EECS 598-005: Theoretical Foundations of Machine Learning **Fall 2015**

Lecture 23: Online convex optimization

Lecturer: Jacob Abernethy *Scribes: Vikas Dhiman*

23.1 Online convex optimization: generalization of several algorithms

Many of the algorithms that we studied in this course are special cases of online convex optimization. We recall a few algorithms studied so far. Later on we will argue how they form a special case of online convex optimization.

Expert Setting

for $t \leftarrow 1$ **to** T **do**
  Algorithm chooses $\mathbf{w}_t \in \Delta_n$;
  Nature chooses loss $\boldsymbol{\ell}_t \in [0,1]^n$;
end

Algorithm 23.1: Expert setting

Linear Perceptron

for $t \leftarrow 1$ **to** T **do**
  Algorithm chooses $\mathbf{w}_t \in \mathbb{R}^n$;
  Nature chooses $(\mathbf{x}_t, y_t) \in \mathbb{R}^n \times \{-1, 1\}$;
  Algorithm experiences loss $\boldsymbol{\ell}_t = \mathbb{1}[(\mathbf{w}_t \mathbf{x}_t) y_t < 0] = \max(0, -y_t(\mathbf{w}_t \mathbf{x}_t))$;
end

Algorithm 23.2: Linear Perceptron

Online linear regression

for $t \leftarrow 1$ **to** T **do**
  Algorithm chooses $\mathbf{w}_t \in \mathbb{R}^n$;
  Nature chooses $\mathbf{x}_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}$;
  Algorithm experiences loss $\boldsymbol{\ell}_t = (\mathbf{w}_t \mathbf{x}_t - y_t)^2$;
end

Algorithm 23.3: Online linear regression

Portfolio selection

1 **for** $t \leftarrow 1$ **to** T **do**
2 | Algorithm chooses $\mathbf{w}_t \in \Delta_n$;
3 | Nature chooses price $\text{Price}_t(i)$ for all i;
4 | Algorithm experiences gain $g_t = \mathbf{w}_t \mathbf{x}_t$ where $\mathbf{x}_t(i) = \frac{\text{Price}_t(i)}{\text{Price}_{t-1}(i)}$;
5 **end**

Algorithm 23.4: Portfolio selection

23.2 Online convex optimization

Now we introduce the generalized form of convex optimization that generalizes the above given examples.

Given: A convex compact decision set: $X \subset \mathbb{R}^n$
A class of loss functions: $\mathcal{L} = \{l : l : X \to \mathbb{R}, l \text{ is convex and Lipshitz continuous } \}$
1 **for** $t \leftarrow 1$ **to** T **do**
2 | Algorithm chooses $\mathbf{w}_t \in \mathbb{X}$;
3 | Nature chooses $l_t \in \mathcal{L}$;
4 | Algorithm experiences loss $l_t(\mathbf{w}_t)$;
5 | Algorithm updates $\mathbf{w}_{t+1}$;
6 **end**

Algorithm 23.5: Generalized pattern of online convex optimization

Our goal is to design a rule—Alg. 23.5 step 5—that minimizes regret

$$\text{Regret}_T(\text{Alg}) := \sum_{t=1}^{T} l_t(\mathbf{w}_t) - \min_{\mathbf{w}^* \in \mathbb{X}} \sum_{t=1}^{T} l_t(\mathbf{w}^*) \ . \tag{23.1}$$

All problems listed above are the same convex optimization. [1] proposed online gradient descent (OGD) with the regret bound

$$\text{Regret}_T(\text{OGD}) \leq DG\sqrt{T} \ ,$$

where D is the L_2 diameter of convex compact set $\mathbb{X}$ and G is the L_2 Lipschitz bound on $\mathcal{L}$.

23.3 Online gradient descent

Initialize: $\mathbf{w}_t$ is the center of the set $\mathbb{X}$. Let η be a parameter.
1 **for** $t \leftarrow 1$ **to** T **do**
2 | Algorithm chooses $\mathbf{w}_t \in \mathbb{X}$;
3 | Nature chooses $l_t \in \mathcal{L}$;
4 | Algorithm experiences loss $l_t(\mathbf{w}_t)$;
5 | $\tilde{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t - \eta \nabla l_t(\mathbf{w}_t)$;
6 | $\mathbf{w}_{t+1} \leftarrow \text{Projection}_{\mathbb{X}}(\tilde{\mathbf{w}}_{t+1}) := \arg\inf_{\mathbf{w} \in X} \|\mathbf{w} - \tilde{\mathbf{w}}_{t+1}\|_2$;
7 **end**

Algorithm 23.6: Generalized pattern of online convex optimization

Theorem 23.1. *Regret—defined in* (23.1)*—for Algorithm 23.6 has an upper bound of* $DG\sqrt{T}$.

Proof:

Define potential function $\phi_t = -\frac{1}{2\eta}\|\mathbf{w}^* - \mathbf{w}_t\|_2^2$. The change in potential at every iteration is given by

$$\begin{aligned}
\phi_{t+1} - \phi_t &= -\frac{1}{2\eta}(\|\mathbf{w}^* - \mathbf{w}_{t+1}\|_2^2 - \frac{1}{2\eta}\|\mathbf{w}^* - \mathbf{w}_t\|_2^2) && (23.2)\\
&= -\frac{1}{2\eta}(\|\mathbf{w}^* - \text{Projection}_{\mathbb{X}}(\mathbf{w}_t - \eta\nabla l_t(\mathbf{w}_t))\|_2^2 - \|\mathbf{w}^* - \mathbf{w}_t\|_2^2) && (23.3)\\
&\geq -\frac{1}{2\eta}(\|\mathbf{w}^* - \mathbf{w}_t + \eta\nabla l_t(\mathbf{w}_t)\|_2^2 - \|\mathbf{w}^* - \mathbf{w}_t\|_2^2) && (23.4)\\
&= -\frac{1}{2\eta}(\eta^2\|\nabla l_t(\mathbf{w}_t)\|_2^2 - 2\eta\nabla l_t(\mathbf{w}^* - \mathbf{w}_t)) && (23.5)\\
&= -\nabla l_t(\mathbf{w}_t)(\mathbf{w}^* - \mathbf{w}_t) - \frac{\eta}{2}\|\nabla l_t(\mathbf{w}_t)\|_2^2 && (23.6)\\
&\geq -(l_t(\mathbf{w}^*) - l_t(\mathbf{w}_t)) - \frac{\eta}{2}\|\nabla l_t(\mathbf{w}_t)\|_2^2 && (23.7)\\
&\geq l_t(\mathbf{w}_t) - l_t(\mathbf{w}^*) - \frac{\eta}{2}G^2\ . && (23.8)
\end{aligned}$$

Here (23.4) is due to the fact that projection of point is to $\mathbb{X}$ always closer to any point inside $\mathbb{X}$. And (23.7) is due to convexity of $l_t \in \mathcal{L}$ because for any convex function f, $\nabla f(x)(y-x) \leq f(y) - f(x)$. Equation (23.8) is because of G-Lipschitz continuity of $l_t \in \mathcal{L}$.

Summing both sides of (23.8) from $t = 1$ to $t = T$, we get

$$\begin{aligned}
\text{Regret}_T(\text{OGD}) = \sum_{t=1}^{T} l_t(\mathbf{w}_t) - l_t(\mathbf{w}^*) \quad &\leq \sum_{t=1}^{T}\left(\phi_{t+1} - \phi_t + \frac{\eta}{2}G^2\right) && (23.9)\\
&= \phi_{T+1} - \phi_1 + \frac{\eta}{2}TG^2 && (23.10)\\
&\leq -\phi_1 + \frac{\eta}{2}TG^2 \quad = \frac{1}{2\eta}\|\mathbf{w}^* - \mathbf{w}_1\|_2^2 + \frac{\eta}{2}TG^2 && (23.11)\\
&\leq \frac{1}{2\eta}D^2 + \frac{\eta}{2}TG^2\ . && (23.12)
\end{aligned}$$

Equation (23.12) is because D is the diameter of the convex set $\mathbb{X}$. Since (23.12) is true for every η, we chose $\eta = \frac{D}{G\sqrt{T}}$ to get

$$\text{Regret}_T(\text{OGD}) \leq DG\sqrt{T}\ . \tag{23.13}$$

■

Note that Perceptron is OGD with $l_t(\mathbf{w}_t) = \max(0, -y_t(\mathbf{w}_t^\top \mathbf{x}_t)$ and gradient given by

$$\nabla l_t(\mathbf{w}_t) = \begin{cases} 0 & \text{if } y_t(\mathbf{w}_t^\top \mathbf{x}_t) > 0 \\ -y_t\mathbf{x}_t & \text{otherwise} \end{cases} \tag{23.14}$$

23.4 Further Generalization

A more generic algorithm is a modification of Follow the Regularized Leader (FTRL). In FTRL the update step is

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{X}} \sum_{s=1}^{t} l_s(\mathbf{w}) + \frac{1}{\eta}R(\mathbf{w})\ , \tag{23.15}$$

where $R(\mathbf{w})$ is the regulariser. Define a new algorithm FTRL'with the update step

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{X}} \sum_{s=1}^{t} \nabla l_s(\mathbf{w}) + \frac{1}{\eta} R(\mathbf{w}) \ . \tag{23.16}$$

Note that OGD is a special case of FTRL'with $R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$. Similarly Exponential Weights Algorithm (EWA) is also a FTRL'with $R(\mathbf{w}) = \sum_{i=1}^{k} \mathbf{w}_i \log(\mathbf{w}_i)$. The above observations are very recent. Only recently since 2008, we got to understand about the equivalence of OGD, EWA and FTRL'. The upper bound on regret of FTRL'is given by

$$\text{Regret}_T(\text{FTRL}) \leq \frac{R(\mathbf{w}^*) - R(\mathbf{w}_t)}{\eta} + \sum_{t=1}^{T} D_R(\mathbf{w}_t, \mathbf{w}_{t+1}) \ , \tag{23.17}$$

where $D_R(\mathbf{w}_t, \mathbf{w}_{t+1})$ is the Bragman Divergence. For more details please refer to the surveys on the course website.

23.5 Online to Batch Conversion

Standard stochastic setting. Assume that we have unknown distribution D on $X \times Y$. Let $(\mathbf{x}_t, y_t) \sim D$ for all $t = \{1, \ldots, T\}$. Use only online learning algorithm on this sequence

input : $(\mathbf{x}_t, y_t) \sim D$ for all $t = \{1, \ldots, T\}$.
Batch output: $\bar{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$
for $t \leftarrow 1$ **to** T **do**
  Algorithm chooses $\mathbf{w}_t \in \mathbb{X}$;
  Algorithm observes $(\mathbf{x}_t, y_t)$;
  Algorithm experiences loss given by convex loss function $l(\mathbf{w}_t|\mathbf{x}_t, y_t)$;
  $\mathbf{w}_{t+1} \leftarrow \text{Projection}_{\mathbb{X}}(\mathbf{w}_t - \eta\nabla l(\mathbf{w}_t|\mathbf{x}_t, y_t))$;
end

Algorithm 23.7: Online to batch conversion

Note that we are choosing the average weight $\bar{\mathbf{w}}_T$ rather than the last weight $\mathbf{w}_T$. This differences arises because of a different objective function than regret. For batch problem we want to minimize the risk,

$$\text{Risk}(\mathbf{w}) := \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} l(\mathbf{w}|\mathbf{x}, y) \tag{23.18}$$

Theorem 23.2. *The output of Online to Batch Conversion enjoys:*

$$\textit{Risk}(\bar{\mathbf{w}}_T) - \textit{Risk}(\mathbf{w}^*) \leq \mathbb{E}\left[\frac{\textit{Regret}_T(\textit{Alg})}{T}\right]$$

Proof: From the definition of risk of averaged weights we get,

$$\text{Risk}(\bar{\mathbf{w}}_T) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} l(\bar{\mathbf{w}}_T|\mathbf{x},y) \quad = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} l\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t \mid \mathbf{x},y\right) \tag{23.19}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} l(\mathbf{w}_t|\mathbf{x},y) \tag{23.20}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(\mathbf{x}_t,y_t)\sim D} l(\mathbf{w}_t|\mathbf{x}_t,y_t) \quad = \mathop{\mathbb{E}}_{(\mathbf{x}_t,y_t)\sim D}\left[\frac{1}{T}\sum_{t=1}^{T}\underbrace{l(\mathbf{w}_t|\mathbf{x}_t,y_t)}_{l_t(\mathbf{w}_t)}\right] \tag{23.21}$$

$$\leq \mathop{\mathbb{E}}_{(\mathbf{x}_t,y_t)\sim D}\left[\frac{1}{T}\sum_{t=1}^{T} l(\mathbf{w}^*|\mathbf{x}_t,y_t) + \frac{\text{Regret}_t(\text{Alg})}{T}\right] \tag{23.22}$$

$$= \text{Risk}(\mathbf{w}^*) + \mathbb{E}\left[\frac{\text{Regret}_T(\text{Alg})}{T}\right] . \tag{23.23}$$

Here Eq. (23.20) is due to convexity of the loss function with respect to $\mathbf{w}$. Note that random variables $(\mathbf{x},y)\sim D$ are replaced with identically distributed $(\mathbf{x}_t,y_t)\sim D$ in (23.21). Equation (23.22) uses definition of $\text{Regret}_T(\text{Alg})$ as given by (23.1).

■

References

[1] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.