Lecture Notes Online Learning, Part 2

Online Convex Optimization

March the 15th, 2018

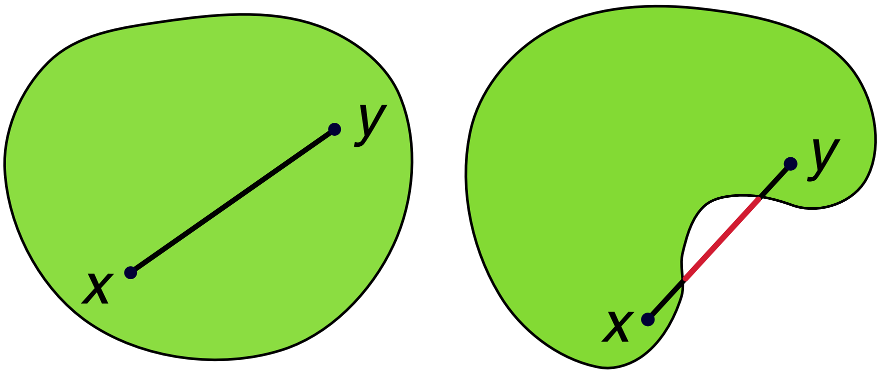
1 Convex Sets and convex Functions

We start with a short reminder of some basic properties of convex sets and convex functions.

1.1 Convex Sets

Convex sets have the property that if we take two points in them and form the intersecting line between them, then all points on that line are again in this set as illustrated in Figure 1. The formal definition is the following.

Definition 1 *A set $A \subset \mathbb{R}^d$ is called convex if for every $a, b \in A$ and $\lambda \in [0,1]$ the point $\lambda a + (1-\lambda)b$ is also in A.*



(a) Convex Set (b) Non-Convex Set

Figure 1: [1]Illustration of a convex and a non-convex subset of $\mathbb{R}^2$

For the online gradient descent algorithm introduced in this lecture we will need the following notation. Let $A \subset \mathbb{R}^d$ be a closed and convex set, and $w \in \mathbb{R}^d$. We want to be able to project the point w onto the convex set A. We do that by replacing w with the closest point in the convex set A. Formally we introduce the projection as follows.

Definition 2 *The projection $\Pi_A(w)$ of an point $w \in \mathbb{R}^d$ onto a closed, convex set $A \subset \mathbb{R}^d$ is defined as*

$$\Pi_A(w) = \arg\min_{a \in A} ||w - a||_2$$

Where $||.||_2$ is the euclidean norm in $\mathbb{R}^d$.

[1]pictures by CheCheDaWaff, distributed under a CC-BY 4.0 license.

The reason we use convex sets for our algorithm is a property known as Pythagoras theorem. It basically states that the distance from any point in a convex set is smaller to the projected point in comparison to the not projected point.

Theorem 1 ***(Pythagoras)***
Let $A \subset \mathbb{R}^d$ be a closed, convex set. Then for all points $a \in A$ and $w \in \mathbb{R}^d$ the following inequality holds.

$$||\Pi_A(w) - a||_2 \leq ||w - a||_2$$

1.2 Convex Functions

If a function $l : \mathbb{R} \to \mathbb{R}$ is differentiable then it is convex if the line defined by the gradient in each point is below the function. Formally this function is convex if the following inequality holds.

$$l(b) \geq l(a) + \frac{\partial}{\partial a} l(a)(b - a) \tag{1}$$

If we fix a and keep b variable then we can think of the right hand side of this inequality as the line segment through $l(a)$ and with a slope defined by the gradient. This intuition is captured in Figure 2.

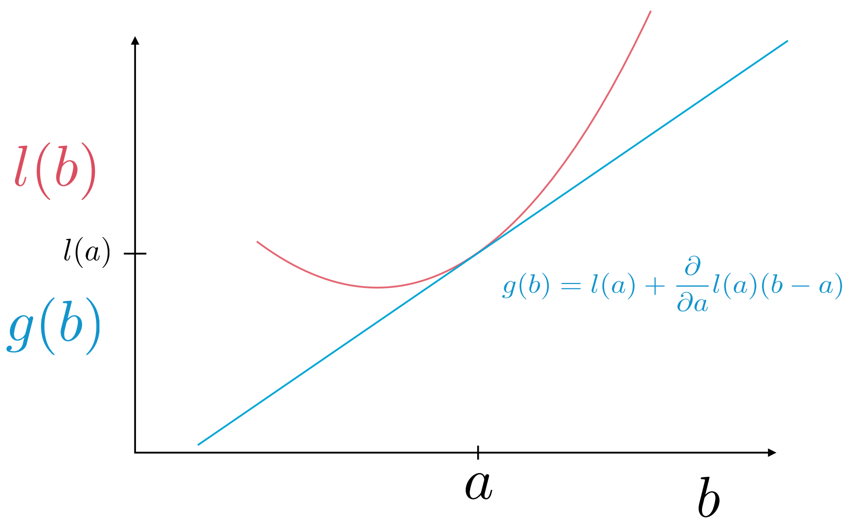


Figure 2: The red graph illustrates a convex and differentiable one dimensional function. The blue graph depicts the gradient in point a and we see that it is always below the convex function.

This idea can be generalized for functions in multiple variables $l : \mathbb{R}^d \to \mathbb{R}$. We then change the derivative $\frac{\partial}{\partial a} l(a) : \mathbb{R} \to \mathbb{R}$ for the gradient $\nabla l(a) : \mathbb{R}^d \to \mathbb{R}^d$. Generalizing to multiple dimensions and reordering (1) we get the following property of convex, differentiable functions.

Lemma 1 *If $l : \mathbb{R}^d \to \mathbb{R}$ is a convex and differentiable function then the following inequality holds.*

$$l(a) - l(b) \leq \nabla^T l(a)(a - b) \tag{2}$$

Here $\nabla^T l(b)$ denotes the transpose of the gradient $\nabla l(b)$.

2 The setup

Similar to the expert setting we play an ongoing game against an adversary. In each time step we take an action, then the adversary or nature takes an action, and then we observe a loss based on our and the adversaries action. More precisely we assumed that we are given d experts and we can pick at each time step t a distribution over the d experts. In other words; We took in each time step an action in the space Δ_d. The first difference in online convex optimization is that we generalize this idea, and at time t we take an action a_t in a closed, convex set $\mathcal{A} \subset \mathbb{R}^d$. We thus play the following game.

- At time t we pick an action $a_t \in \mathcal{A}$
- An adversary picks an action $z_t \in \mathcal{Z}$
- We observe the loss $l(a_t, z_t)$

We ask for the following properties of the loss function $l : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$: Fix $z \in \mathcal{Z}$ and define a new function $l^z : \mathcal{A} \to \mathbb{R}$ by $l^z(a) = l(a, z)$. We assume that for all z the function l^z is convex and differentiable. If we say that l is convex or differentiable, we mean that it has this property with respect to the variable a, i.e. l^z has this property for all $z \in \mathcal{Z}$. The next difference in comparison to the expert setting is the regret. While in the expert regret we compared to the best expert, we now compare to the best constant action.

Definition 3 *Let $\mathcal{A}$ be a convex and closed action space and $l : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ be a convex and differentiable loss function. The* ***OCO regret*** *after n rounds for a given player strategy $a_t \in \mathcal{A}$ and adversary strategy $z_t \mathcal{Z}$ with $1 \leq t \leq n$ is given by*

$$R_n = \sum_{t=1}^{n} l(a_t, z_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} l(a, z_t).$$

3 Online Gradient Descent

As in the expert setting, our goal is now to achieve a bound for the regret R_n that grows slower than linear in n. The way to achieve this is a very simple algorithm called Online Gradient Descent (OGD). The method of gradient descent is a well known technique in convex optimization for finding extremum points. The basic idea is to start with an initial guess for an extremum and then follow the direction of the negative gradient as illustrated in Figure 3. The gradient will provide us the right direction to walk into, but we have to decide how far we go that direction. This will depend on a parameter $\eta \in \mathbb{R}$ set by us, the learning rate. Note that the actions we are allowed to take are defined by the action space $\mathcal{A}$. When we use gradient descent it can happen that we end up with a new solution that is not in $\mathcal{A}$ anymore. To solve this problem we project this new solution then back to $\mathcal{A}$. So formally OGD with learning rate η iterates the following two steps.

Definition 4 *Fix a convex, closed action space $\mathcal{A}$, a convex, differentiable loss function $l : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ and an initial action $a_1 \in \mathcal{A}$. The strategy given by* ***Online Gradient Descent*** *with learning rate η is defined by the following two iterating steps.*

- *Set $w_{t+1} = a_t - \eta \nabla_a l(a_t, z_t)$*
- *Set $a_{t+1} = \Pi_A(w_{t+1})$*

The term $\Pi_A(w)$ here is the projection of w onto the set A.

This strategy achieves the following regret bound.

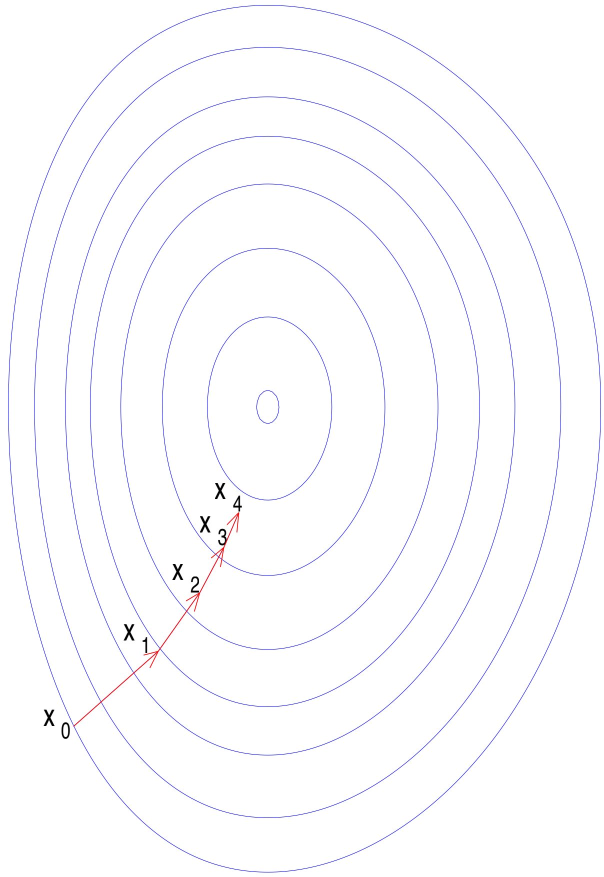


Figure 3: This picture shows the level sets of a three dimensional graph, so basically the projection onto the x-y plane. The values of the function decrease with each circle going closer to the center. The minimum is thus found in the little most inner circle. The sequence from x_0 to x_5 depicts the behavior of the gradient descent algorithm.

Theorem 2 *Assume the setting from the previous definition. Assume further that* $||a||_2 \leq R$ *for all* $a \in \mathcal{A}$ *and* $||\nabla l(a, z)||_2 \leq G$ *for all* $(a, z) \in \mathcal{A} \times \mathcal{Z}$. *The strategy given by the Online Gradient Descent with learning rate* η *and* $a_1 = 0$ *achieves a bound of the regret given by*

$$R_n \leq \frac{R^2}{2\eta} + \frac{\eta G n}{2}. \tag{3}$$

For a learning rate $\eta = \frac{R}{G\sqrt{n}}$ *we can thus bound*

$$R_n \leq RG\sqrt{n} \tag{4}$$

The proof of this Theorem is found in [Bub11].

References

[Bub11] Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.