Introduction to Algorithms (CS 482)	Cornell University
Instructor: Bobby Kleinberg	Lecture Notes, 30 April 2008

Online Prediction Algorithms

1 Binary prediction with one perfect expert

As a warm-up for the algorithms to be presented below, let's consider the following "toy problem." The algorithm's goal is to predict the bits of an infinite binary sequence $B = (b_1^*, b_2^*, \ldots)$, whose bits are revealed one at a time. Just before the t-th bit is revealed, a set of n experts make predictions $b_{1t}, b_{2t}, \ldots, b_{nt} \in \{0, 1\}$. The algorithm is allowed to observe all of these predictions, then it makes a guess denoted by $a_t \in \{0, 1\}$, and then the truth, b_t^*, is revealed. We are given a promise that there is at least one expert whose predictions are always accurate, i.e. we are promised that $\exists i \, \forall t \; b_{it} = b_t^*$.

Consider the following algorithm, which we will call the "Majority algorithm": at each time t, it consults the predictions of all experts who did not make a mistake during one of the first $t-1$ steps. (In other words, it considers all experts i such that $b_{is} = b_s^*$ for all $s < t$.) If more of these experts predict 1 than 0, then $a_t = 1$; otherwise $a_t = 0$.

Theorem 1. *Assuming there is at least one expert i such that $b_{it} = b_t^*$ for all t, the Majority algorithm makes at most $\lfloor \log_2(n) \rfloor$ mistakes.*

Proof. Let S_t denote the set of experts who make no mistakes before time t. Let $W_t = |S_t|$. If the Majority algorithm makes a mistake at time t, it means that at least half of the experts in S_t made a mistake at that time, so $W_{t+1} \leq \lfloor W_t/2 \rfloor$. On the other hand, by assumption we have $|W_t| \geq 1$ for all t. Thus the number of mistakes made by the algorithm is bounded above by the number of iterations of the function $x \mapsto \lfloor x/2 \rfloor$ required to get from n down to 1. This is $\lfloor \log_2(n) \rfloor$. □

Remark 2. The bound of $\lfloor \log_2(n) \rfloor$ in Theorem 1 is information-theoretically optimal, i.e. one can prove that no deterministic algorithm makes strictly fewer than $\lfloor \log_2(n) \rfloor$ mistakes on every input.

Remark 3. Although the proof of Theorem 1 is very easy, it contains the two essential ingredients which will reappear in the analysis of the Weighted Majority and Hedge algorithms below. Namely, we define a number W_t which measures the "remaining amount of credibility" of the set of experts at time t, and we exploit two key properties of W_t:

- When the algorithm makes a mistake, there is a corresponding multiplicative decrease in W_t.

Algorithm $\mathsf{WMA}(\varepsilon)$

/* Initialization */
$w_i \leftarrow 1$ **for** $i = 1, 2, \ldots, n$.

/* Main loop */
for $t = 1, 2, \ldots$
    /* Make prediction by taking weighted majority vote */
    if $\sum_{i\,:\,b_{it}=0} w_i > \sum_{i\,:\,b_{it}=1} w_i$
        output $a_t = 0$;
    else
        output $a_t = 1$.

    Observe the value of b_t^*.

    /* Update weights multiplicatively */
    $E_t \leftarrow$ {experts who predicted incorrectly}
    $w_i \leftarrow (1 - \varepsilon) \cdot w_i$ **for all** $i \in E_t$.
end

Figure 1: The weighted majority algorithm

- The assumption that there is an expert whose predictions are close to the truth implies a lower bound on the value of W_t for all t.

The second property says that W_t can't shrink too much starting from its initial value of n; the first property says that if W_t doesn't shrink too much then the algorithm can't make too many mistakes. Putting these two observations together results in the stated mistake bound. Each of the remaining proofs in these notes also hinges on these two observations, although the manipulations required to justify the two observations become more sophisticated as the algorithms we are analyzing become more sophisticated.

2 Deterministic binary prediction: the Weighted Majority Algorithm

We now present an algorithm for the same binary prediction problem discussed in Section 1. This new algorithm, the Weighted Majority algorithm, satisfies a provable mistake bound even when we don't assume that there is an expert who never makes a mistake. The algorithm is shown in Figure 1. It is actually a one-parameter family of algorithms $\mathsf{WMA}(\varepsilon)$, each with a preconfigured parameter $\varepsilon \in (0, 1)$.

Theorem 4. *Let M denote the number of mistakes made by the algorithm* $\mathsf{WMA}(\varepsilon)$*. For every integer m, if there exists an expert i which makes at most m mistakes, then*

$$M < \left(\frac{2}{1-\varepsilon}\right) m + \left(\frac{2}{\varepsilon}\right) \ln(n).$$

Proof. Let w_{it} denote the value of w_i at the beginning of the t-th iteration of the main loop, and let $W_t = \sum_{i=1}^{n} w_{it}$. The hypothesis implies that there is an expert i such that $w_{iT} \geq (1-\varepsilon)^m$ for all T, so

$$W_T > w_{iT} \geq (1-\varepsilon)^m \tag{1}$$

for all T. On the other hand, if the algorithm makes a mistake at time t, it implies that

$$\sum_{i \in E_t} w_{it} \geq \frac{W_t}{2},$$

hence

$$\begin{aligned} W_{t+1} &= \sum_{i \in E_t} (1-\varepsilon) \cdot w_{it} + \sum_{i \notin E_t} w_{it} \\ &= \sum_{i=1}^{n} w_{it} - \varepsilon \sum_{i \in E_t} w_{it} \\ &\leq W_t \left(1 - \frac{\varepsilon}{2}\right). \end{aligned}$$

For any $T > 0$, we find that

$$\frac{W_T}{W_0} = \prod_{t=0}^{T-1} \frac{W_{t+1}}{W_t} \leq \left(1 - \frac{\varepsilon}{2}\right)^M \tag{2}$$

where M is the total number of mistakes made by the algorithm $\mathsf{WMA}(\varepsilon)$. Combining (1) with (2) and recalling that $W_0 = \sum_{i=1}^{n} w_{i0} = \sum_{i=1}^{n} 1 = n$, we obtain

$$\frac{(1-\varepsilon)^m}{n} < \frac{W_T}{W_0} \leq \left(1 - \frac{\varepsilon}{2}\right)^M.$$

Now we take the natural logarithm of both sides.

$$\begin{aligned} \ln(1-\varepsilon)m - \ln(n) &< \ln\left(1 - \frac{\varepsilon}{2}\right) M && (3) \\ \ln(1-\varepsilon)m - \ln(n) &< -(\varepsilon/2)M && (4) \\ \ln\left(\frac{1}{1-\varepsilon}\right) m + \ln(n) &> (\varepsilon/2)M && (5) \\ \left(\frac{2}{\varepsilon}\right) \ln\left(\frac{1}{1-\varepsilon}\right) m + \left(\frac{2}{\varepsilon}\right) \ln(n) &> M && (6) \\ \left(\frac{2}{1-\varepsilon}\right) m + \left(\frac{2}{\varepsilon}\right) \ln(n) &> M && (7) \end{aligned}$$

Algorithm Hedge(ε)

/* Initialization */
$w_x \leftarrow 1$ **for** $x \in [n]$

/* Main loop */
for $t = 1, 2, \ldots$
    /* Define distribution for sampling random strategy */
    for $x \in [n]$
        $p_t(x) \leftarrow w_x \Big/ \left(\sum_{y=1}^{n} w_y\right)$
    end
    Choose $x_t \in [n]$ at random according to distribution p_t.
    Observe cost function c_t.

    /* Update score for each strategy */
    for $x \in [n]$
        $w_x \leftarrow w_x \cdot (1-\varepsilon)^{c_t(x)}$
    end
end

Figure 2: The algorithm Hedge(ε).

where (4) was derived from (3) using identity (21) from the appendix of these notes, and (7) was derived from (6) using identity (22) from the appendix. □

3 Randomized prediction: the Hedge Algorithm

We now turn to a generalization of the binary prediction problem: the "best expert" problem. In this problem, there is again a set of n experts, which we will identify with the set $[n] = \{1, 2, \ldots, n\}$. In each time step t, the adversary designates a cost function c_t from $[n]$ to $[0, 1]$, and the algorithm chooses an expert $x_t \in [n]$. The cost function c_t is revealed to the algorithm only after it has chosen x_t. The algorithm's objective is to minimize the sum of the costs of the chosen experts, i.e. to minimize $\sum_{t=1}^{\infty} c_t(x_t)$. Observe that the binary prediction problem is a special case of the best expert problem, in which we define $c_t(x) = 1$ if $b_{xt} \neq b_t^*$, 0 otherwise.

Figure 2 presents a randomized online algorithm for the best expert problem. As before, it is actually a one-parameter family of algorithms Hedge(ε) with a preconfigured parameter $\varepsilon \in (0, 1)$. Note the algorithm's similarity to WMA(ε): it maintains a vector of weights, one for each expert, and it updates these weights multiplicatively using a straightforward generalization of the multiplicative update rule in WMA. The

main difference is that WMA makes its decisions by taking a weighted majority vote of the experts, while Hedge makes its decisions by performing a weighted random selection of a single expert.

Theorem 5. *For every randomized adaptive adversary, for every $T > 0$, the expected cost suffered by* Hedge(ε) *satisfies*

$$\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t)\right] < \left(\frac{1}{1-\varepsilon}\right)\mathbf{E}\left[\min_{x\in[n]}\sum_{t=1}^{T} c_t(x)\right] + \left(\frac{1}{\varepsilon}\right)\ln(n). \tag{8}$$

Proof. Let w_{xt} denote the value of w_x at the beginning of the t-th iteration of the main loop, and let $W_t = \sum_{x=1}^{n} w_{xt}$. Note that w_{xt}, W_t are random variables, since they depend on the adversary's choices which in turn depend on the algorithm's random choices in previous steps. For an expert $x \in [n]$, let $c_{1..T}(x)$ denote the total cost

$$c_{1..T}(x) = \sum_{t=1}^{T} c_t(x).$$

Let $x^* = \arg\min_{x\in[n]} c_{1..T}(x)$. We have

$$W_T > w_{x^*t} = (1-\varepsilon)^{c_{1..T}(x^*)}$$

and after taking logarithms of both sides this becomes

$$\ln(W_T) > \ln(1-\varepsilon)c_{1..T}(x^*) \tag{9}$$

On the other hand, we can bound the expected value of $\ln(W_T)$ from above, using an

inductive argument. Let w_{*t} denote the vector of weights $(w_{1t}, \ldots, w_{nt})$.

$$\begin{aligned}
\mathbf{E}(W_{t+1} \mid w_{*t}) &= \sum_{x=1}^{n} \mathbf{E}\left((1-\varepsilon)^{c_t(x)} w_{xt} \mid w_{*t}\right) && (10)\\
&\leq \sum_{x=1}^{n} \mathbf{E}\left((1-\varepsilon c_t(x)) w_{xt} \mid w_{*t}\right) && (11)\\
&= \sum_{x=1}^{n} w_{xt} - \varepsilon \mathbf{E}\left(\sum_{x=1}^{n} c_t(x) w_{xt} \mid w_{*t}\right) && (12)\\
&= W_t \cdot \left(1 - \varepsilon \mathbf{E}\left(\sum_{x=1}^{n} c_t(x) p_t(x) \mid w_{*t}\right)\right) && (13)\\
&= W_t \cdot (1 - \varepsilon \mathbf{E}(c_t(x_t) \mid w_{*t})) && (14)\\
\mathbf{E}(\ln(W_{t+1}) \mid w_{*t}) &\leq \ln(W_t) + \ln(1 - \varepsilon \mathbf{E}(c_t(x_t) \mid w_{*t})) && (15)\\
&\leq \ln(W_t) - \varepsilon \mathbf{E}(c_t(x_t) \mid w_{*t}) && (16)\\
\varepsilon \mathbf{E}(c_t(x_t) \mid w_{*t}) &\leq \ln(W_t) - \mathbf{E}(\ln(W_{t+1}) \mid w_{*t}) && (17)\\
\varepsilon \mathbf{E}(c_t(x_t)) &\leq \mathbf{E}(\ln(W_t)) - \mathbf{E}(\ln(W_{t+1})) && (18)\\
\varepsilon \mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right) &\leq \ln(n) - \mathbf{E}(\ln(W_T)). && (19)
\end{aligned}$$

Here, (11) is derived using identity (23) from the appendix, (13) is derived using the fact that $p_t(x) = w_{xt}/W_t$, (14) is derived using the observation that x_t is a random element sampled from the probability distribution $p_t(\cdot)$ on $[n]$, (15) and (16) are derived using the identities (24) and (21) respectively, (18) is derived by taking the unconditional expectation of both sides of the inequality, and (19) is derived by summing over t and recalling that $W_0 = n$.

Combining (9) and (19) we obtain

$$\begin{aligned}
\varepsilon \mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right) &< \ln(n) - \ln(1-\varepsilon)\mathbf{E}(c_{1..T}(x^*))\\
\mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right) &< \left(\frac{1}{\varepsilon}\right)\ln(n) + \frac{1}{\varepsilon}\ln\left(\frac{1}{1-\varepsilon}\right)\mathbf{E}(c_{1..T}(x^*))\\
\mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right) &< \left(\frac{1}{\varepsilon}\right)\ln(n) + \left(\frac{1}{1-\varepsilon}\right)\mathbf{E}(c_{1..T}(x^*))
\end{aligned}$$

where the last line is derived using identity (22) from the appendix. □

4 Appendix: Some useful inequalities for logarithms and exponential functions

In various steps of the proofs given above, we applied some useful inequalities that follow from the convexity of exponential functions or the concavity of logarithms. In this section we collect together all of these inequalities and indicate their proofs.

Lemma 6. *For all real numbers x,*

$$1 + x \leq e^x \tag{20}$$

with equality if and only if $x = 0$.

Proof. The function e^x is strictly convex, and $y = 1 + x$ is the tangent line to $y = e^x$ at $(0, 1)$. □

Lemma 7. *For all real numbers $x > -1$,*

$$\ln(1 + x) \leq x \tag{21}$$

with equality if and only if $x = 0$.

Proof. Take the natural logarithm of both sides of (20). □

Lemma 8. *For all real numbers $y \in (0, 1)$,*

$$\frac{1}{y} \ln\left(\frac{1}{1-y}\right) < \frac{1}{1-y}. \tag{22}$$

Proof. Apply (21) with $x = \frac{y}{1-y}$, then divide both sides by y. □

Lemma 9. *For every pair of real numbers $x \in [0, 1], \varepsilon \in (0, 1)$,*

$$(1 - \varepsilon)^x \leq 1 - \varepsilon x \tag{23}$$

with equality if and only if $x = 0$ or $x = 1$.

Proof. The function $y = (1 - \varepsilon)^x$ is strictly convex and the line $y = 1 - \varepsilon x$ intersects it at the points $(0, 1)$ and $(1, 1 - \varepsilon)$. □

Lemma 10. *For every random variable X, we have*

$$\mathbf{E}(\ln(X)) \leq \ln(\mathbf{E}(X)) \tag{24}$$

with equality if and only if there is a constant c such that $\Pr(X = c) = 1$.

Proof. Jensen's inequality for convex functions says that if f is a convex function and X is a random variable,

$$\mathbf{E}(f(X)) \geq f(\mathbf{E}(X)),$$

and that if f is strictly convex, then equality holds if and only if there is a constant c such that $\Pr(X = c) = 1$. The lemma follows by applying Jensen's inequality to the strictly convex function $f(x) = -\ln(x)$. □