

Lecture 16: Gradient descent – modifications

Ciaran Evans

Recall: gradient descent

- ▶ **Goal:** Minimize $f(\mathbf{x})$
- ▶ Iterative updates: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$
- ▶ Choosing step size α_k :
 - ▶ one option is backtracking line search with sufficient decrease condition

Limitations of gradient descent

Motivating example: Data on med school admissions for 55 students

- ▶ GPA: student's undergraduate GPA
- ▶ MCAT: student's MCAT score

Function to minimize:

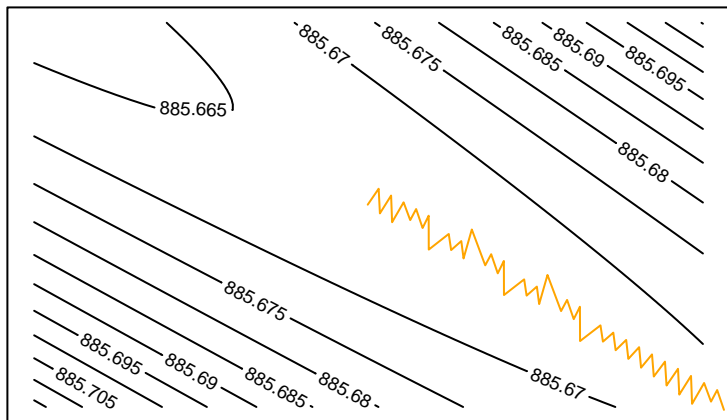
$$f(\beta_0, \beta_1) = \sum_{i=1}^n (\text{MCAT}_i - \beta_0 - \beta_1 \text{GPA}_i)^2$$

- ▶ Gradient descent with backtracking linear search beginning at (0, 0): 6517 iterations

Question: from the activity last time, why does gradient descent need so many iterations here?

Limitations of gradient descent

Gradient descent struggles to traverse long, narrow valleys



Question: What do you notice about the gradient descent path?

Limitations of gradient descent

We should expect gradient descent to have a zig-zag path; this makes long, narrow valleys slow.

- ▶ $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$
- ▶ Suppose α_k is chosen to minimize

$$f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}))$$

Claim: $\nabla f(\mathbf{x}^{(k+1)})$ is **orthogonal** (perpendicular) to $\nabla f(\mathbf{x}^{(k)})$

Overview: modifications of gradient descent

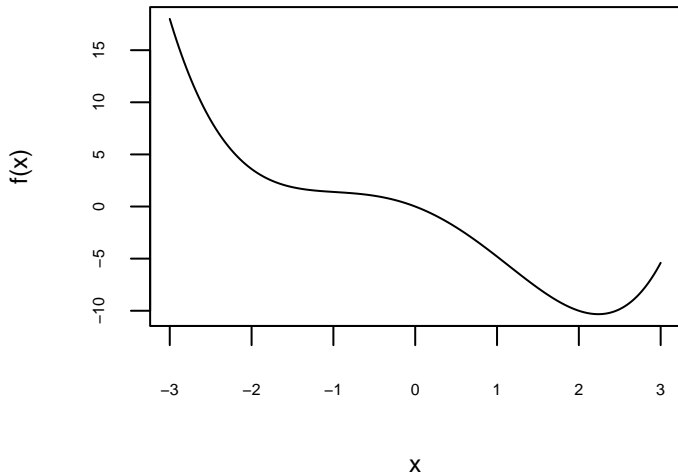
Basic idea: don't always move in the direction of steepest descent

- ▶ Momentum methods: direction of movement is combination of current gradient and previous gradients
- ▶ Subgradient methods: step size is different for each entry in gradient vector
- ▶ Second-order methods: use information about curvature of function (second derivative), not just gradient

It would be impossible to cover all of the possible modifications. My goal is to give you an introduction to some of the common ideas, and their motivation.

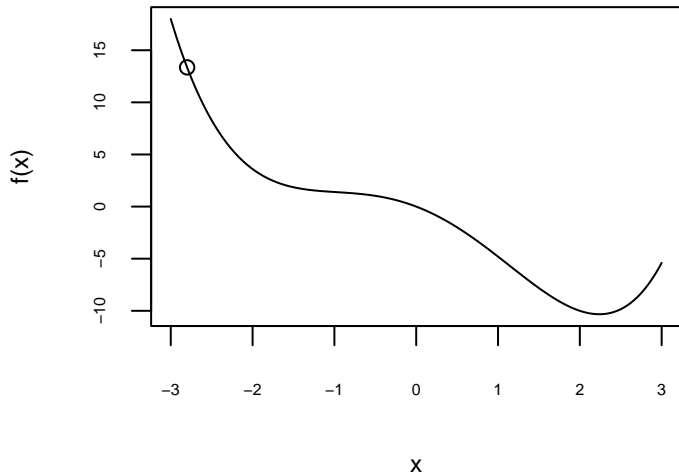
Momentum: motivation

Suppose we wish to minimize this function:



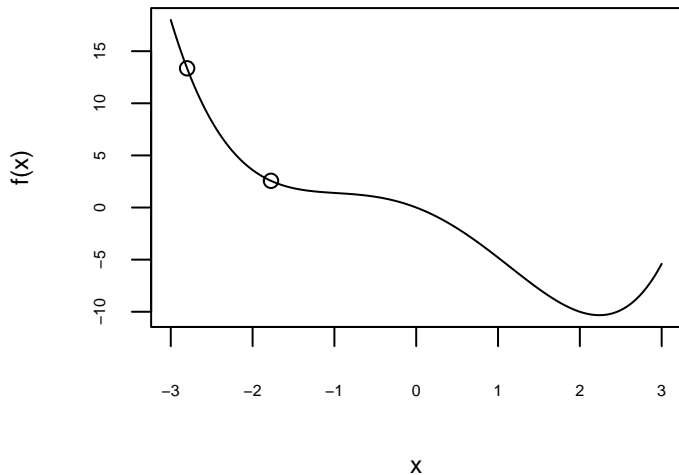
Momentum: motivation

We start at $x^{(0)} = -2.8$:



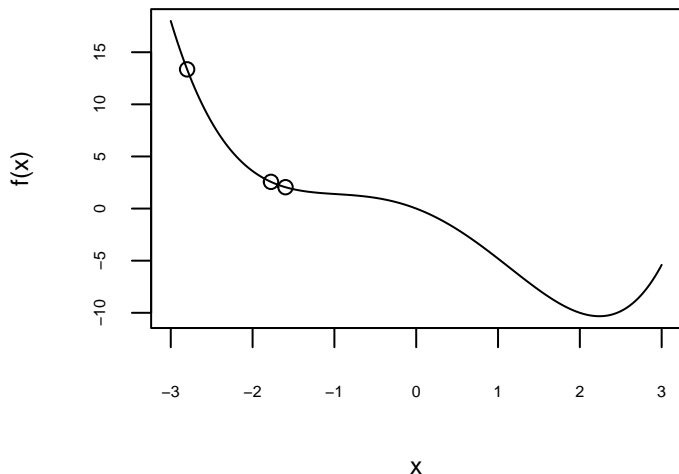
Momentum: motivation

And now we perform gradient descent. After the first iteration:



Momentum: motivation

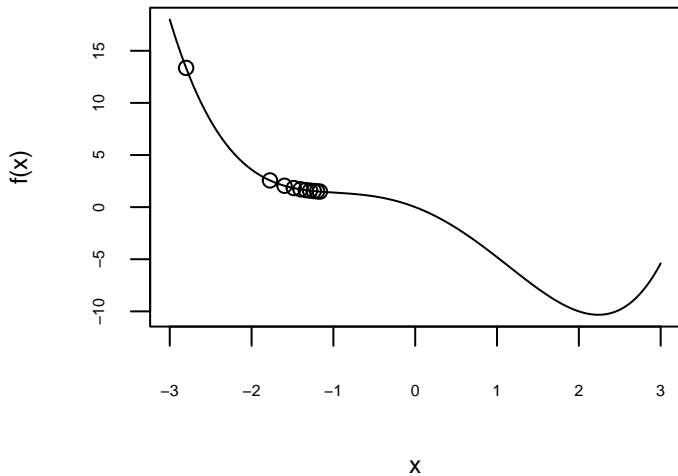
And now we perform gradient descent. After the second iteration:



Question: Why is the second step smaller than the first step?

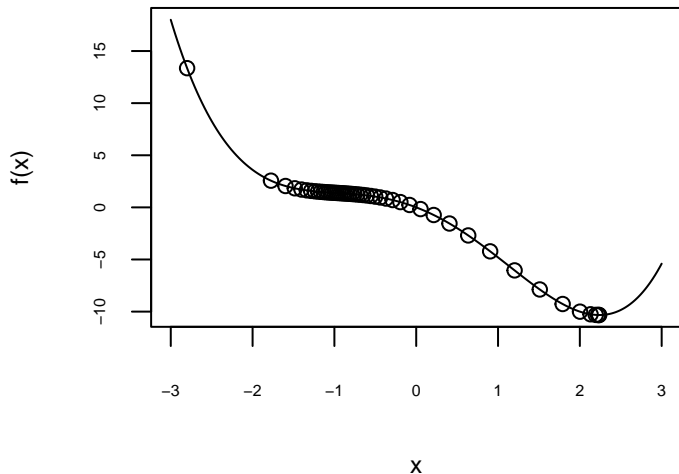
Momentum: motivation

Several more iterations:



Momentum: motivation

Finally, after 50 iterations:



Question: At which part of this function is gradient descent slowest?

Momentum

Gradient descent can be slow in flat areas. Idea: use the momentum from previous gradients.

- ▶ Let $0 \leq \beta < 1$
- ▶ Update rule:
 - ▶ $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$
 - ▶ $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)}) - \beta \alpha \nabla f(\mathbf{x}^{(0)})$
 - ▶ $\mathbf{x}^{(3)} =$

Momentum

Gradient descent can be slow in flat areas. Idea: use the momentum from previous gradients.

► Let $0 \leq \beta < 1$

► Update rule:

► $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$

► $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)}) - \beta \alpha \nabla f(\mathbf{x}^{(0)})$

► $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} - \alpha \nabla f(\mathbf{x}^{(2)}) - \beta \alpha \nabla f(\mathbf{x}^{(1)}) - \beta^2 \alpha \nabla f(\mathbf{x}^{(0)})$

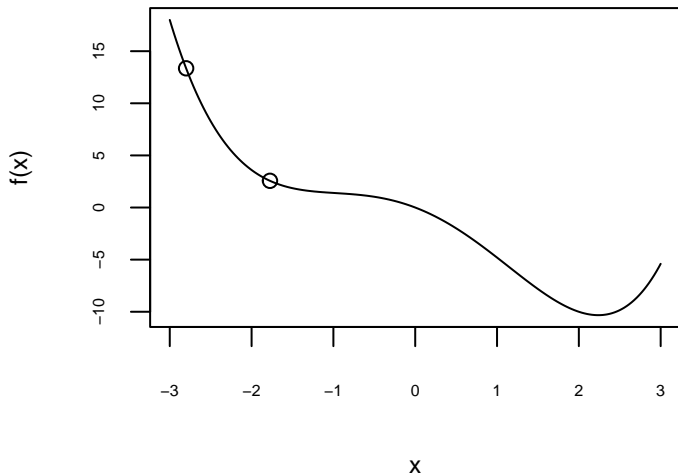
Written another way:

Momentum

- ▶ Let $0 \leq \beta < 1$
- ▶ $\mathbf{v}^{(0)} = -\alpha \nabla f(\mathbf{x}^{(0)})$
- ▶ $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{v}^{(0)}$
- ▶ $\mathbf{x}^{(k)} = -\alpha \nabla f(\mathbf{x}^{(k)}) + \beta \mathbf{v}^{(k-1)}$
- ▶ $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}$

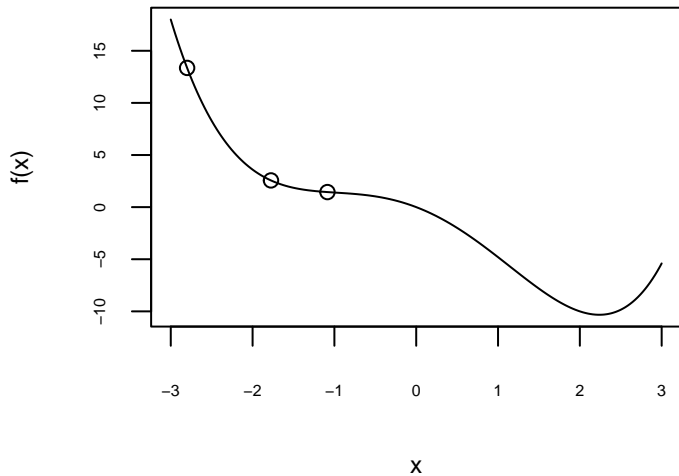
Momentum in action

Starting again at $x^{(0)} = -2.8$. After the first iteration:



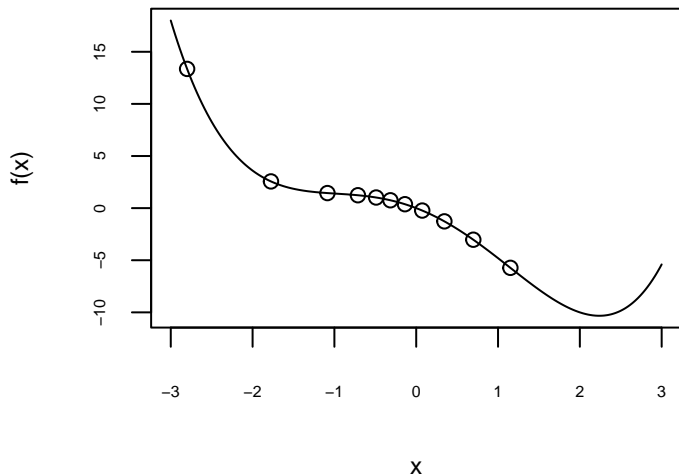
Momentum in action

After the two iterations:



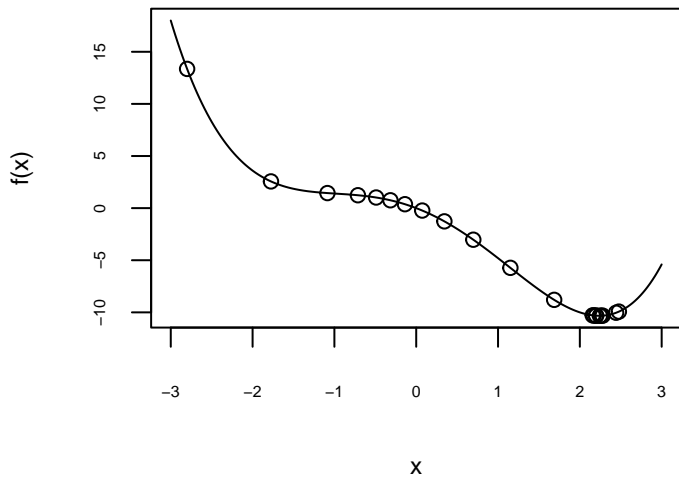
Momentum in action

After 10 iterations:



Momentum in action

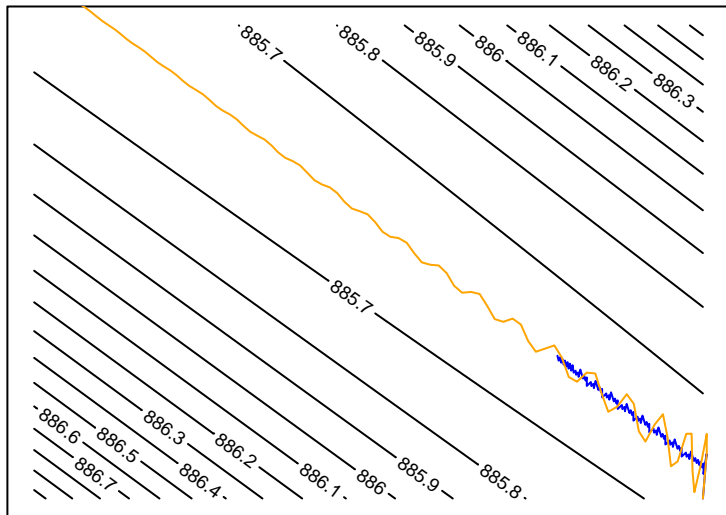
After 20 iterations:



Momentum in action

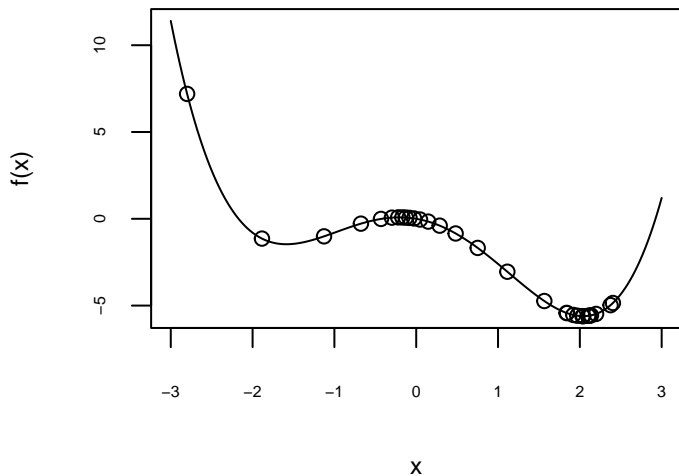
Blue: 100 iterations of gradient descent

Orange: 100 iterations of descent with momentum



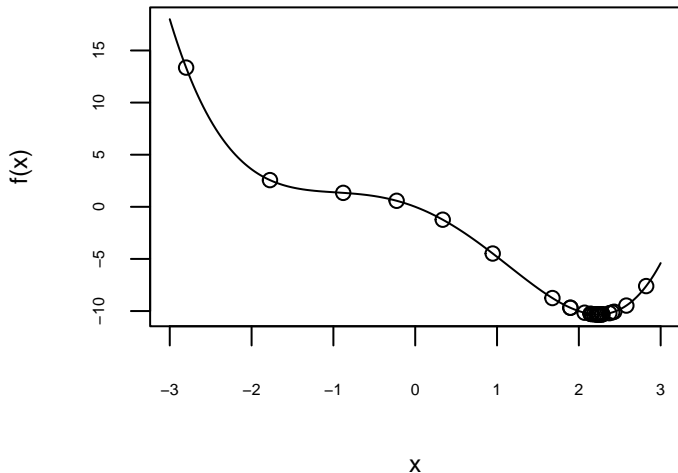
Another example

Momentum can also help overcome local minima:



Drawback of momentum

Question: What do you notice about the iterations near the minimum?



Nesterov momentum

“Heavy ball” momentum:

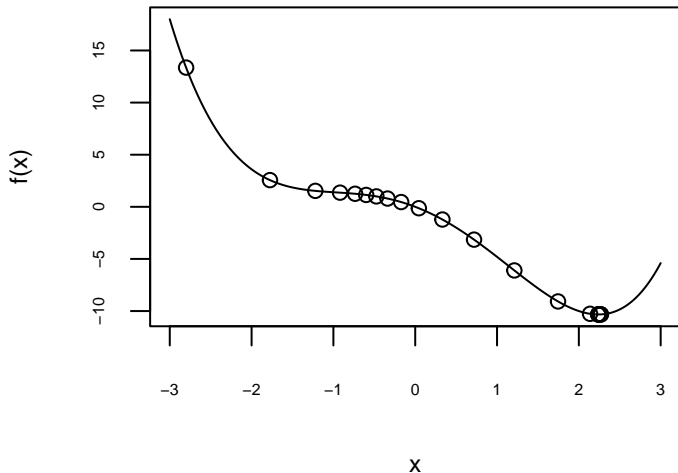
- ▶ $\mathbf{v}^{(k)} = -\alpha \nabla f(\mathbf{x}^{(k)}) + \beta \mathbf{v}^{(k-1)}$
- ▶ $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}$

Nesterov momentum:

- ▶ $\mathbf{v}^{(k)} = -\alpha \nabla f(\mathbf{x}^{(k)} + \beta \mathbf{v}^{(k-1)}) + \beta \mathbf{v}^{(k-1)}$
- ▶ $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}$

Nesterov momentum in action

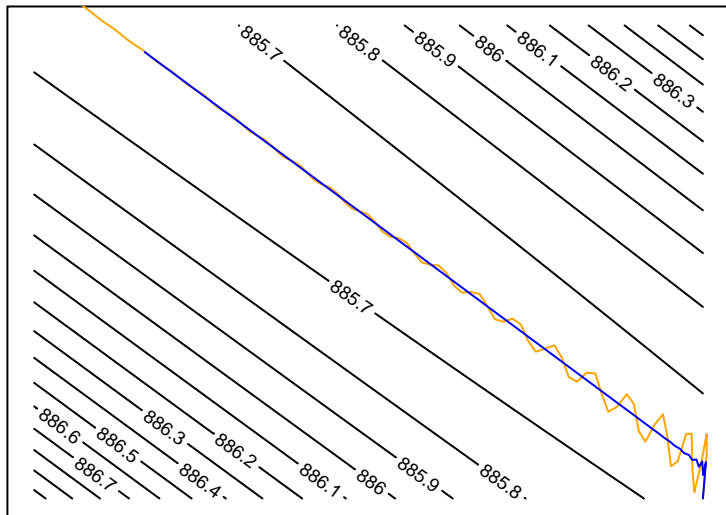
Descent with Nesterov momentum can slow itself down when it gets to the bottom:



Nesterov momentum in action

Blue: Nesterov momentum

Orange: Heavy ball momentum



Subgradient methods

Gradient descent: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$

- ▶ The same step size α_k is applied to each element of the gradient $\nabla f(\mathbf{x}^{(k)})$
- ▶ Example gradient from the MedGPA example:

$$\nabla f((4.1, 9.3)) = \begin{pmatrix} 146.1 \\ 521.7 \end{pmatrix}$$

Question: Why might we not want to use the same value of α_k for each element of the gradient?

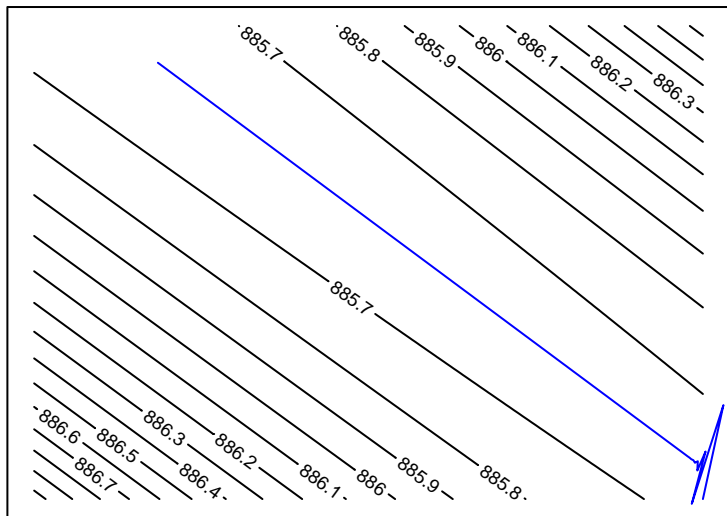
Adagrad

Adagrad (*adaptive subgradient*) uses a different step size for each element of the gradient:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\alpha}{\varepsilon + \sqrt{s_i^{(k)}}} \nabla f(\mathbf{x}^{(k)})_i$$

$$s_i^{(k)} = \sum_{j=1}^k (\nabla f(\mathbf{x}^{(j)})_i)^2$$

Adagrad



Your turn

Practice questions on the course website:

https://sta379-s25.github.io/practice_questions/pq_16.html

- ▶ Try heavy ball momentum
- ▶ Start in class. You are welcome to work with others
- ▶ Practice questions are to help you practice. They are not submitted and not graded
- ▶ Solutions are posted on the course website