Disclaimer: These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.

Logarithmic regret for strongly convex functions

We next revisit the OGD algorithm for special cases of convex functions. Namely, we consider the OCO setting when the functions to be observed are strongly convex.

**Definition 19.1.** A convex function $f$ is said to be $\alpha$-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\alpha}{2} \| y - x \|^2$$ (19.1)

### 19.0.1 OGD for strongly convex functions

We next analyse the OGD algorithm for strongly convex functions.

**Theorem 19.2.** For $\alpha$-strongly convex functions (and $G$-Lipschitz), OGD with step size $\eta_t = \frac{1}{\alpha t}$ achieves the following guarantee for all $T \geq 1$

$$\text{Regret}_T \leq \frac{G^2}{2\alpha} (1 + \log T)$$

**Proof.** Define $\nabla_t = \nabla f(x_t)$ and let $x^* = \arg\min \sum_{t=1}^T f(x^*)$. Applying the definition of strong convexity to the pair of points $x^*, x_t$ we have that

$$2f(x_t) - f(x^*) \geq 2\nabla f(x_t)^T (x^* - x) + \alpha \| x^* - x \|^2$$ (19.2)

Using the update rule we have that

$$\| x_{t+1} - x^* \|^2 = \| \Pi_K(x_t - \eta_t \nabla_t) - x^* \|^2 \leq \| x_t - \eta_t \nabla_t - x^* \|^2$$

Hence,

$$\| x_{t+1} - x^* \|^2 - \| x_t - x^* \|^2 \leq \eta_t^2 \| \nabla_t \|^2 - 2\eta_t \nabla_t \cdot (x_t - x^*)$$

and,

$$2\nabla_t \cdot (x_t - x^*) \leq \frac{\| x_t - x^* \|^2 - \| x_{t+1} - x^* \|^2}{\eta_t} + \eta_t G^2$$ (19.3)

Considering Eq. 19.2, Eq. 19.3 and summing up while considering $\eta_t = \frac{1}{\alpha t}$ (define $\frac{1}{\eta_0} = 0$):

$$2 \sum_{t=1}^T f_t(x_t) - f_t(x^*)$$

$$\leq \sum_{t=1}^T \| x_t - x^* \|^2 \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \alpha \right) + G^2 \sum_{t=1}^T \eta_t$$

$$= 0 + G^2 \sum_{t=1}^T \frac{1}{\alpha t} \leq \frac{G^2}{\alpha} (1 + \log T)$$
19.0.2 Learning Regularized Objectives

We have so far applied SGD algorithm to optimize convex problems of the form
\[
\minimize \sum f_t(w^*) \\
\text{s.t. } \|w^*\| \leq B
\]

We’ve discussed that sometimes it is more convenient to consider unconstrained regularized problems of the form
\[
\minimize \lambda \frac{1}{2} \|w\|^2 + \frac{1}{T} \sum_{t=1}^T f_t(w)
\]

The following claim is left as an exercise:

**Claim 19.3.** For any convex function \( f_t \): the function \( \frac{1}{2} \|w\|^2 + f_t(w) \) is \( \lambda \)-strongly convex

**Theorem 19.4.** Apply OGD without projection (or with \( B = \infty \)), and set \( \eta_t = \frac{1}{T\lambda} \) on a sequence of functions \( \{ \frac{1}{2} \|w\|^2 + f_t(w) \}_{t=1}^T \) and then for any \( \|w^*\| \leq B_0 \) we have that:

\[
\sum_{t=1}^T f_t(w_t) - f_t(w^*) \leq \frac{G^2 \log T}{T\lambda} + \lambda B_0
\]

**Proof.** First we bound the Lipschitness of the functions \( \frac{1}{2} \|w\|^2 + f_t(w) \): Note that \( \nabla f_t = \lambda w_t + \nabla f_t(w_t) \), hence

\[
w_{t+1} = w_t - \frac{1}{\lambda} \nabla f_t
\]

\[
= (1 - \frac{1}{T})w_t + \frac{1}{T} \left( \frac{1}{\lambda} \nabla f_t(w_t) \right)
\]

Hence we have that:

\[
\|w_{t+1}\| = \| \frac{1}{T} \sum \frac{1}{\lambda} \nabla f_t(w_t) \| \leq \frac{G}{\lambda}
\]

We thus have that
\[
\|\nabla f_t\| \leq 2G
\]

Thus, the result follows from applying the regret bound for strongly convex functions we get that for any \( w^* \):

\[
\sum_{t=1}^T \frac{1}{2} \|w_t\|^2 + f_t(w_t) - \frac{1}{2} \|w^*\| + f_t(w^*) \leq \frac{4G^2 \log T}{2\lambda}
\]

Hence
\[
\frac{1}{T} \sum_{t=1}^T f_t(w_t) - f_t(w^*) \leq \frac{2G^2 \log T}{T\lambda} + \frac{1}{2} \lambda B^2
\]

Applying the Online to Batch theorem we can thus obtain an algorithm for learning regularized objective. Note that if we want to achieve \( \frac{1}{\sqrt{T}} \) error, we would need to set \( \lambda = O(\frac{1}{\sqrt{T}}) \) which would still have the same overall learning rate, but this algorithm avoids making projection steps.
19.1 Second Order Methods

19.1.1 Motivation – Online Portfolio Selection

To motivate the construction of second order method, we return to the problem of online portfolio selection. In online portfolio selection, at each iteration the learner chooses to distribute her wealth amongst $n$ stocks. Similarly to the online expert model her decision is a vector $x_t \in \Delta_n$. Nature then chooses a vector $\|r_t\|_{\infty} \leq 1$, where $r_t(i) =$ price of stock $i$ at time $t + 1$. For exposition, will assume for simplicity that $\min_i r_t(i) \geq \frac{1}{T}$ (i.e. no stock ever loses its whole value). The revenue of the learner is measured in terms of $\log x_t \cdot r_t$. The overall gain of the learner is then

$$\sum \log x_t \cdot r_t = \log \prod x_t \cdot r_t = \log \frac{\text{Wealth at time } T}{\text{Wealth at time } 1}.$$  

The regret is measured against the best constant rebalance portfolio (such a portfolio strategy is called universal)

$$\text{Regret}_T = \max_{x^* \in \Delta_n} \sum \log(x^* \cdot r_t) - \sum \log(x_t \cdot r_t)$$

Note that we have a maximization problem and not a minimization problem, yet log is a concave and not convex problem therefore the problem is equivalent to

$$\text{Regret}_T = \sum -\log(x_t \cdot r_t) - \min_{x^* \in \Delta_n} \sum -\log(x^* \cdot r_t)$$

As before we can apply OGD to obtain $O(\sqrt{T})$ regret (Note that our assumption that $r_t \geq \frac{1}{T^2}$ implies that the loss vectors are always Lipschitz). However, the log function is in fact, an $exp-concave$ function which allows us to improve on these results using second order methods.

19.1.2 Second Order Methods

The idea behind second order methods is that if gradient descent linearizes the function and chooses a step according to the first order approximation, a second order method considers always the Hessian of the function (i.e. its second derivative). For example the Hessian of the log($x_t \cdot r_t$) function is given by

$$\nabla^2 f_t(x) = \frac{r_t \cdot r_t^\top}{(r_t \cdot x)^2},$$

and it is the derivative of the gradient function. The crucial property that we wish to exploit, is that the Hessian of the function is large in the direction of the gradient:

**Definition 19.5.** A convex function $f$ is called $\alpha - exp$-concave over $K$ if the function $g$ is concave, where $g$ is defined as

$$g(x) = e^{-\alpha f(x)}.$$  

The following Lemma is left as an exercise:

**Lemma 19.6.** A twice-differentiable function $f$ is $\alpha - exp$ concave if and only if

$$\nabla^2 f > \alpha \nabla f(x) \nabla f(x)^\top$$

In other words $\nabla^2 f - \alpha \nabla f(x) \nabla f(x)^\top$ is a p.s.d matrix.
19.1.3 Online Newton Step

We next describe the online newton step algorithm. For that we add one more notation, we will denote by $\Pi^A_K$ the projection over the set $K$, with respect to the metric $A$: i.e.

$$\Pi^A_K(y) = \arg\min_{x \in K} (x - y)^{\top} A (x - y)$$

**Algorithm 1 Online Newton Step ONS**

**Initialization** $x_1 \in K$, parameters $\gamma, \epsilon > 0$, $A_0 = \epsilon \text{Id}$.

for $t = 1, 2 \ldots T$ do

- Play $x_t$ and observe cost $f_t(x_t)$
- Rank 1 update $A_t = A_{t-1} + \nabla_t \nabla_t^\top$
- Newton step and projection:

$$y_{t+1} = x_t - \frac{1}{\gamma} A_t^{-1} \nabla_t$$

$$x_{t+1} = \Pi^A_K(y_{t+1})$$

end for

return

**Theorem 19.7.** Alg. 1 with parameters $\gamma = \min\left\{ \frac{1}{4GD}, \alpha \right\}$ and $\epsilon = \frac{1}{\gamma^2 D^2}$, guarantees (for $T > 4$):

$$\text{Regret}_T \leq 5\left(\frac{1}{\alpha} + GD\right)n \log T$$