Online Convex Optimization and Mirrored Descent

Julie Nutini

MLRG - Summer Term

June 13^{th}, 2017

Online Convex Optimization

- In online convex optimization (OCO), learn from experience.
  - At each iteration t, player chooses x_t from decision set $\mathcal{K}$.
  - Convex cost/loss function $f_t \in \mathcal{F} : \mathcal{K} \mapsto \mathbb{R}$ is revealed.
    - $\mathcal{F}$ bounded family of cost functions.
    - Unknown to decision maker beforehand.
    - Can be adversarially chosen.
    - Can depend on action taken by decision maker.
  - Cost incurred by player is $f_t(x_t)$.
- **Applications**: online routing, ad selection for search engines, spam filtering, prediction from expert advice, online shortest paths, portfolio selection, matrix completion and recommendation systems, etc. ...

Examples

- **Online Linear Spam Filtering**:
  - $\mathcal{K} = \{x \in \mathbb{R}^d \mid \|x\| \leq \omega\}$, norm-bounded linear filters.
  - Features (words) $a \in \mathbb{R}^d$, labels $b \in \{-1, 1\}$.
  - At time t, select pair (a_t, b_t).
  - Loss function: $f_t(x) = (\text{sign}(a_t^T x) - b_t)^2$.
- **Online Matrix Completion** (recommendation systems):
  - $\mathcal{K} \subseteq \{X \mid X \in \{0, 1\}^{n \times m}\}$, n people, m movies.
  - $X(i, j) = 1$ implies person i likes song j (0 otherwise).
  - At time t, select $a_t = (i_t, j_t)$ and corresponding $b_t \in \{0, 1\}$.
  - Loss function: $f_t(X) = (X(i_t, j_t) - b_t)^2$.

Restrictions

- Losses f_t must be bounded.
  - Otherwise, adversary could keep decreasing scale of loss.

    → Possibly never recover from loss of first step.
- Decision set $\mathcal{K}$ must be bounded/structured (not necessarily finite).
  - Consider decision making with an infinite set of possible decisions.
  - Adversary can assign high loss to all strategies chosen by player indefinitely, while setting apart some strategies with zero loss.

    → Precludes any meaningful performance metric.

Goal of Offline vs Online Convex Optimization

- **Offline**: To minimize optimization error,

$$h_t = f(x_t) - f(x^*).$$

- **Online**: To minimize regret,

$$\text{regret} = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

$\rightarrow$ Optimization error ill-defined in online setting (objective changes at each t).

- Assume all $f_t := f$ and $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$ (average decision).
- Then $f(\bar{x}_T) \rightarrow f(x^*)$ at a rate at most the average regret,

$$f(\bar{x}_T) - f(x^*) \leq \frac{1}{T}\sum_{t=1}^{T}[f(x_t) - f(x^*)] = \frac{\text{regret}}{T}.$$

General Regret

- Given an algorithm $\mathcal{A}$, which maps certain game history to decision in $\mathcal{K}$.
- Formally define regret of $\mathcal{A}$ after T iterations as

$$\text{regret}_T(\mathcal{A}) = \sup_{\{f_1,\ldots,f_t\}\subseteq\mathcal{F}} \left\{ \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \right\}.$$

- Algorithm performs well if its regret is sublinear as a function of T, i.e., $o(T)$.
  - $\rightarrow$ On average, the algorithm performs as well as best fixed strategy in hindsight.

Projections Onto Convex Sets

- **Projection onto a convex set**:
  - Defined as the closest point inside the convex set to a given point,

$$\Pi_{\mathcal{K}}(y) \triangleq \operatorname*{argmin}_{x \in \mathcal{K}} \|x - y\|.$$

→ Projection of a given point over a compact convex set exists and is unique.

Theorem (Pythagoras, circa 500 BC)

Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, $y \in \mathbb{R}^d$ and $x = \Pi_{\mathcal{K}}(y)$. Then for any $z \in \mathcal{K}$ we have

$$\|y - z\| \geq \|x - z\|.$$

Offline Gradient Descent - Algorithm

Algorithm 2 Basic gradient descent

1: Input: f, T, initial point $\mathbf{x}_1 \in \mathcal{K}$, sequence of step sizes $\{\eta_t\}$
2: **for** $t = 1$ to T **do**
3: Let $\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)$, $\mathbf{x}_{t+1} = \Pi_{\mathcal{K}}(\mathbf{y}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_{T+1}$

- Take a step in direction of negative gradient of cost.
- May result in point outside underlying convex set.
- Algorithm projects back onto the convex set.

Online Gradient Descent - Algorithm

Algorithm 6 online gradient descent

1: Input: convex set $\mathcal{K}$, T, $\mathbf{x}_1 \in \mathcal{K}$, step sizes $\{\eta_t\}$
2: **for** $t = 1$ to T **do**
3: Play $\mathbf{x}_t$ and observe cost $f_t(\mathbf{x}_t)$.
4: Update and project:

$$\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)$$
$$\mathbf{x}_{t+1} = \prod_{\mathcal{K}}(\mathbf{y}_{t+1})$$

5: **end for**

- Take a step in direction of negative gradient of previous cost.
- May result in point outside underlying convex set.
- Algorithm projects back to the convex set.

Assumptions

- All f_t are convex and differentiable.
- Decision set $\mathcal{K} \in \mathbb{R}^d$ is a compact convex set.
- Denote by D an upper bound on the diameter of $\mathcal{K}$:

$$\forall\, x, y \in \mathcal{K}, \quad \|x - y\| \leq D.$$

- Denote by G an upper bound on the norm of the subgradients of f over $\mathcal{K}$:

$$\|\nabla f(x)\| \leq G \text{ for all } x \in \mathcal{K}.$$

Regret Bound for Online Gradient Descent

Theorem

Online gradient descent (GD) with step sizes $\eta_t = \frac{D}{G\sqrt{t}}$ *guarantees the following for all* $T \geq 1$*:*

$$\textit{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(x^*) \leq \frac{3}{2} GD\sqrt{T}$$

Proof.

Let $x^* \in \text{argmin}_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$. By the convexity of f,

$$f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t)^T (x_t - x^*). \tag{1}$$

Using the update rule for x_{t+1} and the Pythagorean theorem,

$$\|x_{t+1} - x^*\|^2 = \|\Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t)) - x^*\|^2 \leq \|x_t - \eta_t \nabla f_t(x_t) - x^*\|^2.$$

Hence,

$$\begin{aligned} & \|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)^T (x_t x^*) \\ \iff \quad & 2\nabla f_t(x_t)^T (x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t G^2. \end{aligned} \tag{2}$$

Proof cont'd ...

Summing (1) and (2) from $t = 1$ to T, and setting $\eta_t = \frac{D}{G\sqrt{t}}$ (with $\frac{1}{\eta_0} \triangleq 0$):

$$
\begin{aligned}
2\left(\sum_{t=1}^{T} f_t(x_t) - f_t(x^*)\right) &\le 2\sum_{t=1}^{T} \nabla f_t(x_t)^T (x_t - x^*) \\
&\le \sum_{t=1}^{T} \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + G^2 \sum_{t=1}^{T} \eta_t \\
\text{(since } 1/\eta_0 \triangleq 0, \|x_{T+1} - x^*\|^2 \ge 0) \quad &\le \sum_{t=1}^{T} \|x_t - x^*\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + G^2 \sum_{t=1}^{T} \eta_t \\
&\le D^2 \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + G^2 \sum_{t=1}^{T} \eta_t \\
&\le D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t \le 3DG\sqrt{T}.
\end{aligned}
$$

Last inequality follows from $\eta_t = \frac{D}{G\sqrt{t}}$ and $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \le 2\sqrt{T}$. □

Better Regret Bounds?

- Online GD is a linear-time algorithm for the most general case.
  - Tight regret bounds and elementary proofs.
- Do regret bounds in OCO vary as much as the convergence bounds in offline convex optimization over different classes of convex cost functions?

  → Yes!
- Some problems admit better regret:
  - Least squares linear regression
  - Soft-margin SVM
  - Portfolio selection

Assumptions

- A function is α-strongly convex if

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

- If a function is twice differentiable and admits a second derivative (Hessian), above condition equivalent to

$$\alpha I \preceq \nabla^2 f(x),$$

where $A \preceq B$ if the matrix $B - A$ is positive semidefinite.

Example of Strongly Convex Cost Function

- **Online Soft-Margin SVM**:
  - Features $a \in \mathbb{R}^d$, labels $b \in \{-1, 1\}$
  - $\mathcal{K}$ is a set of linear predictors:
    - Weight vectors $x \in \mathbb{R}^d$.
    - Prediction of weight vector x for feature vector a is $a^T x$.
  - Loss function: $f_t(x) = \underbrace{\max\{0, 1 - b_t a_t^T x\}}_{\text{convex, non-smooth (hinge loss)}} + \underbrace{\lambda \|x\|^2}_{\text{strongly-convex}}$

$\rightarrow$ Online GD with step-sizes $\eta_t \approx \frac{1}{t}$ has $O(\log T)$ regret.

Logarithmic Regret Bound for Online GD

Theorem

For α-strongly convex loss functions, online GD with step sizes $\eta_t = \frac{1}{\alpha t}$ achieves the following guarantee for all $T \geq 1$:

$$\textit{regret}_T \leq \frac{G^2}{2\alpha}(1 + \log(T)).$$

Proof.

Let $x^* \in \text{argmin}_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$.

Applying the definition of α-strong convexity to the pair of points $x_t,\ x^*$, we have

$$2\left(f_t(x_t) - f_t(x^*)\right) \leq 2\nabla f_t(x_t)^T \left(x_t - x^*\right) - \alpha\|x^* - x_t\|^2. \quad (3)$$

Using the update rule for x_{t+1} and the Pythagorean Theorem, we get

$$\|x_{t+1} - x^*\|^2 = \|\Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t)) - x^*\|^2 \leq \|x_t - \eta_t \nabla f_t(x_t) - x^*\|^2.$$

Hence,

$$\begin{aligned} & \|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 + \eta_t^2\|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)^T(x_t - x^*) \\ \iff \quad & 2\nabla f_t(x_t)^T(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t G^2. \quad (4) \end{aligned}$$

Proof cont'd ...

Summing (3) and (4) from $t = 1$ to T, and setting $\eta_t = 1/\alpha t$ (define $\frac{1}{\eta_0} \triangleq 0$):

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

(since $1/\eta_0 \triangleq 0, \|x_{T+1} - x^*\|^2 \geq 0$)

$$= 0 + G^2 \sum_{t=1}^{T}\frac{1}{\alpha t}$$

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

□

Asymptotic Convergence Rates

- **Offline**: Establish convergence rates of optimization error, $f(x_t) - f(x^*)$.

	β-smooth	α-strongly convex
Gradient descent	$\frac{\beta}{T}$	$\frac{1}{\alpha T}$
Accelerated GD	$\frac{\beta}{T^2}$	-

- **Online**: Establish asymptotic regret bounds.

	β-smooth	α-strongly convex
Upper bound	$\sqrt{T}$	$\frac{1}{\alpha}\log T$
Average regret	$\frac{1}{\sqrt{T}}$	$\frac{\log T}{\alpha T}$

$\rightarrow$ Smoothness does not improve asymptotic regret rates.

- β-smooth $\equiv$ β-Lipschitz ∇f_t.

$\rightarrow$ Despite potentially different cost functions, the regret attained is sublinear.

Online Mirrored Descent

- **Mirrored Descent**: Class of first order methods, generalize gradient descent.
- **Online Mirrored Descent** (OMD): Online version of Mirrored Descent.
    - Computes current decision using gradient update rule and previous decision.
    - Update carried out in *dual* space.
        - Duality notion defined by the choice of a regularization function R.
    - Using regularization:
        - Ensures stability of decision $\rightarrow$ unlike Follow-The-Leader (see last week's slides).
        - Transforms space in which gradient updates are performed $\rightarrow$ unlike Follow-The-Regularized-Leader, which updates in Euclidean space.
        - Enables better bounds in terms of geometry of the space.

Bregman Divergence

- Bregman divergence with respect to regularization function R defined by

$$B_R(x||y) := R(x) - R(y) - \nabla R(y)^T(x - y).$$

- For twice differentiable functions,

$$B_R(x||y) = \frac{1}{2}\|x - y\|_z^2 \triangleq \frac{1}{2}\|x - y\|_{\nabla^2 R(z)}^2,$$

for some point $z \in [x, y]$,

$$\|\cdot\|_{x,y}^* \triangleq \|\cdot\|_z^* \triangleq \|\cdot\|_{\nabla^{-2} R(z)}.$$

- Thus,

$$B_R(x||y) = \frac{1}{2}\|x - y\|_{x,y}^2.$$

- Projection of a point y according to the Bregman divergence is given by

$$\Pi_{\mathcal{K}}^R(y) = \underset{x \in \mathcal{K}}{\text{argmin}}\, B_R(x||y).$$

Versions of Online Mirrored Descent

- There are two versions of OMD:
  - **Lazy**: Keeps track of a point in Euclidean space and projects onto convex decision set $\mathcal{K}$ only at decision time.
  - **Agile**: Maintains feasible point at all times, much like online GD.

Online Mirrored Descent - Algorithm

Algorithm 11 Online Mirrored Descent

Input: parameter $\eta > 0$, regularization function $R(\mathbf{x})$.
Let $\mathbf{y}_1$ be such that $\nabla R(\mathbf{y}_1) = \mathbf{0}$ and $\mathbf{x}_1 = \arg\min_{\mathbf{x} \in \mathcal{K}} B_R(\mathbf{x} \| \mathbf{y}_1)$.
for $t = 1$ to T **do**
    Play $\mathbf{x}_t$.
    Observe the payoff function f_t and let $\nabla_t = \nabla f_t(\mathbf{x}_t)$.
    Update $\mathbf{y}_t$ according to the rule:

$$\begin{aligned} &[\text{Lazy version}] && \nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{y}_t) - \eta\, \nabla_t \\ &[\text{Agile version}] && \nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{x}_t) - \eta\, \nabla_t \end{aligned}$$

    Project according to B_R:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} B_R(\mathbf{x} \| \mathbf{y}_{t+1})$$

end for

→ **Main point**: Can replace Euclidean norm with other divergence functions.

Regularized Follow-The-Leader

- Let R be a strongly convex, smooth and twice differentiable.

Algorithm 10 Regularized Follow The Leader

1: Input: $\eta > 0$, regularization function R, and a convex compact set $\mathcal{K}$.
2: Let $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \{R(\mathbf{x})\}$.
3: **for** $t = 1$ to T **do**
4: Predict $\mathbf{x}_t$.
5: Observe the payoff function f_t and let $\nabla_t = \nabla f_t(\mathbf{x}_t)$.
6: Update

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \left\{ \eta \sum_{s=1}^{t} \nabla_s^\top \mathbf{x} + R(\mathbf{x}) \right\}$$

7: **end for**

- Regularization improves stability of prediction (see last week).
- Yields asymptotically optimal regret bounds.

Online Mirrored Descent and Regularized Follow-The-Leader

- For linear cost functions, RFTL and lazy-OMD algorithms are equivalent.
  - → Get regret bounds for free.

Lemma

Let $f_1, \ldots, f_T$ be linear cost functions. The lazy OMD and RFTL algorithms produce identical predictions, i.e.,

$$\underset{x \in \mathcal{K}}{\text{argmin}}\, B_R(x \| y_t) = \underset{x \in \mathcal{K}}{\text{argmin}} \left(\eta \sum_{s=1}^{t-1} \nabla_s^T x + R(x) \right).$$

Discussion

- **Online convex optimization**:
  - Learns from experience as more aspects of the problem are observed.
  - Many applications.
  - Goal is to minimize regret.
- **Online Gradient Descent**:
  - Takes step in direction of negative gradient of previous cost function.
  - Asymptotic regret bound:
    - Convex cost functions: $O(\sqrt{T})$ regret.
    - α-strongly convex cost function: $O(\log T)$ regret.
- **Online Mirrored Descent**:
  - Replaces Euclidean norm with Bregman divergence function.
  - Lazy version equivalent to Regularized-Follow-The-Leader.

References

- Hazan, Elad. *Introduction to Online Convex Optimization*, 2016.