Notes 9: Occam's Razor

1. Hypothesis class

Hypothesis class $\mathcal{H}$ = set of hypotheses the learning algorithm may output
Usually $\mathcal{H} \supseteq \mathcal{C}$, but can sometimes be bigger
  e.g. Winnow1 learns $\mathcal{C} = \{k\text{-sparse monotone disjunctions}\}$ using $\mathcal{H} = \{\text{LTFs with} \geqslant 0 \text{ weights}\}$
Proper learning: Algorithm required to output $h \in \mathcal{C}$, i.e. $\mathcal{H} = \mathcal{C}$
Improper learning: Algorithm allowed to output $h \notin \mathcal{C}$, i.e. $\mathcal{H} \supsetneq \mathcal{C}$

---

2. Consistent hypotheses

Fix concept class $\mathcal{C}$ and finite hypothesis class $\mathcal{H}$

Consistent Hypothesis Algorithm

Given labelled samples, output any $h \in \mathcal{H}$ consistent with all samples

Call hypothesis h bad if $\mathrm{err}_{\mathcal{D}}(h, c) \geqslant \varepsilon$

Theorem 1. *For any distribution $\mathcal{D}$ over instance space X, given m independent samples from $\mathrm{EX}(c, \mathcal{D})$, if $m \geqslant \frac{1}{\varepsilon}\ln(|\mathcal{H}|/\delta)$, then*

$$\mathbb{P}[\textit{some bad hypothesis in } \mathcal{H} \textit{ consistent with all samples}] \leqslant \delta$$

Better bound than Halving Algorithm + Online-to-PAC conversion

Proof. For any bad $h \in \mathcal{H}$

$$\mathbb{P}[h \text{ consistent with all } m \text{ samples}] \leqslant (1-\varepsilon)^m \leqslant e^{-\varepsilon m} = \delta/|\mathcal{H}|$$

Union bound:

$$\mathbb{P}[\text{some bad hypothesis in } \mathcal{H} \text{ consistent with all samples}] \leqslant |\mathcal{H}| \cdot (\delta/|\mathcal{H}|) = \delta$$ □

In other words, $|\mathcal{H}| \leqslant \delta e^{\varepsilon m}$

Occam's Razor: Scientific principle to favour simpler hypotheses
PAC learning algorithm due to small hypothesis class
  Simple hypothesis ≈ hypothesis with short description ≈ small number of hypotheses

---

3. PAC learning sparse disjunctions

$\mathcal{C} = \{\text{disjunctions}\}$ over $X = \{0,1\}^n$ $s \stackrel{\text{def}}{=} \text{size}(c)$
How to PAC learn $\mathcal{C}$ efficiently?

(1) Elimination Algorithm + Online-to-PAC conversion: $O\left(\frac{n}{\varepsilon}\ln\frac{n}{\delta}\right)$ samples
  $\approx \frac{n}{\varepsilon}$ ignoring log factors
(2) Winnow1 + Online-to-PAC conversion: $O\left(\frac{s\ln n}{\varepsilon}\ln\frac{s\ln n}{\delta}\right)$ samples
  $\approx \frac{s}{\varepsilon}$ ignoring log factors
  Better dependence on n; Good for small s
  But improper
(3) Consistent Hypothesis Algorithm: $O\left(\frac{s}{\varepsilon}\ln\frac{n}{\delta}\right)$ samples
  Because $|\mathcal{H}| = \binom{n}{s}2^s \leqslant (2n)^s$ ($\mathcal{H} \stackrel{\text{def}}{=} \{s\text{-sparse disjunctions}\}$)
  Even better dependence on n and s
  But inefficient! (need $|\mathcal{H}| \approx n^s$ time, not $\mathrm{poly}(n, 1/\varepsilon, 1/\delta, s)$)
(4) (Below) efficient algorithm using $O\left(\frac{1}{\varepsilon}(\ln\frac{1}{\delta} + s\ln\frac{1}{\varepsilon}\ln n)\right)$ samples
  $\approx \frac{s}{\varepsilon}$ ignoring log factors; Good dependence on s and n
  Idea 1: Find consistent disjunction quickly using Greedy Heuristic for Set Cover
  Idea 2: Further reduce $|\mathcal{H}|$ by hypothesis testing

---

4. Set Cover

A computational problem (not originated from learning)
Input: Universe $U = \{1, \ldots, m\}$ of m elements and subsets $S_1, \ldots, S_r \subseteq U$
Goal: Find smallest collection $S_{i_1}, \ldots, S_{i_k}$ of given subsets to cover U (i.e. $S_{i_1} \cup \cdots \cup S_{i_k} = U$)

Set Cover is NP-hard (as hard as thousands other problems conjectured to be intractable)
We settle for an approximation algorithm that outputs a nearly optimal solution

Greedy Heuristic

For $t = 1, 2, \ldots$ until U is covered
Pick largest subset S_{i_t}
Remove from every subset S_j all elements in S_{i_t} (i.e. S_j becomes $S_j \setminus S_{i_t}$)

Theorem 2. *Greedy Heuristic always outputs a cover with* $\leqslant \mathrm{OPT} \cdot \ln m$ *many sets*

Proof. Let $T_t \subseteq U$ denote set of uncovered elements after iteration t (initially $T_0 = U$)
Claim: Largest subset S_{i_t} at iteration t covers $\geqslant 1/\mathrm{OPT}$ fraction of T_{t-1}
Reason: Uncovered elements are covered by OPT sets; largest set must cover $\geqslant 1/\mathrm{OPT}$ fraction
Using Claim,

$$
\begin{aligned}
|T_t| &\leqslant \left(1 - \frac{1}{\mathrm{OPT}}\right) |T_{t-1}| \leqslant \ldots \leqslant \left(1 - \frac{1}{\mathrm{OPT}}\right)^t m < e^{-t/\mathrm{OPT}} m \\
&\leqslant 1 \qquad \text{if } t \geqslant \mathrm{OPT} \cdot \ln m
\end{aligned}
$$

□

Elimination Algorithm + conversion only uses negative samples
Keep removing literals x_i or $\overline{x}_i$ that contradicts a negative sample
All literals in c are also in h (h automatically consistent with all positive samples)
Improved algorithm further uses positive samples to shorten h (and hence shrink $\mathcal{H}$)
Find a few literals to "explain" (i.e. cover) all positive samples
c contains s literals, all positive samples can be "covered" with s literals
Can quickly find a cover using $s \ln m$ literals ($m = \#$positive samples)

Improved algorithm

$\{y_1, \ldots, y_r\}$ = set of literals that are consistent with all negative examples
i.e. if literal y_i is true in some negative sample, then y_i is excluded
For $1 \leqslant i \leqslant r$, let S_i = set of positive samples where y_i is true
Find a set cover $S_{i_1}, \ldots, S_{i_k}$ using $k = s \ln m$ sets
Hypothesis $h = y_{i_1} \vee \cdots \vee y_{i_k}$

$|\mathcal{H}| = \binom{n}{s \ln m} 2^{s \ln m} \leqslant (2n)^{s \ln m}$
Need $|\mathcal{H}| \leqslant \delta e^{\varepsilon m}$
True if $(2n)^{s \ln m} \leqslant \delta e^{\varepsilon m} \iff s(\ln m) \ln 2n + \ln(1/\delta) \leqslant \varepsilon m$
Can show that $m \geqslant \Omega\left(\frac{1}{\varepsilon}(\ln(1/\delta) + s(\ln n) \ln(s \ln n))\right)$ suffices (details omitted)