Efficient PAC Learning from the Crowd

Reported by:

Yao Xiao
Zhuosheng Zhang
Shiwei Zeng

Background Introduction

What is crowdsourcing?

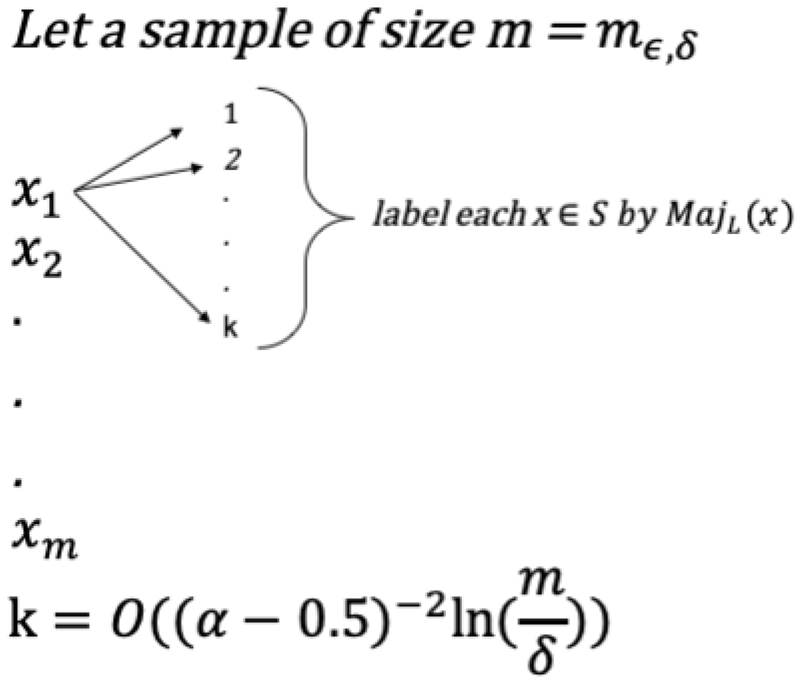
In machine learning, we have a set of hypothesis and we want to learn the true model $f^* \in \mathcal{F}$. However, in reality, we don't know the true label for each example, so, we want to find people who can label them. In this case, we can query from the crowd to get efficient labels.

Pros and Cons:

+ Efficient, cheap, query anytime.

- Not guaranteed, need to deal with noises.

Baseline Algorithm and its limitation



Limitation:
The average cost per sample is changing according to the total sample size.
It is not efficient and does not make sense in human’s intuition.
Because when the sample size is increasing, some samples in this set could remain the same but become more expensive.

Baseline Algorithm and Proof

BASELINE: Draw a sample $m = m_{\epsilon,\delta}$ from $D_{|\chi}$ and label each $x \in S$ by $Maj_L(x)$, where $L \sim P^k$ for $k = O((\alpha - 0.5)^{-2} \ln(\frac{m}{\delta}))$ is a set of randomly drawn labels. Return classifier $\mathcal{O}_{\mathcal{F}}(S)$.

Here we prove the complexity of k:

Given that we request k labels to label each $x \in S$. Within those labels, there are a fraction α will give us the right label. Therefore, the probability that $Maj_L(x)$ will give the right label is implied by Hoeffding's Inequalities. The $Maj_L(x)$ is wrong with the probability:

$$P\left(y\sum_{i=1}^{k}\hat{y}_i \le 0\right) = P\left(\sum_{i=1}^{k}\hat{y}_i \le 0, y = 1\right) + P\left(\sum_{i=1}^{k}\hat{y}_i \ge 0, y = -1\right)$$

$$\le P\left(\sum_{i=1}^{k}\hat{y}_i \le 0 | y = 1\right) + P\left(\sum_{i=1}^{k}\hat{y}_i \ge 0 | y = -1\right)$$

Baseline Algorithm and Proof

According to the properties of hoeffding's inequalities:

$$P\left(\sum_{i=1}^{k} \hat{y}_i \leq 0 | y = 1\right) = P\left(\sum_{i=1}^{k} \hat{y}_i \geq 0 | y = -1\right)$$

When $y = -1$ is the ground truth, $\mu = \alpha(-1) + (1-\alpha)(1) = 1 - 2\alpha < 0$

Hoeffding's inequalities: (where $b - a = 1 - (-1) = 2$)

$$P\left(\sum_{i=1}^{k} (y_i - \mu) \geq t\right) \leq e^{-\frac{2t^2}{k(b-a)^2}}$$

$$P\left(\sum_{i=1}^{k} (y_i) - k(1-2\alpha) \geq t\right) \leq e^{-\frac{t^2}{2k}}$$

$$P\left(\sum_{i=1}^{k} (y_i) \geq t + k(1-2\alpha)\right) \leq e^{-\frac{t^2}{2k}}$$

Baseline Algorithm and Proof

Let $t + k(1 - 2\alpha) = 0$, $t = -k(1 - 2\alpha)$,

$$P\left(\sum_{i=1}^{k}(y_i) \geq 0\right) \leq e^{-\frac{k(1-2\alpha)^2}{2}}$$

$$P(Maj_L(x)\ is\ wrong) = P\left(y\sum_{i=1}^{k}\hat{y}_i \leq 0\right) \leq 2e^{-\frac{k(1-2\alpha)^2}{2}}$$

For all samples x_j from S with size $m_{\epsilon,\delta}$:

$$1 - P\big(all\ of\ Maj_L(x_j)\ is\ right\big) = P\big(any\ one\ of\ Maj_L(x_j)\ is\ wrong\big)$$

$$\leq \sum_{j=1}^{m_{\epsilon,\delta}} P\big(Maj_L(x_j)\ is\ wrong\big) \leq m_{\epsilon,\delta} \cdot 2e^{-\frac{k(1-2\alpha)^2}{2}} = \delta$$

$$k = O\left((0.5 - \alpha)^{-2}\log\frac{m_{\epsilon,\delta}}{\delta}\right) > O(1)$$

Boosting

Early work showed that one can combine 3 classifiers of error p to get a classifier of error $O(p^2)$ for any $p > 0$.

Theorem 4.1

For any $p < \frac{1}{2}$ and distribution D, consider three classifiers:

1) classifier h_1 such that $err_D\ (h_1) \leq p$;

2) classifier h_2 such that $err_{D_2}\ (h_2) \leq p$, where $D_2 = \frac{1}{2} D_C + \frac{1}{2} D_I$ for distributions D_C and D_I that denote distribution D conditioned on $\{x | h_1(x) = f^*(x)\}$ and $\{x | h_1(x) \neq f^*(x)\}$, respectively;

3) classifier h_3 such that $err_{D_3}(h_3) \leq p$, where D_3 is D conditioned on $\{x | h_1(x) \neq h_2(x)\}$. Then, $err_D(MAH(h_1, h_2, h_3)) \leq 3p^2 - 2p^3$.

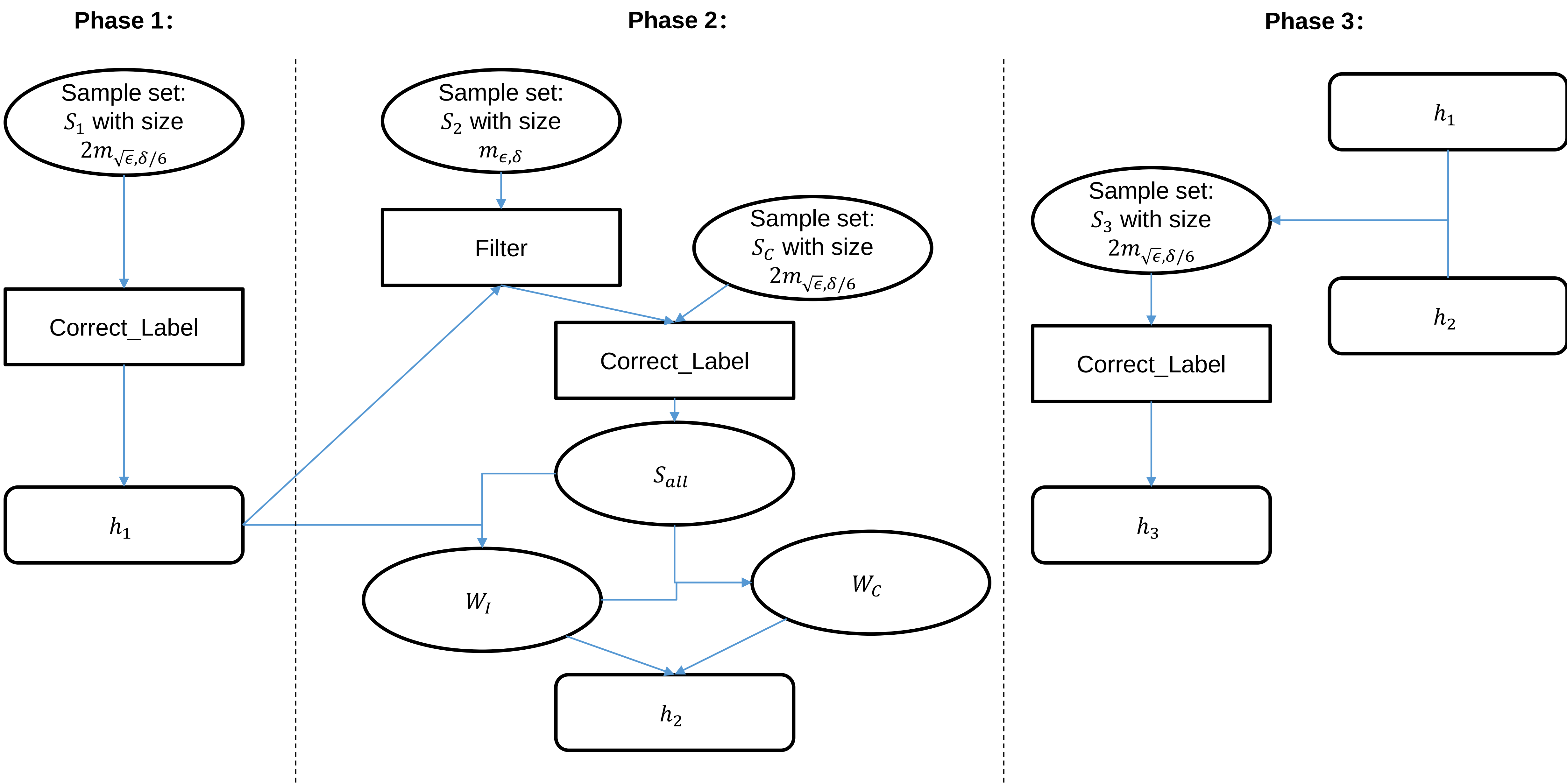
We will show that the improved Algorithm 2 is based on this theorem from boosting algorithm.

An Improved Algorithm: Algorithm 2

- Interleaving the process of learning and acquiring high quality labels.
- Boosting by probabilistic filtering for $\alpha = \frac{1}{2} + \Theta(1)$, giving that more than half of the labelers are perfect.

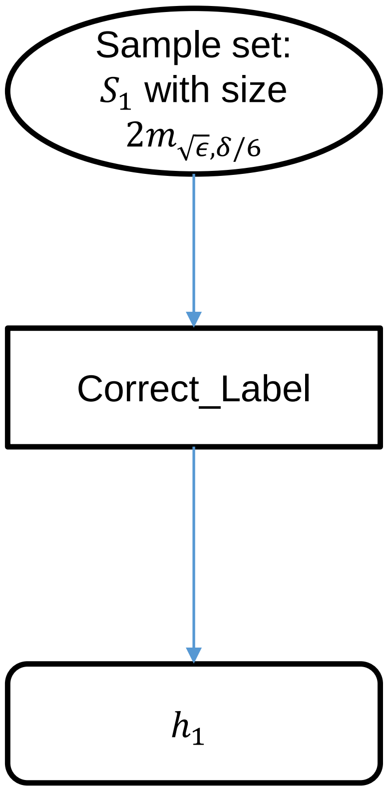
Algorithm 2 flow chart:

Base on the Boosting Algorithm， The paper contribute an algorithm that produce 3 classifier h_1, h_2, h_3 that satisfy **Theroem 4.1:**



Phase 1:

Phase 1:



Phase 1:

Let $\bar{S_1} = CORRECT - LABEL(S_1, \frac{\delta}{6})$, for a set of sample S_1 of size $2m_{\sqrt{\epsilon},\frac{\delta}{6}}$ from $D_{|\chi}$.
Let $h_1 = \mathcal{O}_{\mathcal{F}}(\bar{S_1})$.

First, we get a set of sample S_1 of size $2m_{\sqrt{\epsilon},\frac{\delta}{6}}$ from $D_{|\chi}$. According to the idea of super-sampling and Lemma 4.2:

$$m_{c\epsilon,\delta} = O(\frac{1}{c} m_{\epsilon,\delta}),$$

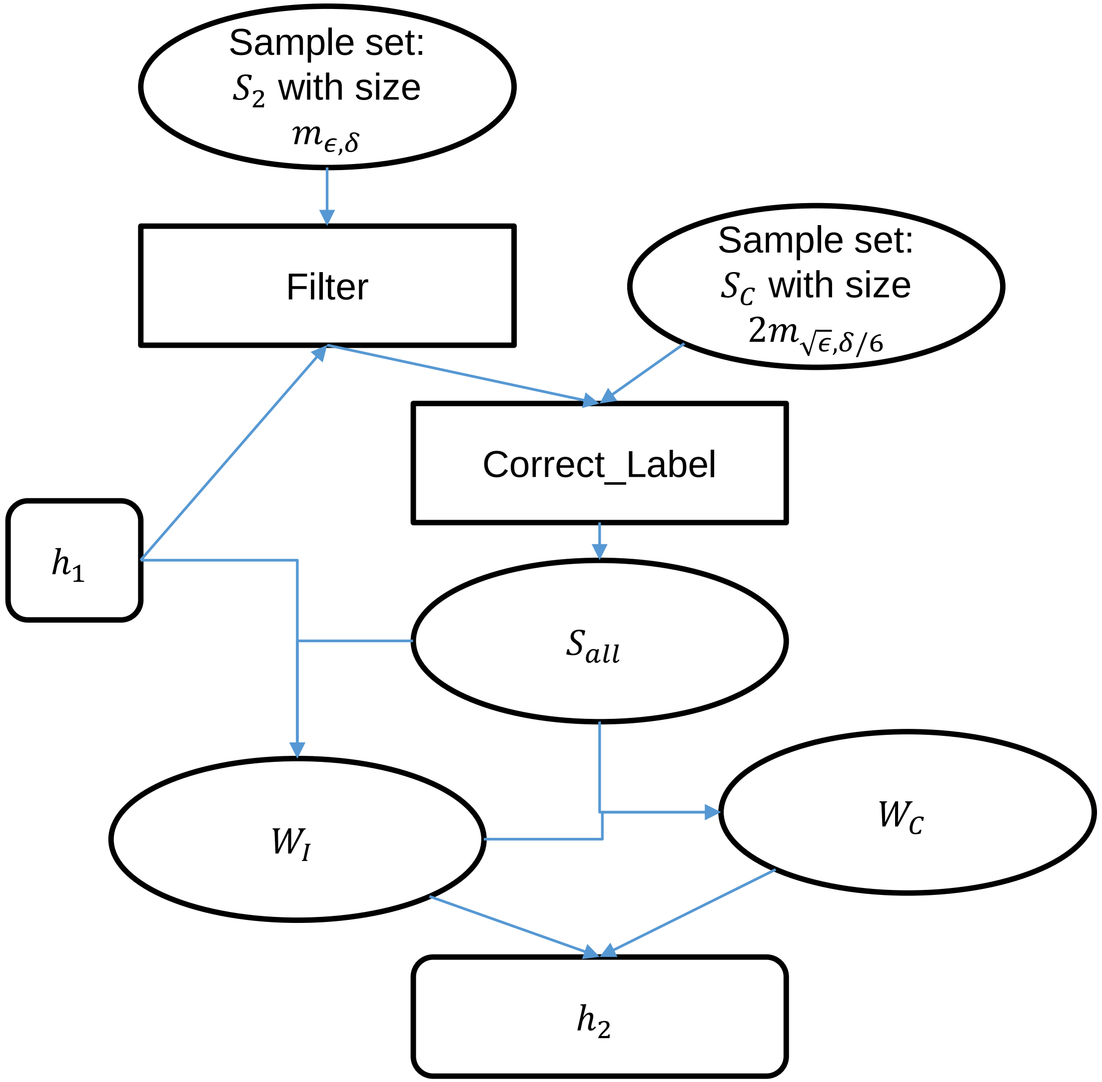
the size $2m_{\sqrt{\epsilon},\frac{\delta}{6}} = O(m_{\frac{\sqrt{\epsilon}}{2},\frac{\delta}{6}})$.

Then use $CORRECT - LABEL$ get labeled sample, base on that we can get the first classifier h_1.

The $CORRECT - LABEL$ is based on Baseline algorithm, as what we proved before, the classifier h_1 we obtained has error of at most $\frac{\sqrt{\epsilon}}{2} \in (0, \frac{1}{2})$ with high probability, and the labels queried in this step is $O(m_{\sqrt{\epsilon},\delta} \log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}))$.

Phase 2:

Phase 2:



Phase 2:

Base on the Theorem 4.1, we want a classifier that have two properties:

- Learned from a dataset that half of it is classified by h_1 correctly and half of it is misclassified.
- The error rate of this classifier is smaller than $\frac{\sqrt{\epsilon}}{2}$

So the algorithm is:

Let $S_I = FILTER(S_2, h_1)$, for a set of samples S_2 of size $\Theta(m_{\epsilon,\delta})$ drawn from $D_{|\chi}$.We will prove that, in phase 2, the size of S_I after filtering sample set S_2 with classifier h_1 is $\Theta\left(m_{\sqrt{\epsilon},\delta}\right)$ with high probability. Which is the query times.

Let S_C be a sample set of size $\Theta(m_{\sqrt{\epsilon},\delta})$ drawn from $D_{|\chi}$.

Let $\overline{S_{All}} = CORRECT - LABEL(S_I \cup S_C, \frac{\delta}{6})$. The query times here is $O(m_{\sqrt{\epsilon},\delta} \log\left(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}\right)$.

Let $\overline{W_I} = \{(x,y) \in \overline{S_{All}} \mid y \neq h_1(x)\}$ and let $\overline{W_C} = \overline{S_{All}} \backslash \overline{W_I}$.

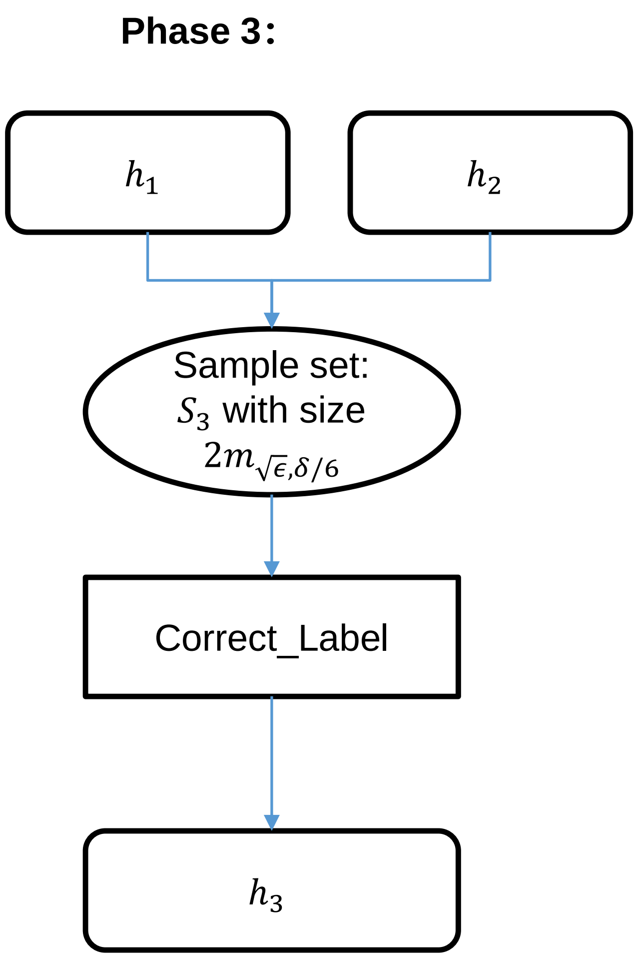
Draw a sample set $\overline{W}$ of size $\Theta(m_{\sqrt{\epsilon},\delta})$ from a distribution that equally weights $\overline{W_I}$ and $\overline{W_C}$.

Let $h_2 = \mathcal{O}_{\mathcal{F}}(\overline{W})$.

So the classifier h_2 we obtained has error of at most $\frac{\sqrt{\epsilon}}{2} \in (0, \frac{1}{2})$ with high probability.

And the labels queried in this step is $O(m_{\sqrt{\epsilon},\delta} \log\left(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}\right) + m_{\sqrt{\epsilon},\delta})$.

Phase 3:



Phase 3:

Let $\overline{S_3} = CORRECT - LABEL\left(S_3, \frac{\delta}{6}\right)$,for a sample set S_3 of size $2m_{\sqrt{\epsilon},\frac{\delta}{6}}$ drawn from $D_{|\chi}$ conditioned on $h_1(x) \neq h_2(x)$.

Let $h_3 = \mathcal{O}_{\mathcal{F}}(\overline{S_3})$.

Because The S_3 is drawn from dataset that satisfy $h_1(x)$ $\neq h_2(x)$, and use $CORRECT - LABEL$ to label it.

So, similar to phase 1, labels queried in phase 3 is attributed to the labels queried by function $CORRECT - LABEL(S, h)$, which is $O(m_{\sqrt{\epsilon},\delta}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}))$.

The classifier h_3 we obtained in this step has error of at most $\frac{\sqrt{\epsilon}}{2} \in (0, \frac{1}{2})$ with high probability.

Algorithm 2 Conclusion

After obtained h_1 h_2 h_3, using $Maj(h_1, h_2, h_3)$, based on theorem 4.1, we can obtain a classifier that uses oracle $\mathcal{O}_{\mathcal{F}}$, runs in time $poly(d, \frac{1}{\epsilon}, \ln\frac{1}{\delta})$ and with probability $1-\delta$ returns $f \in \mathcal{F}$ with $err_D(f) \leq \epsilon$. The cost per labeled sample is $\Lambda = O(\sqrt{\epsilon}\log\left(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}\right) + 1)$, when $\frac{1}{\sqrt{\epsilon}} \leq \log\left(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}\right)$, $\Lambda = O(1)$. To prove this:

As shown in Algorithm 2, labels queried by phase 1 and 3 is $O(m_{\sqrt{\epsilon},\delta}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}))$. Since $E[|S_I|] = \Theta\left(m_{\sqrt{\epsilon},\delta}\right)$, $|S_I \cup S_C| \leq O\left(m_{\sqrt{\epsilon},\delta}\right)$, therefore $CORRECT-LABEL(S_I \cup S_C, \frac{\delta}{6})$ contributes $O(m_{\sqrt{\epsilon},\delta}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}))$ labels. In addition, $FILTER(S_2, h_1)$ in phase 2 also contributes to $O(m_{\epsilon,\delta})$ labels (lemma 4.9). This leads to

$$\frac{O(m_{\sqrt{\epsilon},\delta}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}) + m_{\epsilon,\delta})}{m_{\epsilon,\delta}} = \frac{O(\frac{1}{\sqrt{\epsilon}}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}) + \frac{1}{\epsilon})}{O(\frac{1}{\epsilon})} = O(\sqrt{\epsilon}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}) + 1)$$

cost per labeled example.

Probabilistic Filtering

Algorithm 1 FILTER(S, h)

Let $S_I = \emptyset$ and $N = \log\frac{1}{\epsilon}$

for $x \in S$ **do**

  for $t = 1, \ldots, N$ **do**

    Draw a random labeler $i \sim P$ and let $y_t = g_i(x)$

    If t is odd and $Maj(y_{1:t}) = h(x)$, **then** break.

  end

  Let $S_I = S_I \cup \{x\}$. //Reaches this step when for all $t, Maj(y_{1;t}) \neq h(x)$

end

return S_I

FILTER (S, h) returns a set $S_I \subseteq S$ such that for any $x \in S$ that is mislabeled by h_1, $x \in S_I$, the majority of the labels never agree with$h(x)$.

Probabilistic Filtering

According to lemma 4.9, $err_D(h) \leq \sqrt{\epsilon}$ with probability $1 - exp(-\Omega(|S|\sqrt{\epsilon}))$, FILTER (S, h) makes $O(|S|)$ label queries.

We can using Chernoff bound to prove the result. The total number of points in S where h disagrees with f^* is $O(|S|\sqrt{\epsilon})$. Since $N = \log\frac{1}{\epsilon}$, the number of queries spent on these points is at most $O(|S|\sqrt{\epsilon}\, log(1/\epsilon)) \ \leq \ O(|S|)$.

Prove the size of S_I after filtering

We here prove that, in phase 2, the size of S_I after filtering sample set S_2 with classifier h_1 is $\Theta\left(m_{\sqrt{\epsilon},\delta}\right)$ with high probability. According to Lemma 4.6., given any sample set S and classifier h, for every $x \in S$:

1. If $h(x) = f^*(x)$, then $x \in FILTER(S,h)$ with probability $p_1 \in (0, \sqrt{\epsilon})$.
2. If $h(x) \neq f^*(x)$, then $x \in FILTER(S,h)$ with probability $p_2 \in [0.5,1]$.

From phase 1, we obtain a classifier h_1 with error at most $\frac{\sqrt{\epsilon}}{2}$. Therefore, the algorithm here falls into case 1 with probability $1 - \frac{\sqrt{\epsilon}}{2}$, into case 2 with probability $\frac{\sqrt{\epsilon}}{2}$ respectively.

$$E[|S_I|] = (1 - \frac{\sqrt{\epsilon}}{2}) \cdot p_1|S_2| + \frac{\sqrt{\epsilon}}{2} \cdot p_2|S_2|$$

$$\left(1 - \frac{\sqrt{\epsilon}}{2}\right) \cdot \sqrt{\epsilon}|S_2| + \frac{\sqrt{\epsilon}}{2} \cdot 1 \cdot |S_2| \geq E[|S_I|] \geq (1 - \frac{\sqrt{\epsilon}}{2}) \cdot 0 \cdot |S_2| + \frac{\sqrt{\epsilon}}{2} \cdot 0.5 \cdot |S_2|$$

$$O\left(m_{\sqrt{\epsilon},\delta}\right) \geq \left(\sqrt{\epsilon} - \frac{\epsilon}{2}\right) \cdot |S_2| + \frac{\sqrt{\epsilon}}{2}|S_2| \geq E[|S_I|] \geq \frac{\sqrt{\epsilon}}{4} \cdot |S_2| \geq \Omega(m_{\sqrt{\epsilon},\delta})$$

Super-sampling

This technique means that as long as we have the correct label of the sampled points and we are in the realizable setting, more samples never hurt the algorithm.

In our algorithm, we have set samples $S_1$of size $2m_{\sqrt{\varepsilon},\delta/6}$ drawn from $D_{|x}$.

First, notice that because D and D' are both labeled according to $f^* \in F$, for any $f \in F$ we have,

$$err_{D'}(f) = \sum_x d'(x) 1_{f(x)\neq f^*(x)} \geq \sum_x cd(x) 1_{f(x)\neq f^*(x)} = cerr_{D(f)}$$

Therefore, if $err_{D(f)} \leq c\epsilon$, then $err_{D(f)} \leq \epsilon$. Let $m' = m_{c\epsilon,\delta}$, we have

$$\begin{aligned} \delta &> Pr_{S' \sim D'm'}[\exists f \in \mathcal{F}, s.t. err_{S'}(f) = 0 \wedge err_{D'}(f) \geq c\epsilon] \\ &\geq Pr_{S' \sim D'm'}[\exists f \in \mathcal{F}, s.t. err_{S'}(f) = 0 \wedge err_{D}(f) \geq \epsilon] \end{aligned}$$

The claim follows by the fact that $m_{c\epsilon,\delta}= O(\frac{1}{c} m_{\epsilon,\delta})$.

So, the size of S_1 is $m_{\frac{\sqrt{\epsilon}}{2},\delta/6}$.

Correct-Label Algorithm

$\mathbf{CORRECT-LABEL}(S, \delta)$:
$\mathbf{for}\ x \in S\ \mathbf{do}$

Let $L \sim P^k$ for a set of $k = O\left(\log\left(\frac{|S|}{\delta}\right)\right)$ labelers drawn from P and $\bar{S} \to \bar{S} \cup \{(x, Maj_L(x))\}$

$\mathbf{end}$
$\mathbf{return}\ \bar{S}$

Comparing new algorithm with the old one

The average cost per sample:

$$\Lambda_{BASELINE} = O\left((0.5-\alpha)^{-2}\log\frac{m_{\epsilon,\delta}}{\delta}\right)\text{ in black.}$$

v.s.

$$\Lambda_{Algorithm2} = O(\sqrt{\epsilon}\log(\frac{m_{\sqrt{\epsilon},\delta}}{\delta}) + 1)\text{ in red.}$$