Economics, AI, and Optimization
Lecture Note 4: Online Convex Optimization and Sion's Minimax Theorem

Christian Kroer*

February 1, 2022

1 Recap

Last time we learned about general-sum games, dominant-strategy solutions, Nash equilibrium, and the special case of zero-sum games. At the end of the lecture we learned the basics of regret minimization. Today we will dig deeper into that topic, and learn the more general *online convex optimization* (OCO) framework. We will then finish by proving Sion's minimax theorem via OCO.

2 Regret Minimization

Now we'll get started on how to compute Nash equilibrium. The fastest methods for computing large-scale zero-sum Nash equilibrium are based on what's called *regret minimization*. In the simplest regret-minimization setting we imagine that we are faced with the task of choosing among a finite set of n actions. After choosing an action, a loss between 0 and 1 is revealed for each action. This scenario is then repeated iteratively. The key is that the losses may be adversarial, and we would like to come up with a decision-making procedure that does at least as well as the single best action in hindsight. We will be allowed to choose a distribution over actions, rather than a single action, at each decision point. Classical example applications would be picking stocks, picking which route to take to work in a routing problem, or weather forecasting. To be concrete, imagine that we have n weather-forecasting models that we will use to daily forecast the weather. We would like to decide which model is best to use, but we're not sure how to pick the best one. In that case, we may run a regret-minimization algorithm, where our "action" is to pick a model, or a probability distribution over models, to forecast the weather with. If we spend enough days forecasting, then our average prediction will be as good as the best single model in hindsight.

As can be seen from the above examples, regret minimization methods are widely applicable beyond equilibrium. The next 2-3 lectures will be on regret minimization, with connections to equilibrium computation covered as we go along.

2.1 Setting

Formally, we are faced with the following problem: at each time step $t = 1, \dots, T$:

*Department of Industrial Engineering and Operations Research, Columbia University. Email: christian.kroer@columbia.edu.

1. We recommend a decision $x_t \in \Delta^n$
2. A loss vector $g_t \in [0,1]^n$ is revealed to us, and we pay the loss $\langle g_t, x_t \rangle$

Our goal is to develop an algorithm that recommends good decisions. A natural goal would be to do as well as the best sequence of actions in hindsight. But this turns out to be too ambitious, as the following example shows

Example 1. *We have 2 actions a_1, a_2. At timestep t, if our algorithm puts probability greater than $\frac{1}{2}$ on action a_1, then we set the loss to $(1,0)$, and vice versa we set it to $(0,1)$ if we put less than $\frac{1}{2}$ on a_1. Now we face a loss of at least $\frac{T}{2}$, whereas the best sequence in hindsight has a loss of 0.*

Instead, our goal will be to minimize *regret*. The regret at time t is how much worse our sequence of actions is, compared to the best single action in hindsight:

$$R_t = \sum_{\tau=1}^{t} \langle g_\tau, x_\tau \rangle - \min_{x \in \Delta^n} \sum_{\tau=1}^{t} \langle g_\tau, x \rangle$$

We say that an algorithm is a *no-regret algorithm* if for every $\epsilon > 0$, there exists a sufficiently-large time horizon T such that $\frac{R_T}{T} \leq \epsilon$.

Let's see an example showing that randomization is necessary. Consider the following natural algorithm: at time t, choose the action that minimizes the loss seen so far, where e_i is the vector of all zeroes except index i is 1:

$$x_{t+1} = \operatorname*{argmin}_{x \in \{e_1, \ldots, e_n\}} \sum_{\tau=1}^{t} \langle g_\tau, x \rangle. \tag{FTL}$$

This algorithm is called *follow the leader* (FTL). Note that it always chooses a deterministic action. The following example shows that FTL, as well as any other deterministic algorithm, cannot be a no-regret algorithm

Example 2. *At time t, say that we recommend action i. Since the adversary gets to choose the loss vector after our recommendation, let them choose the loss vector be such that $g_i = 1$, $g_j = 0 \forall j \neq i$. Then our deterministic algorithm has loss T at time T, whereas the cost of the best action in hindsight is at most $\frac{T}{n}$.*

It is also possible to derive a lower bound showing that any algorithm must have regret at least $O(\sqrt{T})$ in the worst case, see e.g. [8] Example 17.5.

2.2 The Hedge Algorithm

We now show that, while it is not possible to achieve no-regret with deterministic algorithms, it is possible with randomized ones. We will consider the *Hedge* algorithm. It works as follows:

- At $t = 1$, initialize a weight vector w^1 with $w_i^1 = 1$ for all actions i
- At time t, choose actions according to the probability distribution $p_i = \frac{w_i^t}{\sum_j w_j^t}$
- After observing g_t, set $w_i^{t+1} = w_i^t \cdot e^{-\eta g_{t,i}}$, where η is a stepsize parameter

The stepsize η controls how aggressively we respond to new information. If $g_{t,i}$ is large then we decrease the weight w_i more aggressively.

Theorem 1. *Consider running Hedge for T timesteps. Hedge satisfies*

$$R_T \leq \frac{\eta T}{2} + \frac{\log n}{\eta}$$

Proof. Let g_t^2 denote the vector of squared losses. Let $Z_t = \sum_j w_j^t$ be the sum of weights at time t. We have

$$\begin{aligned} Z_{t+1} &= \sum_{i=1}^{n} w_i^t e^{-\eta g_{t,i}} \\ &= Z_t \sum_{i=1}^{n} x_{t,i} e^{-\eta g_{t,i}} \\ &\leq Z_t \sum_{i=1}^{n} x_{t,i}(1 - \eta g_{t,i} + \frac{\eta^2}{2} g_{t,i}^2) \\ &= Z_t(1 - \eta \langle x_t, g_t \rangle + \frac{\eta^2}{2} \langle x_t, g_t^2 \rangle) \\ &\leq Z_t e^{-\eta \langle x_t, g_t \rangle + \frac{\eta^2}{2} \langle x_t, g_t^2 \rangle} \end{aligned}$$

where the first inequality uses the second-order Taylor expansion $e^{-x} \leq 1 - x + \frac{x^2}{2}$ and the second inequality uses $1 + x \leq e^x$.

Telescoping and using $Z_1 = n$, we get

$$Z_{T+1} \leq n \prod_{t=1}^{T} e^{-\eta \langle x_t, g_t \rangle + \frac{\eta^2}{2} \langle x_t, g_t^2 \rangle} = n e^{-\eta \sum_{t=1}^T \langle x_t, g_t \rangle + \frac{\eta^2}{2} \sum_{t=1}^T \langle x_t, g_t^2 \rangle}$$

Now consider the best action in hindsight i^*. We have

$$e^{-\eta \sum_{t=1}^T g_{t,i^*}} = w_{i^*}^{T+1} \leq Z_{T+1} \leq n e^{-\eta \sum_{t=1}^T \langle x_t, g_t \rangle + \frac{\eta^2}{2} \sum_{t=1}^T \langle x_t, g_t^2 \rangle}$$

Taking logs gives

$$-\eta \sum_{t=1}^{T} g_{t,i^*} \leq \log n - \eta \sum_{t=1}^{T} \langle x_t, g_t \rangle + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle x_t, g_t^2 \rangle.$$

Now we rearrange to get

$$R_T \leq \frac{\log n}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle x_t, g_t^2 \rangle \leq \frac{\log n}{\eta} + \frac{\eta T}{2},$$

where the last inequality follows from $x_t \in \Delta^n$ and $g_t \in [0,1]^n$. □

If we know T in advance we can now set $\eta = \frac{1}{\sqrt{T}}$ to get that Hedge is a no-regret algorithm.

3 Online Convex Optimization

In OCO, we are faced with a similar, but more general, setting than in the regret-minimization setup from last time. In the OCO setting, we are making decisions from some compact convex set $X \in \mathbb{R}^n$ (analogous to the fact that we were previously choosing probability distributions from Δ^n). After choosing a decision x_t, we suffer a convex loss $f_t(x_t)$. We will assume that f_t is differentiable for convenience, but this assumption is not necessary.

As before, we would like to minimize the regret:

$$R_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in X} \sum_{t=1}^{T} f_t(x)$$

We saw in the last lecture that the follow-the-leader (FTL) algorithm, which always picks the action that minimizes the sum of losses seen so far, does not work. That same argument carries over to the OCO setting. The basic problem with FTL is that it is too unstable: If we consider a setting with $X = [-1, 1]$ and $f_1(x) = \frac{1}{2}x$ and f_t alternates between $-x$ and x then we get that FTL flip-flops between -1 and 1, since they become alternately optimal, and always end up being the wrong choice for the next loss.

This motivates the need for a more stable algorithm. What we will do is to smooth out the decision made at each point in time. In order to describe how this smoothing out works we need to take a detour into *distance-generating functions*.

4 Distance-Generating Functions

A distance-generating function (DGF) is a function $d : X \to \mathbb{R}$ which is continuously differentiable on the interior of X, and strongly convex with modulus 1 with respect to a given norm $\|\cdot\|$, meaning

$$d(x) + \langle \nabla d(x), x' - x \rangle + \frac{1}{2}\|x' - x\|^2 \leq d(x') \forall x, x' \in X$$

If d is twice differentiable on int X then the following definition is equivalent:

$$\langle h, \nabla^2 d(x) h \rangle, \ \forall x \in X, h \in \mathbb{R}^n$$

Intuitively, strong convexity says that the gap between d and its first-order approximation should grow at a rate of at least $\|x - x'\|^2$. Graphically, we can visualize the 1-dimensional version of this as follows:

We will use this gap to construct a distance function. In particular, we say that the *Bregman divergence* associated with a DGF d is the function:

$$D(x' \| x) = d(x') - d(x) - \langle \nabla d(x), x' - x \rangle.$$

Intuitively, we are measuring the distance going from x to x'. Note that this is not symmetric, the distance from x' to x may be different, and so it is not a true distance metric.

Given d and our choice of norm $\|\cdot\|$, the performance of our algorithms will depend on the *set width* of X with respect to d:

$$\Omega_d = \max_{x, x' \in X} d(x) - d(x'),$$

and the dual norm of $\|\cdot\|$:

$$\|g\|_* = \max_{\|x\| \leq 1} \langle g, x \rangle.$$

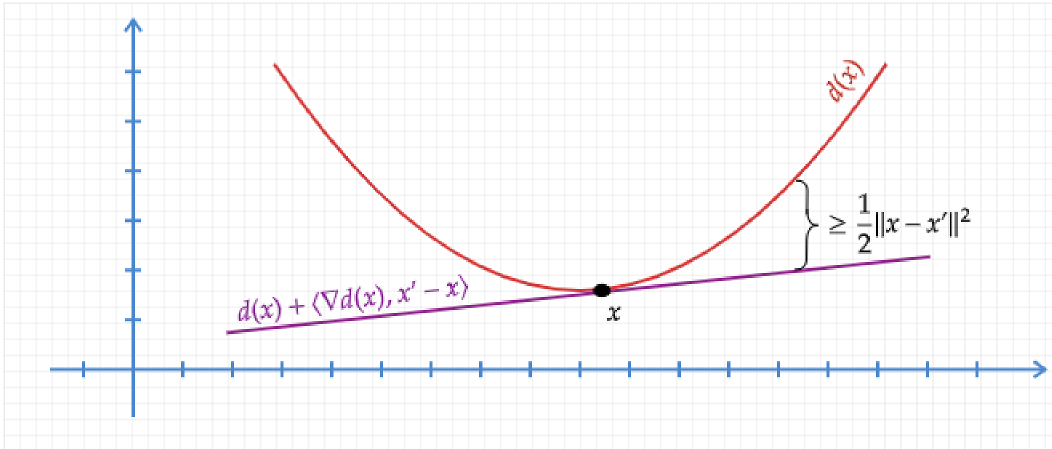

Figure 1: Strong convexity illustrated. The gap between the distance function and its first-order approximation should grow at least as $\|x - x'\|^2$.

In particular, we will care about the largest possible loss vector g that we will see, as measured by the dual norm $\|g\|_*$.

Norms and their dual norm satisfy a useful inequality that is often called the Generalized Cauchy-Schwarz inequality:

$$\langle g, x \rangle = \|x\| \left\langle g, \frac{x}{\|x\|} \right\rangle \leq \|x\| \max_{\|x'\| \leq 1} \langle g, x' \rangle \leq \|x\| \|g\|_*$$

What's the point of these DGFs, norms, and dual norms? The point is that we get to choose all of these in a way that fits the "geometry" of our set X. This will become important later when we will derive convergence rates that depend on Ω and L, where L is an upper bound on the dual norm $\|g\|_{X,*}$ of all loss vectors.

Consider the following two DGFs for the probability simplex $\Delta^n = \{x : \sum_i x_i = 1, x \geq 0\}$:

$$d_1(x) = \sum_i x_i \log(x_i), \quad d_2(x) = \frac{1}{2} \sum_i x_i^2.$$

The first is the *entropy DGF*, the second is the *Euclidean DGF*. First let us check that they are both strongly convex on Δ^n. The Euclidean DGF is clearly strongly convex wrt. the ℓ_2 norm. It turns out that the entropy DGF is strongly-convex wrt. the ℓ_1 norm. Using the second-order definition of strong convexity and any $h \in \mathbb{R}^n$:

$$
\begin{aligned}
\|h\|_1^2 &= \left(\sum_i |h_i|\right)^2 \\
&= \left(\sum_i \sqrt{x_i}\frac{|h_i|}{\sqrt{x_i}}\right)^2 \\
&\leq \left(\sum_i x_i\right)\left(\sum_i \frac{|h_i|^2}{x_i}\right) && \text{by Cauchy-Schwarz} \\
&= \left(\sum_i \frac{|h_i|^2}{x_i}\right) && \text{because } x \in \Delta^n \\
&= \langle h, \nabla^2 d_1(x) h\rangle
\end{aligned}
$$

But now imagine that our losses are in $[0,1]^n$. The maximum dual norm for the Euclidean DGF is then

$$\max_{\|x\|_2 \leq 1} \langle \vec{1}, x\rangle = \left\langle \vec{1}, \frac{\vec{1}}{\sqrt{n}} \right\rangle = \sqrt{n},$$

while $\Omega_{d_2} = 1$.

In contrast, the maximum dual norm for the ℓ_1 norm is

$$\max_{\|x\|_1 \leq 1} \langle \vec{1}, x\rangle = \|\vec{1}\|_\infty = 1.$$

and the set width of the entropy DGF is $\Omega_{d_1} = \log n$.

Thus if our convergence rate is of the form $O\left(\frac{\Omega L}{\sqrt{T}}\right)$, then the entropy DGF gives us a $\log n$ dependence on the dimension n of the simplex, whereas the Euclidean DGF leads to a $\sqrt{n}$ dependence. This shows the well-known fact that the entropy DGF is the "right" DGF for the simplex (from a theoretical standpoint, things turn out to be quite different in numerical performance as we shall see later in the course).

We will need the following inequality on a given norm and its dual norm:

$$\langle g, x\rangle \leq \frac{1}{2}\|g\|_*^2 + \frac{1}{2}\|x\|^2. \tag{1}$$

which follows from

$$\langle g, x\rangle - \frac{1}{2}\|x\|^2 \leq \|g\|_*\|x\| - \frac{1}{2}\|x\|^2 \leq \frac{1}{2}\|g\|_*^2$$

where the first step is by the generalized Cauchy-Schwarz inequality and the second step is by maximizing over x.

We will also need the following result concerning Bregman divergences. Unfortunately it's not clear what intuition one can give about this, except to say that the left-hand side is analogous to a triangle inequality.

Lemma 1 (Three-point lemma)**.** *For any three points x, u, z, we have*

$$D(u\|x) - D(u\|z) - D(z\|x) = \langle \nabla d(z) - \nabla d(x), u - z\rangle$$

The proof is direct from expanding definitions and canceling terms.

5 Online Mirror Descent

We now cover one of the canonical OCO algorithms: *Online Mirror Descent* (OMD). In this algorithm, we smooth out the choice of x_{t+1} in FTL by penalizing our choice by the Bregman divergence $D(x\|x_t)$ from x_t. This has the effect of stabilizing the algorithm, where the stability is essentially due to the strong convexity of d. We pick our iterates as follows:

$$x_{t+1} = \operatorname*{argmin}_{x \in X} \langle \eta \nabla f_t(x), x\rangle + D(x\|x_t).$$

where $\eta > 0$ is the stepsize.

For this algorithm to be well-defined we also need one of the following assumptions:

$$\lim_{x \to \partial X} \|\nabla d(x)\| = +\infty \tag{2}$$

$$\tag{3}$$

or d should be continuously differentiable on all of X.

Let $g_t = \nabla f_t(x_t)$. By first-order optimality of x_{t+1} we have

$$\langle \eta g_t + \nabla d(x_{t+1}) - \nabla d(x_t), x - x_{t+1}\rangle \geq 0, \forall x \in X \tag{4}$$

We first prove what is sometimes called a *descent lemma* or *fundamental inequality* for OMD[1].

Theorem 2. *For all $x^* \in X$, we have*

$$\eta(f_t(x_t) - f_t(x^*)) \leq \eta\langle g_t, x_t - x^*\rangle \leq D(x^*\|x_t) - D(x^*\|x_{t+1}) + \frac{\eta^2}{2}\|g_t\|_*^2$$

Proof. The first inequality in the theorem is direct from convexity of f_t. Thus we only need to prove the second inequality.

$$\begin{aligned}
\langle \eta g_t, x_t - x^*\rangle =& \langle \nabla d(x_t) - \nabla d(x_{t+1}) - \eta g_t, x^* - x_{t+1}\rangle + \langle \nabla d(x_{t+1}) - \nabla d(x_t), x^* - x_{t+1}\rangle \\
&+ \langle \eta g_t, x_t - x_{t+1}\rangle \\
\leq& \langle \nabla d(x_{t+1}) - \nabla d(x_t), x^* - x_{t+1}\rangle + \langle \eta g_t, x_t - x_{t+1}\rangle; \quad \text{by (4)} \\
=& D(x^*\|x_t) - D(x^*\|x_{t+1}) - D(x_{t+1}\|x_t) + \langle \eta g_t, x_t - x_{t+1}\rangle; \quad \text{by three-points lemma} \\
\leq& D(x^*\|x_t) - D(x^*\|x_{t+1}) - D(x_{t+1}\|x_t) + \frac{\eta^2}{2}\|g_t\|_*^2 + \frac{1}{2}\|x_t - x_{t+1}\|^2; \quad \text{by (1)} \\
\leq& D(x^*\|x_t) - D(x^*\|x_{t+1}) + \frac{\eta^2}{2}\|g_t\|_*^2; \quad \text{by strong convexity of } d,
\end{aligned}$$

which proves the theorem. □

Assume that we have a bound L on the gradient norm $\|g_t\|$. Then we have that

Theorem 3. *The OMD algorithm with DGF d achieves the following bound on regret:*

$$R_T \leq \frac{D(x^*\|x_1)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_*^2$$

[1] Our proof follows the one from the excellent lecture notes of Orabona [7]. See also Beck [1] for a proof of the offline variant of mirror descent.

Proof. Consider any $x^* \in X$. Now dividing the inequality from Theorem 2, through by η, and summing from $t = 1, T$ we get

$$\begin{aligned}\sum_{t=1}^{T} \langle g_t, x^* - x_t \rangle &\leq \sum_{t=1}^{T} \frac{1}{\eta} \left(D(x^* \| x_t) - D(x^* \| x_{t+1}) + \frac{\eta^2}{2} \|g_t\|_*^2 \right) \\ &\leq \frac{D(x^* \| x_1) - D(x^* \| x_{T+1})}{\eta} + \sum_{t=1}^{T} \frac{\eta}{2} \|g_t\|_*^2 \\ &\leq \frac{D(x^* \| x_1)}{\eta} + \sum_{t=1}^{T} \frac{\eta}{2} \|g_t\|_*^2\end{aligned}$$

where the second inequality is by noting that the term $D(x^* \| x_t)$ appears with a positive sign at the t'th part of the sum, and negative sign at the $t-1$'th part of the sum. □

Suppose that each f_t is Lipschitz in the sense that $\|g_t\|_* \leq L$, using our bound Ω on DGF differences, and supposing we initialize x_1 at the minimizer of d, then we can set $\eta = \frac{\sqrt{2\Omega}}{L\sqrt{T}}$ to get

$$R_T \leq \frac{\Omega}{\eta} + \frac{\eta T L^2}{2} \leq \sqrt{2\Omega T} L$$

A related algorithm is the *follow-the-regularizer-leader* algorithm. It works as follows:

$$x_{t+1} = \operatorname*{argmin}_{x \in X} \eta \langle \sum_{\tau=1}^{t} g_t, x \rangle + d(x).$$

Note that it is more directly related to FTL: it uses the FTL update, but with a single smoothing term $d(x)$, whereas OMD re-centers a Bregman divergence at $D(\cdot \| x_t)$ at every iteration. FTRL can be analyzed similarly to OMD. It gives the same theoretical properties for our purposes, but we'll see some experimental performance from both algorithms later. For a convergence proof see Orabona [7].

6 Minimax theorems via OCO

In the previous lecture we saw von Neumann's minimax theorem, which was:

Theorem 4 (von Neumann's minimax theorem)**.** *Every two-player zero-sum game has a unique value v, called the* value of the game*, such that*

$$\min_{x \in \Delta^n} \max_{y \in \Delta^m} \langle x, Ay \rangle = \max_{y \in \Delta^m} \min_{x \in \Delta^n} \langle x, Ay \rangle = v.$$

We will now prove a generalization of this theorem.

Theorem 5 (Generalized minimax theorem)**.** *Let $X \in \mathbb{R}^n, Y \in \mathbb{R}^m$ be compact convex sets. Let $f(x, y)$ be continuous, convex in x for a fixed y, and concave in y for a fixed x, with subgradients . Then there exists a value v such that*

$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y) = v.$$

Proof. We will view this is a game between a player choosing the minimizer and a player choosing the maximizer. Let y^* be the y chosen when y is chosen first. When y is chosen second, the maximizer over y can, in the worst case, pick at least y^* every time. Thus we get

$$\max_{y \in Y} \min_{x \in X} f(x,y) \le \min_{x \in X} \max_{y \in Y} f(x,y)$$

For the other direction we will use our OCO results. We run a repeated game where the players choose a strategy x_t, y_t at each iteration t. The x player chooses x_t according to a no-regret algorithm (say OMD), while y_t is always chosen as $\operatorname{argmax}_{y \in Y} f(x_t, y)$. Let the average strategies be

$$\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t, \quad \bar{y} = \frac{1}{T} \sum_{t=1}^{T} y_t.$$

Using OMD with the Euclidean DGF (since X is compact this is well-defined), we get the following bound:

$$R_T = \sum_{t=1}^{T} f(x_t, y_t) - \min_{x \in X} \sum_{t=1}^{T} f(x, y_t) \le O\left(\sqrt{\Omega T} L\right) \tag{5}$$

Now we bound the value of the min-max problem as

$$\min_{x \in X} \max_{y \in Y} \ f(x,y) \le \max_{y \in Y} f(\bar{x}, y) \le \frac{1}{T} \max_{y \in Y} \sum_{t=1}^{T} f(x_t, y) \le \frac{1}{T} \sum_{t=1}^{T} f(x_t, y_t),$$

where the first inequality follows because $\bar{x}$ is a valid choice in the minimization over X, the second inequality follows by convexity, and the third inequality follows because y_t is chosen to maximize $f(x_t, y_t)$. Now we can use the regret bound (5) for OMD to get

$$\begin{aligned}
\min_{x \in X} \max_{y \in Y} \ f(x,y) &\le \frac{1}{T} \min_{x \in X} \sum_{t=1}^{T} f(x, y_t) + O\left(\frac{\sqrt{\Omega} L}{\sqrt{T}}\right) \\
&\le \min_{x \in X} f(x, \bar{y}) + O\left(\frac{\sqrt{\Omega} L}{\sqrt{T}}\right) \\
&\le \max_{y \in Y} \min_{x \in X} f(x,y) + O\left(\frac{\sqrt{\Omega} L}{\sqrt{T}}\right)
\end{aligned}$$

Now taking the limit $T \to \infty$ we get

$$\min_{x \in X} \max_{y \in Y} f(x,y) \le \max_{y \in Y} \min_{x \in X} f(x,y)$$

which concludes the proof. □

For simplicity we assumed continuity of f. The argument did not really need continuity, though. The same proof works for f which is lower/upper semicontinuous in x and y respectively.

7 Historical notes

When applied to the offline setting where $f_t = f \forall t$, OMD is equivalent to the *mirror descent* algorithm which was introduced by Nemirovsky and Yudin [5], with the more modern variant introduced by Beck and Teboulle [2]. There's a functional-analytic interpretation of OMD and mirror descent where one views d as a *mirror map* that allows us to think of f and x in terms of the dual space of linear forms. This was the original motivation for mirror descent, and allows one to apply the algorithm in broader settings, e.g. Banach spaces. This is described in several textbooks and lecture notes e.g. Orabona [7] or Bubeck et al. [3].

The FTRL algorithm run on an offline setting with $f_t = f$ becomes equivalent to *Nesterov's dual averaging* [6].

The term "von Neumann's minimax theorem" usually refers to theorem 4. The more general theorem 5, as well as even more general versions that allow quasi-concavity and quasi-convexity, are often referred to as *Sion's minimax theorem.* von Neumann supposedly already proved a generalization of his original minimax theorem in 1937, which covered this case [4]. A quite general version of what's usually referred to as Sion's minimax theorem can be found on Wikipedia at `https://en.wikipedia.org/wiki/Sion%27s_minimax_theorem`.

References

[1] Amir Beck. *First-order methods in optimization*, volume 25. SIAM, 2017.

[2] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[3] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[4] Tinne Hoff Kjeldsen. John von neumann's conception of the minimax theorem: a journey through different mathematical contexts. *Archive for history of exact sciences*, 56(1):39–68, 2001.

[5] Arkadi Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[6] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[7] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[8] Tim Roughgarden. *Twenty lectures on algorithmic game theory.* Cambridge University Press, 2016.