Notes 14: Weak and strong learning

1. Weak learning

Recall PAC learning definition (henceforth strong PAC learning):
Algorithm A PAC learns C if
for any concept \( c \in C \) and any distribution \( D \) over \( X \)
for any confidence parameter \( \delta > 0 \) and any accuracy parameter \( \varepsilon > 0 \)
when \( A \) takes \( m \) samples from \( \text{EX}(c, D) \)
with prob. \( \geq 1 - \delta \), \( A \) outputs hypothesis with error \( \leq \varepsilon \)
\( A \) needs to work for arbitrarily small \( \delta > 0 \) and \( \varepsilon > 0 \): stringent requirement!
What if \( A \) only is guaranteed to work for some \( \delta > 0 \) and \( \varepsilon > 0 \)? (much weaker guarantee)
Turns out \( A \) can be boosted to a strong learning algorithm

2. Boosting confidence

Suppose algorithm \( A \), with probability \( \geq 2/3 \), outputs hypothesis with error \( \leq \varepsilon \) (for any \( \varepsilon > 0 \))
\( A \)'s confidence \( \delta \) bounded away from 0
Can be converted to strong PAC algorithm (with arbitrarily small \( \delta \) and \( \varepsilon \)):

Repeat \( t = 1, \ldots, R \) times:
Run \( A \) on independent samples, with accuracy being \( \varepsilon/2 \), to get hypothesis \( h_t \)
Draw \( m' \) more samples \( S \) to evaluate hypotheses \( h_1, \ldots, h_R \)
Output the hypothesis with least empirical error on \( S \)

\( R \) def \( = \frac{3}{2} \ln \frac{2}{\delta} = O \left(\ln \frac{1}{\delta}\right) \) so that
\[ \Pr \left[ \text{none of } h_1, \ldots, h_R \text{ has error } \leq \frac{\varepsilon}{2} \right] \leq \left(1 - \frac{2}{3}\right)^{3/2\ln(2/\delta)} \leq e^{-\ln(2/\delta)} = \frac{2}{\delta} \]

\( m' \) def \( = O \left(\frac{1}{\varepsilon} \ln \frac{1}{\delta}\right) \) so that
Chernoff + Union Bound: with prob. \( \geq 1 - \delta/2 \),
all bad hypotheses among \( h_1, \ldots, h_R \) have empirical error \( \geq \frac{5}{6} \varepsilon \); and
some \( \frac{5}{6} \)-accurate hypothesis among \( h_1, \ldots, h_R \) has empirical error \( \leq \frac{4}{6} \varepsilon \)
Hence any hypothesis with least empirical error must have (true) error \( \leq \varepsilon \)
Algorithm \( B \) succeeds with prob \( \geq 1 - \delta \)
\( A \) uses \( m = \text{poly} \left(\frac{1}{\varepsilon}\right) \) samples \( \implies \) \( B \) uses \( Rm + m' = \text{poly} \left(\frac{1}{\varepsilon}, \ln \frac{1}{\delta}\right) \) samples
\( A \) runs in \( T = \text{poly} \left(\frac{1}{\varepsilon}\right) \) time \( \implies \) \( B \) runs in \( RT + m' \text{ poly} \left(\frac{1}{\varepsilon}, \ln \frac{1}{\delta}\right) \) time
Summary: \( O \left(\frac{1}{\varepsilon} \ln \frac{1}{\delta}\right) \) calls to \( A \); \( O \left(\frac{1}{\varepsilon} \ln \frac{1}{\delta}\right) \) further samples to test the hypotheses

3. Boosting accuracy

Call algorithm \( A \) weak PAC learning algorithm with advantage \( \gamma \) if
for any \( c \in C \), for any distribution \( D \), for any \( \delta > 0 \)
with probability \( \geq 1 - \delta \), output hypothesis \( h \) with \( \text{err}_D(h, c) \leq \frac{1}{2} - \gamma \)
Getting advantage \( \gamma = 0 \) (i.e. \( \text{err}_D(h, c) = \frac{1}{2} \)) is trivial: just output uniformly random guess
Goal: Turn any weak PAC algorithm \( A \) with advantage \( \gamma \) into strong PAC algorithm
with \( \text{poly} \left(\frac{1}{\varepsilon}, \frac{1}{\gamma}, \frac{1}{\delta}\right) \) overhead in \#samples and running time
Will show efficient boosting algorithm \( B \) with following structure
Boosting algorithm $B$

- Draw independent training samples $S = \{(x^1, c(x^1)), \ldots, (x^m, c(x^m))\}$ from $\text{EX}(c, D)$
- Initially set $D_1 = \text{uniform distribution over } S$
- Repeat $t = 1, \ldots, R$ times:
  - Run $A$ on independent samples from $\text{EX}(c, D_t)$ to get hypothesis $h_t$
  - Adjust $D_t$ according to $h_t$ to get updated distribution $D_{t+1}$ over $S$
- Combine hypotheses $h_1, \ldots, h_R$ to get hypothesis $h$

Missing details:
- What are $D_2, D_3, \ldots$?
- How to combine $h_1, \ldots, h_R$ into $h$?
- Why $\text{err}_D(h, c) \leq \varepsilon$?

History: Theory influenced practical algorithms!
- Kearns and Valiant (1989): introduced weak learning, showing weakening learning may still be hard
- Freund and Schapire (1990): weak and strong learning are equivalent in distribution-free setting
- Freund and Schapire (1995): AdaBoost, now part of many machine learning libraries