SOME THOUGHTS ABOUT THE BINARY WEIGHTING ALGORITHM

1. BW can support an arbitrary prior over experts

When modified to support a prior, the main theorem is:

Theorem 1.1. *For all k, all priors $P(c)$, and all sequences $s \in (X \times \{0,1\})^+$; if any classifier c makes less than k mistakes, then the total number of mistakes of $BW(k,P)$ on s is at most*

$$m_c \equiv \max q : q \leq \frac{1}{P(c)} + \log \binom{q}{\leq k}$$

Proof. Suppose the master algorithm makes q mistakes. The measure on the consistent variant of c must be at least $\frac{P(c)}{\binom{q}{\leq k}}$. The measure on inconsistent variants must decrease by a factor of 2 after each mistake. This implies that after making m mistakes, the total measure is at most $\frac{1}{2^m}$. We must have $\frac{1}{2^q} \geq \frac{P(c)}{\binom{q}{\leq k}}$ because any consistent variant never loses measure. Consequently, we have $\log \binom{q}{\leq k} + \log \frac{1}{P(c)} \geq q$ $\square$

2. BW Can work with exactly k mistakes

To see this, note that the above argument works when $\binom{q}{\leq k}$ is substituted with $\binom{q}{k}$.

Theorem 2.1. *For all k, all priors $P(c)$, and all sequences $s \in (X \times \{0,1\})^+$; if any classifier c makes less than k mistakes, then the total number of mistakes of $BW_{exact}(k,P)$ on s is at most*

$$m_c \equiv \max q : q \leq \frac{1}{P(c)} + \log \binom{q}{k}$$

Proof. Exactly the same as above. $\square$

3. The Prior can Easily Eliminate Knowledge of k

In particular, we can state the following theorem.

Theorem 3.1. *For all priors $P(c)$, and all sequences $s \in (X \times \{0,1\})^+$; if any classifier c makes less than k mistakes, then the total number of mistakes of $BW_{exact}(P)$ on s is at most*

$$m_c \equiv \max q : q \leq \frac{1}{P(c)} + \log \binom{q}{k} + \log m$$

Proof. The rough idea is that we place weight $\frac{1}{m}$ on every possible value of k. This weakens the measure on the correct k by only a factor of m. We may be able to make this work for $\frac{1}{\sqrt{m}}$ by noting that 'nearby' k give 'similar' weightings.

By being clever, we can always use $m=$ the number of the next element in the sequence. $\square$

4. The Randomization Issue

The original BW paper left the issue of randomization open.

For the version space algorithm, switching to the probabilistic setting saves a factor of 2 in the efficiency. If the measure of the vote is x and $1-x$, the exact algorithm plays the smaller vote class with probability $\frac{1}{\log_2 \frac{1}{x} + \log_2 \frac{1}{1-x}}$.

The difficulty with this improvement for the BW algorithm is that it seems to be incompatible with the 'conservative' strategy. In fact, an optimal adversary against a conservative probabilistic algorithm could force $N-1$ mistakes.

How can we gain the advantage of conservative play without suffering the drawback of a purely conservative strategy?