COMS 6998-4 Fall 2017 Presenter: Arushi Gupta
November 20, 2017 Scribe: Jinyi Zhang

Learning from Partial Correction

1 Introduction

In active learning, given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and a set of hypothesis $\mathcal{H}$ that maps $\mathcal{X}$ to $\mathcal{Y}$, the learner observes an data point $x_t \in \mathcal{X}$ and decides whether or not to query its label at time t. In this interactive learning model, the learner asks a question and receives a complete answer at a time. In a binary classification problem, a complete answer is a single bit. However, as the learning problem becomes much more complex than predicting a single bit, it might be too expensive to give a complete answer at each round of the interaction. Thus, we introduce the idea of *partial feedback* for interactive learning, in which an expert examines the predictions of a learner and partially fixes them if they are wrong. For example, let's consider a hierarchical clustering problem. Given a huge set of species, the learner aims to build a tree over this set. Instead of giving the entire tree to the expert and asking the expert to label every corner of the tree, the learner actually takes a small subset of species, builds the subtree over this subset, and shows the subtree to the expert for feedback. We denote the small subset of species as a question q and denote the learner's prediction as $h(q)$. If $h(q)$ is correct, the expert accepts it; otherwise, the expert provides a partial feedback such as a minor structure that the correct tree must satisfy. This kind of feedback is easier than fixing the entire $h(q)$. As shown in Figure 1, a small subset (question) $q = \{\text{dolphin, elephant, mouse, rabbit, whale, zebra}\}$ is randomly chosen, and the learner returns $h(q)$ as the left part of the figure. The expert provides feedback, as the right part of the figure, of the form $h^*(x)$, where $x = \{\text{dolphin, whale, zebra}\} \subset q$ on which h is not correct, and h^* is the target hierarchy.

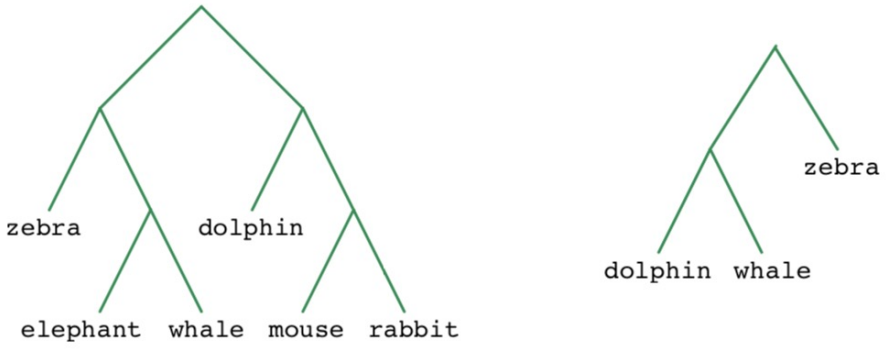

Figure 1: Left: The learner makes a prediction on a subset of species. Right: The expert gives a feedback to correct a part of this subtree.

Formally, we propose the partial feedback interactive learning framework as the following. There is a hypothesis class $\mathcal{H}$ and a target hypothesis $h^* \in \mathcal{H}$. Any hypothesis $h \in \mathcal{H}$ can be uniquely identified by its answer of a set of questions $\mathcal{Q}$. For each round of learning,

- The learner selects a hypothesis $h \in \mathcal{H}$ based on all previously received feedback.
- A question $q \in \mathcal{Q}$ 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 corrects a part of it.

Now, new problems pops up. What do we mean by "part of it"? And how to we choose the question q for each round? We assume that q has c atomic components, and we denote components as $(q, 1), ..., (q, c)$. Correcting a part of q means that the expert picks an index j from 1 to c where $h(q, j) \neq h^*(q, j)$ and reveals $h^*(q, j)$ to the learner. Also, we define μ as the probability distribution on $\mathcal{Q}$ and write $q \in_\mu \mathcal{Q}$ to indicate q was chosen according to probability distribution μ from $\mathcal{Q}$ and $[c] = \{1, 2, ..., c\}$. Thus, we could measure the error of a hypothesis either by the full question q like

$$\text{err}(h) = \Pr_{q \in_\mu \mathcal{Q}}[h(q) \neq h^*(q)]$$

or in terms of components like

$$\text{err}_c(h) = \Pr_{q \in_\mu \mathcal{Q}, j \in_R [c]}[h(q, j) \neq h^*(q, j)]$$

where R is some probability distribution over $[c]$.

2 Threshold functions

Let's consider a concrete example. Suppose we have $\mathcal{X} = [0, 1]$ and we aim to learn a threshold function. Therefore, we can write our hypothesis class $\mathcal{H}$ as $\mathcal{H} = \{h_v : v \in [0, 1]\}$ and we have $h_v(x) = \mathbb{1}_{\{x > v\}}$.

Figure 2: A threshold function h_v over $[0, 1]$.

Suppose our target hypothesis $h^* = h_0$. This means the threshold classifier we want to learn returns 0 if $x = 0$ and returns 1 otherwise. And our queries will consist of c numbers

in $[0,1]$ ($\mathcal{Q} = \mathcal{X}^c$). These numbers are our atomic components. Assume μ is the uniform distribution over the components, then we have

$$\mathrm{err}_c(h_v) = v$$

for any $v \in [0,1]$ in this particular case $h^* = h_0$. This is obvious since the disagreement region between h^* and h_v is $(0, v]$. And we can compute the error of h_v by the full answer as

$$\mathrm{err}(h_v) = 1 - (1-v)^c,$$

where $(1-v)^c$ is the probability that all the c components are correct.

Let v_t be the threshold learned so far by the learner. It matters that which component the expert chooses to label during the interactive learning process. At the beginning, the learner labels $[0, 1)$ as 0 and point 1 as 1 because he picks threshold classifier that consistent with all data samples seen so far and he hasn't seen any samples yet. At each round, the learner picks c points to label and the expert chooses one of them to correct if there exists at least one error. There are two policies that the expert can use for picking a component.

- Largest. For the c points, the expert picks the largest point in error to correct.
- Smallest. For the c points, the expert picks the smallest point in error to correct.

The "largest" policy seems to be intuitive because the expert tries to fix the largest error in the c components. However, this would be the least informative correction and it would take much more steps for the learner to realize the target threshold. On the other hand, the "smallest" policy is more helpful as the learner could approach the target much faster.

Let V_{t+1} be the random variable that is the threshold value the learner determines at step $t+1$. And we denote the c components as $x_1, ..., x_c \in [0,1]$. We are looking for how the two policies compare in terms of the expectation value of $v_t - V_{t+1}$ as the benefit from step t.

Let's consider a point v in $[0, v_t)$. If the expert chooses the "largest" policy, V_{t+1} can exceed v only if there is at least one component x_i lies in (v, v_t) or all the c components are to the right of v_t. We show this in Figure 3. On the other hand, if the expert chooses the "smallest" policy, V_{t+1} can exceed v only if none of x_i lies in $[0, v]$ as shown in Figure 4. Then, we have

$$\mathbb{E}[V_{t+1}|V_t = v_t] = \int_0^{v_t} \Pr(V_{t+1} > v | V_t = v_t)$$

as the expectation of V_{t+1} given v_t. For the "largest" policy, we can calculate the probability and then determine the integral for the expectation value as

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

Similarly, we have

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

for the "smallest" policy. If $c = 1$, then the two policies yield the same value as $v_t - v_t^2/2$. Then, the expected reduction (benefit) from step t, that is $\mathbb{E}[v_t - V_{t+1}]$ would be $v_t^2/2$. However, when $c \neq 1$, the two policies apparently provide different amount of information. In Figure 5, we show the ratio of the expected reduction with c-point queries to the expected reduction with 1-point queries for $c = 4, 8$. As shown in the figure, c-point queries always provide more information than 1-point queries under the "smallest" policy, while this is true when v_t is sufficiently small under the "largest" policy. If v_t closes to zero, a c-point query under either policies yields information roughly same as c 1-point queries.

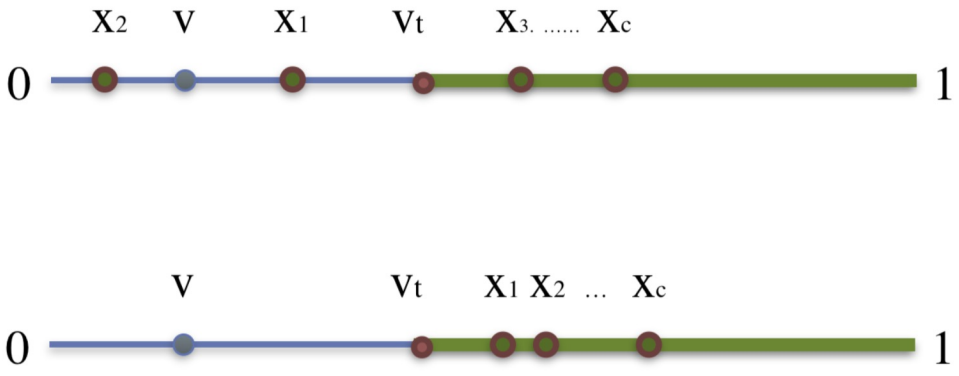

Figure 3: Top: One component lies in (v, v_t). Bottom: All components are to the right of v_t.

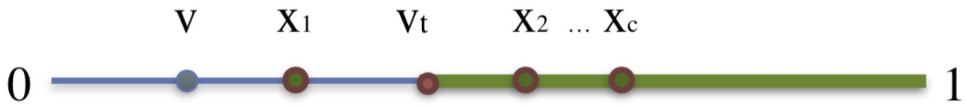

Figure 4: All components are to the right of v.

We have seen which component the expert chooses to correct does matter. Now, we consider the different distributions on $\mathcal{Q}$. Suppose we have a question distribution μ, instead of being supported on $[0, 1]$, it is supported on a single point $(1/c, 2/c, ..., c/c = 1)$. And we assume the expert corrects the most glaring error – the "largest" policy. Therefore, the expert corrects the point $x = 1$ at the first, the point $x = (c-1)/c$ at the second, and so on. Thus it takes $c/2$ rounds to bring the error down to $1/2$.

Let's pick any $\epsilon > 0$ and consider another distribution μ over $\mathcal{Q}$ that is supported on two points:

$$\left(\frac{1}{2c}, \frac{2}{2c}, ..., \frac{1}{2}\right)$$

with probability 2ϵ, and

$$\left(\frac{1}{2} + \frac{1}{2c}, \frac{1}{2} + \frac{2}{2c}, ..., 1\right)$$

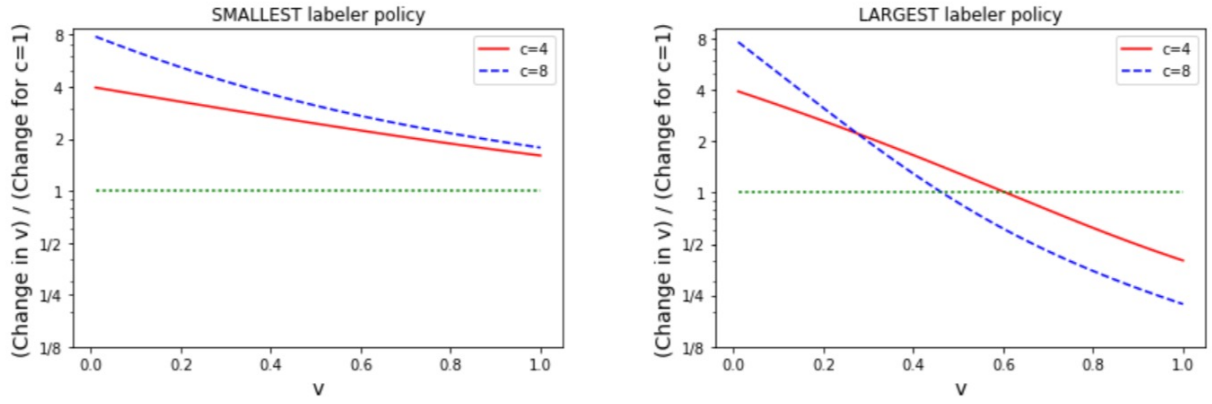

Figure 5: The ratio of the expected reduction with c-point queries to the expected reduction with 1-point queries for $c = 4, 8$.

with probability $1 - 2\epsilon$. We want to achieve $\text{err}_c(h) \leq \epsilon$. Thus, we want

$$\mathbb{E}_{q \in_\mu \mathcal{Q}, j \in_R [c]}[\mathbb{1}_{h(q,j) \neq h^*(q,j)}] \leq \epsilon.$$

For any h_v, it will always agree with h^* on the region $[v, 1]$. So we want

$$\Pr[\text{pick } x_i \in [0, v]] \leq \epsilon.$$

This implies that $v \leq 1/4$. So the learner must see the first point at least $c/2$ times which requires $\Omega(c/\epsilon)$ examples. Now, we introduce the first theorem based on all the previous analysis.

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 consistent hypotheses have error $\leq \epsilon$.*

3 Main results

Now, we present our main theorem. Even if the expert is not helpful, we can still achieve some lower bound of learning complexity. For any $h \in \mathcal{H}$, let

$$B(h) = \{q \in \mathcal{Q} \text{ s.t. } h \text{ is incorrect on } q\},$$

$$G(h) = \{q \in \mathcal{Q} \text{ s.t. } h \text{ is correct on } q\}.$$

Theorem 2. *The base algorithm of partial feedback interactive learning produces an ϵ-good hypothesis within $2N$ steps with probability at least $1-\delta$, where $N = c \cdot \left(\frac{l}{\epsilon'} + 1\right)$, $l = \log(|\mathcal{H}|/\delta)$, and $\epsilon' = \epsilon/2$.*

It's clear that what happens to the next step depends on which question is sampled. Once a question q in $G(h)$ is sampled, it's in there forever. On the other hand, if q is not sampled, we don't necessarily have information on it. If a question is sampled from $B(h)$, the expert will correct one component of it. For this component, the learner will not make mistake again. All the h that have this bad component will be corrected. The intuition behind this theorem is if we sample a particular question or a particular component a lot, then we can be reasonably certain that the hypothesis h we learned is a good hypothesis because it has to be consistent with all previously received questions and answers. Before formally proving our theorem, let's first introduce some notations.

- h_t is the hypothesis selected at the beginning of step t.
- $\bar{\mathcal{Q}} = \mathcal{Q} \times [c]$.
- $\bar{B}(h) = B(h) \times [c] = \{(q, j) \in \bar{\mathcal{Q}} : q \in B(h) \text{ and } h(q, j) \neq h^*(q, j)\}$.
- $\bar{G}(h) = G(h) \times [c]$.
- $\gamma(q, j)$ is the conditional probability that the expert provides feedback on j given that q is queried. And we define $w_t(q, j) = \mu(q) \cdot \gamma(q, j)$, as the product of the probability that a question q is chosen and the component j is fixed by the expert. For all $q \in G(h_t)$, we have the summation of $w_t(q, 1), ..., w_t(q, c)$ equals to $\mu(q)$.
- $W_t(q, j) = w_1(q, j) + w_2(q, j) + ... + w_t(q, j)$ is the sum of the individual distributions up to step t.

Now, we show some lemmas.

3.1 How to pick the weights

Lemma 1. *For all $q \in G(h_t)$, non-negative values $w(q, 1), ..., w(q, c)$ summing up to $\mu(q)$ can be calculated such that*

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

Proof. First, we have $W_t(q, [c]) = t \cdot \mu(q)$. The average of $W_t(q, j)$ over $[c]$ is $\frac{t \cdot \mu(q)}{c}$. We pick components $j_1, ..., j_c$ such that

$$W_{t-1}(q, j_1) \leq W_{t-1}(q, j_2) \leq ... \leq W_{t-1}(q, j_c).$$

Let $\Delta = \mu(q)$. We initialize all the $w_t(q, j_i)$ to 0, repeat

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

till $\Delta = 0$, and reset $\Delta = \Delta - w_t(q, j_i)$. □

3.2 Eliminating inconsistent hypotheses

Lemma 2. *With probability at least $1-\delta$, the following holds. $\forall h \in \mathcal{H}$: If there is a step t for which $W_t(\bar{B}(h)) \geq l$, then h is not consistent with the feedback received up to that step.*

Proof. First, any hypothesis h is eliminated with probability at least $w_t(\bar{B}(h))$. Let t be the first step for which $W_t(\bar{B}(h)) \geq l$. Then the probability that h is not eliminated by the end of step t is

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

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

3.3 Analyzing the first N steps

Let $\tau = \frac{N}{c} = \frac{l}{\epsilon'} + 1$ be a threshold value. We will think of an atomic component as having been adequately sampled when W_t reaches $\tau \cdot \mu(q)$. At the beginning of step t, let

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

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

$$\bar{L}'_{t-1} = \{(q,j) \in \bar{Q} : W_{t-1(q,j)} \leq (\tau-1) \cdot \mu(q) = \frac{l}{\epsilon'} \cdot \mu(q)\}.$$

Lemma 3. *Any any step t, if $W_{t-1}(\bar{B}(h_t)) < l$, then*

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

Proof. Note that

$$\mu(B(h_t)) = w_t(\bar{B}(h_t)) = w_t(\bar{B}(h_t) \wedge L'_{t-1}) + w_t(\bar{B}(h_t) \setminus L'_{t-1}),$$

then we can see that

$$l > W_{t-1}(\bar{B}(h_t)) \geq W_{t-1}(\bar{B}(h_t) \setminus L'_{t-1}) \geq \frac{l}{\epsilon'} \cdot w_t(\bar{B}(h_t) \setminus L'_{t-1}).$$

It follows that

$$w_t(\bar{B}(h_t) \setminus L'_{t-1}) \leq \epsilon'.$$

So, we have

$$\mu(B(h_t)) \leq w_t(\bar{B}(h_t) \wedge L'_{t-1}) + \epsilon', \text{ and thus}$$
$$w_t(\bar{B}(h_t) \wedge \bar{L}'_{t-1}) \geq \mu(B(h_t)) - \epsilon'.$$

□

Lemma 4. *For any step $t \leq N$, $w_t(\bar{L}_t) \geq 1 - \epsilon'$.*

Proof. Note that

$$w_t(\bar{L}_t) = w_t(\bar{B}(h_t) \wedge \bar{L}_t) + w_t(G(h_t) \wedge \bar{L}_t).$$

Since any $(q, j) \in \bar{B}(h_t) \wedge \bar{L}'_{t-1}$ satisfies $(q, j) \in \bar{B}(h_t) \wedge \bar{L}'_t$, the previous lemma 3 implies that $w_t(\bar{B}(h_t) \wedge \bar{L}_t) \geq \mu(B(h_t)) - \epsilon'$. For $q \in G(h_t)$, any (q, j) with $w_t(q, j) > 0$ satisfies

$$W_t(q, j) = \frac{t \cdot \mu(q)}{c} \leq \tau \cdot \mu(q).$$

Thus $(q, j) \in \bar{L}_t$ and it follows that

$$w_t(\bar{G}(h_t) \wedge \bar{L}_{t-1}) = \mu(G(h_t)).$$

Overall, we have

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

□

Corollary 1. *Let $\widehat{W}_t(q, j) = \min\{W_t(q, j), \tau \cdot \mu(q)\}$. As we have seen, $\widehat{W}_t(\bar{\mathcal{Q}}) \leq N$. We can see as a corollary to before that*

$$\widehat{W}_N(\bar{\mathcal{Q}}) \leq (1 - \epsilon')N.$$

3.4 Analysis of the next N steps

If $\mu(B(ht)) \geq 2\epsilon'$, then $\mu(B(h_t)) - \epsilon' \geq \frac{1}{\epsilon}$. During one of the steps in the second N steps, $\mu(B(h_t)) < 2 \cdot \epsilon' = \epsilon$ at which point the algorithm can return h_t.

Bibliographic notes

Partial feedback in interactive learning and all its analysis is due to Dasgupta & Luby [1].

References

[1] Sanjoy Dasgupta, Michael Luby. Learning from partial correction. *at arXiv*, 2017.