CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 22**

Online Convex Optimization

Lecturer: Alexander Rakhlin *Scribe: Jiening Zhan*

1 Recap: Prediction with Expert Advice

In the last lecture, the following online prediction algorithm based on expert advice was presented. Given a convex loss function $l(\cdot)$,

For $t = 1, ..., T$
player observes $f_{1,t}, ..., f_{1,N}$
player predicts p_t
adversary reveals outcome y_t
player suffers loss $l(p_t, y_t)$
experts suffer loss $l(f_{i,t}, y_t)$
End

The goal is to minimize the regret,

$$R_T = \sum_{t=1}^{T} l(p_t, y_t) - \min_{i \in [N]} \sum_{i=1}^{T} l(f_{i,t}, y_t) \tag{1}$$

By using exponential weights $e^{-\eta L_{i,t}}$, and predictions $p_t = \frac{\sum_i wi,tf_{i,t}}{\sum_i w_{i,t}}$, an upper bound to the regret was found to be $R_T \leq \sqrt{\frac{T}{2} \log N}$.

Based on this algorithm, the online linear optimization and the online convex optimization algorithm are derived.

2 Online Linear Optimization

Let Δ_N denote the N dimensional simplex.

For $t = 1, ..., T$
player predicts $\mathbf{w_t} \in \Delta_N$ ($\mathbf{w_t}$ is essentially a probability distribution)
adversary reveals $\mathbf{l_t} \in \mathbb{R}^N$
player suffers loss $\mathbf{w_t} \cdot \mathbf{l_t}$ where $\mathbf{l_t}(i) = l(f_{i,t}, y_t)$
End

The regret is defined as,

$$R_T = \sum_{t=1}^{T} \mathbf{w_t} \cdot \mathbf{l_t} - \min_{\mathbf{w}^* \in \Delta_N} \sum_{t=1}^{T} \mathbf{w}^* \cdot \mathbf{l_t} \tag{2}$$

Note that for the simplex, the distribution $\mathbf{w}^*$ will place all the probability on the best expert.

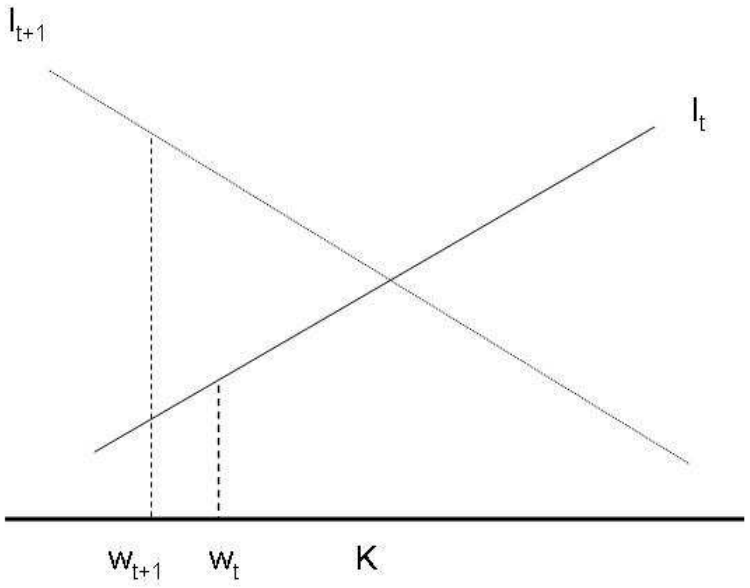

Figure 1: Online Linear Optimization (K is simplex)

3 Online Convex Optimization

Let K be a convex region. Also, $\forall t \in [T]$, let $l_t : \mathbb{R}^N \to \mathbb{R}$ be conxex.

For $t = 1, ..., T$
player predicts $\mathbf{w_t} \in K$
adversary reveals $\mathbf{l_t}(\cdot)$

player suffers loss $\mathbf{l_t}(\mathbf{w_t})$
End

The regret is

$$R_T = \sum_{t=1}^{T} l_t(\mathbf{w_t}) - \min_{\mathbf{w^* \in K}} \sum_{\mathbf{t=1}}^{\mathbf{T}} \mathbf{l_t}(\mathbf{w}^*) \tag{3}$$

Figure 3 gives an example of the online convex optimization algorithm. In this algorithm, the adversary is quite 'handicapped.' The worst that an adversary can do is change the loss function as demonstrated in Figure 3. However, we can always choose a w^* such that the adversary cannot produce too much loss.

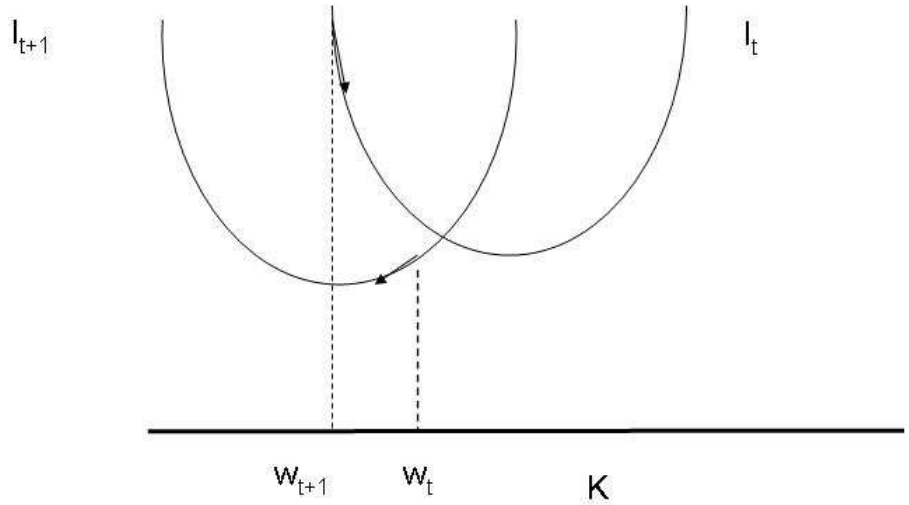

Figure 2: Online Convex Optimization

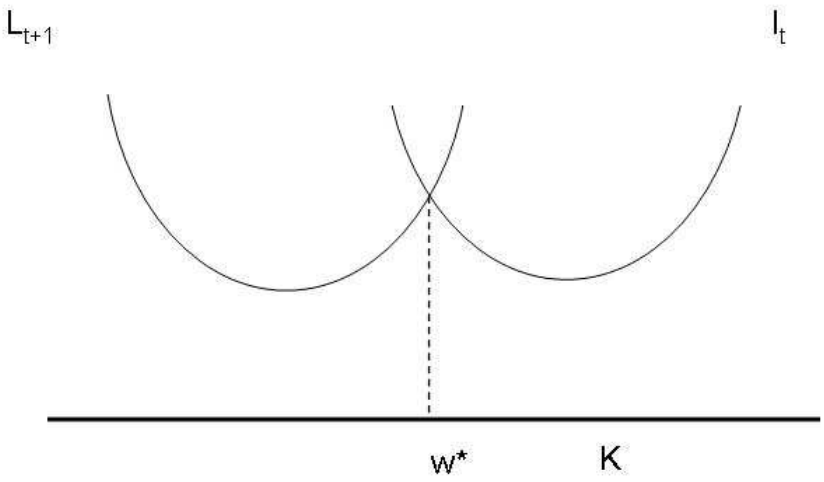

Figure 3: Adversary is quite handicapped in the algorithm

4 Algorithm: Online Gradient Descent, Zinkevich 03

For $t = 1, ..., T$
Predict w_t
Observe $l_t(\cdot)$
Update $w_{t+1} = \Pi_k(w_t - \eta \nabla l_t(w_t))$ where $\Pi_k(\cdot)$ is the euclidean projection onto the set K.
End

Theorem 4.1. Let $G = \max_{t \in [T]} \|\Delta l_t(w_t)\|$ and $D =$ diameter K. The online gradient descent algorithm attains regret $R_T \leq GD\sqrt{T}$.

PROOF. Let $\bar{w}_{t+1} = w_t - \eta \Delta l_t(w_t)$, $w* = \arg\min_{w \in K} \sum_{t=1}^{T} l_t(w)$ and $\Delta_t = \Delta l_t(w_t)$.

$$\|w_{t+1} - w^*\| \leq \|\bar{w}_{t+1} - w^*\| \tag{4}$$
$$= \|\bar{w}_t - \eta \Delta l_t(w_t) - w^*\|^2 \tag{5}$$
$$= \|w_t - w^*\|^2 + \eta^2 \|\Delta_t\|^2 - 2\eta \Delta_t (w_t - w^*) \tag{6}$$

Rearranging terms, it can be seen that

$$\Delta_t(w_t - w^*) \leq \frac{\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2}{2\eta} + \frac{\eta}{2} \|\Delta_t\|^2 \tag{7}$$

For a convex loss function l_t,

$$l_t(w_t) - l_t(w^*) \leq \Delta_t(w_t - w^*) \tag{8}$$

Summing over $t = 1, ..., T$,

$$\sum_{t=1}^{T} (l_t(w_t) - l_t(w^*)) \leq \Delta_t(w_t - w^*) \tag{9}$$
$$\leq \frac{\|w_1 - w^*\|^2}{2\eta} + \frac{\eta}{2} \|\Delta_t\|^2 \tag{10}$$
$$\leq \frac{D^2}{2\eta} + \frac{\eta}{2} T G^2 \tag{11}$$

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

Let K be a simplex of dimension N. It follows that $D = 1$, and $G \leq \sqrt{N}$. Therefore, the regret from the online gradient descent is $R_T \leq \sqrt{TN}$. Compared to the regret bound from the prediction using exponential weights $R_T \leq \sqrt{\frac{T}{2} \log N}$, this bound scales much more quickly in N.

The online descent algorithm behaves differently from the prediction with exponential weights algorithm. Assume that we are choosing weights from a three dimensional simplex. At time t, we choose weight w_t. If the there is no loss $l_t = (0, 0, 0)$, then for both algorithms, $w_{t+1} = w_t$. That is, our position does not change. However, if $l_t = (1, 0, 0)$, then at time $t + 1$, in the case of online gradient descent, we will move a distance of η away while in the case of exponential weights, we will move an exponential distance away. This is demonstated in Figure 4.

5 Bergman Divergence

Suppose $R : \mathbb{R}^N \to \mathbb{R}$ is strictly convex with continuous 1st order partial derivatives.

Definition. Bregman Divergence between x and y with respect to R is

$$D_R(x, y) = R(x) - R(y) - \Delta R(y)(x - y) \tag{12}$$

Note that the Bregman distance is in general not symmetric.

Example.

$$R(x) = \frac{1}{2} \|x\|^2 \Rightarrow D_R(x, y) = \frac{1}{2} \|x - y\|^2 \tag{13}$$

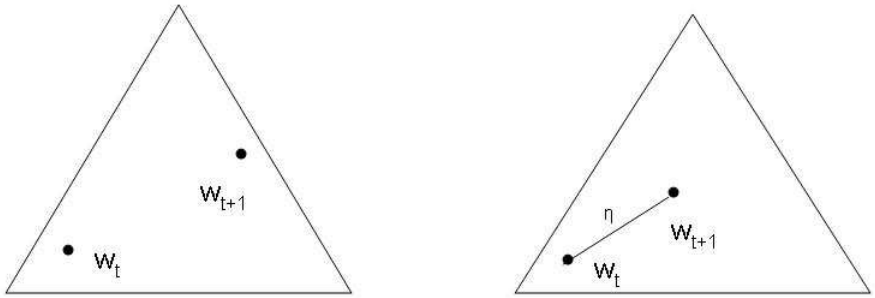

Figure 4: Exponential weights algorithm vs. online gradient descent

Example.

$$R(x) = \sum_{i=1}^{N}(x_i \log x_i - x_i) \Rightarrow D_R(x,y) = KL(x,y) - \sum_{i=1}^{N} x_i \log \frac{x_i}{y_i} + \sum_{i=1}^{N}(y_i - x_i) \tag{14}$$

Property.

$$D_{A+B}(x,y) = D_A(x,y) + D_B(x,y) \tag{15}$$

Property.

$$D_R(x,v) + D_R(v,w) = D_R(u,v) + (u-v)(\Delta R(w) - \Delta R(v)) \tag{16}$$

Property. The Bergman projection unto a convex set K exists and is unique. Let w' be the Bergman projection of the point w unto the convex set K. It follows

$$w' = \arg\min_{v \in K} D_R(v,w) \tag{17}$$

Property. Generalized Pythagorean Theorem: for all $u \in K$

$$D_R(u,w) \geq D_R(u,w') + D_R(w',w) \tag{18}$$