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:

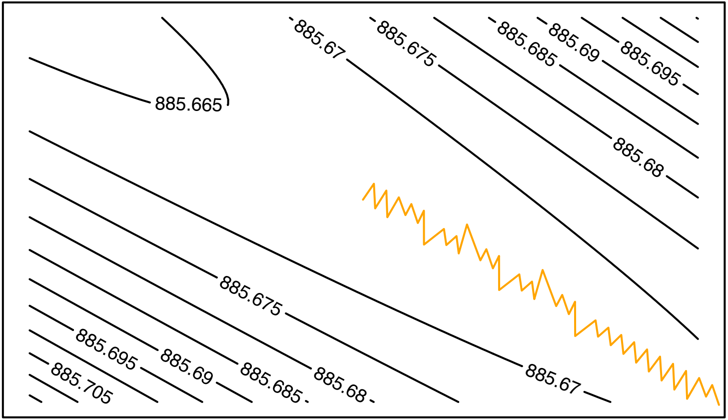
$$f(\beta_0, \beta_1) = \sum_{i=1}^{n} (\mathrm{MCAT}_i - \beta_0 - \beta_1 \mathrm{GPA}_i)^2$$

- 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

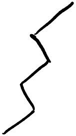


Question: What do you notice about the gradient descent path?

Zig-Zags!

Limitations of gradient descent

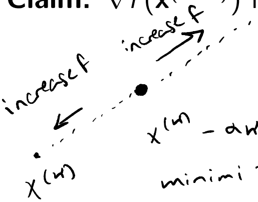
We should expect gradient descent to have a zig-zag path; this makes long, narrow valleys slow.



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

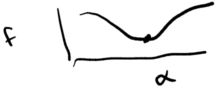
$$f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}))$$

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



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

minimize f in direction of $-\nabla f$



i.e. $x^{(k)} - \alpha_k \nabla f(x^{(k)})$ minimizes f along direction $-\nabla f(x^{(k)})$

$\Rightarrow D_{-\nabla f(x^{(k)})} f(x^{(k+1)}) = 0$

$\Rightarrow \nabla f(x^{(k+1)})^T \nabla f(x^{(k)}) = 0$

$\Rightarrow \nabla f(x^{(k+1)}) \perp \nabla f(x^{(k)})$

Overview: modifications of gradient descent

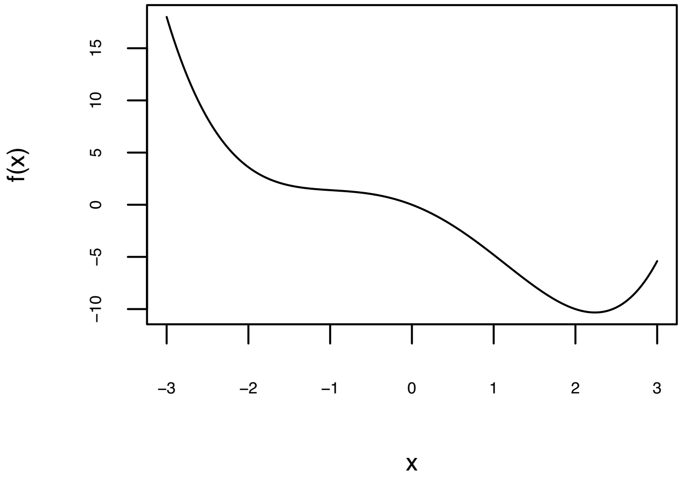
Basic idea: don't always move in the direction of steepest descent

- 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.

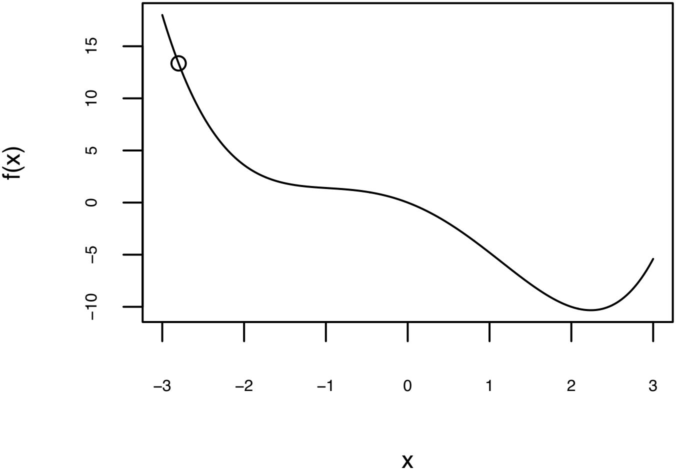
Momentum: motivation

Suppose we wish to minimize this function:



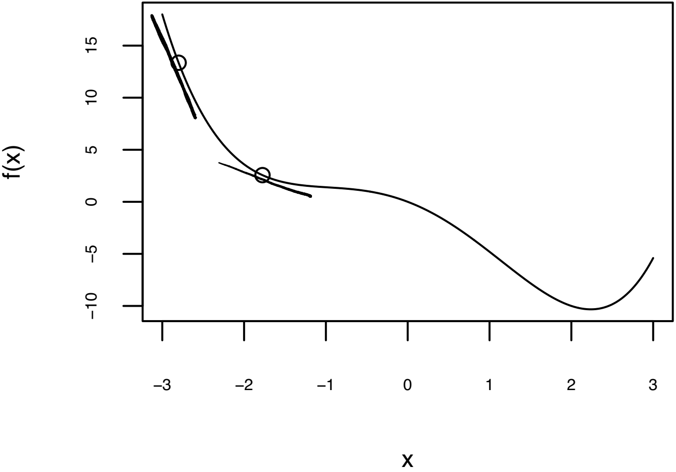
Momentum: motivation

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



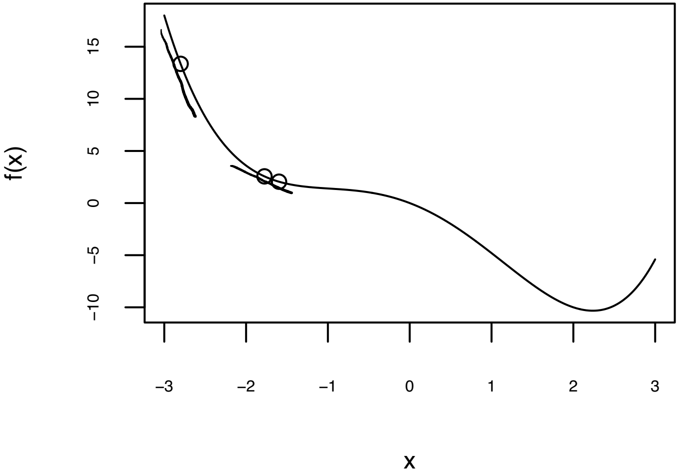
Momentum: motivation

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



Momentum: motivation

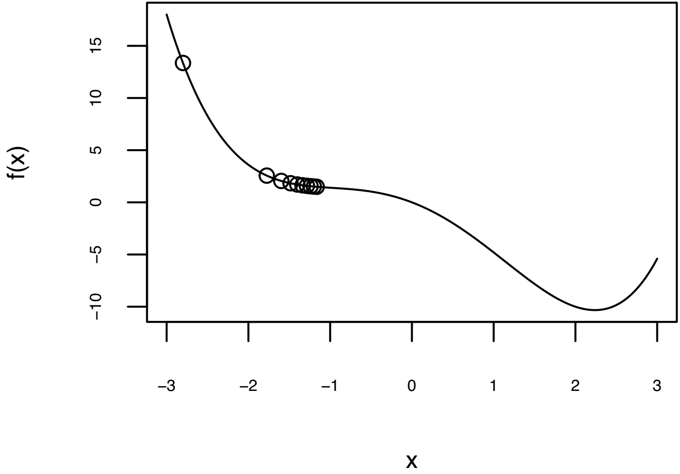
And now we perform gradient descent. After the second iteration:



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

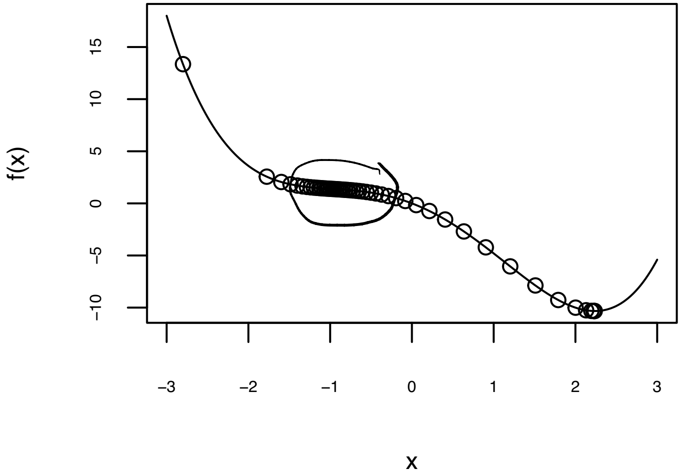
Momentum: motivation

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 $0 \leq \beta < 1$ (weight on previous gradients)
- Update rule:
  - $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$

    some weight on previous movement

  - $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)}) - \beta \underbrace{\alpha \nabla f(\mathbf{x}^{(0)})}_{\text{previous}}$

    previous direction of movement

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

    decaying weight on previous gradients

Momentum

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

- Let $0 \leq \beta < 1$

  $v^{(0)}$ ← movement at first step

- Update rule:
  - $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} \overbrace{- \alpha \nabla f(\mathbf{x}^{(0)})}^{v^{(0)}} = x^{(0)} + v^{(0)}$
  - $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} \underbrace{- \alpha \nabla f(\mathbf{x}^{(1)}) - \overbrace{\beta \alpha \nabla f(\mathbf{x}^{(0)})}^{\beta v^{(0)}}}_{v^{(1)}} = x^{(1)} + v^{(1)}$ ↙ movement at second step
  - $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} - \underbrace{\alpha \nabla f(\mathbf{x}^{(2)}) - \underbrace{\beta \alpha \nabla f(\mathbf{x}^{(1)}) - \beta^2 \alpha \nabla f(\mathbf{x}^{(0)})}_{\beta v^{(1)}}}_{v^{(2)}}$

Written another way:

$$x^{(k+1)} = x^{(k)} + v^{(k)}$$ ← current movement

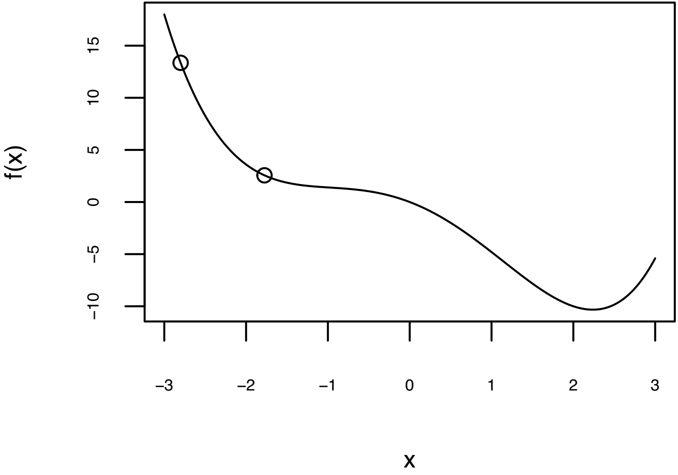
$$v^{(k)} = -\alpha \nabla f(x^{(k)}) + \beta v^{(k-1)}$$ ↖ previous movement

Momentum

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

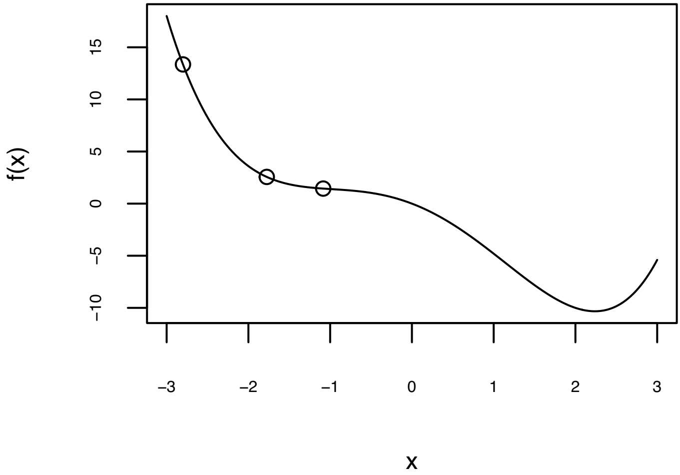
Momentum in action

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



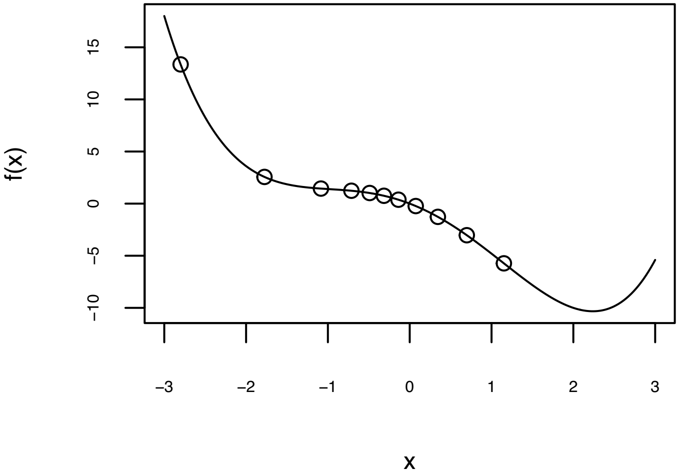
Momentum in action

After the two iterations:



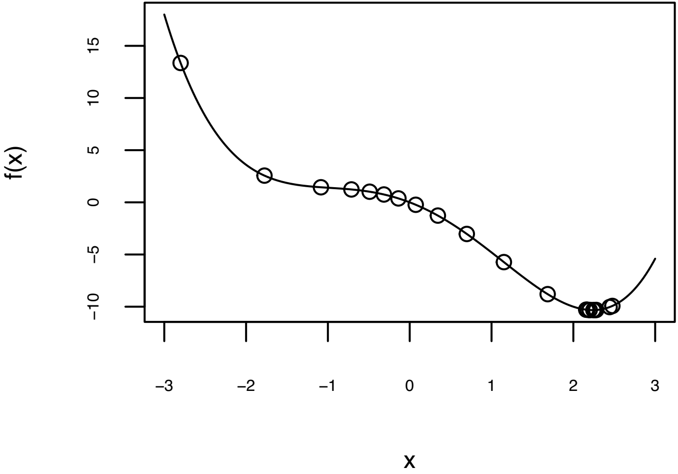
Momentum in action

After 10 iterations:



Momentum in action

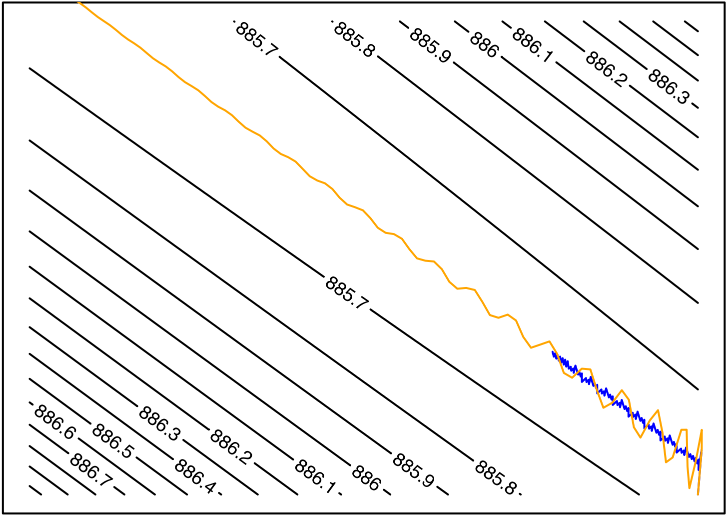
After 20 iterations:



Momentum in action

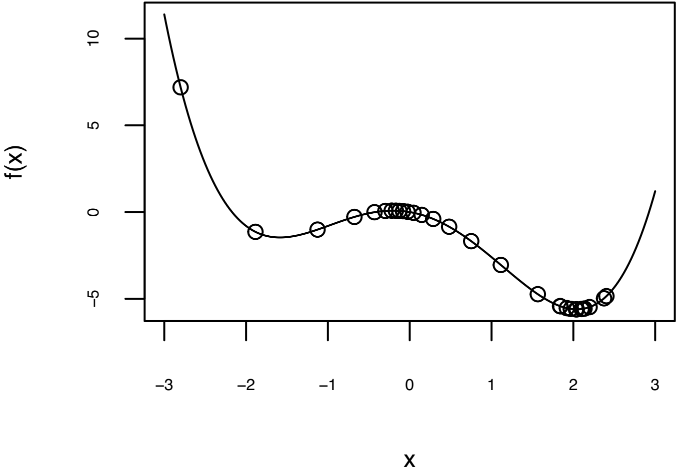
Blue: 100 iterations of gradient descent

Orange: 100 iterations of descent with momentum



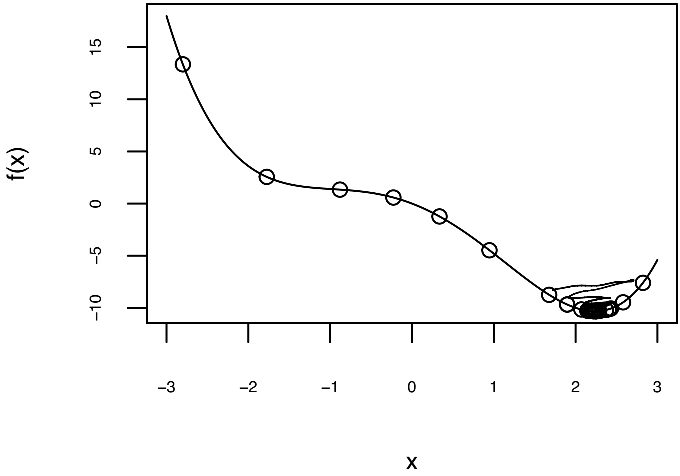
Another example

Momentum can also help overcome local minima:



Drawback of momentum

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



bounces around

Nesterov momentum

"Heavy ball" momentum:

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

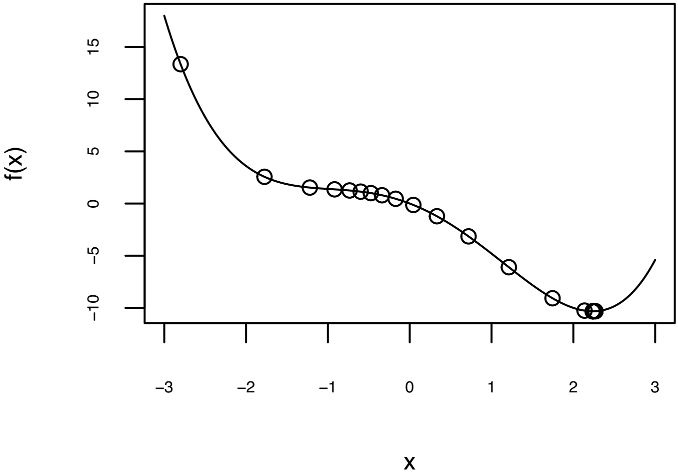
look at gradient in our future direction of movement, not just current point

Nesterov momentum:

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

Nesterov momentum in action

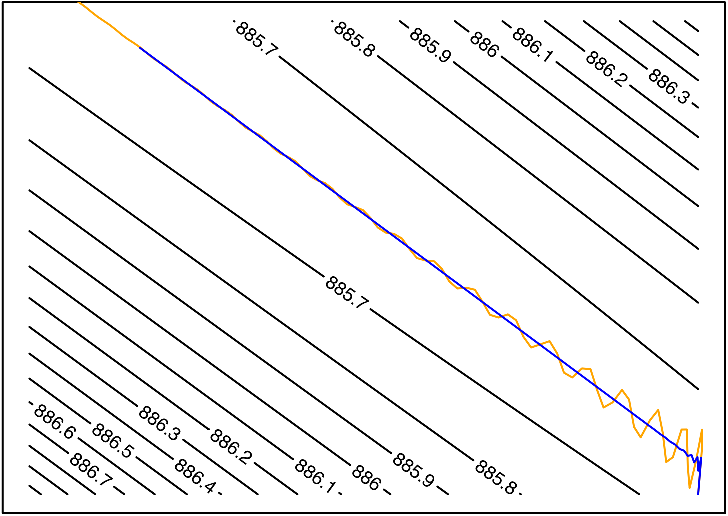
Descent with Nesterov momentum can slow itself down when it gets to the bottom:



Nesterov momentum in action

Blue: Nesterov momentum

Orange: Heavy ball momentum



Subgradient methods

$\hat{\beta}_0 = 3.9$, $\hat{\beta}_1 = 9.1$

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:

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

changes in β_1 have a bigger impact on f

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

Adagrad

Adagrad (*ada*ptive sub*grad*ient) uses a different step size for each element of the gradient:

step size for ith coordinate

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\alpha}{\varepsilon + \sqrt{s_i^{(k)}}} \underbrace{\nabla f(\mathbf{x}^{(k)})_i}_{\text{ith entry of } \nabla f(x^{(k)})}$$

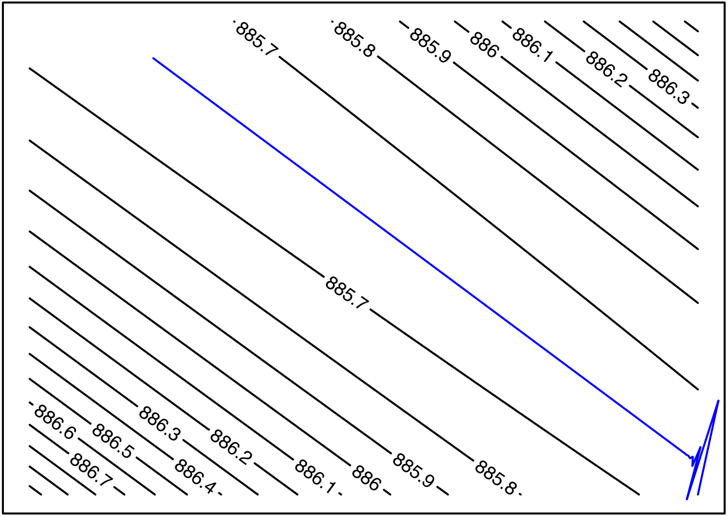
ith entry of $x^{(k+1)}$

ε prevents division by 0

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

idea: take a smaller step if ith coordinate has a consistently large gradient

Adagrad


885.7
885.8
885.9
886
886.1
886.2
886.3
885.7
886.6
886.7
886.5
886.4
886.3
886.2
886.1
886
885.9
885.8

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