CS 6840 Algorithmic Game Theory	**Adaptive Game Playing**
Instructor: Eva Tardos	**March 10, 2010**

Adaptive Game Playing: Weighted Majority

We focus on a single player, who has n options to choose from. Assume that at time step t if he chooses option s he gets a reward of a_s^t and we normalize things so that $1 \leq a_s^t \leq 1$ for all $t \geq 0$ and all $s \in \{1, \ldots, n\}$. The goal is to design an algorithm that gets reward at least the best single option

$$B^T = \max_s \sum_{t=1}^{T} a_s^t$$

without knowing the values a_s^t in advance. We will assume that after the decision is make for time t, all values a_s^t are revealed, not only the value for the choice s used. For a randomized algorithm $\mathcal{A}$ we will use $V^T(\mathcal{A})$ to denote the expected reward till time T, and let

$$R(\mathcal{A}) = V^T(\mathcal{A}) - \max_s \sum_{t=1}^{T} a_s^t$$

be the regret if the algorithm till time T.

The idea is that the player maintains a *weight* for each option, and picks options proportionally to the weights. When he/she sees one of the options is good, he/she increases the weight, so a sto choose it more often in the future.

$w_s^t \geq 0$	the weight of option s for round t
$W^t = \sum_s w_s^t$	the total weight of options in round t
$w_s^1 = 1$	the initial weight of option s, so $W^1 = n$
$p_s^t = w_s^t / W_t$	probability of picking option s in round t

We we set the way weights are updated. We will select a small value $\epsilon > 0$ later, and use $w_s^{t+1} = (1+\epsilon)^{a_s^t} w_s^t$.

Theorem 1 *For a sufficiently large T (depending on ϵ) the above algorithm $\mathcal{A}$ has regret $R(\mathcal{A}) \leq \epsilon T$*

Note that we can think of $\epsilon > 0$ as effecting the learning rate, when ϵ is small, the adjustment are small, and learning will take a long time, but the bound will get better.

Let V^t be the expected reward collected in round t, that is $V^t = \sum_s p_s^t a_s^t$. By definition

$$V^t = \sum\nolimits_s p_s^t a_s^t = \sum\nolimits_s a_s^t \frac{w_s^t}{W^t}$$

so the total expected payoff over all rounds is just $\sum_t V^t = \sum_t \sum_s p_s^t a_s^t$.

The weights are independent of the player's moves, so we can look at how the total weight changes after each round. When $a_s^t \in \{0, 1\}$ (takes values either 0 or 1), we have

$$\begin{aligned} W^{t+1} &= W^t + \epsilon \sum\nolimits_i a_{i,t} w_{i,t} \\ &= W^t + \epsilon W^t \sum\nolimits_i a_{i,t} \frac{w_{i,t}}{W_t} \\ &= W^t + \epsilon W^t V^t \quad = \quad W^t(1 + \epsilon V_t) \end{aligned}$$

The first equation was true as when a_s^t is 0 or 1 $(1+\epsilon)^{a_s^t} = 1 + \epsilon a_s^t$. When $a_s^t \in [0,1]$ (takes values between 0 or 1), we have instead that $W^{t+1} \leq W^t(1+\epsilon V_t)$ as we can use that $(1+\epsilon)^{a_s^t} \leq 1 + \epsilon a_s^t$. This is true as $(1+\epsilon)^x$ is a convex function of x. The right hand side is the line connecting the function values at $x = 0$ and $x = 1$, and convex functions take values less than or equal to the connecting line.

The idea of the analysis is that if there is a single option s with high total reward, that option has high weight, and hence W^T is high. On the other hand we just saw that the weight grows proportional to the expected reward of the algorithm. More formally, we gave that $W^{T+1} \geq \max_s w_s^T = (1+\epsilon)^{B^T}$ on one hand, and

$$W^{T+1} \leq W^1 \prod_t \left(1 + \epsilon V^t\right) = n \prod_t \left(1 + \epsilon V^t\right)$$

on the other hand. Combining these, and recalling that $\epsilon \geq \ln(1+\epsilon) \geq \epsilon - \frac{\epsilon^2}{2}$, we get

$$\begin{aligned}
n \prod_t \left(1 + \epsilon V^t\right) &\geq (1+\epsilon)^{B^T} \\
\ln n + \sum_t \ln(1 + \epsilon V^t) &\geq B^T \ln(1+\epsilon) \\
\ln n + \epsilon \sum_t V^t &\geq B^T (\epsilon - \frac{\epsilon^2}{2}) \\
\sum_t V^t &\geq B^T - \frac{\ln n}{\epsilon} - B^T \frac{\epsilon}{2}
\end{aligned}$$

The left term is exactly the total expected payoff, so the player selects selects ϵ to maximize the right term. This happens when $\epsilon = \sqrt{2\frac{\ln n}{B^T}}$, giving a payoff $\sum_t V^t \geq B^T - 2\sqrt{2B^T \ln n}$ close to B^T. However, there is a slight cheat here: the player does not know B^T at the start of the game, and so cannot select ϵ.

To get our claimed theorem, we only need $\frac{\ln n}{\epsilon} \leq \epsilon T/2$, which we get letting $T \geq 2\frac{\ln n}{\epsilon^2}$. So the regret bound of ϵT is valid for high enough T, as claimed.