Statistical Techniques in Robotics (16-831, S22) Lecture #09 (Wednesday, Febuary 16)

Online Gradient Descent & Norm. Exp. Gradient Descent

Lecturer: Kris Kitani *Scribes: Tzu-Hsuan Yang, Mukun Guo*

1 Review

Last lecture we studied Online Mirror Descent and Duality, together with some mathematical tools and analysis. We will review some of the core contents.

1.1 Online Mirror Descent

Online Mirror Descent provides us a framework that connects many online learning/optimization algorithms. It can be interpreted as a Follow the Regularized leader (FTRL) algorithms with linear loss with convex regularization term, and further gives us a unification of online learning algorithms and more mathematical tools for regret analysis.

In order to establish the connection between OMD and FTRL, we define the following notations.

1. $\boldsymbol{z}^{(1:t)} = \sum_{i=1}^{t} \boldsymbol{z}^{(i)}$ (sum of gradients of linear loss in FTRL-LinLoss)
2. $\boldsymbol{\theta} \triangleq -\boldsymbol{z}^{(1:t)}$ (parameter of the dual space, sum of gradients in the dual space)
3. $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \boldsymbol{z}^{(t)}$ (dual space parameter defined as incremental sum)
4. $\boldsymbol{w} = g(\boldsymbol{\theta})$ (mirror/linking function, connect dual space with primal space)

The OMD algorithm is formally written as:

Algorithm 1 Online Mirror Descent (Convex Set S, $g : \mathbb{R}^D \to S$)

for $t = 1, 2, \ldots, T$ **do**
  RECEIVE $(f^{(t)} : W \to \mathbb{R})$
  $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \mathbf{z}^{(t)}, \quad \mathbf{z} \in \partial f^{(t)}(\mathbf{w}^{(t)})$ ▷ Dual parameter update
  $\mathbf{w}^{(t+1)} = g(\boldsymbol{\theta}^{(t+1)})$ ▷ Mirror projection, Primal parameter update
end for

Note that different choices of regularizer lead to different g, and futher result in different algorithms.

1.2 Duality

1.2.1 Convex conjugate

The motivation for introducing duality is that the dual space provides another way to optimize primal space parameters. Specifically, we can execute optimization in the dual space and mirror

in the primal space. The concepts of primal and dual space can be intuitively understood as two ways to parameterize a function:

1. **Primal:** function / value parameterization: $\{\psi(w), w\}$
2. **Dual:** intercept / slope parameterization: $\{b(\theta), \theta\}$

Assuming we have a smooth convex function, the equation for a convex conjugate is:

$$\psi^*(\theta) = \max_{\mathbf{w}}(\langle \theta, w \rangle - \psi(w)) \tag{1}$$

The convex conjugate has some interesting and useful properties:

1. The derivative of the conjugate function w.r.t. θ is the optimal point on the primal function.

$$\nabla_\theta \psi^*(\theta) = \frac{\partial \psi^*(\theta)}{\partial \theta} = w^* \tag{2}$$

2. The derivative of the primal function is the slope/dual parameter

$$\nabla_w \psi(w) = \frac{\partial \psi(w)}{\partial w}|_{w=w^*} = \theta \tag{3}$$

   Note that Eq.(2) and Eq.(3) together form a nice inverse relationship that links the primal/dual space.

3. Fenchel-Young Inequality: The complex conjugate computed at θ is lower bounded by the line equation that is parameterized by θ

$$\psi^*(\theta) \geq (\langle w, \theta \rangle - \psi(w)) \tag{4}$$

1.2.2 Bregman Divergence

Bregman Divergence describes the error between a linear approximation and a convex function between two points u and w. It is defined as as the difference between the rise of the convex function and the linear approximation function:

$$\psi(w) - \psi(u) = \nabla \psi(u)^T (w - u) + D_\psi(w||u) \tag{5}$$

$$D_\psi(w||u) = \psi(w) - \psi(u) - \nabla \psi(u)^T (w - u) \tag{6}$$

1.3 OMD Regret Bound

With the above definition and mathematical tools, we can derive the regret bound for Online Gradient Descent.

$$R(\mathbf{u}) \leq \psi(\mathbf{u}) - \psi(\mathbf{w}^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(-\mathbf{z}^{(1:t)}|| - \mathbf{z}^{(1:t-1)}) \tag{7}$$

where ψ is the regularization function. Note that this is the general regret bound for any OMD algorithms. With different choices of regularization functions, we will have different algorithms and hence different regret bounds.

2 Summary

2.1 Gradient Descent

Gradient descent is an standard approach for minimizing differentiable convex functions. The Gradient Descent algorithm is formally written as Algorithm 2. To understand how the gradient descent works, there are three perspectives to explain the algorithm.

Algorithm 2 Gradient Descent

1: $\mathbf{w}^{(0)} \leftarrow 0$
2: **for** $t = 1, \cdots, T$ **do**
3: COMPUTE $(\nabla f(w^{(t-1)}))$
4: $w_n^{(t)} = w^{(t-1)} - \eta \nabla f(w^{(t-1)})$
5: **end for**

2.1.1 Perspective 1 : Geometric

An intuitive but not rigorous explanation is that moving in the direction opposite of the gradient and finally we will reach the local minima. Given that f is a function of convex hypothesis class $\boldsymbol{w}$: $f : \mathbb{R}^N \to \mathbb{R}$ at $\boldsymbol{w}$. The gradient $\nabla f(\boldsymbol{w})$ can be denoted as a vector of partial derivatives

$$\nabla f(\boldsymbol{w}) = \left\{ \frac{\partial f(\boldsymbol{w})}{w_1}, \cdots, \frac{\partial f(\boldsymbol{w})}{w_N} \right\} \tag{8}$$

If we want to find the **min** of f, move in the opposite direction of the gradient.

2.1.2 Perspective 2 : Linear Approximation with regularization

From the previous geometric point of view, it is hard to come up the update rule. So in this perspective, we want to take gradient descent from the view of linear approximation. When f is convex, we can use first order Taylor series approximation to find the lower bound of f at u as

$$f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle \tag{9}$$

If we minimize the approximation directly, as the equation shown below, we will get a solution in at negative infinity.

$$\min_u \left\{ f(w) + (u - w)^\top \nabla f(w) \right\} \tag{10}$$

Therefore, we can introduce a squared L2 norm constraint

$$\min_w \left\| \boldsymbol{w} - \boldsymbol{w}^{(t)} \right\|_2^2 \tag{11}$$

to express the closeness to w and the final objective function becomes a linear loss with quadratic regularization

$$w^{(t+1)} = \arg\min_w \frac{1}{2} \left\| w - w^{(t)} \right\|^2 + \eta \left(f\left(w^{(t)}\right) + \left\langle w - w^{(t)}, \nabla f\left(w^{(t)}\right) \right\rangle \right). \tag{12}$$

To get the optimal w, taking the partial derivative of objective function wrt w and set to 0. then we can get the Gradient Descent update rule.

$$\frac{\partial \frac{1}{2}\left\|\mathbf{w}-\mathbf{w}^{(t)}\right\|_2^2+\eta\left(f\left(\mathbf{w}^{(t)}\right)+\left\langle\left(\mathbf{w}-\mathbf{w}^{(t)}\right), \nabla f\left(\mathbf{w}^{(t)}\right)\right\rangle\right)}{\partial \mathbf{w}}=0 \tag{13}$$

$$\left(\mathbf{w}-\mathbf{w}^{(t)}\right)+\eta\left(0+\nabla f\left(\mathbf{w}^{(t)}\right)=0\right. \tag{14}$$

$$\mathbf{w}^{(t+1)}=\mathbf{w}^{(t)}-\eta \nabla f\left(\mathbf{w}^{(t)}\right) \tag{15}$$

2.1.3 Perspective 3 : Isometic Quadratic Approximation

Similar as previous perspective, with Taylor series approximation, we can then use second order approximation to yield

$$f(\boldsymbol{u}) \approx f(\boldsymbol{w})+(\boldsymbol{u}-\boldsymbol{w})^{\top} \nabla f(\boldsymbol{w})+\frac{1}{2}(\boldsymbol{u}-\boldsymbol{w})^{\top} \nabla^2 f(\boldsymbol{w})(\boldsymbol{u}-\boldsymbol{w}) \tag{16}$$

Since the $\nabla^2 f(\boldsymbol{w})$ may be hard to compute, we then use Isometric quadratic approximation(unit scaling in all dimension/directions) to rewrite our approximation as

$$f(\boldsymbol{u}) \approx f(w)+(u-w)^{\top} \nabla f(w)+\frac{1}{2 \eta}(u-w)^{\top} \mathbf{I}(u-w) \tag{17}$$

where η is a tunable scaling parameter. Then the objective function could be written as

$$\arg \min _w \frac{1}{2}\|u-w\|^2+\eta\left(f(w)+(u-w)^T \nabla f(w)\right) \tag{18}$$

By replacing the u with w and w with $w^{(t)}$ we can then obtain the objective function written as

$$w^{(t+1)}=\arg \min _w \frac{1}{2}\left\|w-w^{(t)}\right\|^2+\eta\left(f\left(w^{(t)}\right)+\left\langle w-w^{(t)}, \nabla f\left(w^{(t)}\right)\right\rangle\right). \tag{19}$$

which is the same equation shown in perspective 2. Therefore, we will get the gradient descent update rule by doing the same steps in perspective 2.

$$\mathbf{w}^{(t+1)}=\mathbf{w}^{(t)}-\eta \nabla f\left(\mathbf{w}^{(t)}\right) \tag{20}$$

We have seen this update rule may times such as weighted majority, online perceptron, and FTRL. In conclusion, this update rule comes from **a quadratic approximation** of the loss function f

2.2 Stochastic Gradient Descent

Sometimes, the update rule in the gradient descent is expensive to compute because you have to run through ALL the samples in your data set to do a single update for a parameter in a particular iteration. Therefore, to speed up computation, we can use stochastic gradient descent.The algorithm of stochastic gradient descent is formally written as:

Algorithm 3 Stochastic Gradient Descent

1: $\mathbf{w}^{(0)} \leftarrow 0$
2: $\eta > 0$
3: **for** $t = 1, \cdots, T$ **do**
4: $\quad z \sim D$ ▷ sample from the data distribution(single sample or mini-batch)
5: $\quad v^{(t)} = \nabla f_z(w^{(t-1)})$ ▷ fast to compute(expectation is the true gradient)
6: $\quad w^{(t)} = w^{(t-1)} - \eta v^{(t)}$
7: **end for**

We can observe from the algorithm that instead of running through all the samples points, stochastic gradient sampled a single or mini-bathed from the data(line 4) to calculate the gradient. By doing so, this random approximation of the data set removes the computational burden associated with gradient descent while achieving iteration faster and at a lower convergence rate.

2.3 Online (Projected Sub-) Gradient Descent (OGD) as Online Mirror Descent (OMD)

In the lecture, we learned that OGD is OMD with a linear loss and quadratic regularization. To proof this, we start with defining the regularization function as

$$\psi(\boldsymbol{w}) = \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2 \tag{21}$$

and the loss function as

$$f(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle \tag{22}$$

Then the prediction rule derives from the mirror function becomes

$$w = \arg\min_{\boldsymbol{w}} \left\langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \right\rangle + \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2 \tag{23}$$

and $\frac{1}{2\eta}\|\boldsymbol{w}\|_2^2$ can be written as $\frac{1}{2\eta}\sum_n w_n^2$ So, let $\mathcal{L} = \langle w, -\theta \rangle + \frac{1}{2\eta}\sum_n w_n^2$, we can obtain the optimal parameter by calculating the gradient.

$$\frac{\partial \mathcal{L}}{\partial w_n} = -\theta_n + \frac{1}{2\eta} 2 w_n = 0 \tag{24}$$

$$w_n = \eta \theta_n \tag{25}$$

which is a projection of the parameter in the dual space to a parameter in the primal space. Hence this is a mirror function and we can define the mirror function for OGD as

$$g(\theta) = \eta\theta \tag{26}$$

There are two versions of OGD mirror function. One of them assumes constrained range of sub-gradients with no projection to the weight update rule called online sub-gradient descent, whose algorithm can be found in Algorithm 4. The other one updates the weight with some explicit

condition on the feasibility set via projection called online project sub-gradient descent. As shown in Algorithm 5.

Algorithm 4 Online Sub-Gradient Descent

for $t = 1, \cdots, T$ **do**
  $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}\left(\boldsymbol{w}^{(t)}\right)$ ▷ Dual parameter update
  $w^{(t+1)} = \eta\boldsymbol{\theta}^{(t+1)}$ ▷ Mirror projection
end for

Algorithm 5 Online Proj Sub-Gradient Descent

for $t = 1, \cdots, T$ **do**
  $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}\left(\boldsymbol{w}^{(t)}\right)$ ▷ Dual parameter update
  $w^{(t+1)} = \prod_{\theta \to S} \eta\theta^{(t+1)}$ ▷ Mirror projection
end for

2.4 Analysis of Online Gradient Descent

To discuss the regret bound of OGD, we start from recalling the general OMD regret bound

$$R(\boldsymbol{u}) \leq \psi(\boldsymbol{u}) - \psi\left(\boldsymbol{w}^{(1)}\right) + \sum_{t=1}^{T} D_{\psi^*}\left(-\boldsymbol{\theta}^{(t+1)} \| - \boldsymbol{\theta}^{(t)}\right) \tag{27}$$

With Bregman divergence, the RHS of OMD regret will become

$$\psi(\boldsymbol{u}) - \psi\left(\boldsymbol{w}^{(1)}\right) + \sum_{t=1}^{T} \psi^*\left(\boldsymbol{\theta}^{(t+1)}\right) - \psi^*\left(\boldsymbol{\theta}^{(t)}\right) - \nabla\psi^*\left(\boldsymbol{\theta}^{(t)}\right)\left(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\right) \tag{28}$$

For the L2 norm, the convex conjugate is also L2 norm, then we can rewritten equation 28 as

$$\frac{1}{2\eta}\|u\|_2^2 - \frac{1}{2\eta}\left\|w^{(1)}\right\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}\left\|\theta^{(t+1)}\right\|_2^2 - \frac{1}{2\eta}\left\|\theta^{(t)}\right\|_2^2 - \nabla\frac{1}{2\eta}\left\|\theta^{(t)}\right\|_2^2\left(\theta^{(t+1)} - \theta^{(t)}\right) \tag{29}$$

since we know that $\theta^{(t+1)} - \theta^{(t)}$ is constant, we can then compute the gradient to get

$$\frac{1}{2\eta}\|u\|_2^2 - \frac{1}{2\eta}\left\|w^{(1)}\right\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}\left\|\theta^{(t+1)}\right\|_2^2 - \frac{1}{2\eta}\left\|\theta^{(t)}\right\|_2^2 - \frac{1}{\eta}\theta^{(t)}\left(\theta^{(t+1)} - \theta^{(t)}\right) \tag{30}$$

by rearranging the summation term and plug in definition of dual parameter(equation 32 $\rightarrow$ equation 33) and removing the always positive term(equation 33 $\rightarrow$ equation 34) , we can get

$$\frac{1}{2\eta}\|u\|_2^2 - \frac{1}{2\eta}\left\|w^{(1)}\right\|_2^2 + \sum_{t=1}^{T}\frac{1}{2\eta}\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2^2 \tag{31}$$

$$= \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\left\|\boldsymbol{w}^{(1)}\right\|_2^2 + \sum_{t=1}^{T}\frac{1}{2\eta}\left\|\boldsymbol{\theta}^{(t)} - \eta\boldsymbol{z}^{(t)} - \boldsymbol{\theta}^{(t)}\right\|_2^2 \tag{32}$$

$$= \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\left\|\boldsymbol{w}^{(1)}\right\|_2^2 + \sum_{t=1}^{T}\frac{1}{2\eta}\left\|-\eta\boldsymbol{z}^{(t)}\right\|_2^2 \tag{33}$$

$$\leq \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 + \sum_{t=1}^{T}\frac{\eta}{2}\left\|z^{(t)}\right\|_2^2 \tag{34}$$

Define $D = \max \|u\|_2, \boldsymbol{u} \in S$ and $G = \max \|\boldsymbol{z}\|_2, \boldsymbol{z} \in \partial f(\boldsymbol{w})$

$$R_{OGD}(\boldsymbol{u}) \leq \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 + \sum_{t=1}^{T}\frac{\eta}{2}\left\|\boldsymbol{z}^{(t)}\right\|_2^2 \tag{35}$$

$$\leq \frac{D^2}{2\eta} + \frac{\eta}{2}TG^2 \tag{36}$$

To find optimal step size η

$$\frac{d}{d\eta}\left\{\frac{1}{2\eta}D^2 + \frac{\eta}{2}G^2T\right\} = 0 \tag{37}$$

$$\eta = \frac{D}{G\sqrt{T}} \tag{38}$$

By Replacing η with $\frac{D}{G\sqrt{T}}$, we get the **Regret Bound of Online Gradient Descent**

$$R_{OGD} \leq DG\sqrt{T} \tag{39}$$

2.5 Online Normalized Exponentiated Gradient Descent

The algorithm for **Online Normalized Exponentiated Gradient Descent** (ONEGD) is defined as:

Algorithm 6 Norm-Exponentiated-Gradient (η)

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \boldsymbol{z} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
3: $\quad \boldsymbol{w}^{(t+1)} \propto \boldsymbol{w}^{(t)} \exp\left(\eta \boldsymbol{z}^{(t)}\right)$ ▷ Mirror Projection
4: **end for**

We will now show how this algorithm can be derived from OMD with a specific regularization term.

Recall that in OMD, different choices of g (i.e. mirror function) lead to different algorithms. If we define the regularization function as negative entropy, that is:

$$\psi(\boldsymbol{w}) = \sum_{k=1}^{K} w_k \log w_k \quad \boldsymbol{w} \in \mathbb{S}^K \tag{40}$$

where $\mathbb{S}^K$ is the K-simplex constraint (further discussed in Appendix), and keep the loss function linear:

$$f(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle \tag{41}$$

Then our prediction rule becomes:

$$\begin{aligned} \boldsymbol{w}^{(t+1)} &= \operatorname*{argmin}_{\boldsymbol{w}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \psi(\boldsymbol{w}) \\ &= \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{S}^K} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k \end{aligned} \tag{42}$$

Add the simplex constraint:

$$\boldsymbol{w}^{(t+1)} = \operatorname*{argmin}_{\boldsymbol{w}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k + \lambda \left(1 - \sum_k \boldsymbol{w}_k \right) \tag{43}$$

If we view λ as the Lagrangian multiplier, then we can define the objective function as:

$$\mathcal{L} = \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \frac{1}{\eta} \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k + \lambda \left(1 - \sum_k \boldsymbol{w}_k \right) \tag{44}$$

Now, in order to solve the original optimization problem, we would like to find the minimum of the Lagrangian equation. We will take the partial derivative w.r.t. w_n, note that above is only the partial derivative w.r.t one element inside $\boldsymbol{w}$:

$$\frac{\partial \mathcal{L}}{\partial w_n} = -\theta_n + \frac{1}{\eta}(1 + \log w_n) - \lambda \tag{45}$$

Set the partial derivative to 0:

$$\begin{aligned} 0 &= -\theta_n + \frac{1}{\eta} + \frac{1}{\eta} \log w_n - \lambda \\ \frac{1}{\eta} \log w_n &= \theta_n - \frac{1}{\eta} + \lambda \\ \log w_n &= \eta\theta_n - 1 + \eta\lambda \\ &= \eta\theta_n - (1 - \eta\lambda) \\ w_n &= \exp(\eta\theta_n - (1 - \eta\lambda)) \\ &= \frac{\exp(\eta\theta_n)}{\exp(1 - \eta\lambda)} \end{aligned} \tag{46}$$

where w_n is the minimizer for the objective function, thus the minimizer for linear loss and entropic regularization. From the above equation, since the mirror function enforces the geometry of the

problem (e.g. probability simplex) we could derive the mirror function that links dual space with primary space, which is:

$$g(\boldsymbol{\theta}) = \frac{\exp(\eta\boldsymbol{\theta})}{\sum_{n'} \exp(\eta\theta_{n'})} \tag{47}$$

And the dual parameter update is the same for all OMD algorithms:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}(\boldsymbol{w}^{(t)}) \tag{48}$$

Now let's make the mirror projection clearer. From Eq. 47 we have:

$$\begin{aligned}
w_n^{(t+1)} =& \frac{\exp\left(\eta\theta_n^{(t+1)}\right)}{\sum_{n'} \exp(\eta\theta_{n'}^{(t+1)})} \\
=& \frac{\exp\left(\eta(\theta_n^{(t)} - z_n^{(t)})\right)}{\sum_k \exp\left(\eta(\theta_k^{(t)} - z_k^{(t)})\right)} && \text{(Recall the definition of } \boldsymbol{\theta} \text{ as incremental sum)} \\
=& \frac{\exp(\eta\theta_n^{(t)})\exp(-\eta z_n^{(t)})}{\sum_k \exp(\eta\theta_k^{(t)})\exp(-\eta z_k^{(t)})} \cdot \frac{\sum_j \exp(\eta\theta_j^{(t)})}{\sum_j \exp(\eta\theta_j^{(t)})}
\end{aligned} \tag{49}$$

By the definition of the mirror function (Eq. 47):

$$w_n^{(t)} = \frac{\exp(\eta\theta_n^{(t)})}{\sum_j \exp(\eta\theta_j^{(t)})} \tag{50}$$

$$\sum_k w_k^{(t)} \exp(-\eta z_k^{(t)}) = \sum_k \frac{\exp(\eta\theta_n^{(t)}) \cdot \exp(-\eta z_k^{(t)})}{\sum_j \exp(\eta\theta_j^{(t)}} \tag{51}$$

Plug Eq. 50 and Eq. 51 into Eq. 49, we get:

$$w_n^{(t+1)} = \frac{w_n^{(t)} \exp(-\eta z_n^{(t)})}{\sum_k w_k^{(t)} \exp(-\eta z_k^{(t)})} \tag{52}$$

$$\downarrow$$

$$w_n^{(t+1)} \propto w_n^{(t)} \exp(-\eta z_n^{(t)}) \tag{53}$$

Lastly, it should be easy to see that Hedge algorithm (Algorithm 7) is an unnormalized exponentiated gradient descent algorithm since it has a linear loss function and exponential weights update.

Algorithm 7 Hedge Algorithm (β)

$\mathbf{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^N$
for $t = 1, \cdot; T$ **do**
  RECEIVE $(\boldsymbol{x}^(t) \in \{-1, 1\})$
  $i \sim \text{MULTINOMIAL}(\boldsymbol{w}^{(t)}/\Phi^{(t)})$
  $\hat{y} = h_i(\boldsymbol{x}^{(t)})$ ▷ Expected 0-1 loss is linear
  RECEIVE $y^t \in \{-1, 1\}$
  $w_n^{(t+1)} = w_n^{(t)} e^{-\beta(\mathbf{1}[y^{(t)} \neq h_n(\boldsymbol{x}^{(t)})])}$ ▷ Exponential update
end for

References

[1] F. Orabona. *A Modern Introduction to Online Learning.* 2021.

[2] Wikipedia contributors. Simplex — Wikipedia, the free encyclopedia, 2022.

3 Appendix

3.1 Comparison between OGD and ONEGD

We will conduct a simple comparison between **Online Gradient Descent** and **Online Exponentiated Gradient Descent**. Note that this comparison was shown in Lecture 7, but it is a good thing to refresh our memory here.

	Online Gradient Descent	**Online Exponentiated Gradient Descent**
Example	Perceptron	Hedge/GWM, Winnow
Predictor	linear: $\boldsymbol{w} \cdot \boldsymbol{x}^{(t)}$	linear: $\boldsymbol{w} \cdot \boldsymbol{x}^{(t)}$
Regularization	quadratic regularization: $\frac{1}{\eta}\|\|\boldsymbol{w}\|\|_2^2$	entropic regularization: $\frac{1}{\eta}\sum_n w_n \log(w_n)$
Update	additive update: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \nabla l(y^{(t)}, \hat{y}^{(t)})$	exponential update: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} \exp(-\eta \nabla l(y^{(t)}, \hat{y}^{(t)}))$
Regret Bound	no regret: $BL\sqrt{2T}$	no regret: $\sqrt{(T/2)\log N}$

Note that though both Gradient Descent and Exponentiated Gradient Descent are no-regret algorithm, Exponentiated Gradient Descent has a better bound theoretically.

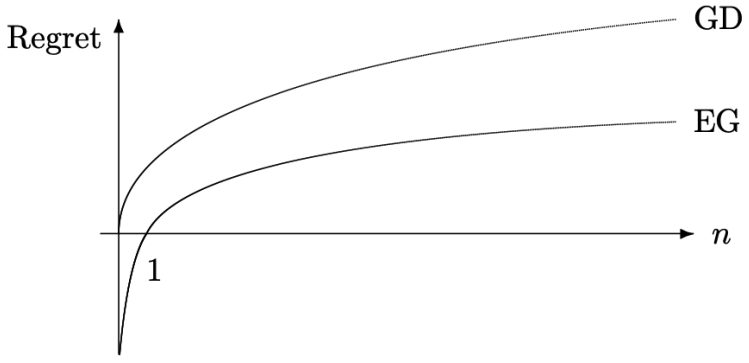

Figure 1: Regret Bound Comparison with n experts

3.2 K-Simplex and Simplex Constraint[2]

In geometry, a simplex is a generalization of a triangle to arbitrary dimensions. For example,

1. 0-simplex is a *point*
2. 1-simplex is a *line segment*
3. 2-simplex is a *triangle*
4. 3-simplex is a *tetrahedron*

Formally, a **k-simplex** is a defined as:

$$C = \left\{ \theta_0 u_0 + \cdots + \theta_k u_k \mid \sum_{i=1}^{k} \theta_i = 1 \text{ and } \theta_i \geq 0 \text{ for } i = 0, \cdots, k \right\} \tag{54}$$

And the **probability simplex** or **standard simplex** is a simplex whose vertices are the k standard unit vectors:

$$\left\{ x \in \mathbb{R}^k : x_0 + \cdots + x_{k-1} = 1, x_i \geq 0 \text{ for } i = 0, \ldots, k \right\} \tag{55}$$