Active learning for Binary classification

Consider example space $\mathcal{X}$ - space of all images (e.g. satellite photos) (binary) labeled space $\{0, 1\}$ - contains "human-made object" or not. Assume: for every $x \in \mathcal{X}$ a corresponding label $y_x \in \{0, 1\}$

Hypothesis class $\mathcal{H}$: the $h : \mathcal{X} \rightarrow \{0, 1\}$.

"Traditional" passive learning (for supervised learning)

*Distribution $\mathcal{D}$ over $\mathcal{X}$ and we observe $\{(x_t, y_t)\}_{t=1}^{\infty} \sim \mathcal{D}$, $x_t \in \mathcal{X}$, $y_t \in \{0, 1\}$

Given dataset, learn $\hat{h}_n = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{t=1}^{n} \mathcal{D}(h(x_t) \neq y_t)$

Reason about true risk $\mathbb{E}_{(x, y) \sim \mathcal{D}} \left[ \mathbb{I}\{\hat{h}_n(x) \neq y\} \right]$.

Active learning selects examples to be labelled and we evaluate an algorithm based on both # labels requested, # unlabelled looked at

and $\mathbb{E}_{(x, y) \sim \mathcal{D}} \left[ \mathbb{I}\{\hat{h}_n(x) \neq y\} \right]$.

**Question**: Can active learning achieve same accuracy as passive learning w/ far fewer labels?

### Settings of interest

<table>
<thead>
<tr>
<th>Pool-based setting</th>
<th>Streaming setting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Separable</strong></td>
<td><strong>Agnostic</strong></td>
</tr>
</tbody>
</table>

**Today**
**Separable setting:** $\exists h \in \mathcal{H} : Y_x = h^*(x) \forall x \in \mathcal{X}$

i.e. $\exists$ hypothesis that perfectly labels all data

**Agnostic setting:** Not the separable setting, when the label for $X \in \mathcal{X}$ is requested, we observe $Y \sim Bernoulli( \theta_x )$ where $\theta : \mathcal{X} \to [0,1]$ is arbitrary.

---

**Pool-based setting**

Example space $\mathcal{X}$ is finite and fixed.

Game proceeds in rounds.

Input: $\mathcal{H}, \mathcal{X}$

for $t = 1, 2, \ldots, n$

Learner chooses $x_t \in \mathcal{X}$

Nature reveals $y_t$

Learner outputs $\hat{h}_n \in \mathcal{H}$ and receives loss

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left[ \hat{h}_n(x_i) \neq y_i \right]$$

over entire pool $\mathcal{X}$

---

**Streaming setting** $\mathcal{X}$ can be uncountable. Exists a distribution $\mathcal{D}_x$ over $\mathcal{X}$.

for $t = 1, 2, \ldots, n$

Nature reveals $x_t \sim \mathcal{D}_x$

Learner decides request label or not

If yes, nature reveals $y_t$. Else round ends.

Learner receives loss $\mathbb{E}_{(x,y) \sim p_{xy}} \left[ \mathbb{1} \left[ \hat{h}_n(x) \neq y \right] \right]$. 

---
An algorithm for streaming setting can always be applied to the pool-based setting.

**Separable, pool-based setting** (Exact setting: identify $h^*$)

- $X$ is finite, and can be enumerated $X = \{1, \ldots, n\}$
- $n = |X|$

$X$ finite $\Rightarrow \mathcal{H}$ is finite wlog

For $h^* \in \mathcal{H}$:

$h^*(i) = h^*(x_i) = y_i \forall x_i \in X \ (i \in X)$

(1) \quad \forall i \in [n]$

Ex. Thresholds

$h_j(x_i) = \chi \{i \leq j\} \quad \mathcal{H} = \{h_j : j \in \mathcal{N}_n\}$

For this class I can use bisection search to learn $h^* \in \mathcal{H}$ using just \lceil \log_2 |\mathcal{H}| \rceil \ labels.

Ex. Needle in a haystack $h_j(x_i) = \chi \{i = j\}$

To identify $h^*$, nothing is better than exhaustive search.
**Question:** Given arbitrary hypothesis class $\mathcal{H}$, how many queries are necessary and sufficient to identify $h^* \in \mathcal{H}$?

For a deterministic algorithm $A$ let $S(X, \mathcal{H}, A, h)$ be the number of labels requested under $h^* = h$, until all other hypotheses are ruled out.

WLOG any deterministic algorithm $A$ is a binary tree:

$\#$ leaves = $|2^{|\mathcal{H}|}|$

$\Rightarrow$ depth of tree $\geq \lceil \log_2 |\mathcal{H}| \rceil$

$\Rightarrow$ Some $h \in \mathcal{H}$ requires $\geq \lceil \log_2 |\mathcal{H}| \rceil$ labels.

**Proposition** For any hypothesis class $\mathcal{H}$ we have:

$$\min_{A \in \mathcal{A}} \max_{h \in \mathcal{H}} S(X, \mathcal{H}, A, h) \geq \lceil \log_2 |\mathcal{H}| \rceil.$$
Extended teaching dimension (Hegedus 1995)

Def. We say $S \subseteq X$ is a specifying set for $b \in \{0,1\}^n$ with $\mathcal{H}$ if $|\{h \in \mathcal{H} : h(x) = b(x) \forall x \in S\}| \leq 1$.

Note: $S$ is not necessarily in $\mathcal{H}$.

When $b \in \mathcal{H}$ then a specifying set is sufficient to "teach" the concept $b \in \mathcal{H}$.

When $b \notin \mathcal{H}$ "to prove that $b \notin \mathcal{H}$"

Def. For any $X$, $\mathcal{H}$ define the extended teaching dimension

$$\text{EXT-TD}(\mathcal{H}) = \min \{k : \forall b \in \{0,1\}^n, \exists \text{spec. set } S \text{ for } b \text{ w/ } |S| \leq k\}$$

Theorem. For any $\mathcal{H}$ we have

$$\text{EXT-TD}(\mathcal{H}) = \min \max_{\mathcal{H}} \text{EXT-TD}(X, \mathcal{H}, A, h) \leq \text{EXT-TD}(\mathcal{H}) + \lceil \log_2 |\mathcal{H}| \rceil$$

Moreover, the Halving algorithm achieves the upper bound.

Ex. $\text{EXT-TD}(\mathcal{H}_{\text{thresholds}}) = 2$.

If $b \in \mathcal{H}$, just need to give example to left/right of threshold value.

If $b \notin \mathcal{H}$ then $b$ a "0" before a "1" and so choose any pair

or $b = 0^n$ and just return $S = \{1\}$ since $h(1) = 1$

Ex. $\text{EXT-TD}(\mathcal{H}_{\text{haythorn}}) = |\mathcal{H}| - 1$.

Consider $b = 0^n$