Tensors for optimization(?)

Higher order and accelerated methods

Based on

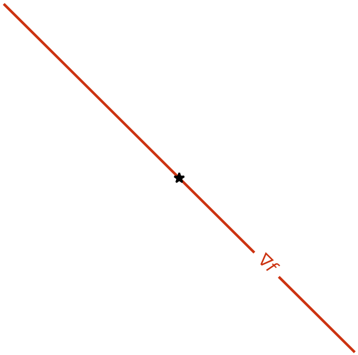
Estimate sequence methods: extensions and approximations, Michel Baes, 2009

MLRG Fall 2020 – Tensor basics and applications – Nov 18

Tensors in optimization

Goal: find the minimum of a function f

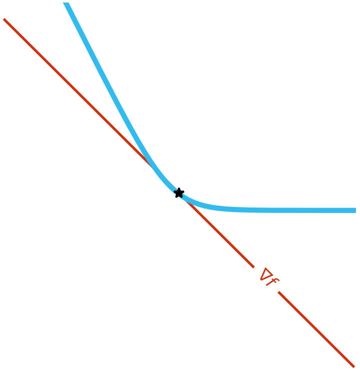
Only available information is local: $x, f(x), \nabla f(x), \ldots$



Tensors in optimization

Goal: find the minimum of a function f

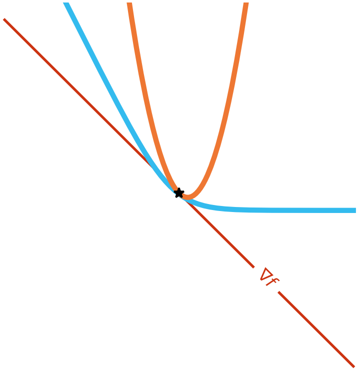
Only available information is local: $x, f(x), \nabla f(x), \ldots$



Tensors in optimization

Goal: find the minimum of a function f

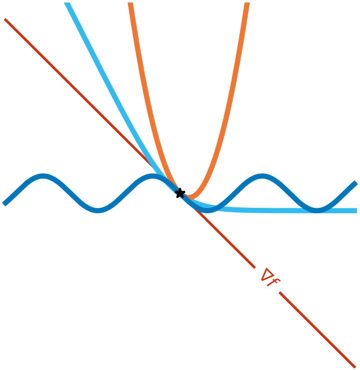
Only available information is local: $x, f(x), \nabla f(x), \ldots$



Tensors in optimization

Goal: find the minimum of a function f

Only available information is local: $x, f(x), \nabla f(x), \ldots$

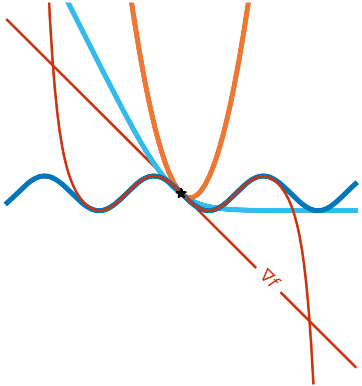


Which one is it?

Tensors in optimization

Goal: find the minimum of a function f

Only available information is local: $x, f(x), \nabla f(x), \ldots$



Higher order derivatives = more information

But... isn't Newton "bad"?

Newton's method: $\quad x' = x - \alpha[\nabla^2 f(x)]^{-1}\nabla f(x)$

But… isn't Newton "bad"?

Newton's method: $x' = x - \alpha[\nabla^2 f(x)]^{-1}\nabla f(x)$

Less stable than gradient descent

Only works for small problems

Can go up instead of down

Awful scaling with dimension

Why use even higher order information?

But... isn't Newton "bad"?

$$\text{Newton's method:} \quad x' = x - \alpha[\nabla^2 f(x)]^{-1}\nabla f(x)$$

Less stable than gradient descent

Can go up instead of down

Only works for small problems

Awful scaling with dimension

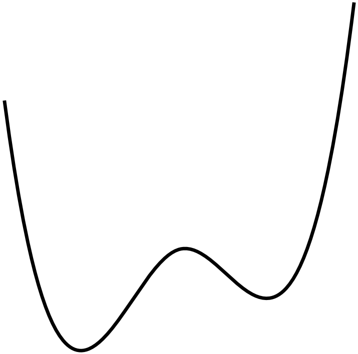
Why use even higher order information?

Today: a primer

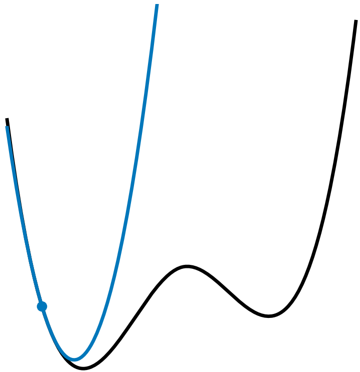
- Newton: the issues and how to fix them
- General recipe for higher order
- Some intuition for faster/approximate methods

Next week: Superfast higher order methods

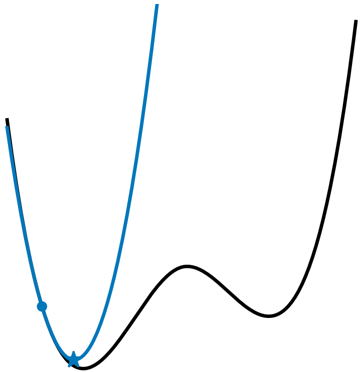
The issues with Newton: Non-convex functions



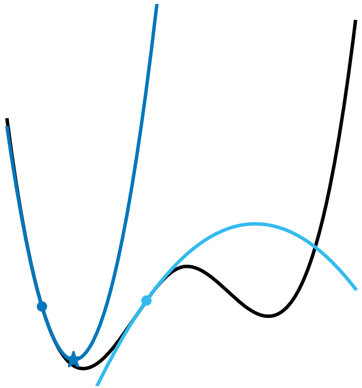
The issues with Newton: Non-convex functions



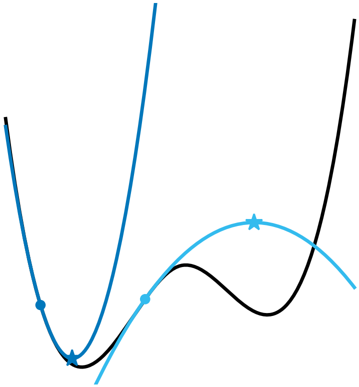
The issues with Newton: Non-convex functions



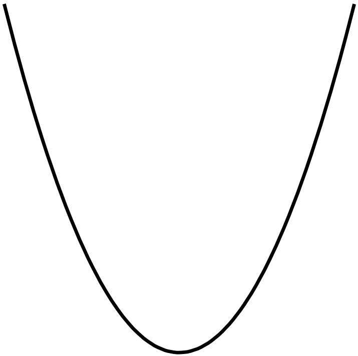
The issues with Newton: Non-convex functions



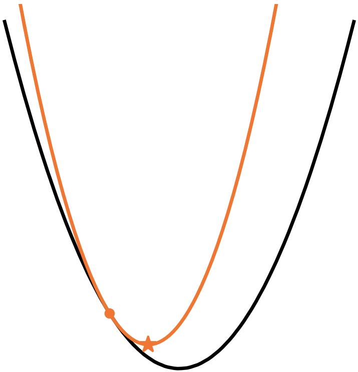
The issues with Newton: Non-convex functions



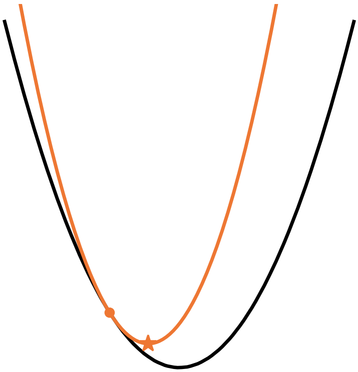
The issues with Newton: Stability



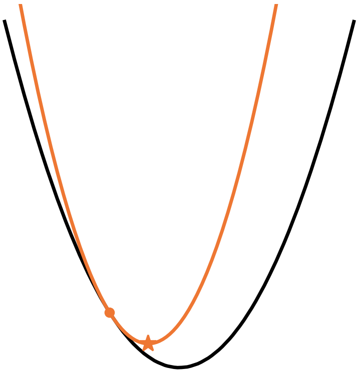
The issues with Newton: Stability



The issues with Newton: Stability

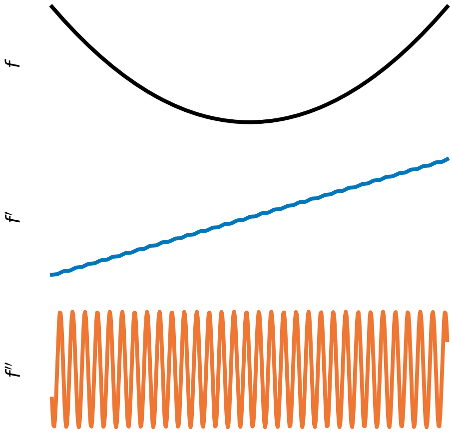


The issues with Newton: Stability



What?

The issues with Newton: Stability

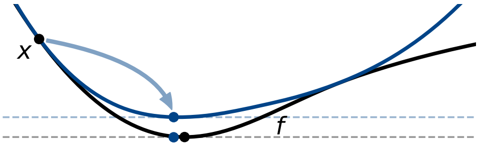


Optimization

Want to minimize f. At x, we know $f(x), \nabla f(x), \ldots$

Optimization

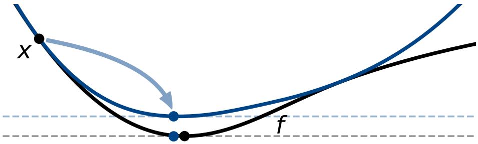
Want to minimize f. At x, we know $f(x), \nabla f(x), \ldots$



Surrogate: Simple(r) to optimize, progress on it leads to progress on f

Optimization

Want to minimize f. At x, we know $f(x), \nabla f(x), \ldots$

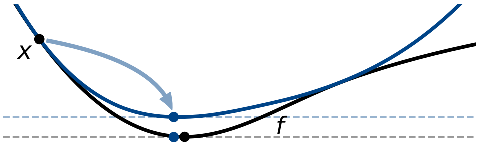


Surrogate: Simple(r) to optimize, progress on it leads to progress on f

Assumptions on f and available information $\rightarrow$ Best algorithm?

Optimization

Want to minimize f. At x, we know $f(x), \nabla f(x), \ldots$



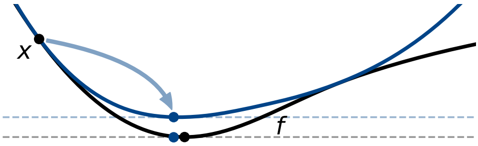
Surrogate: Simple(r) to optimize, progress on it leads to progress on f

Assumptions on f and available information $\rightarrow$ Best algorithm?

Algorithm $\rightarrow$ Why does it work? When does it work?

Optimization

Want to minimize f. At x, we know $f(x), \nabla f(x), \ldots$



Surrogate: Simple(r) to optimize, progress on it leads to progress on f

Assumptions on f and available information $\rightarrow$ Best algorithm?

Algorithm $\rightarrow$ Why does it work? When does it work?

Gradient descent $\rightarrow$ Modified Newton $\rightarrow$ Arbitrary order

First order

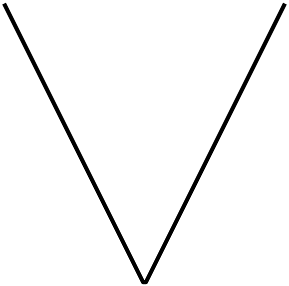
Gradient Descent (fixed α): $$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

Need continuity: if the gradient changes too fast, not informative

First order

Gradient Descent (fixed α): $$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

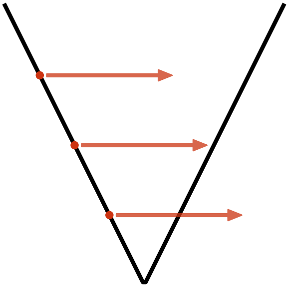
Need continuity: if the gradient changes too fast, not informative



First order

Gradient Descent (fixed α): $$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

Need continuity: if the gradient changes too fast, not informative



First order

Gradient Descent (fixed α): $x_{t+1} = x_t - \alpha \nabla f(x_t)$

The Hessian is bounded $\rightarrow$ The gradient does not change too fast

First order

Gradient Descent (fixed α): $x_{t+1} = x_t - \alpha \nabla f(x_t)$

The Hessian is bounded $\rightarrow$ The gradient does not change too fast

Taylor expansion:

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2!}f''(x)(y-x)^2 + \frac{1}{3!}f'''(x)(y-x)^3 + \dots$$

First order

Gradient Descent (fixed α): $x_{t+1} = x_t - \alpha \nabla f(x_t)$

The Hessian is bounded $\rightarrow$ The gradient does not change too fast

Taylor expansion:

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2!} f''(x)(y-x)^2 + \frac{1}{3!} f'''(x)(y-x)^3 + \dots$$

Truncated error:

$$f(y) = f(x) + f'(x)(y-x) + O((y-x)^2)$$

First order

Gradient Descent (fixed α): $x_{t+1} = x_t - \alpha \nabla f(x_t)$

The Hessian is bounded $\rightarrow$ The gradient does not change too fast

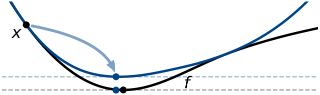
Taylor expansion:

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2!}f''(x)(y-x)^2 + \frac{1}{3!}f'''(x)(y-x)^3 + \dots$$

Truncated error:

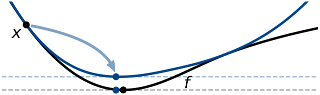
$$\begin{aligned} f(y) &= f(x) + f'(x)(y-x) + O((y-x)^2) \\ &\leq f(x) + f'(x)(y-x) + \left[\frac{1}{2}\max_x f''(x)\right](y-x)^2 \end{aligned}$$

First order



$$\tilde{f}(y) = f(x) + \nabla f(x)(y - x) + \frac{L}{2}\|y - x\|^2$$

First order

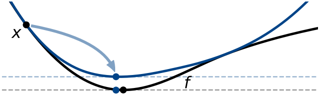


$$\tilde{f}(y) = f(x) + \nabla f(x)(y - x) + \frac{L}{2}\|y - x\|^2$$

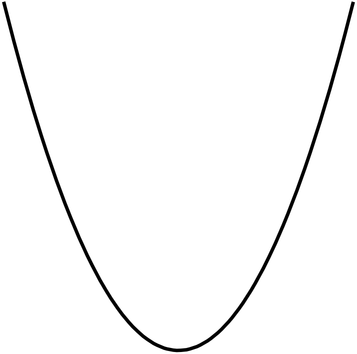
Convex quadratic upper bound on f:

$$
\begin{aligned}
& & 0 &= \nabla \tilde{f}(y) \\
&\Longrightarrow & 0 &= \nabla f(x) + L(y - x) \\
&\Longrightarrow & y &= x - \frac{1}{L}\nabla f(x)
\end{aligned}
$$

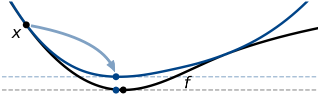
First order



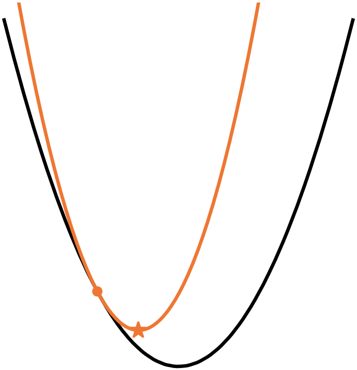
$$\tilde{f}(y) = f(x) + \nabla f(x)(y - x) + \frac{L}{2}\|y - x\|^2$$



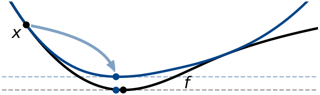
First order



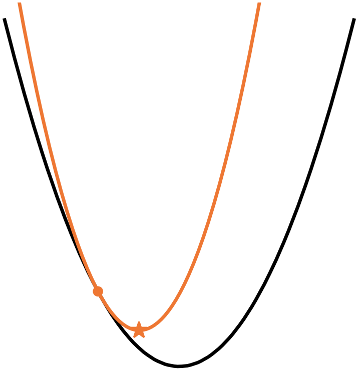
$$\tilde{f}(y) = f(x) + \nabla f(x)(y - x) + \frac{L}{2}\|y - x\|^2$$



First order



$$\tilde{f}(y) = f(x) + \nabla f(x)(y - x) + \frac{L}{2}\|y - x\|^2$$



Second order

The Hessian does not change too fast

$$f(x) + f'(x)(y-x) + \frac{1}{2}f''(x)(y-x)^2 + O((y-x)^3)$$

Second order

The Hessian does not change too fast

$$f(x) + f'(x)(y-x) + \frac{1}{2}f''(x)(y-x)^2 + \left[\frac{1}{6}\max_x f'''(x)\right](y-x)^3$$

The third derivative is bounded

Second order

The Hessian does not change too fast

$$f(x) + f'(x)(y - x) + \tfrac{1}{2} f''(x)(y - x)^2 + \left[\tfrac{1}{6} \max_x f'''(x)\right] (y - x)^3$$

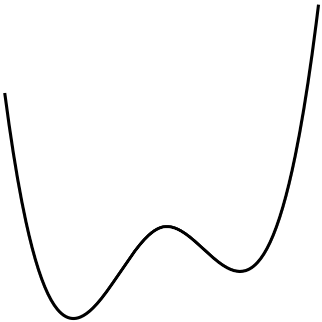
The third derivative is bounded

$$x' = \arg\min_y \left\{ \langle \nabla f(x), y - x \rangle + \frac{1}{2} \nabla^2 f(x)[y - x, y - x] + \frac{M}{6} \|y - x\|^3 \right\}$$

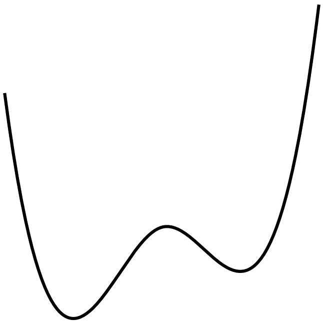
Cubic regularization of Newton's method

Cubic regularization: non-convex

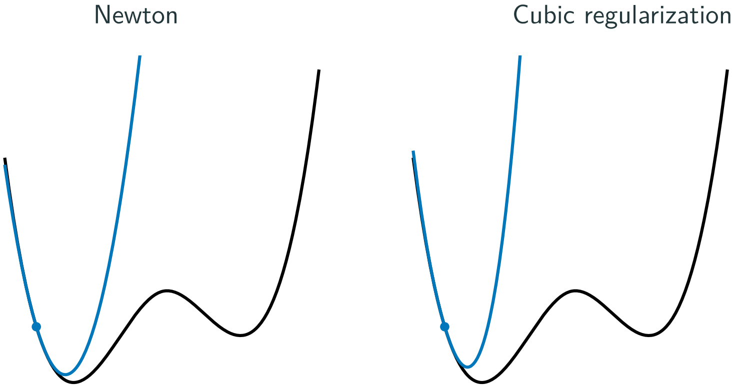
Newton



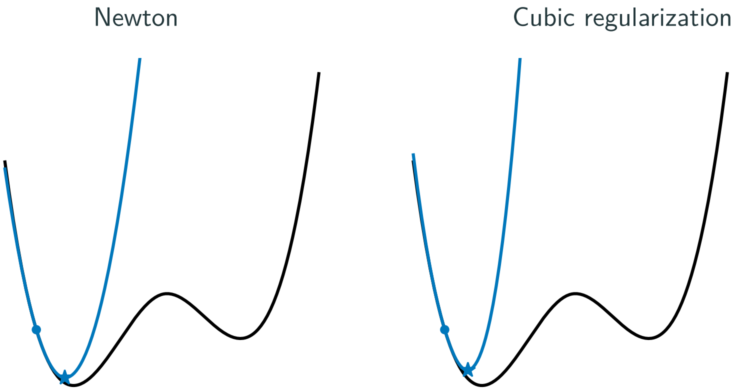
Cubic regularization



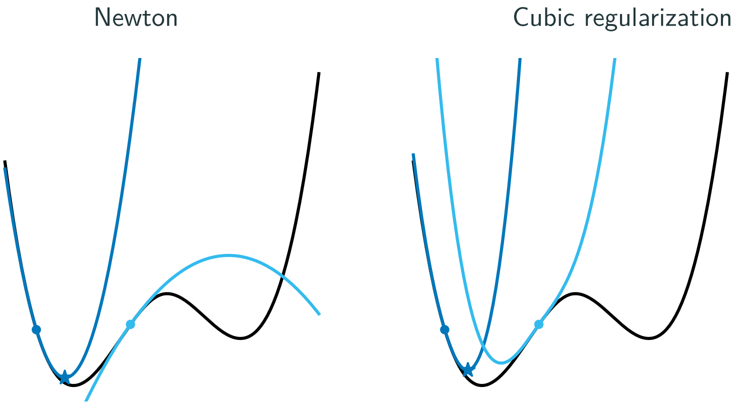
Cubic regularization: non-convex



Cubic regularization: non-convex

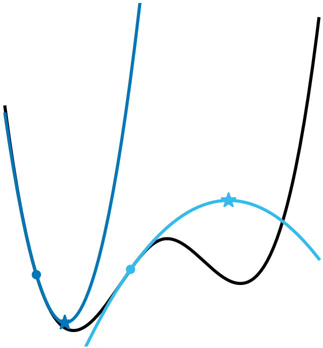


Cubic regularization: non-convex

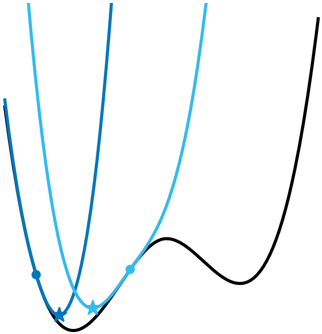


Cubic regularization: non-convex

Newton

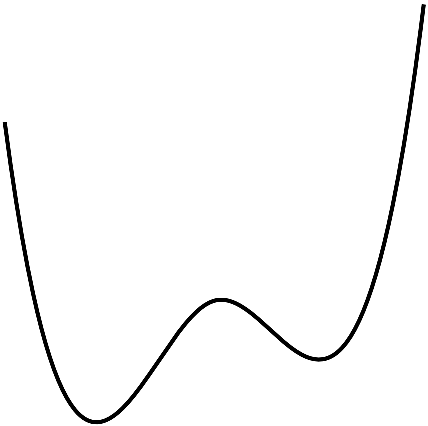


Cubic regularization



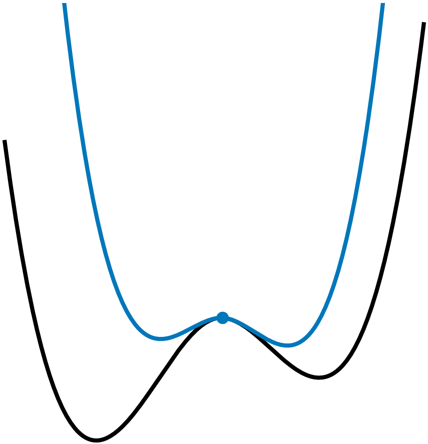
Cubic regularization: non-convex

Stationary points



Cubic regularization: non-convex

Stationary points



General strategy

For mth order methods: If bounded $(m+1)$th derivative

$$\text{Minimize} \quad \{m\text{th order Taylor expansion} + C\|x-y\|^{m+1}\}$$

General strategy

For mth order methods: If bounded $(m+1)$th derivative

$$\text{Minimize} \quad \{m\text{th order Taylor expansion} + C\|x-y\|^{m+1}\}$$

So far:

Issues with naïve Newton

Regularity assumptions for GD and higher order methods

Cubic regularization

General strategy

For mth order methods: If bounded $(m+1)$th derivative

$$\text{Minimize} \quad \{m\text{th order Taylor expansion} + C\|x-y\|^{m+1}\}$$

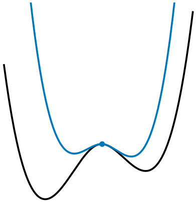
So far:

- Issues with naïve Newton
- Regularity assumptions for GD and higher order methods
- Cubic regularization

Next:

- How to solve the subproblem
- Some caveats
- Is it faster? Fastest?

Time per iteration



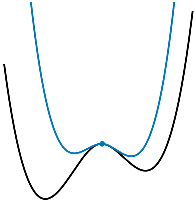
Cubic regularization update:

$$x' = x - d$$

$$\min_d \left\{ \langle g, d \rangle + \frac{1}{2} H[d,d] + \frac{M}{6} \|d\|^3 \right\}$$

... It's not convex

Time per iteration



Cubic regularization update:

$$x' = x - d$$

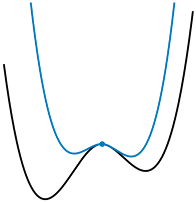
$$\min_d \left\{ \langle g, d \rangle + \frac{1}{2} H[d,d] + \frac{M}{6} \|d\|^3 \right\}$$

... It's not convex (but it is simpler)

If $\|d\| = r$ $\implies$ minimizing a quadratic (with simple constraints)

$$d = \left[H + \frac{Mr}{2} I \right]^{-1} g$$

Time per iteration



Cubic regularization update:

$$x' = x - d$$

$$\min_d \left\{ \langle g, d \rangle + \frac{1}{2} H[d, d] + \frac{M}{6} \|d\|^3 \right\}$$

... It's not convex (but it is simpler)

If $\|d\| = r$ $\Longrightarrow$ minimizing a quadratic (with simple constraints)

$$d = \left[H + \frac{Mr}{2} I \right]^{-1} g$$

Find the fixed point of $r = \left\| \left[H + \frac{Mr}{2} I \right]^{-1} g \right\|$ (1D, convex)

Time: Matrix inverse (once, then reuse) $O(n^3)$
$+$ a few iterations of a convex 1D solver (only matrix-vector products)

Caveats

GD works well $\neq$ Cubic regularization works well

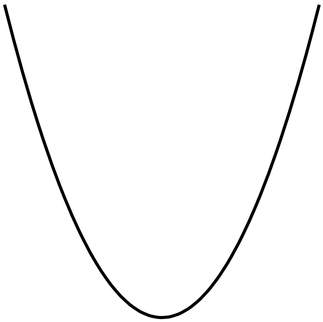
Quality of approximation goes up **if** higher derivatives are smooth enough

Caveats

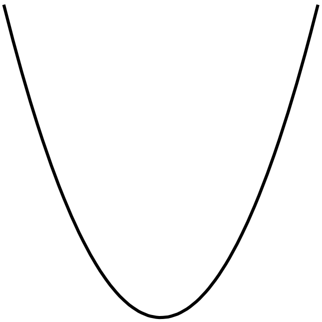
GD works well $\neq$ Cubic regularization works well

Quality of approximation goes up **if** higher derivatives are smooth enough

GD



Cubic regularization

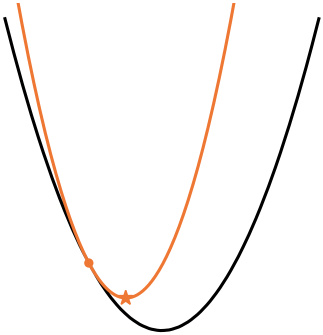


Caveats

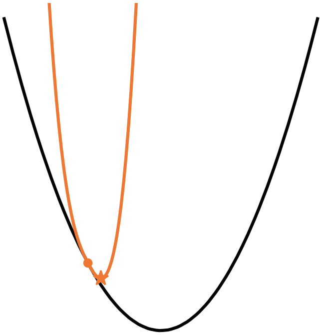
GD works well $\neq$ Cubic regularization works well

Quality of approximation goes up **if** higher derivatives are smooth enough

GD



Cubic regularization

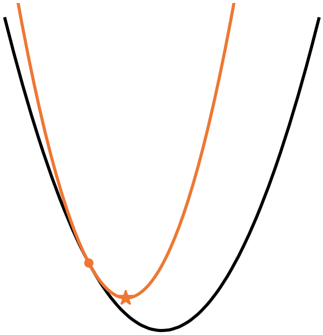


Caveats

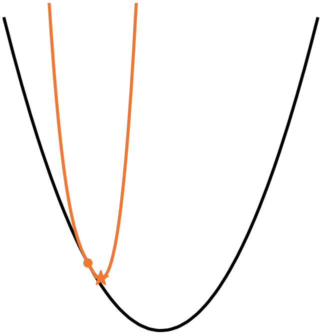
GD works well $\neq$ Cubic regularization works well

Quality of approximation goes up **if** higher derivatives are smooth enough

GD



Cubic regularization



Caveats

GD works well $\neq$ Cubic regularization works well

Quality of approximation goes up **if** higher derivatives are smooth enough

Bounds on $f, f', f'', f''', \ldots$

Some functions get "smoother" with higher derivatives, some less so

Caveats

GD works well $\neq$ Cubic regularization works well

Quality of approximation goes up **if** higher derivatives are smooth enough

Bounds on $f, f', f'', f''', \ldots$

Some functions get "smoother" with higher derivatives, some less so

$$\begin{array}{ccccc} f & f' & f'' & f''' & f'''' \\ \sin(cx) & c\cos(cx) & -c^2\sin(cx) & -c^3\cos(cx) & c^4\sin(cx) \end{array}$$

$$c < 1 \implies \max f^{(m)}(x) \to 0 \qquad c > 1 \implies \max f^{(m)}(x) \to \infty$$

Is it faster?

After T steps, $f(x_T) - f^* \leq$?

(in convex world)

Is it faster?

After T steps, $f(x_T) - f^* \leq$?

(in convex world)

C_m depends on bound on $f^{(m+1)}$ (and initial error)

Gradient descent:	$f(x_t) - f^* \leq C_1/T$
Cubic regularization:	$f(x_t) - f^* \leq C_2/T^2$
mth-order (regularized):	$f(x_t) - f^* \leq C_m/T^m$

Is it faster?

After T steps, $f(x_T) - f^* \leq$?

(in convex world)

C_m depends on bound on $f^{(m+1)}$ (and initial error)

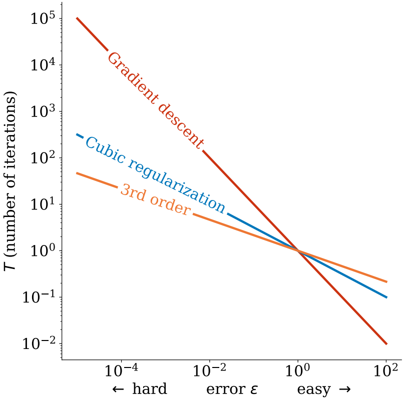
Gradient descent:	$f(x_t) - f^* \leq C_1/T$
Cubic regularization:	$f(x_t) - f^* \leq C_2/T^2$
mth-order (regularized):	$f(x_t) - f^* \leq C_m/T^m$

How many iterations to reach $f(x_T) - f^* \leq \epsilon$?

Gradient descent:	$T \geq C_1/\epsilon$
Cubic regularization:	$T \geq (C_2/\epsilon)^{1/2}$
mth-order (regularized):	$T \geq (C_m/\epsilon)^{1/m}$

Is it faster?

Time to reach $f(x_T) - f^* \leq \epsilon$



Plot caveats

- Height depends on constants
- Only slopes are accurate
- Worst case, if assumptions hold
- Log-log scale

Main takeaway:
For tiny ϵ, higher order methods are better *even* if more expensive/iteration

“Actual time”:

Number of iterations:

Is it fastest? (main part of the paper)

"Actual time": nope

Only need to solve subproblem approximately (at first)

$$\tilde{f}_t(x_{t+1}) - \tilde{f}_t^* \leq \epsilon_t, \qquad \epsilon_t = O\left(\frac{1}{t^c}\right)$$

Number of iterations:

"Actual time": nope

Only need to solve subproblem approximately (at first)

$$\tilde{f}_t(x_{t+1}) - \tilde{f}_t^* \leq \epsilon_t, \qquad \epsilon_t = O\left(\frac{1}{t^c}\right)$$

Number of iterations: nope (for convex functions)

Gradient descent: $1/T$

Cubic regularization: $1/T^2$

Is it fastest? (main part of the paper)

"Actual time": nope

Only need to solve subproblem approximately (at first)

$$\tilde{f}_t(x_{t+1}) - \tilde{f}_t^* \leq \epsilon_t, \qquad \epsilon_t = O\left(\frac{1}{t^c}\right)$$

Number of iterations: nope (for convex functions)

Gradient descent: $1/T$

Cubic regularization: $1/T^2$

Accelerated Gradient Descent: $1/T^2$

Is it fastest? (main part of the paper)

"Actual time": nope

Only need to solve subproblem approximately (at first)

$$\tilde{f}_t(x_{t+1}) - \tilde{f}_t^* \leq \epsilon_t, \qquad \epsilon_t = O\left(\frac{1}{t^c}\right)$$

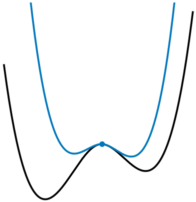
Number of iterations: nope (for convex functions)

Gradient descent: $1/T$

Cubic regularization: $1/T^2$ Accelerated Gradient Descent: $1/T^2$

Accelerated cubic regularization: $1/T^3$ mth order: $1/T^{m+1}$

Main ideas



Optimization with higher order approximations
Regularity assumptions
Constructing upper bounds
Solving polynomials

Next week: Super fast accelerated higher order methods (and maybe a tensor)

Thanks!