies, Inc.

Abstract

We introduce a new model of interactive learning in which an expert examines the predictions of a learner and partially fixes them if they are wrong. Although this kind of feedback is not i.i.d., we show statistical generalization bounds on the quality of the learned model.

1 Introduction

Partial correction is a natural paradigm for interactive learning. Suppose, for example, that a taxonomy is to be constructed on a large set of species \( \mathcal{I} \), using steps of interaction with an expert. To see how one such step might go, let’s say the learner’s current model is some hierarchy \( h \). Since it is likely too large to be fathomed in its entirety, a small set of species \( q \subset \mathcal{I} \) is chosen at random (for instance, \( q = \{ \text{dolphin, elephant, mouse, rabbit, whale, zebra} \} \)), and the biologist is shown the restriction of \( h \) to just these species, denoted \( h(q) \). See Figure 1. If this subtree is correct, the biologist accepts it. Otherwise, the expert fixes some part of it. This is easier than fixing the entire subtree.

Earlier models of interactive learning have typically adopted a question-answer paradigm: the learner asks a question and the expert answers it completely. In active learning of binary classifiers, for example, the question is a data point and the answer is a single bit, its label. When learning broader families of structures, however, partial correction can be more convenient and intuitive. In the tree case, the minimal question would consist of three species, and the expert would need to provide the restriction of the target hierarchy to these three leaves. But seeing a larger snapshot is helpful: it provides more context, and thus more guidance about the levels of granularity of clusters; it allows the expert to select one especially egregious flaw to fix, rather than having to correct minor mistakes that might in any case go away once the bigger problems are resolved; and, by allowing choice, it also potentially produces more reliable feedback. Finally, if the subtree is correct, the expert can accept it with a single click, and is saved the nuisance of having to enter it.

Formally, we assume that there is a space of structures \( \mathcal{H} \) (for instance, trees over a fixed set of species), of which some \( h^* \in \mathcal{H} \) is the target. Any \( h \in \mathcal{H} \) can be specified by its answers to a set of questions \( \mathcal{Q} \) (for instance, all subsets of six species). On each step of learning:

- The learner selects some hypothesis \( h \in \mathcal{H} \) based on feedback received so far.
- Some \( q \in \mathcal{Q} \) is chosen at random.
- The learner displays \( q \) and \( h(q) \) to an expert.
- If \( h(q) \) is correct, the expert accepts it. Otherwise, the expert fixes some part of it.

To formalize this partial correction, we assume that each \( h(q) \) contains up to \( c \) atomic components, individual pieces that can be corrected. In the tree example, these are triples of species, so \( c = \binom{6}{3} = 20 \). We index these components as \( (q,1), \ldots, (q,c) \). The expert picks some \( j \) for which \( h(q,j) \neq h^*(q,j) \) and provides \( h^*(q,j) \).

One case of technical interest, which we will later return to, is when the components of \( q \) are independently chosen from the same distribution. We will call such a distribution on queries component independent.
As another example, suppose each \( q \) is a sequence of \( c \) video frames of the driver’s view in a car, and \( h^*(q, j) \) is the appropriate driving action for the \( j \)th frame. On each step of interaction, a human labeler is shown \( c \) frames, each labeled with an action, and either accepts all these actions as reasonable or corrects one of them. In this case it is unlikely that the distribution on queries is component independent.

Formally, on each step of interaction, the learner either finds out that its prediction \( h(q) = (h(q, 1), \ldots, h(q, c)) \) is entirely correct, or receives the correct value \( h^*(q, j) \) for just one atomic component \( j \). This kind of feedback is not i.i.d.: first, the feedback is constrained to be only one component on which \( h \) is incorrect if there is such a component; and second, among possibly several such components, the expert chooses one in some haphazard arbitrary manner. Ideally, the expert’s choices are illustrative and help the learning process, and we will soon see a simple example of this kind. But in this paper we also study the other extreme: is it true that even if the expert adversarially chooses what feedback to give, the same rate of convergence as i.i.d. sampling is always assured? We show that this is indeed the case, and this is a crucial sanity check for the partial correction model. Furthermore, we show that our algorithms are optimal with respect to natural metrics.

### 1.1 Learning procedure

Let \( \mu \) be a probability distribution on \( Q \), and let \( q \in \mu Q \) indicate that \( q \) is chosen independently from \( Q \) according to \( \mu \); in the tree example above, \( Q \) is all subsets of six species and \( \mu \) is the uniform distribution on \( Q \). On step \( t = 1, 2, \ldots \) of learning,

1. Learner selects some \( h_t \in H \) consistent with all feedback received so far
2. Choose \( q \in \mu Q \), where \( q \) has \( c \) atomic components, \( (q, 1), \ldots, (q, c) \).
3. Learner displays \( q \) and \( h_t(q) \) to expert
4. If \( h_t(q) \) is correct:
   - Expert feeds back that \( h_t(q) \) is correct
   - Feedback implicitly provides, for all \( j \in [c] \), \( h^*(q, j) = h_t(q, j) \).

Else \( h_t(q) \) is incorrect:
   - Expert chooses \( 1 \leq j \leq c \) for which \( h_t(q, j) \neq h^*(q, j) \)
   - Expert feeds back \( j \) and \( h^*(q, j) \).
1.2 Results

The error of a hypothesis \( h \in \mathcal{H} \) can be measured in two ways: in terms of full questions \( q \in \mathcal{Q} \),

\[
err(h) = \Pr_{q \in \mathcal{Q}}[h(q) \neq h^*(q)].
\]

or in terms of atomic components \((q,j)\):

\[
err_c(h) = \Pr_{q \in \mathcal{Q}, j \in [c]}[h(q,j) \neq h^*(q,j)].
\]

These are related by \( err_c(h) \leq err(h) \leq c \cdot err_c(h) \). Note that \( err_c(h) \approx err(h)/\mu \) if \( \mu \) is component independent and \( err(h) \) is small.

An important complexity metric is the expert cost per step to provide feedback. This cost can be substantially lower in the new model: The expert can choose a component that is easiest to determine is incorrect amongst a set of \( c \) components, instead of being required to provide feedback for a particular component. We leave to future work the study of this metric in more detail.

Another crucial complexity metric is the number of steps of feedback required to learn. We start with a simple one-dimensional example (Section 2) that illustrates how the expert’s choice of feedback can significantly affect this metric. In the example, one feedback strategy reduces the number of steps needed for learning by a factor of up to \( c \) (so that each feedback component is about as valuable as \( c \) randomly chosen components), while a different strategy increases the number of steps by a factor of \( \Omega(c) \) (slows down learning).

The example demonstrates that the number of steps needed to learn can vary by wide margins depending on the expert policy. Our main result (Theorem 2) shows that, despite this, there is a reasonable bound on the number of steps to learn no matter how adversarial the expert policy: For any expert policy, for any \( 0 < \delta, \epsilon < 1 \), with probability \( 1 - \delta \) the base algorithm of Section 1.1 produces a hypothesis \( h \) with \( err(h) \leq \epsilon \) within \( O((c/\epsilon) \cdot \log(|\mathcal{H}|/\delta)) \) steps of feedback. Moreover (Theorem 8), after the same number of steps, all remaining consistent hypotheses have \( err_c(h) \leq \epsilon \). Section 6 shows that this number of steps is needed for at least some examples.

In the standard supervised learning model, labeled data is provided in advance, after which a consistent hypothesis is sought. In our protocol, feedback is obtained in steps, and the learner needs to maintain a consistent hypothesis throughout the process. Because it can be expensive to continually select a consistent hypothesis, we introduce the stick-with-it algorithm, a variant of the base algorithm, that might be preferable in practice (Section 5). Rather than always having to select a hypothesis that is consistent with all feedback received so far at each step, it only updates its hypothesis \( O(c) \) times during the entire learning process. It works for any concept class of bounded VC dimension \( d \), it obtains a hypothesis \( h \) with \( err(h) \leq \epsilon \) within \( O((c/\epsilon) \cdot (d + \log(1/\delta))) \) steps of feedback.

To obtain these sample complexity bounds, we look at the effective distribution \( w_t \) over atomic components \((q,j)\) at each time step \( t \), which is a function of previous feedback, the learning algorithm’s choice of current model \( h_t \), and the expert’s criterion for selecting what to correct. This can be quite different from the distribution that would be easy to analyze, where \( q \in \mu \mathcal{Q} \) and \( j \) is chosen at random; in particular, \( w_t \) can be zero at many \((q,j)\) with \( \mu(q) > 0 \). Nonetheless, we show that over time, no matter what policy the expert chooses, \( w_t \) cannot avoid covering the whole \( \mathcal{Q} \times [c] \) space in some suitably amortized sense.

The growing area of interactive learning raises many new problems and challenges. Here we have formalized an interactive protocol that is quite natural and intuitive in terms of human-computer interface, but breaks the statistical assumptions that underlie generalization results in other settings like the PAC model. Our key technical contribution is to establish sample complexity bounds in this novel framework.

2 An illustrative example

Suppose \( \mathcal{X} = [0,1] \) and the goal is to learn a threshold classifier:

\[
\mathcal{H} = \{ h_v : v \in [0,1] \}, \quad h_v(x) = 1(x > v).
\]
Say the target threshold is 0 (that is, \( h^* = h_0 \)), so that the correct label for all points in \((0, 1]\) is 1. If we were learning from random examples \((x, h^*(x))\) then, independent of the distribution on \(X\), after \(O(1/\epsilon)\) samples all remaining consistent hypotheses would (with probability close to one) consist entirely of classifiers \(h\) with \(\epsilon(h) \leq \epsilon\). Thus, after \(O(1)\) instances, the error would be lower than any pre-specified constant.

### 2.1 Uniformly distributed, component independent queries

Let \(\mu_c\) be the uniform distribution on atomic components \(X\), and let \(\mu\) be the component independent distribution over the space of queries \(Q = X^c\) induced by \(\mu_c\). Thus, for any \(v \in [0, 1]\), \(\epsilon_c(h_v) = v\) and \(\epsilon(h_v) = 1 - (1 - v)^c\), and thus \(\epsilon_c(h) \approx \epsilon(h)/c\) if \(\epsilon(h)\) is small. We’ll try to understand how the rate of convergence of \(v_t\) to zero is affected by \(c\) and by the expert labeler’s policy for which errors to correct, where \(v_t\) is the threshold the learner has determined at step \(t\) steps, i.e., \(v_t\) is the smallest component value for which feedback has been provided by the expert and thus only classifiers with threshold at most \(v_t\) are still consistent with the feedback through \(t\) steps.

We consider two expert policies:

- “Largest”: the expert picks the largest component in error. This corresponds to a natural tendency to fix the biggest mistake, but happens to be the least informative correction.

- “Smallest”: the expert picks the smallest component in error. This is the most informative correction.

The next query is chosen by picking \(x_1, \ldots, x_c \in R [0, 1]\). Let \(V_{t+1}\) be the random variable that is the threshold value the learner determines at step \(t + 1\). What is the expected value of \(V_{t+1}\)?

**When the labeling policy is “largest”:** For any \(v \in [0, v_t]\), the only way \(V_{t+1}\) can exceed \(v\) is if either all the \(x_i\) are \(\geq v_t\) (and thus they are all correctly labeled by \(h_{v_t}\)) or if at least one of the \(x_i\) lies in \((v, v_t)\) (in which case, there is at least one error, but the largest component in error exceeds \(v_t\)):

\[
\Pr(V_{t+1} > v \mid V_t = v_t) = \Pr(\text{all } x_i \geq v_t) + (1 - \Pr(\text{no } x_i \text{ in } (v, v_t)))
\]

\[
= (1 - v_t)^c + (1 - (1 - (v_t - v))^c)
\]

Therefore, by calculation,

\[
\mathbb{E}[V_{t+1} \mid V_t = v_t] = \int_0^{v_t} \Pr(V_{t+1} > v \mid V_t = v_t)dv = v_t - \frac{1 - (1 - v_t)^c \cdot (1 + c \cdot v_t)}{c + 1}.
\]

**When the labeling policy is “smallest”:** For \(v \in [0, v_t]\), the only way \(V_{t+1}\) can exceed \(v\) is if none of the \(x_i\) lie in \([0, v]\), so \(\Pr(V_{t+1} > v \mid V_t = v_t) = (1 - v)^c\), whereupon, by a similar integral,

\[
\mathbb{E}[V_{t+1} \mid V_t = v_t] = \frac{1 - (1 - v_t)^{c+1}}{c + 1}.
\]

When \(c = 1\), the two policies coincide and \(\mathbb{E}[V_{t+1} \mid v_t] = v_t - v_t^2/2\), so the expected instantaneous reduction in \(V_t\), that is \(\mathbb{E}[v_t - V_{t+1}]\), from seeing a single-point query is \(v_t^2/2\). How does this compare to the expected instantaneous reduction from queries consisting of \(c\) points? The ratio of the expected reduction with \(c\)-point queries to the expected reduction with 1-point queries is shown in Figure 2 for \(c = 4, 8\) and for the “smallest”, “largest” labeling policies. The ratio is given at each value \(v_t\).

As expected, under the “smallest” labeling policy, \(c\)-point queries are always more helpful than single-point queries. Under the “largest” policy, this is true only when \(v_t\) is sufficiently small. In either case, when \(v_t\) gets close to zero, the single label yielded by a \(c\)-point query is roughly as informative as \(c\) random labeled points. This can be checked using simple approximations to the expressions obtained above.

This toy example shows that the rate of convergence of learning by partial correction depends on the labeler’s policy for deciding which error to fix. Even in a one-dimensional setting with few degrees of freedom, different labeling policies can cause convergence to either slow down or speed up, by factors of up to \(c\). We now formalize lower bounds of this type.
2.2 A lower bound on component-level error

We continue with the one-dimensional example, but now turn to distributions that are not component independent.

We will consider a learner that begins with a threshold of 1, and at any given time, chooses the largest threshold consistent with all feedback so far: namely, the smallest labeled point it has received.

2.2.1 A single query, repeated

To start with an especially simple case, say the distribution $\mu$ over $Q$ is supported on a single point, $(1/c, 2/c, \ldots, 1)$. Suppose moreover that the expert labeler behaves as follows: when presented with a labeling of the points $1/c, 2/c, \ldots, 1$, he/she always chooses to “correct the most glaring flaw”, that is, the highest value for which a 0 label is suggested.

It is clear that $x = 1$ is labeled in the first round, $x = (c - 1)/c$ in the second round, $x = (c - 2)/c$ in the third round, and so on. The labeler’s behavior is hardly pathological. And yet, it takes $c/2$ rounds of interaction to bring the error down to $1/2$. If the feedback were on random components, then $O(1)$ rounds would have been sufficient.

2.2.2 Lower bound

Pick any $\epsilon > 0$, and now consider a distribution $\mu$ over $Q$ that is supported on just two points:

$$
\begin{align*}
\left(\frac{1}{2c}, \frac{2}{2c}, \ldots, \frac{1}{2}\right) & \quad \text{probability } 2\epsilon \\
\left(\frac{1}{2} + \frac{1}{2c}, \frac{1}{2} + \frac{2}{2c}, \ldots, 1\right) & \quad \text{probability } 1 - 2\epsilon
\end{align*}
$$

Any hypothesis with $\text{err}_c(h_o) \leq \epsilon$ must have $v \leq 1/4$. In order to achieve this, the learner must see the first point at least $c/2$ times, which requires seeing $\Omega(c/\epsilon)$ samples overall, with high probability.

We have established the following.

**Theorem 1** There is a concept class $\mathcal{H}$ of VC dimension 1 such that for any $\epsilon > 0$, it is necessary to have $\Omega(c/\epsilon)$ rounds of feedback in order to be able to guarantee that with high probability, all hypotheses $h$ consistent with this feedback have $\text{err}_c(h) \leq \epsilon$. 

---

Figure 2: The ratio between the expected reduction in error from a query consisting of $c$ points versus a query consisting of a single point, for two labeling policies (“smallest” and “largest”) and values $c = 4, 8$. 
3 Main result

For each \( h \in \mathcal{H} \), let

\[
B(h) = \{ q \in \mathcal{Q} : h \text{ is incorrect on } q \}, \\
G(h) = \{ q \in \mathcal{Q} : h \text{ is correct on } q \}
\]

Note that \( \text{err}(h) = \mu(B(h)) \) is the probability that \( h \) is incorrect on a randomly chosen query. We say that hypothesis \( h \) is \( \epsilon \)-good if \( \mu(B(h)) \leq \epsilon \). On input \( (\epsilon, \delta) \), the goal is to find an \( h \in \mathcal{H} \) that is \( \epsilon \)-good with probability at least \( 1 - \delta \).

Let \( \ell = \log(H/\delta) \), let \( \epsilon' = \epsilon/2 \), and let \( N = c \cdot \left( \frac{\ell}{\epsilon'} + 1 \right) \).

**Theorem 2** The base algorithm of Section 1,4 produces an \( \epsilon \)-good hypothesis within \( 2 \cdot N \) steps with probability at least \( 1 - \delta \).

It is interesting to compare Theorem 2 when \( \mu \) is component independent to classical learning theory. Theorem 2 shows that after at most \( 2 \cdot N = O(c \cdot \log(|\mathcal{H}|)/\epsilon) \) steps the output hypothesis \( h \) satisfies \( \text{err}(h) \leq \epsilon \), which implies \( \text{err}_c(h) \leq \epsilon/c \) if \( \mu \) is component independent. This is the same number of steps the classical model would take to achieve component error \( \epsilon/c \) when each question is a single component and the expert provides complete feedback for each question. Of course, the bound of Theorem 2 applies whether or not \( \mu \) is component independent.

The remainder of this section concentrates on proving Theorem 2. The analysis proceeds in two phases: the first phase considers the first \( N \) steps, and the second phase considers the subsequent \( N \) steps. Let

\[
\hat{\mathcal{Q}} = \mathcal{Q} \times [c], \\
\hat{B}(h) = \{ (q,j) \in \hat{\mathcal{Q}} : q \in B(h) \text{ and } h(q,j) \neq h^*(q,j) \}, \\
\hat{G}(h) = G(h) \times [c].
\]

3.1 Effective sampling distribution

At the beginning of step \( t \), some hypothesis \( h_t \) is selected. The analysis defines an effective sampling distribution \( w_t \) over \( \mathcal{Q} \), as follows:

- For all \( (q,j) \in \hat{B}(h_t) \), let \( \gamma(q,j) \) denote the conditional probability that the expert provides feedback on \( (q,j) \) when query \( q \) is made. Define \( w_t(q,j) = \mu(q) \cdot \gamma(q,j) \).
- For all \( q \in \hat{G}(h_t) \) calculate \( w_t(q,1), \ldots, w_t(q,c) \), summing to \( \mu(q) \), as specified below in Lemma 3.

Finally, let

\[
W_t(q,j) = w_t(q,j) + \cdots + w_t(q,j)
\]

denote the sum of the individual distributions up to step \( t \). Note that at each step \( t \), for each \( q \in \mathcal{Q} \), we have \( w_t(q,[c]) = \mu(q) \) and thus \( W_t(q,[c]) = t \cdot \mu(q) \).

**Lemma 3** For all \( q \in \hat{G}(h_t) \), non-negative values for \( w_t(q,1), \ldots, w_t(q,c) \), summing to \( \mu(q) \), can be calculated such that if \( w_t(q,j) > 0 \) then

\[
W_t(q,j) = W_{t-1}(q,j) + w_t(q,j) \leq t \cdot \frac{\mu(q)}{c}.
\]

**Proof:** Since \( W_t(q,[c]) \) will be equal to \( t \cdot \mu(q) \), \( \frac{t \cdot \mu(q)}{c} \) will be the average over \( j \in [c] \) of \( W_t(q,j) \). Thus, we can choose \( w_t(q,1), \ldots, w_t(q,c) \) as follows. Let \( j_1, \ldots, j_c \) be an ordering of the elements of \([c]\) such that

\[
W_{t-1}(q,j_1) \leq W_{t-1}(q,j_2) \leq \cdots \leq W_{t-1}(q,j_c).
\]
Initialize $\Delta = \mu(q), w_t(q,j_1) = \cdots = w_t(q,j_{c}) = 0$. Repeat the following for $i = 1, \ldots, c$ until $\Delta = 0$: Reset

$$w_t(q,j_i) = \min \left\{ \frac{t \cdot \mu(q)}{c} - W_{t-1}(q,j_i), \Delta \right\}$$

and reset $\Delta = \Delta - w_t(q,j_i)$. □

3.2 Eliminating inconsistent hypotheses

**Lemma 4** With probability at least $1 - \delta$, the following holds for all $h \in \mathcal{H}$: if there is a step $t$ at which $W_t(\bar{B}(h)) \geq \ell$, then $h$ is not consistent with the feedback received up to that step.

**Proof:** Pick any $h \in \mathcal{H}$. It is eliminated if feedback is received on any $(q,j) \in \bar{B}(h)$. The probability that this happens at step $t$ is at least $w_t(\bar{B}(h))$.

Let $t$ be the first step at which $W_t(\bar{B}(h)) \geq \ell$. The probability that $h$ is not eliminated by the end of step $t$ is at most

$$1 - w_1(\bar{B}(h)) \cdot (1 - w_2(\bar{B}(h))) \cdots (1 - w_t(\bar{B}(h))) \leq \exp(-W_t(\bar{B}(h))) \leq \exp(-\ell) = \frac{\delta}{|\mathcal{H}|}.$$

Taking a union bound over $\mathcal{H}$, with probability at least $1 - \delta$, any hypothesis $h$ is eliminated from the version space by the step at which $W_t(\bar{B}(h)) \geq \ell$. □

At the outset of any step $t$, we may therefore assume that $h_t$ satisfies $W_{t-1}(\bar{B}(h_t)) < \ell$.

3.3 Analysis for Phase 1

Consider a first phase consisting of the first $N$ steps. Let

$$\tau = \frac{N}{c} = \frac{\ell}{c} + 1$$

be a threshold value. We will think of an atomic question $(q,j)$ as having been adequately sampled when $W_t(q,j)$ reaches $\tau \cdot \mu(q)$.

At the beginning of step $t$, let

$$\bar{L}_{t-1} = \{ (q,j) \in \bar{Q} : W_{t-1}(q,j) \leq \tau \cdot \mu(q) \}.$$

Thus

$$W_{t-1}(\bar{L}_{t-1}) = \sum_{(q,j) \in \bar{L}_{t-1}} W_{t-1}(q,j) \leq c \cdot \tau = N$$

(substituting a suitable integral over $\bar{Q}$ if necessary). Let

$$\bar{L}_{t-1} = \{ (q,j) \in \bar{Q} : W_{t-1}(q,j) \leq (\tau - 1) \cdot \mu(q) = \frac{\ell}{c} \cdot \mu(q) \}.$$

This is the set of $(q,j)$ in which there is still some remaining capacity.

**Lemma 5** At any step $t$, if $W_{t-1}(\bar{B}(h_t)) < \ell$, then

$$w_t(\bar{B}(h_t) \cap \bar{L}_{t-1}) \geq \mu(B(h_t)) - \epsilon'.$$

**Proof:** Note that

$$\mu(B(h_t)) = w_t(\bar{B}(h_t)) = w_t(\bar{B}(h_t) \cap \bar{L}_{t-1}) + w_t(\bar{B}(h_t) \setminus \bar{L}_{t-1}). \quad (1)$$

Since

$$\ell > W_{t-1}(\bar{B}(h_t)) \geq W_{t-1}(\bar{B}(h_t) \setminus \bar{L}_{t-1}) \geq \frac{\ell}{c} \cdot w_t(\bar{B}(h_t) \setminus \bar{L}_{t-1}),$$

it follows that $w_t(\bar{B}(h_t) \setminus \bar{L}_{t-1}) \leq \epsilon'$ and thus $w_t(\bar{B}(h_t) \cap \bar{L}_{t-1}) \geq \mu(B(h_t)) - \epsilon'$ follows from Equation (1). □
Lemma 6 At any step \( t \leq N \), \( w_t(\bar{L}_t) \geq 1 - \epsilon' \).

Proof: Note that
\[
w_t(\bar{L}_t) = w_t(\bar{B}(h_t) \cap \bar{L}_t) + w_t(\bar{G}(h_t) \cap \bar{L}_t).
\]
Since any \((q,j) \in \bar{B}(h_t) \cap \bar{L}_{t-1}\) satisfies \((q,j) \in \bar{B}(h_t) \cap \bar{L}_t\), Lemma 5 implies \( w_t(\bar{B}(h_t) \cap \bar{L}_t) \geq \mu(B(h_t)) - \epsilon' \).

For \( q \in \bar{G}(h_t) \), from Lemma 3 and noting that \( t \leq N \), any \((q,j)\) with \( w_t(q,j) > 0 \) satisfies
\[
W_t(q,j) \leq \frac{t \cdot \mu(q)}{c} \leq \tau \cdot \mu(q),
\]
and thus \((q,j) \in \bar{L}_t\), and it follows that \( w_t(\bar{G}(h_t) \cap \bar{L}_t) = \mu(G(h_t)) \). Overall,
\[
w_t(\bar{L}_t) \geq \mu(B(h_t)) - \epsilon' + \mu(G(h_t)) = 1 - \epsilon'.
\]

Let \( \hat{W}_t(q,j) = \min\{W_t(q,j), \tau \cdot \mu(q)\} \). As we have seen, \( \hat{W}_t(\bar{Q}) \leq N \).

Corollary 7 \( \hat{W}_N(\bar{Q}) \geq (1 - \epsilon') \cdot N \).

Proof: An immediate consequence of Lemma (6).

3.4 Analysis for Phase 2

We now finish the proof of Theorem 2. Proof: Consider a second phase of \( N \) additional steps. Let \( h_t \) be the current hypothesis for one of these steps. If \( \mu(B(h_t)) \geq 2 \cdot \epsilon' \) then \( \mu(B(h_t)) - \epsilon' \geq \epsilon' \), and the first half of the proof of Lemma 6 implies that \( \hat{W}_t(\bar{Q}) \) increases by at least \( \epsilon' \) during this step. However, since \( \hat{W}_t(\bar{Q}) \leq N \), and since \( \hat{W}_N(\bar{Q}) \geq (1 - \epsilon') \cdot N \) at the beginning of the second phase from Corollary 7, there can be at most \( N \) steps in the second phase where \( \hat{W}_t(\bar{Q}) \) increases by at least \( \epsilon' \). Thus, during one of the steps in the second phase \( \mu(B(h_t)) \leq 2 \cdot \epsilon' = \epsilon \), at which point the base algorithm can select \( h_t \) as an \( \epsilon \)-good hypothesis and terminate. This concludes the proof of Theorem 2.

4 Generalization bound

The following generalization bound holds at the end of Phase 1.

Theorem 8 With probability at least \( 1 - \delta \), any \( h \in \mathcal{H} \) that remains in the version space at the end of Phase 1 (that is, after \( N = c \cdot (\frac{t}{t} + 1) \) rounds of feedback) has \( \text{err}_e(h) < \epsilon \).

Proof: Let \( \bar{\mu} \) be the distribution over \( \bar{Q} \) that corresponds to picking \( q \) from \( \mu \) and then picking a feature at random: \( \bar{\mu}(q,j) = \mu(q)/c \). Thus for any \( h \in \mathcal{H} \), we have \( \text{err}_e(h) = \bar{\mu}(\bar{B}(h)) \).

At the end of Phase 1, \( \hat{W}_N(\bar{Q}) \geq (1 - \epsilon') \cdot N \). Thus for any \( h \in \mathcal{H} \),
\[
\hat{W}_N(\bar{B}(h)) \geq \left( \sum_{(q,j) \in \bar{B}(h)} \tau \cdot \mu(q) \right) - \epsilon' \cdot N = \left( \sum_{(q,j) \in \bar{B}(h)} N \cdot \bar{\mu}(q,j) \right) - \epsilon' \cdot N = N \cdot (\bar{\mu}(\bar{B}(h)) - \epsilon').
\]

If \( \bar{\mu}(\bar{B}(h)) \geq \epsilon = 2 \cdot \epsilon' \), we get
\[
W_N(\bar{B}(h)) \geq \hat{W}_N(\bar{B}(h)) \geq N \cdot \epsilon' > c \cdot \ell.
\]
By Lemma 4 with probability at least \( 1 - \delta \), any such \( h \) is eliminated by the end of the \( N \)th step.

Recall from Theorem 1 that this \( c/\epsilon \) dependence is inevitable.
5 Stick-with-it algorithm

There are some practical issues with the base algorithm of Section 1.1. One issue is that at the beginning of each step a current hypothesis needs to be selected that is consistent with all previous steps. This results in a number of selections equal to the number of steps.

A second issue is that a separate procedure is needed to evaluate whether a given hypothesis is $\epsilon$-good in order to terminate the base algorithm with a hypothesis that is verified to be $\epsilon$-good, since a current hypothesis may be the current hypothesis for only one step.

A third issue is that the analysis doesn’t obviously carry over to the case where $H = |H|$ is unbounded but instead the VC-dimension of $H$ is bounded.

We introduce the stick-with-it algorithm, a variant of the base algorithm, that addresses all these issues.

We use an integer $k \geq 1$ to describe the following changes to the base algorithm:

- Instead of selecting a current hypothesis at each step, a current hypothesis is selected each $k$ consecutive steps. Thus, once a current hypothesis is selected it is used as the current hypothesis for $k$ consecutive steps. (This is where "stick-with-it" comes from.)

- Redefine $N = c \cdot (\frac{\ell}{\epsilon} + k)$. All parameters defined in terms of $N$ are similarly redefined, e.g., $\tau = \frac{\ell}{\epsilon} + k$.

- Redefine $L'_t = \{ (q,j) \in \bar{Q} : W_t(q,j) \leq (\tau - k) \cdot \mu(q) = \frac{\ell}{\epsilon} \cdot \mu(q) \}$.

Essentially the same analysis as described for the base algorithm shows that the stick-with-it algorithm produces an $h \in H$ that is $\epsilon$-good with probability at least $1 - \delta$ within $2 \cdot N$ steps.

Setting $k = \frac{\ell}{\epsilon} = \frac{2 c \cdot \ell}{\epsilon}$ results in a stick-with-it algorithm with the following properties:

- The number of steps is at most $2 \cdot N \leq \frac{8 c \cdot \ell}{\epsilon}$.

- The number of steps a current hypothesis needs to be selected that is consistent with all previous steps is at most $\frac{2 N}{k} \leq 4 \cdot c$.

- A selected current hypothesis is the current hypothesis for enough steps ($k$) to determine if it is $\epsilon$-good, and if it is $\epsilon$-good then the stick-with-it algorithm terminates.

- Let $d$ be the VC-dimension of $H$. The same analysis applies when $\ell = d + \log(1/\delta)$.

The stick-with-it algorithm is optimal in the following metrics:

- The bound on the number of steps, including steps to verify that the output hypothesis is $\epsilon$-good, is within a constant factor of the number needed for some examples. (See Section 6)

- The bound on the number of selections of a current hypothesis is within a constant factor of the number needed for some examples. (See Section 6)

- The same optimality holds when the VC-dimension of $H$ is bounded.

6 Lower bound on number of steps and selected hypotheses

An example showing the number of needed steps can be proportional to $\frac{c \cdot \log |H|}{\epsilon}$ is obtained as follows. Let $Q^*$ be any fixed subset of $Q$ such that $|Q^*| = \epsilon \cdot |Q|$. Let $h^*$ be the correct hypothesis. For $i = 1, \ldots, c/2$, let $H_i$ consist of all classifiers $h$ with the following properties:

- $h(q,j) = h^*(q,j)$ if $q \notin Q^*$ or $j \notin \{2i, 2i - 1\}$.

- For each $q \in Q^*$, either
  - $h(q, 2i - 1) \neq h^*(q, 2i - 1)$ and $h(q, 2i) = h^*(q, 2i)$, or
\[- h(q, 2i - 1) = h^*(q, 2i - 1) \text{ and } h(q, 2i) \neq h^*(q, 2i) \]

Notice that any such \( h \) has \( \text{err}(h) = \epsilon \), and that \( |H_i| = 2^{|Q^*|} \). Finally, define

\[
H = \bigcup_{i=1}^{c/2} H_i \cup \{h^*\}.
\]

Write \( \ell = |Q^*| = \epsilon \cdot Q \). Then \( \log |H| \approx \ell + \log c \approx \ell \).

Let’s say the learner always prefers to select a hypothesis in \( H_1 \), then \( H_2 \), and so on, and will only select \( h^* \) when all these options have been exhausted. When a hypothesis in \( H_i \) is selected, it will make a mistake whenever the query lies in \( Q^* \). This mistake will provide no information about any other \( q \in Q^* \), or about any other hypothesis in \( H_{i'} \) for \( i' \neq i \).

Thus, on average \( \ell/\epsilon \) queries will be needed to eliminate \( H_i \) for each \( i = 1, \ldots, c/2 \), for a total query complexity of roughly \( \frac{c\ell}{2\epsilon} \).

In this example the number of selected hypotheses is at least \( c/2 \), i.e., at least one hypothesis from each of \( H_1, \ldots, H_{c/2} \) is selected as the current hypothesis at some step.

## 7 Acknowledgments

This work is a direct result of the Foundations of Machine Learning program at the Simons Institute, UC Berkeley.

## References

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