CS269: Machine Learning Theory
Lecture 10: Weighted Majority and Online Convex Optimization
October 27, 2010

Lecturer: Jake Abernethy
Scribe: Karan Chaudhry and Sendie Hudaya

1 Randomized Weighted Majority

In the last lecture, we introduced the Randomized Weighted Majority algorithm and began to prove a regret bound. Recall that Randomized Weighted Majority assigns an exponential weight $w_t(i)$ to each expert i at time t defined as follows:

$$w_t(i) = e^{-\eta L_{t-1}(i)}$$

where η is a learning parameter, $l_t \in [0,1]^n$ is the vector of expert losses at time t, and L_t is the vector of cumulative losses,

$$L_t = \sum_{s=1}^{t} l_s$$

We initialize each $w_1(i)$ to 1 and for each round t, the weight $w_t(i)$ is updated according to learning parameter η and loss at $t-1$. We wish to decrease the weight of an expert if it has high loss, and increase the weight if its loss is low. Also recall that in Randomized Weighted Majority algorithm, the player chooses predictions at random among experts. The probability of choosing an expert is simply the weight of that expert's prediction normalized by the sum of all expert weights:

$$p_t(i) = \tfrac{w_t(i)}{W}$$
$$W = \sum_{i=1}^{n} w_t(i)$$

In the last lecture, we started to prove the following regret bound for Randomized Weighted Majority.

Theorem 1. *Let L^* be a known upper bound on the loss of the best performing expert, and assume $L^* \geq 2\log n$. The regret of Randomized Weighted Majority run with parameter $\eta = \sqrt{2\log n / L^*}$ is no more than $2\sqrt{2L^* \log n}$.*

Note that we can set $L^* = T$ if no other information is known.[1] This bound implies that the regret of Randomized Weighted Majority is $O(\sqrt{T \log n})$. Last time we introduced two tricks that will help us prove the bound on regret:

Fact 1. $\log(1+x) \leq x, \forall x$ *(Note this is equivalent to $1 + x \leq e^x$ from lecture 2)*

[1] Of course we still need to know T to do this. There are tricks for achieving low regret when T is not known in advance, including the common "doubling trick", but we won't get to those here.

Fact 2. $e^{\alpha x} - 1 \leq (e^{\alpha} - 1)x, \forall \alpha, \forall x \in [0, 1]$

Let us now complete the proof of the regret bound for Randomized Weighted Majority.
Proof: By the end of last lecture, we arrived at the following inequality, where i^* is the "best" expert, that is, the expert with the lowest cumulative loss:

$$\sum_{t=1}^{T} p_t \cdot l_t \leq \frac{1}{1 - e^{-\eta}} \left(\log \left(\sum_{i=1}^{n} w_1(i) \right) - \log \left(\sum_{i=1}^{n} w_{T+1}(i) \right) \right)$$

Using this, we can bound the cumulative loss of the algorithm as follows:[2]

$$\begin{aligned}
\sum_{t=1}^{T} p_t \cdot l_t &\leq \frac{1}{1 - e^{-\eta}} \left(\log \left(\sum_{i=1}^{n} w_1(i) \right) - \log \left(\sum_{i=1}^{n} w_{T+1}(i) \right) \right) \\
&= \frac{1}{1 - e^{-\eta}} \left(\log(n) - \log \left[\sum_{i=1}^{n} e^{-\eta L_T(i)} \right] \right) && (1) \\
&\leq \frac{1}{1 - e^{-\eta}} \left(\log(n) - \log \left(e^{-\eta L_T(i^*)} \right) \right) \\
&= \frac{1}{1 - e^{-\eta}} \left(\log(n) + \eta L_T(i^*) \right) \\
&\leq \frac{1}{1 - e^{-\log(1+\eta)}} \left(\log(n) + \eta L_T(i^*) \right) && (2) \\
&= \frac{1}{1 - \frac{1}{1+\eta}} \left(\log(n) + \eta L_T(i^*) \right) \\
&= \frac{1 + \eta}{\eta} \left(\log(n) + \eta L_T(i^*) \right)
\end{aligned}$$

Remember that w_1 is initialized to 1 for all i, so $\sum_{i=1}^{n} w_1(i) = n$. Equation 2 follows from the fact that the sum of expert losses is less than the loss of any single expert. Equation 2 is true by Fact 1.

Assume that $\eta \leq 1$. (This will be true by the assumptions we have made in the theorem statement.) In this case, the above gives us that

$$\sum_{t=1}^{T} p_t \cdot l_t \leq L_T(i^*) + \eta L_T(i^*) + \frac{2}{\eta} \log(n)$$

and so

$$\sum_{t=1}^{T} p_t \cdot l_t - L_T(i^*) \leq \eta L_T(i^*) + \frac{2}{\eta} \log(n) \leq \eta L^* + \frac{2}{\eta} \log(n) \,.$$

This holds for any value of $\eta \in (0, 1]$. It is minimized when we set $\eta = \sqrt{2 \log n / L^*}$, so we will use this value of η. The bound above then becomes

$$\sum_{t=1}^{T} p_t \cdot l_t - L_T(i^*) \leq 2\sqrt{2L^* \log n}.$$

$\square$

[2] Note that this bound is presented slightly differently than it was in class. This presentation is a bit more simple, and doesn't require introducing an additional parameter β.

1.1 Using a Prior Over the Experts

Alternatively, we can change the initial weights to be non-uniform. Let $w_1 \in \Delta_n$ be the initial weights. Then $w_t(i) = e^{-\eta L_{t-1}(i)} w_1(i)$. The calculation for new total loss becomes:

$$\frac{1}{1-e^{-\eta}}\left(\log(1) - \log\left(\sum_{i=1}^{n} w_{T+1}(i)\right)\right) = \frac{1}{1-e^{-\eta}}\left(0 - \log\left[\sum_{i=1}^{n} e^{-\eta L_T(i)} w_1(i)\right]\right)$$
$$\leq \frac{1}{1-e^{-\eta}}\left(\log(1/w_1(i^*)) + \eta L_T(i^*)\right)$$

Now, the new bound on regret becomes:

$$2\sqrt{2L^* \log\left(\frac{1}{w_1(i^*)}\right)}$$

This bound is better if the best expert has a high initial weight.

2 Online Convex Optimization

We now consider a generalization of the expert advice setting. A convex programming problem consists of a convex feasible set $K \subseteq \mathbb{R}^n$ and a convex cost function $f : K \to \mathbb{R}$. In online convex optimization, an algorithm faces a sequence of convex programming problems, each with the same feasible set but different cost functions. Each time the algorithm must choose a point before it observes the cost function. This is a generalization of both work in minimizing error online and of the experts problem.

In the experts problem, one has n experts, each of which has a loss in $[0, 1]$ at each round. At each round, the algorithm selects a probability distribution over experts. The set of all probability distributions is a convex set. Also, the cost function on this set is linear, and therefore convex.

The online convex optimization problem can be formulated as a repeated game between a player and an adversary. At round t, the player chooses an action w_t from some convex subset K of $\mathbb{R}^n$, and the adversary chooses a convex loss function f_t. The players goal is to ensure that the total loss, $\sum_{t=1}^{T} f_t(w_t)$, does not differ from the smallest total loss $\sum_{t=1}^{T} f_t(w)$ for any fixed action w by too much. The difference between the total loss and its optimal value for a fixed action is termed $regret$, and denoted as

$$R_T = \sum_{t=1}^{T} f_t(w_t) - \min_{w\in K} \sum_{t=1}^{T} f_t(w)$$

Recall the following definitions, which we will use in our analysis.

Definition 1. *In Euclidean space, a set K is said to be convex if,*
$\forall x, y \in K$, $(1-\alpha)x + \alpha y \in K$ *for any* $\alpha \in [0, 1]$

Definition 2. *A function f is convex, if,*
$\forall x, y \in K$, $\alpha f(x) + (1-\alpha) f(y) \geq f(\alpha x + (1-\alpha) y)$ *for any* $\alpha \in [0, 1]$

Geometrically, this means that if you draw a line segment between the points$(x, f(x))$ and $(y, f(y))$, the function lies below this line segment.

Fact 3. *For a convex differentiable function f,*
$f(x) - f(y) \leq \nabla f(x)(x - y)$

Geometrically, the above statement means that if you draw the tangent plane to the function at point x, the function lies above it. Recall also that $\| a - b \|^2 = \| a \|^2 + \| b \|^2 - 2ab$.

2.1 Online Gradient Descent (Zinkevich '03)

We now introduce and analyze the Online Gradient Descent algorithm for online convex optimization. The algorithm is as follows:

1. Arbitrarily set w_1
2. for $t = 1$ to T do
3. Choose w_t, observe f_t
4. Update $\tilde{w}_{t+1} = w_t - \eta \nabla f_t(w_t)$ where $\eta > 0$ is a learning parameter
5. Set w_{t+1}= Euclidean projection of $\tilde{w}_{t+1}$on the convex set K
6. end for

We set $\tilde{w}_{t+1} = w_t - \eta \nabla f_t(w_t)$ because the gradient of a function gives the direction of steepest increase. Thus a natural minimization algorithm is to go in the direction opposite the gradient a certain amount. The above rule may lead us to choose a point $w \notin K$, which is a problem. A common trick is to move to the closest point to w in the set K and that is why we set w_{t+1} to be the Euclidean projection of $\tilde{w}_{t+1}$on the convex set K.

Note that the Perceptron algorithm is a special case of online gradient descent when loss is measured by the hinge loss $f_t(w_t) = \max(-w_t \cdot x_t y_t, 0)$.

Let D denote the diameter of K. This means that for every $x, x\prime \in K$,$\| x - x\prime \| \leq D$. Zinkevich showed that the regret of this algorithm grows as $\sqrt{T}$, where T is the number of rounds of the game.

Theorem 2. *Let* $G = max_{t \in [T]} \parallel \nabla f_t(w_t) \parallel$ *(i.e. G is an upper bound on the gradient magnitudes) and* $D = diameter\ K$*. The online gradient descent algorithm attains regret* $R_T \leq GD\sqrt{T}$

Proof: Let $w^* = \arg\min_{w \in K} \sum_{t=1}^{T} f_t(w)$
Let $\nabla_t = \nabla f_t(w_t)$
The distance between w_{t+1} and w^* would be less than or equal to the distance
between $\tilde{w}_{t+1}$ and w^*

$$\| w_{t+1} - w^* \|^2 \leq \| \tilde{w}_{t+1} - w^* \|^2$$
$$= \| w_t - \eta \nabla f_t(w_t) - w^* \|^2$$
$$= \| w_t - w^* \|^2 + \eta^2 \| \nabla_t \|^2 - 2\eta \nabla_t (w_t - w^*)$$

Rearranging terms, it can be seen that,

$$\nabla_t (w_t - w^*) \leq \frac{\| w_t - w^* \|^2 - \| w_{t+1} - w^* \|^2}{2\eta} + \frac{\eta}{2} \| \nabla_t \|^2$$

Fact 3 $\Rightarrow f_t(w_t) - f_t(w^*) \leq \nabla_t (w_t - w^*)$

Summing over $t = 1, 2, \ldots, T$,

$\sum_{t=1}^{T} (f_t(w_t) - f_t(w^*)) \leq \sum_{t=1}^{T} \nabla_t (w_t - w^*)$

$\leq \frac{\| w_1 - w^* \|^2}{2\eta} + \sum_{t=1}^{T} \frac{\eta}{2} \| \nabla_t \|^2$

$\leq \frac{D^2}{2\eta} + \frac{\eta}{2} T G^2$

Setting $\eta = \frac{D}{G\sqrt{T}}$, we get $R_T \leq GD\sqrt{T}$ $\square$

Why is this not a good bound for the Expert Setting ?

In the expert setting Diameter of $(\triangle_n) = \sqrt{2}$ and $G = \sqrt{n}$

$\Rightarrow R_T \leq \sqrt{2nT}$

which is not as good a bound when compared with the regret bound we derived for Weighted Majority since the dependence on n is now linear.