Optimization

Fall semester 2022/23

Kristóf Bérczi

Eötvös Loránd University
Institute of Mathematics
Department of Operations Research

Lecture 4: Gradient descent, Mirror descent, and Multiplicative Weights Update

Setting

Objective: $\min_{x \in \mathbb{R}^n} f(x)$ (**unconstrained setting**)

Model: 1st-order oracle is given, i.e., we can query the gradient at any point.

Solution: Given $\varepsilon > 0$, output a point $x \in \mathbb{R}^n$ s.t. $f(x) \leqslant y^* + \varepsilon$, where y^* denotes the optimal value.

- The running time will be proportional to $1/\varepsilon$, hence it is not polynomial. However, we will see that in this setting one cannot obtain polynomial time algorithms.

Remark: As f is convex, a local minimum is a global minimum. So as long as we can find a point to decrease the objective value, we are making progress and we won't get stuck. But how to decrease the objective?

Gradient descent

Not a single method, but a general framework.

Scheme:

1. Choose a starting point $x_1 \in \mathbb{R}^n$.
2. Suppose $x_1, \ldots, x_t$ are computed. Choose x_{t+1} as a linear combination of x_t and $\nabla f(x_t)$.
3. Stop once a certain stopping criterion is met and output the last iterate.

If T is the total number of iterations, then the running time is $O(T \cdot M(x))$, where $M(x)$ is the time of each update.

- The update time $M(x)$ cannot be optimized below a certain level.
- The main goal is to keep T as small as possible.

Why using the gradient? I

We only have local information about $x \Rightarrow$ a reasonable idea is to pick a direction which locally provides the **largest drop** in the function value.

Formally: Pick a unit vector u for which a 'tiny' (δ) step in direction u maximizes

$$f(x) - f(x + \delta u).$$

This leads to the optimization problem

$$\max_{\|u\|=1} \left[\lim_{\delta \to 0^+} \frac{f(x) - f(x + \delta u)}{\delta} \right].$$

By the Taylor approximation of f, the limit is simply the directional derivative of f at x in direction u, thus

$$\max_{\|u\|=1} \left[-\langle \nabla f(x), u \rangle \right].$$

Cauchy-Schwarz inequality

Cauchy-Schwarz inequality

For all $x, y \in \mathbb{R}^n$, we have $\langle x, y \rangle \leqslant \|x\| \|y\|$.

Proof sketch.
Assuming $x, y \in \mathbb{R}^2$, we know that $\langle x, y \rangle = \|x\| \|y\| \cos \theta$, where θ is the angle between x and y. In higher dimensions, intuitively, the two vectors x and y form together a subspace of dimension at most 2 that can be thought of as $\mathbb{R}^2$. $\square$

Why using the gradient? II

Recall: $\max_{\|u\|=1} [-\langle \nabla f(x), u \rangle]$

From the Cauchy-Schwarz inequality, we get

$$-\langle \nabla f(x), u \rangle \leqslant \|\nabla f(x)\| \|u\| = \|\nabla f(x)\|,$$

and equality holds if $u = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$.

$\Rightarrow$ Moving in the direction of the **negative gradient** is an instantaneously good strategy - called the **gradient flow**:

$$\frac{dx}{dt} = -\frac{\nabla f(x)}{\|\nabla f(x)\|}.$$

Question: How to implement the strategy on a computer?

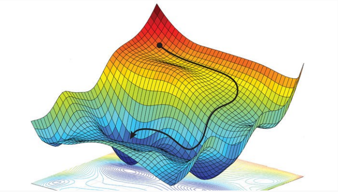
Natural discretization:

$$x_{t+1} = x_t - \alpha \frac{\nabla f(x_t)}{\|\nabla f(x_t)\|},$$

where $\alpha > 0$ is the 'step length'. More generally,

$$x_{t+1} = x_t - \eta \nabla f(x_t),$$

where $\eta > 0$ is a parameter.



Assumptions

Step length: Ideally, we would like to take **big steps**. This results in smaller number of iterations, but the function can change dramatically, leading to a large error.

Solution: Assumptions on certain **regularity parameters**.

1. **Lipschitz gradient.** For every $x, y \in \mathbb{R}^n$ we have
$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|.$$
This is also sometimes referred to as L-**smoothness** of f.
$\Rightarrow$ Around x, the gradient changes in a controlled manner; we can take larger step size.
2. **Bounded gradient.** For every $x \in \mathbb{R}^n$ we have
$$\|\nabla f(x)\| \leqslant G.$$
This implies that f is G-Lipschitz.
$\Rightarrow$ The function can go towards infinity in a controlled manner.
3. **Good initial point.** A point x_1 is provided such that $\|x_1 - x^*\| \leqslant D$, where x^* is some optimal solution.

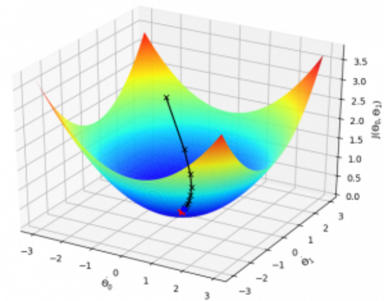
Lipschitz gradient

Thm.

Given a first-order oracle access to an L-Lipschitz convex function $f : \mathbb{R}^n \to \mathbb{R}$, an initial point $x_1 \in \mathbb{R}^n$ with $\|x_1 - x^*\| \leqslant D$, and $\varepsilon > 0$, there is an algorithm the outputs a point $x \in \mathbb{R}^n$ such that $f(x) \leqslant f(x^*) + \varepsilon$. The algorithm makes $T = O\left(\frac{LD^2}{\varepsilon}\right)$ queries to the oracle and performs $O(nT)$ arithmetic operations.

Algorithm

1. Let $T = O(\frac{LD^2}{\varepsilon})$.
2. Let $\eta = \frac{1}{L}$.
3. Repeat for $t = 1, \ldots, T-1$:
   - $x_{t+1} = x_t - \eta \nabla f(x_t)$.
4. **Output** x_T.



Lipschitz gradient

Lower bound

Consider any algorithm for solving the convex unconstrained minimization problem $\min_{x \in \mathbb{R}^n} f(x)$ in the first-order model, when f has Lipschitz gradient with constant L and the initial point $x_1 \in \mathbb{R}^n$ satisfies $\|x_1 - x^*\| \leqslant D$. There is a function f such that

$$\min_{1 \leqslant i \leqslant T} f(x_i) - \min_{x \in \mathbb{R}^n} f(x) \geqslant \frac{LD^2}{T^2}.$$

$\Rightarrow$ The theorem translates to a lower bound of $\Omega(\frac{1}{\sqrt{\varepsilon}})$ iterations to reach an ε-optimal solution.

Is there a method which matches the $\frac{1}{\sqrt{\varepsilon}}$ *iterations bound?* **Yes!**

Nesterov's accelerated gradient descent algorithm

Under the same assumptions, there is an algorithm the outputs a point $x \in \mathbb{R}^n$ such that $f(x) \leqslant f(x^*) + \varepsilon$, makes $T = O(\frac{\sqrt{LD}}{\sqrt{\varepsilon}})$ queries to the oracle, and performs $O(nT)$ arithmetic operations.

Constrained setting - projection

Objective: $\min_{x \in K} f(x)$ (**constrained setting**)

$\Rightarrow$ The next iterate x_{t+1} might fall outside of K, hence we need to project it back onto K, that is,

$$x_{t+1} = \mathrm{proj}_K(x_t - \eta_t \cdot \nabla f(x_t)).$$

Difficulty: The projection may or may not be computationally expensive to perform.

Thm.

Given a first-order oracle access to a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with an L-Lipschitz gradient, oracle access to a projection operator proj_K onto a convex set $K \subseteq \mathbb{R}^n$, an initial point $x_1 \in \mathbb{R}^n$ with $\|x - x^*\| \leqslant D$, and $\varepsilon > 0$, there is an algorithm the outputs a point $x \in \mathbb{R}^n$ such that $f(x) \leqslant f(x^*) + \varepsilon$. The algorithm makes $T = O\left(\frac{LD^2}{\varepsilon}\right)$ queries to the first-order and the projection oracles and performs $O(nT)$ arithmetic operations.

Regularizers I

The Lipschitz gradient algorithm leaves out convex functions which are **non-differentiable**, such as $f(x) = \sum_{i=1}^{n} |x_i|$ or $f(x) = \max\{|x_1|, \ldots, |x_n|\}$.

Let's reconsider how to choose the next point to converge quickly?

Obvious choice: $x^{t+1} = \arg\min_{x \in K} f(x)$

$\Rightarrow$ Coverges quickly to x^* (in one step). Yet, it is not very helpful as x^{t+1} is **hard to compute**.

Idea: Construct a function f^t that **approximates** f in a certain sense and is **easy to minimize**. The update rule becomes

$$x^{t+1} = \arg\min_{x \in K} f^t(x).$$

$\Rightarrow$ Intuitively, if f^t becomes more and more accurate, the sequence of iterates should converge to x^*.

Regularizers II

Example

The Lipschitz gradient algorithm corresponds to the choice

$$f^t(x) = f(x^t) + \langle \nabla f(x^t), x - x^t \rangle + \frac{L}{2}\|x - x^t\|^2.$$

Indeed, $\nabla f^t(x) = \nabla f(x^t) + L(x - x^t) = 0$ if and only if $x = x^t - \frac{1}{L}\nabla f(x^t)$.

In general, when the function is not differentiable, one can try to use the first order approximation of f at x^t, that is,

$$f^t(x) = f(x^t) + \langle \nabla f(x^t), x - x^t \rangle.$$

Then $f^t(x) \leqslant f(x)$ and f^t gives a descent approximation of f in a small neighborhood x^t. The resulting updating rule will be

$$x^{t+1} = \arg\min_{x \in K}\{f(x^t) + \langle \nabla f(x^t), x - x^t \rangle\}.$$

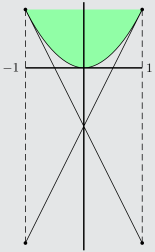
Regularizers III

Example

$K = [-1, 1]$ amd $f(x) = x^2$

$\Rightarrow$ The algorithm is way too aggressive as it jumps between -1 and $+1$ indefinitely.

[**Even worse:** if K is ubounded, then the minimum is not attained at any finite point!]



Idea: Add a term involving a distance function $D : K \times K \to \mathbb{R}$ that does not allow x^{t+1} to land far away from x^t. More precisely,

$$\begin{aligned} x^{t+1} &= \arg\min_{x \in K} \{D(x, x^t) + \eta(f(x^t) + \langle \nabla f(x^t), x - x^t \rangle)\} \\ &= \arg\min_{x \in K} \{D(x, x^t) + \eta \langle \nabla f(x^t), x \rangle\}. \end{aligned}$$

Remark: By picking large η, the significance of the regularizer is reduced. By picking small η, we force x^{t+1} to stay close to x^t.

Kullback-Leibler divergence

Objective: $\min_{p\in\Delta_n} f(p)$, where $\Delta_n = \{p \in [0,1]^n : \sum_{i=1}^n p_i = 1\}$ is the **probability simplex**.

Recall that

$$p^{t+1} = \arg\min_{p\in\Delta_n}\{D(p,p^t) + \eta\langle\nabla f(p^t), p\rangle\}.$$

For two probability distributions , $p, q \in \Delta_n$, their **Kullback-Leibler divergence** is defined as

$$D_{KL}(p,q) = -\sum_{i=1}^n p_i \log \frac{q_i}{p_i}.$$

Remarks:

- D_{KL} is **not** symmetric
- $D_{KL}(p,q) \geqslant 0$

Lemma

Consider any vector $q \in \mathbb{R}^n_{\geqslant 0}$ and a vector $g \in \mathbb{R}^n$. Define $w_i^* = q_i e^{-\eta g_i}$ for $i = 1, \ldots, n$. Then $\arg\min_{p\in\Delta_n}\{D_{KL}(p,q) + \eta\langle g, p\rangle\} = \frac{w^*}{\|w^*\|_1}$.

Exponential gradient descent

Algorithm

1. Initialize $p^1 = \frac{1}{n}\mathbb{1}$ (uniform distribution).
2. Repeat for $t = 1, \ldots, T$:
   - Obtain $g^t = \nabla f(p_t)$.
   - Let $w^{t+1} \in \mathbb{R}^n$ and $p^{t+1} \in \Delta_n$ be defined as
   $$w_i^{t+1} = p_i^t e^{-\eta g_i^t} \text{ and } p_i^{t+1} = \frac{w_i^{t+1}}{\sum_{j=1}^n w_j^{t+1}}.$$
3. **Output** $\bar{p} = \frac{1}{T}\sum_{t=1}^T p^t$.

Thm.

Suppose that $f : \Delta_n \to \mathbb{R}$ is a convex function which satisfies $\|\nabla f(p)\| \leqslant G$ for all $p \in \Delta_n$. If we set $\eta = \Theta\left(\frac{\sqrt{\log n}}{\sqrt{T}G}\right)$, then after $T = \Omega\left(\frac{G^2 \log n}{\varepsilon^2}\right)$ iterations of the algorithm, the point $\bar{p} = \frac{1}{T}\sum_{t=1}^T p^t$ satisfies $f(\bar{p}) \leqslant f(p^*) + \varepsilon$.

Multiplicative weights update

The analysis of the exponential gradient descent algorithm reveals that one can work with arbitrary vectors g^t instead of the gradients of f.

Algorithm

1. Initialize $p^1 = \frac{1}{n}\mathbb{1}$ (uniform distribution).
2. Repeat for $t = 1, \ldots, T$:
   - Obtain g^t from the oracle.
   - Let $w^{t+1} \in \mathbb{R}^n$ and $p^{t+1} \in \Delta_n$ be defined as
   $$w_i^{t+1} = p_i^t e^{-\eta g_i^t} \text{ and } p_i^{t+1} = \frac{w_i^{t+1}}{\sum_{j=1}^n w_j^{t+1}}.$$
3. **Output** $p^1, \ldots, p^T \in \Delta_n$

Thm.

Assume that $\|g^t\| \leqslant G$ for $t = 1, \ldots, T$. If we set $\eta = \Theta\left(\frac{\sqrt{\log n}}{\sqrt{T}G}\right)$, then after $T = \Theta\left(\frac{G^2 \log n}{\varepsilon^2}\right)$ iterations we have $\frac{1}{T}\sum_{i=1}^T \langle g^t, p^t\rangle \leqslant \min_{p\in\Delta_n} \frac{1}{T}\sum_{i=1}^T \langle g^t, p\rangle + \varepsilon$.

Regularizers revisited

Update rule: $x^{t+1} = \arg\min_{x\in K}\{D(x,x^t) + \eta\langle \nabla f(x^t), x\rangle\}$.

The **Bregman divergence** of a function $f : K \to \mathbb{R}$ at $u, w \in K$ is defined to be

$$D_f(u,w) = f(u) - (f(w) + \langle \nabla f(w), u - w\rangle).$$

Remark: The Kullback-Leibler divergence is the Bregman divergence corresponding to the function $H(x) = \sum_{i=1}^n x_i \log x_i - x_i$.

For any convex regularizer $R : \mathbb{R}^n \to \mathbb{R}$, by denoting the gradient at step t by g^t, we have

$$\begin{aligned} x^{t+1} &= \arg\min_{x\in K}\{D_R(x,x^t) + \eta\langle g^t, x\rangle\} \\ &= \arg\min_{x\in K}\{\eta\langle g^t, x\rangle + R(x) - R(x^t) - \langle \nabla R(x^t), x - x^t\rangle\} \\ &= \arg\min_{x\in K}\{R(x) - \langle \nabla R(x^t) - \eta g^t, x\rangle\}. \end{aligned}$$

Suppose that there exists w^{t+1} such that $\nabla R(w^{t+1}) = \nabla R(x^t) - \eta g^t$. Then

$$\begin{aligned} x^{t+1} &= \arg\min_{x\in K}\{R(x) - \langle \nabla R(x^t) - \eta g^t, x\rangle\} \\ &= \arg\min_{x\in K}\{R(x) - R(w^{t+1}) + \langle \nabla R(w^{t+1}), x\rangle\} \\ &= \arg\min_{x\in K}\{D_R(x, w^{t+1})\}. \qquad (D_R\text{-projection of } w^{t+1} \text{ onto } K) \end{aligned}$$

Mirror descent I

Assume that the regularizer $R : \Omega \to \mathbb{R}^n$ has a domain Ω which contains K as a subset. Furthermore, assume that $\nabla R : \Omega \to \mathbb{R}^n$ is a bijection (**mirror map**).

Algorithm

Input: 1st-order oracle access to convex $f : K \to \mathbb{R}$, oracle access to ∇R and its inverse, projection operator w.r.t. $D_R(\cdot, \cdot)$, initial point $x^1 \in K$, parameter $\eta > 0$, integer $T > 0$.

1. Repeat for $t = 1, \ldots, T$:
   - Obtain $g^t = \nabla f(p_t)$.
   - Let w^{t+1} be such that $\nabla R(w^{t+1}) = \nabla R(x^t) - \eta \nabla f(x^t)$.
   - Set $x^{t+1} = \arg\min_{x \in K} D_R(x, w^{t+1})$.
2. **Output** $\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x^t$.

Remarks:

- The mirror map ∇R and its inverse should be efficiently computable.
- The projection step $\arg\min_{x \in K} D_R(x, w^{t+1})$ should be computationally easy to perform.

Mirror descent II

Thm.

Let $f : K \to \mathbb{R}$ and $R : \Omega \to \mathbb{R}$ be convex functions with $K \subseteq \Omega \subseteq \mathbb{R}^n$. Suppose that the gradient map $\nabla R : \Omega \to \mathbb{R}^n$ is a bijection, $\|\nabla f(x)\| \leqslant G$ for $x \in K$ (**bounded gradient**), and that $D_R(x, y) \geqslant \frac{\sigma}{2}\|x - y\|^{*2}$ for $x \in \Omega$ (R is σ-**strongly convex** w.r.t. dual norm $\|\cdot\|^*$).
If we set $\eta = \Theta\left(\frac{\sqrt{\sigma D_R(x^*, x^1)}}{\sqrt{T} G}\right)$, then after $T = \Theta\left(\frac{G^2 D_R(x^*, x^1)}{\sigma \varepsilon^2}\right)$ iterations the point $\bar{x}$ satisfies $f(\bar{x}) \leqslant f(x^*) + \varepsilon$.

Reading assignment

N. Vishnoi. Algorithms for convex optimization.

- Chapter 6
- Chapter 7

L.C. Lau. Convexity and optimization.

- Lecture 7

Exercises

1. Let $G = (V, E)$ be an undirected graph and $s, t \in V$. Consider the following problem:
$$\begin{aligned} \min \sum_{uv \in E} & |x_u - x_v| \\ \text{s.t.} \quad & x_s - x_t = 1 \end{aligned}$$
This is not a linear program in this form. Rewrite it as a linear program. (1pt)

2. Let us consider the following functions:
$$\begin{aligned} f_1(w_1, w_2) =& \frac{1}{2} w_1^2 + \frac{7}{2} w_2^2, \textit{ and} \\ f_2(w_1, w_2) =& 100(w_2 - w_1^2)^2 + (1 - w_1)^2 \qquad \text{(Rosenbrock's function).} \end{aligned}$$
   a. Calculate the gradients of the functions. (2pts)
   b. Are these function convex? (2pts)
   c. Determine the global minimum of the functions. (2pts)
   d. Choose a starting point $w = (w_1, w_2)$ within distance 5 from an optimal solution, and perform one step of the Gradient descent algorithm. (2pts)

3. Given a convex, differentiable function $F : K \to \mathbb{R}$ over a convex subset K of $\mathbb{R}^n$, the Bergman divergence of $x, y \in K$ is defined as
$$D_F(x, y) = F(x) - F(y) - \langle \nabla F(y), x - y \rangle.$$
Prove that $D_F(x, y) \geqslant 0$. (1pt)