Occam and Compression Algorithms: General PAC Learning Results

These notes are slightly edited from scribe notes in previous years. Please consult the handout of slide copies for definitions and theorem statements.

1 Occam Algorithms

1.1 Theorem 0 from Handout

Theorem 0 from Handout Let \( X \) be a domain of examples, and \( C, H \) concept classes over \( X \). Let \( A \) be a learning algorithm such that \( \forall c \in C \ A \) takes a sample of \( m \) examples and outputs a hypothesis \( h \in H \) consistent with sample \( S \). Then when using a sample of size

\[
m \geq \frac{1}{\epsilon} \ln \frac{|H|}{\delta}
\]

\( A \) is a PAC learning algorithm for \( C \) using \( H \).

Note that a similar result has already been proved in lecture 1, in the context of one of the examples.

Proof:

- Define \( h \) is bad if \( \text{err}_D(h) > \epsilon \)

- Fix any bad \( h \). Then

\[
\Pr[h \text{ is consistent with sample}] < (1 - \epsilon)^m < e^{-\epsilon m} \leq e^{-\epsilon \ln |H|} = e^{-\epsilon \ln |H|/|H|} = \frac{\delta}{|H|}
\]

- By the union bound

\[
\Pr[\text{some bad } h \in H \text{ is consistent}] \leq |H| \frac{\delta}{|H|} = \delta
\]

■
1.2 Application of Theorem 0: Learning Conjunctions

We next apply Theorem 0 to the Elimination algorithm for learning conjunctions where $C = H =$conjunctions and we have $|\text{conjunctions}| = 3^n$.

\[
m \geq \frac{1}{\epsilon} \ln \frac{|H|}{\delta} \tag{6}
= \frac{1}{\epsilon} \ln \frac{1}{\delta} + \frac{n}{\epsilon} \ln 3 \tag{7}
= O(\frac{n}{\epsilon} \ln \frac{1}{\delta}). \tag{8}
\]

Previously, the result for the Elimination algorithm was $O(\frac{n}{\epsilon} \ln \frac{n}{\delta})$. So this is a better bound, and it is achieved using a more general analysis!

1.3 Theorem 1 from Handout

Notice that Theorem 1 in these notes above is the same as Theorem 1 in the handout, but with $|H_{n,m}|$ replaced by $|H|$. Conceptually, this allows the learning algorithm to increase its effective hypotheses class when the number of examples is increased.

1.4 Theorem 2 from Handout

Before discussing the theorem, we first note that the definition of Occam algorithms implies some form of compression. In particular, the size of the hypothesis is much smaller than the data set size. To see this, assume that $m$ is sufficiently large that it satisfies

\[m \gg [n \cdot \text{size}(c)]^{\frac{\alpha}{1-\beta}}\]

Then

\[
m^{1-\beta} \gg [n \cdot \text{size}(c)]^\alpha \tag{9}
\]
\[
m \gg m^\beta [n \cdot \text{size}(c)]^\alpha \tag{10}
> \text{size}(h) \tag{11}
\]

Thus, if $m$ is large enough, then $\text{size}(h) \ll m$ so the algorithm compresses the data into the hypothesis representation.

**Theorem 2 from Handout** If $A$ is an $(\alpha, \beta)$-Occam algorithm for learning $C$ by $H$, then when using a sample size

\[
m \geq \max \left\{ \left( \frac{2}{\epsilon} (n \cdot \text{size}(c))^\alpha \right)^{\frac{1}{\beta}}, \frac{2}{\epsilon} \log \frac{1}{\delta} \right\}
\]

$A$ is a PAC learning algorithm for $C$ using $H$.

**Proof:** We apply Theorem 1 (from Handout) which requires:

\[
m \geq \frac{1}{\epsilon} \ln \frac{1}{\delta} + \frac{1}{\epsilon} \ln |H_{n,m}|
\]
By the bound on the size of the hypothesis we get:

$$|H_{n,m}| \leq 2^{[n \cdot \text{size}(c)]^\alpha m^\beta}$$

and therefore it suffices to have

$$m \geq \frac{1}{\epsilon} \ln \frac{1}{\delta} + \frac{1}{\epsilon} (\ln 2)[n \cdot \text{size}(c)]^\alpha m^\beta$$

We simplify the next step using the fact that if $m \geq \max \{2A, 2B\}$ then $m > A + B$. Thus, we require $m \geq \frac{1}{\epsilon} \ln \frac{1}{\delta}$ as well as (we dropped the multiplicative $\ln 2$ which is less than 1):

$$m \geq \frac{2}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta$$

$$m^{1-\beta} \geq \frac{2}{\epsilon} (n \cdot \text{size}(c))^\alpha$$

$$m \geq \left[ \frac{2}{\epsilon} (n \cdot \text{size}(c))^\alpha \right]^{\frac{1}{1-\beta}}$$

\[\boxed{\text{Theorem 3 from Handout}}\]

2 Compression Algorithm

We have so far seen one method to obtain generic convergence bounds for PAC learning via Occam algorithms for finite and enumerable hypotheses classes. Similar results will be developed later for uncountable class using the notion of VC dimension. The following results give an alternative proof method that replies directly on properties of the learning/compression algorithm rather than properties of the concept class (although of course the algorithm depends on the class). This applies, for example, for learning intervals, and for the Perceptron learning algorithm that has a finite mistake bound and which we will discuss in some future lecture.

Fix any learning algorithm $L$ that takes an ordered sample $B$ as input and produces a hypothesis $\text{hyp}(B)$ based on the sample.

Algorithm $A$ is a compression algorithm for concept class $C$ (w.r.t $L$) with size $d$ if when given ordered samples $S = \{x_1, \ldots, x_m\}$, $A$ outputs $B \subseteq S$ s.t.

1. $|B| \leq d$.
2. $\text{hyp}(B)$ is consistent with $S$.

**Theorem 3 from Handout** A compression scheme is a PAC learner for class $C$ when using a sample of size $m \geq \frac{2}{\epsilon} \ln \frac{1}{\delta} + 2d + \left(\frac{2d}{\epsilon} \ln \frac{2}{\epsilon}\right)$

**Proof:** Consider an index set $T \subseteq \{1, \ldots, m\}$ where $|T| = k \leq d$ and define $S_{|T}$ = subset of $S$ with examples at indices $T$.

Define

$$\text{consistent}(T) = \text{hyp}(S_{|T})$$ is consistent with $S$\n
$$\text{bad}(T) = \text{err}(\text{hyp}(S_{|T})) > \epsilon$$
Note that for a fixed index set $T$, the subsamples $S_T$ and $(S \setminus S_T)$ are independent. When algorithm fails, it chooses $T$ with consistent$(T) \cap \text{bad}(T)$.

$$Pr(\text{consistent}(T) \cap \text{bad}(T)) = Pr(\text{bad}(T)) \cdot Pr(\text{consistent}(T)|\text{bad}(T))$$

$$\leq Pr(\text{consistent}(T)|\text{bad}(T))$$

$$\leq (1 - \epsilon)^{|T|}$$

We therefore have

$$Pr(\text{Compression Algorithm outputs hyp with error } > \epsilon)$$

$$= Pr(\text{Algorithm picks } T \text{ with consistent}(T) \cap \text{bad}(T))$$

$$\leq \bigcup_T Pr(\text{consistent}(T) \cap \text{bad}(T))$$

$$\leq \text{ via union bound and bound in previous formula}$$

$$\sum_{k=0}^{d} \binom{m}{k} (1 - \epsilon)^{m-k}$$

$$\leq (1 - \epsilon)^{m-d} \sum_{k=0}^{d} \binom{m}{k}$$

$$\leq (1 - \epsilon)^{m-d} \left( \frac{em}{d} \right)^{d}$$

$$\leq e^{-\epsilon(m-d)} \left( \frac{em}{d} \right)^{d} \leq \delta$$

Here we use the inequality $\sum_{k=0}^{d} \binom{m}{k} \leq \left( \frac{em}{d} \right)^{d}$, that will be proved in a future lecture. The last step, showing $\leq \delta$ is given on pages 12-13 (Lemma 1) of the article [Floyd and Warmuth 1995] linked on the web page.