Projected Gradient Descent

Soham Kalburgi[1]

[1]f20230460@goa.bits-pilani.ac.in
BITS Pilani, KK Birla Goa Campus

Week 3

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

Last Time: Gradient Descent

- **Goal:** Iterative Approach to Approximate Global Minima
- **Definition:** The step of gradient descent is defined by

$$\mathbf{x}_{t+1} := \mathbf{x} - \gamma \nabla f(\mathbf{x_t}) \tag{1}$$

  where $\gamma > 0$ is a fixed *stepsize*.
- Analysis of Lipschitz, Smooth, Strongly, Smooth & Strongly convex functions

Last Time: Convergence Rates

	Lipschitz convex functions	smooth convex functions	strongly convex functions	smooth & strongly convex functions
gradient descent	Thm. 3.1 $\mathcal{O}(1/\varepsilon^2)$	Thm. 3.8 $\mathcal{O}(1/\varepsilon)$		Thm. 3.14 $\mathcal{O}(\log(1/\varepsilon))$
accelerated gradient descent		Thm. 3.9 $\mathcal{O}(1/\sqrt{\varepsilon})$		
projected gradient descent	Thm. 4.2 $\mathcal{O}(1/\varepsilon^2)$	Thm. 4.4 $\mathcal{O}(1/\varepsilon)$		Thm. 4.5 $\mathcal{O}(\log(1/\varepsilon))$
subgradient descent	Thm. 10.20 $\mathcal{O}(1/\varepsilon^2)$		Thm. 10.22 $\mathcal{O}(1/\varepsilon)$	
stochastic gradient descent	Thm. 12.4 $\mathcal{O}(1/\varepsilon^2)$		Thm. 12.4 $\mathcal{O}(1/\varepsilon)$	

Table 3.1: Results on gradient descent. Below each theorem, the number of steps is given which the respective variant needs on the respective function class to achieve additive approximation error at most ε.

Projected Gradient Descent

Definition

Choose $\mathbf{x}_0 \in X$ *arbitrary and for* $t \geq 0$ *define*

$$\mathbf{y}_{t+1} := \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t) \tag{2}$$

$$\mathbf{x}_{t+1} := \Pi_X(\mathbf{y}_{t+1}) := \operatorname*{argmin}_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{y}_{t+1}\|^2 \tag{3}$$

- After each iteration, we project the obtained iterate $\mathbf{y}_{t+1}$ back to X
- Computing $\Pi_{\mathbf{X}}(\mathbf{y}_{t+1})$ means to solve an auxiliary convex constrained minimization problem in each step[2]

[2] the projection is well defined: $d_{\mathbf{y}}(\mathbf{x}) := \|\mathbf{x} - \mathbf{y}\|^2$ is strongly convex, and hence, a unique minimum over the nonempty closed and convex set X exists

Projected Gradient Descent

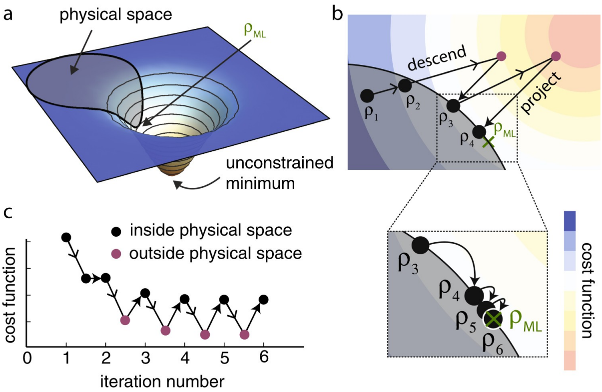

Figure: Illustration of Projected Gradient Descent[3]

[3] image from Bolduc, E., Knee, G.C., Gauger, E.M. *et al.* Projected gradient descent algorithms for quantum state tomography. *npj Quantum Inf* 3, 44 (2017).

Projected Gradient Descent

Fact 1.1

Let $X \subseteq \mathbb{R}^d$ be closed and convex, $\mathbf{x} \in X, \mathbf{y} \in \mathbb{R}^d$. Then

(i)
$$(\mathbf{x} - \Pi_X(\mathbf{y}))^\top (\mathbf{y} - \Pi_X(\mathbf{y})) \leq 0 \tag{4}$$

(ii)
$$\|\mathbf{x} - \Pi_X(\mathbf{y})\|^2 + \|\mathbf{y} - \Pi_X(\mathbf{y})\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2 \tag{5}$$

Equation (4) says that the vectors $\mathbf{x} - \Pi_x(\mathbf{y})$ and $\mathbf{y} - \Pi_X(\mathbf{y})$ form an obtuse angle, and equation (5) equivalently says that the square of the long side $\mathbf{x} - \mathbf{y}$ in the triangle formed by the three points is at least the sum of squares of the two short sides.

Projected Gradient Descent

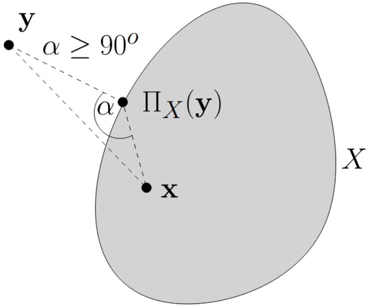

Figure: Illustration of Fact 1.1

What happens when $\mathbf{x}_t = \mathbf{x}_{t+1}$?

- Substitute into equations 2 and 3 to get

$$\mathbf{x}_t = \Pi_X(\mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t)) \tag{6}$$

- This means we project back to the previous iterate
- In this case, $\mathbf{x}_t$ is a *minimizer* of f over the closed and convex set X (f is a convex differentiable function)

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

Analysis: Convergence Rate

Theorem

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex and differentiable, $X \subseteq \mathbf{dom}(f)$ closed and convex, $\mathbf{x}^\star$ a minimizer of f over X; furthermore, suppose that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$, and that $\|\nabla f(\mathbf{x})\| \leq B$ for all $\mathbf{x} \in X$. Choosing the constant stepsize

$$\gamma := \frac{R}{B\sqrt{T}}$$

projected gradient descent with $\mathbf{x}_0 \in X$ yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) \leq \frac{RB}{\sqrt{T}} \tag{7}$$

Analysis: $\mathcal{O}(1/\varepsilon^2)$ steps

- This follows from the analysis of Lipschitz Convex Functions in Gradient Descent with the change being — replace $\mathbf{x}_{t+1}$ by $\mathbf{y}_{t+1}$ as this is the real next (non-projected) gradient descent iterate and then using Fact 1.1(ii) (with $\mathbf{x} = \mathbf{x}^*, \mathbf{y} = \mathbf{y}_{t+1}$)
- In order to achieve $\min_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \varepsilon)$ we need

$$T \geq \frac{R^2 B^2}{\varepsilon^2} \implies \mathcal{O}(1/\varepsilon^2)$$

where T is the number of iterations

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

Smooth Convex Functions

Definition: f is L-smooth over X if

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in X \qquad (8)$$

Analysis: Sufficient Decrease

Lemma

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ *be differentiable and smooth with parameter* L *over a closed and convex set* $X \subseteq \mathbf{dom}(f)$*, according to (8). Choosing stepsize*

$$\gamma := \frac{1}{L}$$

projected gradient descent (2, 3) with arbitrary $\mathbf{x}_0 \in X$ *satisfies*

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{L}{2}\|\mathbf{y}_{t+1} - \mathbf{x}_{t+1}\|^2, \quad t \geq 0 \quad (9)$$

More specifically, this already holds if f *is smooth with parameter* L *over the line segment connecting* $\mathbf{x}_t$ *and* $\mathbf{x}_{t+1}$.

Analysis: Convergence Rate

Theorem

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ *be convex and differentiable. Let* $X \subseteq \mathbf{dom}(f)$ *be a closed convex set, and assume that there is a minimizer* $\mathbf{x}^\star$ *of* f *over* X*; furthermore, suppose that* f *is smooth over* X *with parameter* L*, according to (8). Choosing stepsize*

$$\gamma := \frac{1}{L}$$

projected gradient descent (2, 3) with $\mathbf{x}_0 \in X$ *satisfies*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2, \quad T > 0 \tag{10}$$

Analysis: $\mathcal{O}(1/\varepsilon)$ steps

Implication: With $R^2 := \|\mathbf{x} - \mathbf{x_0}\|^2$ we now only need

$$T \geq \frac{R^2 L}{2\varepsilon} \implies \mathcal{O}(1/\varepsilon)$$

iterations instead of $R^2 B^2/\varepsilon^2$ to achieve absolute error at most ε

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

Strongly Convex Functions

Definition: f is strongly convex with parameter $\mu > 0$ over X if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in X \tag{11}$$

Analysis: Convergence Rate

Theorem

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex and differentiable. Let $X \subseteq \mathbf{dom}(f)$ be a nonempty closed and convex set and suppose that f is smooth over X with parameter L according to (8) and strongly convex over X with parameter $\mu > 0$ according to (11). Choosing

$$\gamma := \frac{1}{L}$$

projected gradient descent (2, 3) with arbitrary $\mathbf{x}_0$ satisfies the following two properties —

Analysis: Convergence Rate

Theorem cont.

(i) Squared distances to $\mathbf{x}^*$ are geometrically decreasing:

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^*\|^2, \quad t \geq 0 \tag{12}$$

(ii) The absolute error after T iterations is exponentially small in T:

$$\begin{aligned} f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \|\nabla f(\mathbf{x}^*)\| \left(1 - \frac{\mu}{L}\right)^{T/2} \|\mathbf{x}_0 - \mathbf{x}^*\| \\ + \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^{T} \|\mathbf{x}_0 - \mathbf{x}^*\|^2, \quad T > 0 \end{aligned} \tag{13}$$

Analysis: $\mathcal{O}(\log(1/\varepsilon))$

- In the constrained case we cannot argue that $\nabla f(\mathbf{x}^*) = 0$, thus the additional term
- The additional term is the dominating one, once the error becomes small
- It has the effect that the required number of steps to reach error at most ε will roughly double compared to the analysis in Gradient Descent $\implies \mathcal{O}(\log(1/\varepsilon))$

Contents[1]

1 The Algorithm

2 Bounded Gradients

3 Smooth Convex Functions

4 Smooth and Strongly Convex Functions

5 Projecting onto ℓ_1-balls

[1] Optimization for Data Science, Lecture Notes, FS23, ETH Zurich

ℓ_1-ball

Let

$$X = B_1(R) := \left\{ \mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i| \leq R \right\} \qquad (14)$$

be the ℓ_1-ball of radius $R > 0$ around $\mathbf{0}$, i.e., the set of all points with 1-norm at most R[4]

[4] geometrically, X is a *cross polytope* (square for $d = 2$, octahedron for $d = 3$), and as such it has 2^d many facets. `https://en.wikipedia.org/wiki/Polytope`

Goal: $\Pi_X(\mathbf{v})$

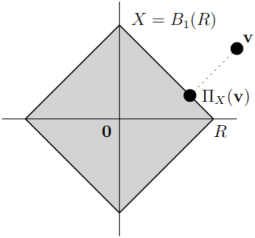

Figure: Projecting onto an ℓ_1-ball

Our goal is to compute $\Pi_X(\mathbf{v})$ for a given vector $\mathbf{v}$, i.e., the projection of $\mathbf{v}$ onto X

Simplifying Observations

Fact 5.1

We may assume WLOG that (i) $R = 1$, (ii) $v_i \geq 0$ for all i, and (iii) $\sum_{i=1}^{d} v_i > 1$

Fact 5.2

Under the assumptions of Fact 5.1, $\mathbf{x} = \Pi_X(\mathbf{v})$ satisfies $x_i \geq 0$ for all i and $\sum_{i=1}^{d} x_i = 1$

Simplifying Observations

Corollary

Under the assumptions of Fact 5.1,

$$\Pi_X(\mathbf{v}) = \operatorname*{argmin}_{x \in \Delta_d} \|\mathbf{x} - \mathbf{v}\|^2$$

where

$$\Delta_d := \left\{ \mathbf{x} \in \mathbb{R}^d : \sum_{i=1}^{d} x_i = 1, x_i \geq 0 \; \forall i \right\}$$

is the standard simplex[5]

[5] https://en.wikipedia.org/wiki/Simplex

Standard Simplex

We have reduced the projection onto an ℓ_1-ball to the projection onto the standard simplex

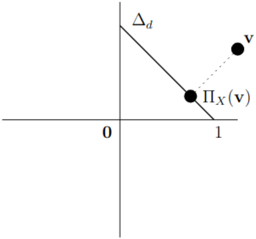

Figure: Projecting onto the standard simplex

Projecting onto the Standard Simplex

Fact 5.3

We may assume WLOG that $v_1 \geq v_2 \geq \cdots \geq v_d$

Lemma 5.1

Let $\mathbf{x}^* := \operatorname{argmin}_{x \in \Delta_d} \|\mathbf{x} - \mathbf{v}\|^2$. *Under the assumption of Fact 5.3, there exists (a unique)* $p \in \{1, \ldots, d\}$ *such that*

$$x_i^* > 0, \quad i \leq p, \tag{15}$$

$$x_i^* = 0, \quad i > p \tag{16}$$

Projecting onto the Standard Simplex

Lemma 5.2

Under the assumption of Fact 5.3, and with p as in Lemma 5.1,

$$x_i^* = v_i - \Theta_p, \quad i \leq p, \tag{17}$$

where

$$\Theta_p = \frac{1}{p}\left(\sum_{i=1}^{p} v_i - 1\right) \tag{18}$$

Projecting onto the Standard Simplex

Summary:

- We have d candidates for $\mathbf{x}^*$, namely the vectors

$$\mathbf{x}^*(p) := (v_1 - \Theta_p, \ldots, v_p - \Theta_p, 0, \ldots, 0), \quad p \in \{1, \ldots, d\} \tag{19}$$

  and we just need to find the right one
- In order for candidate $\mathbf{x}^*(p)$ to comply with Lemma 5.1, we must have

$$v_p - \Theta_p > 0, \tag{20}$$

  and this actually ensures $\mathbf{x}^*(p)_i > 0$ for all $i < p$ by the assumption of Fact 5.3 and therefore $\mathbf{x}^*(p) \in \Delta_d$

Projecting onto the Standard Simplex

- There could still be several values of p satisfying (20)
- Among them, we simply pick the one for which $\mathbf{x}^*(p)$ minimizes the distance to $\mathbf{v}$
- This can be done in $\mathcal{O}(d \log d)$, by first sorting v and then updating the values Θ_p and $\|\mathbf{x}^*(p) - \mathbf{v}\|^2$ as we vary p to check all candidates

Projecting onto the Standard Simplex

There is a simpler criterion that saves us from comparing distances

Lemma 5.3

Under the assumption of Fact 5.3, with $\mathbf{x}^*(p)$, *and with*

$$p^* := \max\{p \in \{1, \ldots, d\} : v_p - \frac{1}{p}\left(\sum_{i=1}^{p} v_i - 1\right) > 0\}, \tag{21}$$

it holds that

$$\operatorname*{argmin}_{x \in \Delta_d} \|\mathbf{x} - \mathbf{v}\|^2 = \mathbf{x}^*(p^*) \tag{22}$$

Projecting onto ℓ_1-balls

Theorem

Let $\mathbf{v} \in \mathbb{R}^d$, $R \in \mathbb{R}_+$, $X = B_1(R)$ *the* ℓ_1*-ball around* $\mathbf{0}$ *of radius* R*. The projection*

$$\Pi_X(\mathbf{v}) = \operatorname*{argmin}_{x \in X} \|\mathbf{x} - \mathbf{v}\|^2 \tag{23}$$

of $\mathbf{v}$ *onto* $B_1(R)$ *can be computed in time* $\mathcal{O}(d \log d)$*.*

This can be improved to time $\mathcal{O}(d)$, based on the observation that a given p can be compared to the value p^* in Lemma 5.3 in linear time, without the need to presort v

Summary

Thank you!