COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

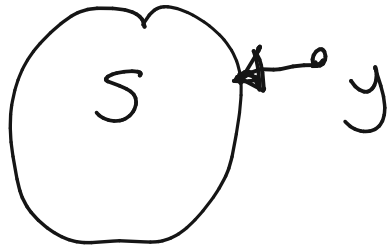
University of Massachusetts Amherst. Fall 2020.

Lecture 24

LOGISTICS

- Problem Set 4 is due tomorrow at 8pm.
- Optional Problem Set 5 will be released tomorrow, due 11/30.
- Exam will span December 3-4. Any two hour period.
- Exam review guide, practice problems, logistical details have been posted under the schedule tab on the course page.
- I am holding an optional SRTI (course reviews) for this class and would really appreciate your feedback (closes Dec 6).
- `http://owl.umass.edu/partners/courseEvalSurvey/uma/`.

SUMMARY



Lipschitz.

Last Class:

- Analysis of gradient descent for optimizing convex functions.
- Introduction to convex sets and projection functions.
- (The same) analysis of projected gradient descent for optimizing under convex functions under (convex) constraints.

$0 \in S$

This Class:

- Online learning, regret, and online gradient descent.
- Application to stochastic gradient descent.

QUIZ REVIEW

$2x + 3y$

$\mathbb{R}^d \to \mathbb{R}$

Consider the function $f(\vec{\theta}) = \vec{x}^T\vec{\theta}$ for $x = [1, -1, -2]$. Give the minimum value of G such that $f(\vec{\theta})$ is G-Lipschitz

$\| \vec{\nabla} f(\vec{\theta}) \|_2 \leq G \quad \forall \quad \theta$

$\nabla f(\vec{\theta}) = \vec{x}$

$= \begin{bmatrix} \frac{\partial f(\theta)}{\partial \theta(1)} \\ \frac{\partial f(\theta)}{\partial \theta(2)} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ \vdots \\ x(d) \end{bmatrix}$

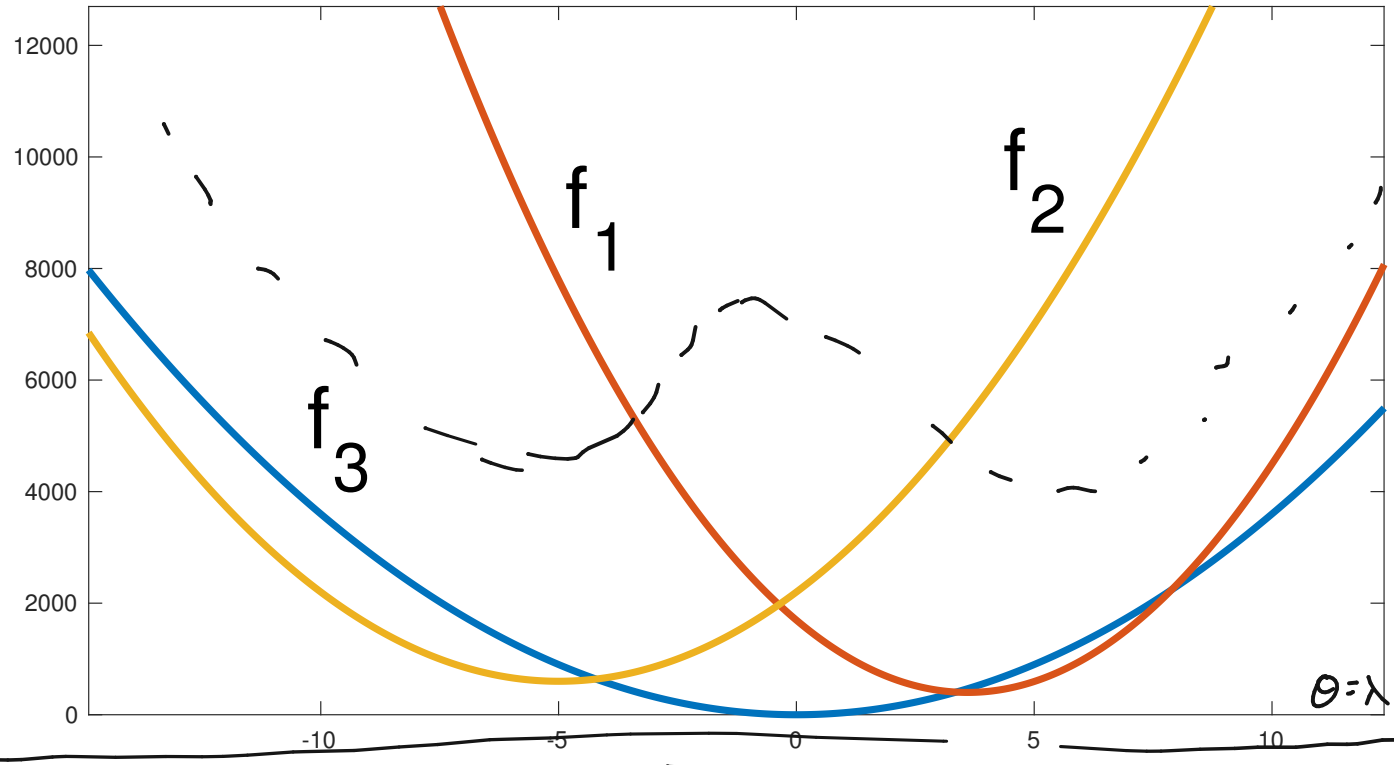
$f(\theta) = x^T\theta = \sum_{i=1}^{d} x(i) \cdot \theta(i)$

$\frac{\partial f(\theta)}{\partial \theta(i)} = x(i)$

$\Rightarrow \| \vec{\nabla} f(\theta) \|_2 = \| \vec{x} \| = \sqrt{6}$

TEST OF INTUITION

What does $f_1(\theta) + f_2(\theta) + f_3(\theta)$ look like?



$\theta_1, \theta_2 \in S_2$, $\lambda \in [0,1]$ s.t. $\lambda\theta_1 + (1-\lambda)\theta_2 \notin S$

$S_1 =$

$S_2 =$

$S_3 =$

$\theta_1 \in S_3$, $\theta_2 \in S_3$

$\theta_1(10) \geq 2$, $\theta_2(10) \geq 2$

$\theta = \lambda\theta_1 + (1-\lambda)\theta_2$

$\theta(10) \geq 2$

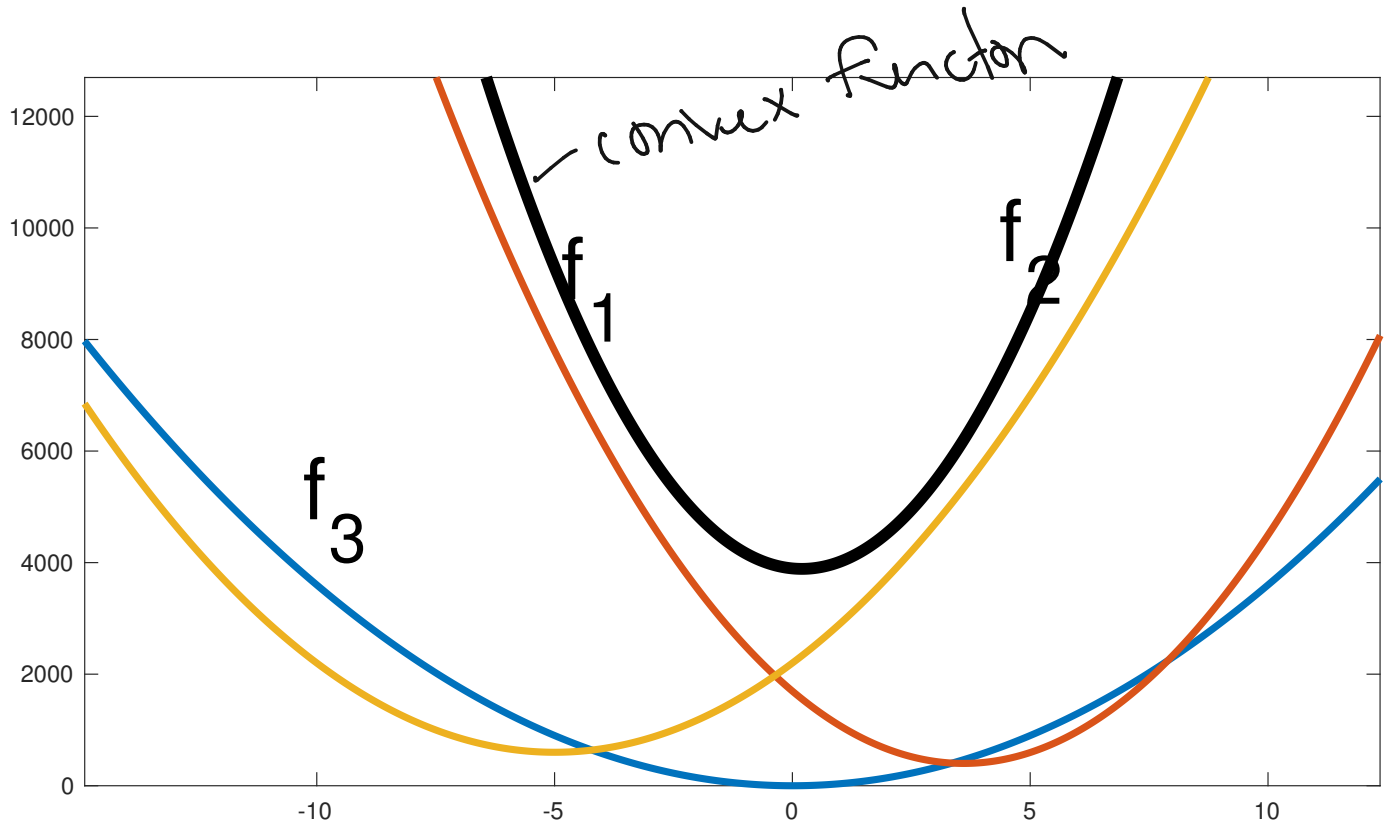
Q2: $\surd S_1: \{x : \|x\|_2 \leq 2\}$ $\surd S_3: \{x : x(10) \geq 2\}$

$\times S_2: \{x : \|x\|_2 \geq 2\}$

$\theta_1 = [10, 0]$, $\theta_2 = [-10, 0]$, $\lambda = 1/2$ $\lambda\theta_1 + (1-\lambda)\theta_2 = 0$

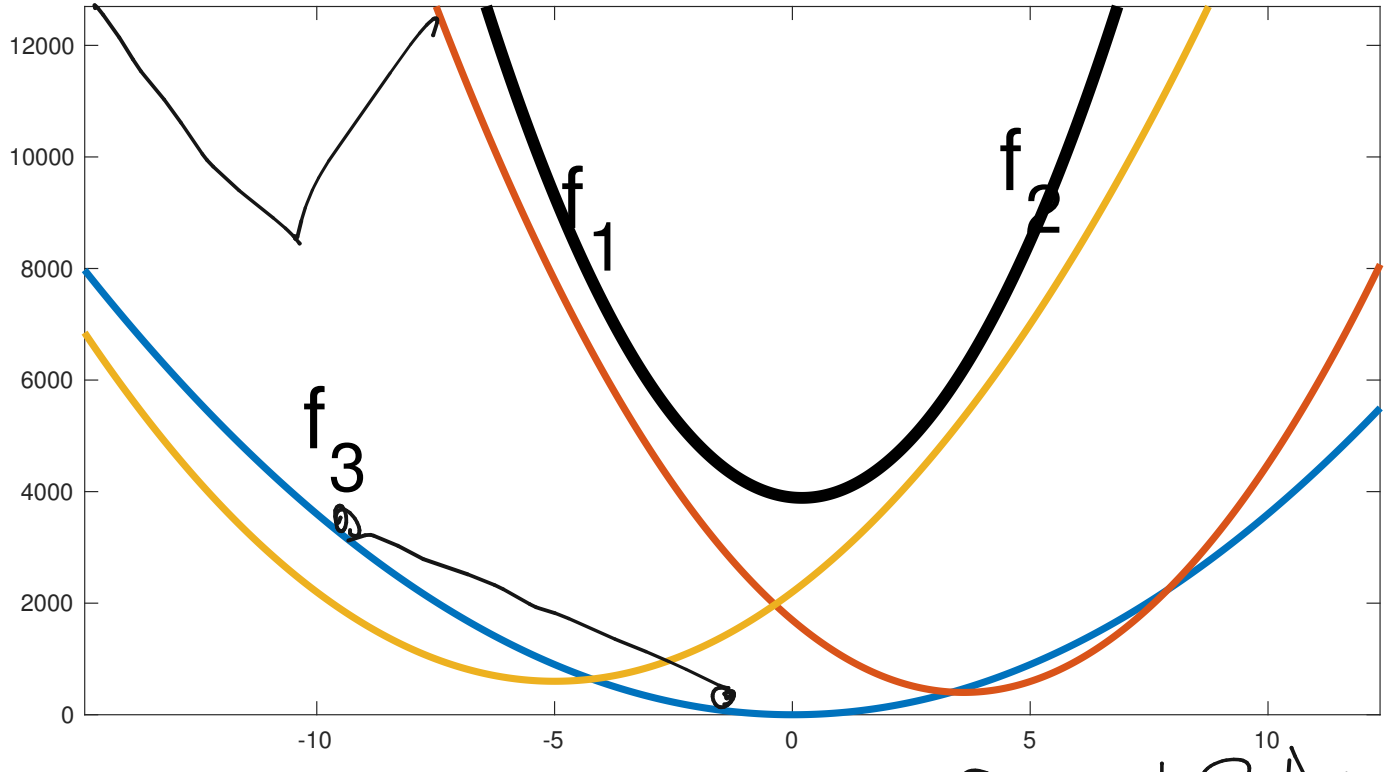
TEST OF INTUITION

What does $f_1(\theta) + f_2(\theta) + f_3(\theta)$ look like?



TEST OF INTUITION

What does $f_1(\theta) + f_2(\theta) + f_3(\theta)$ look like?



A sum of convex functions is always convex (good exercise).

ONLINE GRADIENT DESCENT

In reality many learning problems are online.

- Websites optimize ads or recommendations to show users, given continuous feedback from these users.
- Spam filters are incrementally updated and adapt as they see more examples of spam over time.
- Face recognition systems, other classification systems, learn from mistakes over time.

ONLINE GRADIENT DESCENT

In reality many learning problems are online.

- Websites optimize ads or recommendations to show users, given continuous feedback from these users.
- Spam filters are incrementally updated and adapt as they see more examples of spam over time.
- Face recognition systems, other classification systems, learn from mistakes over time.

Want to minimize some global loss $L(\vec{\theta}, \mathbf{X}) = \sum_{i=1}^{n} \ell(\vec{\theta}, \vec{x}_i)$, when data points are presented in an online fashion $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ (similar to streaming algorithms)

ONLINE GRADIENT DESCENT

In reality many learning problems are online.

- Websites optimize ads or recommendations to show users, given continuous feedback from these users.
- Spam filters are incrementally updated and adapt as they see more examples of spam over time.
- Face recognition systems, other classification systems, learn from mistakes over time.

Want to minimize some global loss $L(\vec{\theta}, \mathbf{X}) = \sum_{i=1}^{n} \ell(\vec{\theta}, \vec{x}_i)$, when data points are presented in an online fashion $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ (similar to streaming algorithms)

Stochastic gradient descent is a special case: when data points are considered a random order for computational reasons.

Online Optimization: In place of a single function f, we see a different objective function at each step:

$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \to \mathbb{R}$$

ONLINE OPTIMIZATION FORMAL SETUP

Online Optimization: In place of a single function f, we see a different objective function at each step:

$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \to \mathbb{R}$$

$\theta^{(1)}$ $\theta^{(2)}$ .. $\theta^{(t)}$

- At each step, first pick (play) a parameter vector $\vec{\theta}^{(i)}$.
- Then are told f_i and incur cost $f_i(\vec{\theta}^{(i)})$.
- **Goal:** Minimize total cost $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$.

ONLINE OPTIMIZATION FORMAL SETUP

Online Optimization: In place of a single function f, we see a different objective function at each step:

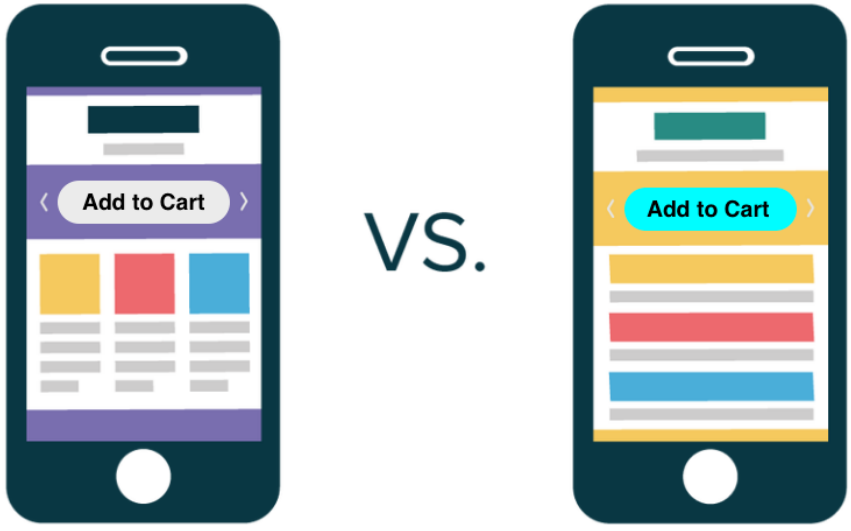
$$f_1, f_2, \ldots, f_t : \mathbb{R}^d \to \mathbb{R}$$

- At each step, first pick (play) a parameter vector $\vec{\theta}^{(i)}$.
- Then are told f_i and incur cost $f_i(\vec{\theta}^{(i)})$.
- **Goal:** Minimize total cost $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$.

Our analysis will make no assumptions on how $f_1, \ldots, f_t$ are related to each other!

ONLINE OPTIMIZATION EXAMPLE

UI design via online optimization.



- Parameter vector $\vec{\theta}^{(i)}$: some encoding of the layout at step i.
- Functions $f_1, \ldots, f_t$: $f_i(\vec{\theta}^{(i)}) = 1$ if user does not click 'add to cart' and $f_i(\vec{\theta}^{(i)}) = 0$ if they do click.
- Want to maximize number of purchases. I.e., minimize $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$.

ONLINE OPTIMIZATION EXAMPLE

Home pricing tools.

$\vec{x} = [\#baths, \#beds, \#floors \ldots]$

linear model

$\langle \vec{x}, \vec{\theta} \rangle$

$275,000

- Parameter vector $\vec{\theta}^{(i)}$: coefficients of linear model at step i.
- Functions $f_1, \ldots, f_t$: $f_i(\vec{\theta}^{(i)}) = (\langle \vec{x}_i, \vec{\theta}^{(i)} \rangle - price_i)^2$ revealed when $home_i$ is listed or sold.
- Want to minimize total squared error $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)})$ (same as classic least squares regression).

REGRET

In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\vec{\theta}} f(\vec{\theta}) + \epsilon.$$

$\leq f(\theta^*) + \epsilon$

REGRET

In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\vec{\theta}} f(\vec{\theta}) + \epsilon.$$

In online optimization we will ask for the same.

$$\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \min_{\vec{\theta}} \sum_{i=1}^{t} f_i(\vec{\theta}) + \epsilon = \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$$

ϵ is called the regret.

In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\vec{\theta}} f(\vec{\theta}) + \epsilon.$$

In online optimization we will ask for the same.

$$\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \min_{\vec{\theta}} \sum_{i=1}^{t} f_i(\vec{\theta}) + \epsilon = \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$$

ϵ is called the regret.

- This error metric is a bit 'unfair'. Why?

- we haven't seen D about of the

REGRET

In normal optimization, we seek $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq \min_{\vec{\theta}} f(\vec{\theta}) + \epsilon.$$

In online optimization we will ask for the same.

$$\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \min_{\vec{\theta}} \sum_{i=1}^{t} f_i(\vec{\theta}) + \epsilon = \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$$

ϵ is called the regret.

- This error metric is a bit 'unfair'. Why?
- Comparing online solution to best fixed solution in hindsight. ϵ can be negative!

INTUITION CHECK

$f_1(\theta) = |\theta - 1000|$ $\quad f_2(\theta) = |\theta + 1000|$, ...

What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in an alternating pattern?

How small can the regret ϵ be? $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$. ~ 0

$\theta = 0$

$\min_{\theta} \sum_{i=1}^{t} f_i(\theta) = \sum_{i=1}^{t} f_i(0) = 1000t.$

$\theta_{10} = -1000 \quad \theta_{11} = +1000 \quad \theta_{12} = -1000$

$f_{10}(\theta_{10}) = 0 \quad f_{11}(\theta_{11}) = 0 \quad \ldots$

INTUITION CHECK

$f_1(\theta) = |\theta - 1000|$ $f_2(\theta) = |\theta + 1000|$ $f_3(\theta) = |\theta + 1000|$

What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in an alternating pattern?

$\theta^{off} = 0$

How small can the regret ϵ be? $\sum_{i=1}^{t} f_i(\vec{\theta}^{(i)}) \leq \sum_{i=1}^{t} f_i(\vec{\theta}^{off}) + \epsilon$.

$\approx 1000t$ $1000t$

What if for $i = 1, \ldots, t$, $f_i(\theta) = |\theta - 1000|$ or $f_i(\theta) = |\theta + 1000|$ in no particular pattern? How can any online learning algorithm hope to achieve small regret?

θ_i f_i

$\| \quad 0$

$\epsilon \geq 0$

but ϵ small.

Assume that:

- $f_1, \ldots, f_t$ are all convex.
- Each f_i is G-Lipschitz (i.e., $\|\vec{\nabla} f_i(\vec{\theta})\|_2 \leq G$ for all $\vec{\theta}$.)
  - $\|\vec{\theta}^{(1)} - \vec{\theta}^{off}\|_2 \leq R$ where $\theta^{(1)}$ is the first vector chosen.

$\theta^{(1)} = 0$

ONLINE GRADIENT DESCENT

Assume that:

- $f_1, \ldots, f_t$ are all convex.
- Each f_i is G-Lipschitz (i.e., $\|\vec{\nabla} f_i(\vec{\theta})\|_2 \leq G$ for all $\vec{\theta}$.)
- $\|\vec{\theta}^{(1)} - \vec{\theta}^{off}\|_2 \leq R$ where $\theta^{(1)}$ is the first vector chosen.

Online Gradient Descent

- Pick some initial $\vec{\theta}^{(1)}$.
- Set step size $\eta = \frac{R}{G\sqrt{t}}$.
- For $i = 1, \ldots, t$
  - Play $\vec{\theta}^{(i)}$ and incur cost $f_i(\vec{\theta}^{(i)})$.
  - $\vec{\theta}^{(i+1)} = \vec{\theta}^{(i)} - \eta \cdot \vec{\nabla} f_i(\vec{\theta}^{(i)})$

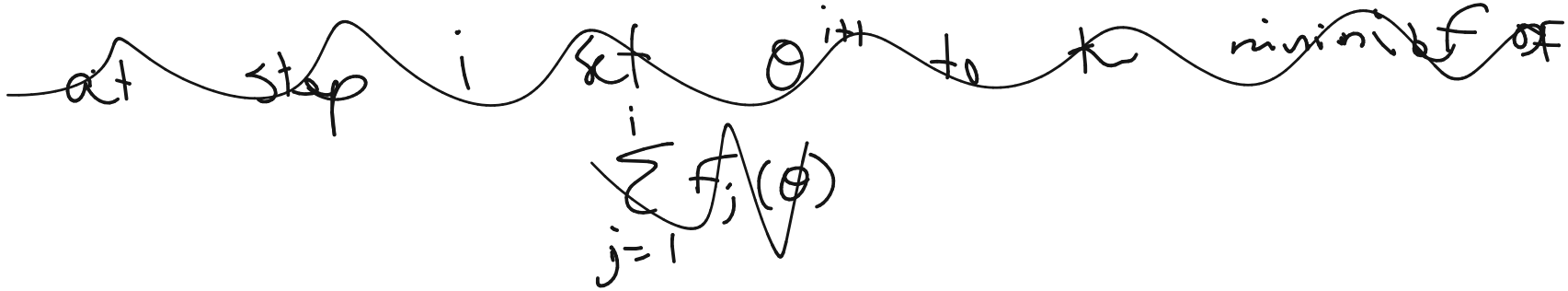
Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius R of θ^{off}, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\frac{1}{t}\left[\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{off})\right] \leq \frac{RG\sqrt{t}}{t} = \frac{RG}{\sqrt{t}}$$

Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius R of θ^{off}, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\frac{1}{t}\left[\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{off})\right] \leq \frac{RG\sqrt{t}}{t} = \frac{RG}{\sqrt{t}}$$

Upper bound on average regret goes to 0 and $t \to \infty$.



Theorem – OGD on Convex Lipschitz Functions: For convex G-Lipschitz $f_1, \ldots, f_t$, OGD initialized with starting point $\theta^{(1)}$ within radius R of θ^{off}, using step size $\eta = \frac{R}{G\sqrt{t}}$, has regret bounded by:

$$\left[\sum_{i=1}^{t} f_i(\theta^{(i)}) - \sum_{i=1}^{t} f_i(\theta^{off})\right] \leq RG\sqrt{t}$$

Upper bound on average regret goes to 0 and $t \to \infty$. No assumptions on $f_1, \ldots, f_t$!