Tensors for optimization (part 2)

3rd-order methods

Based on

[1] Superfast Second-Order Methods for Unconstrained Convex Optimization

[2] Inexact Accelerated High-order Proximal-point Methods

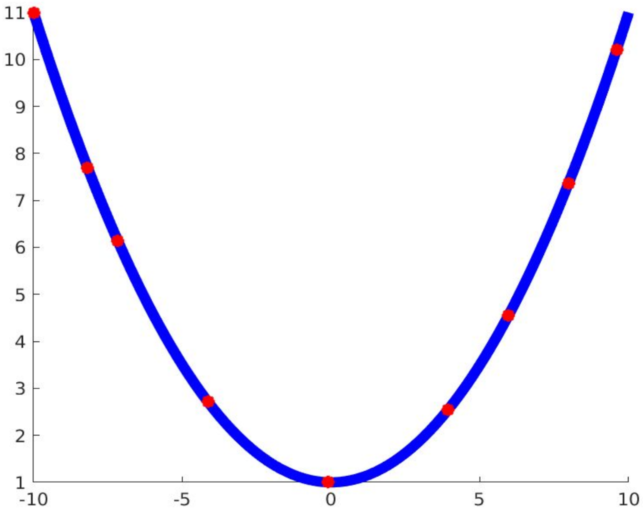
The purpose of this talk

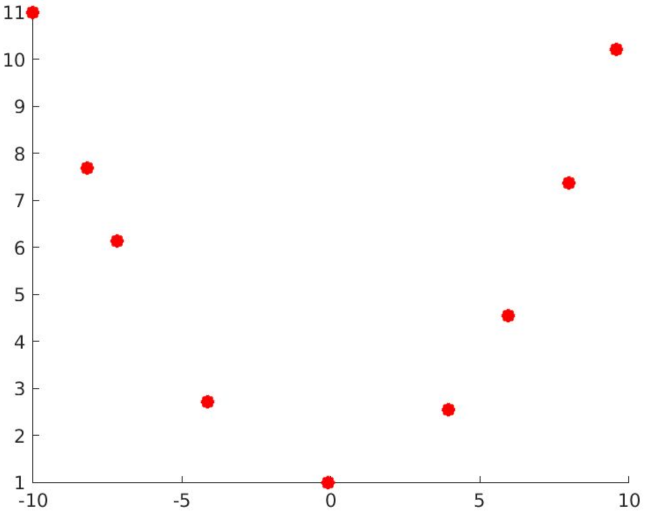
(1) Give an overview of [1][2]

(2) Cover some missing steps in [1] (will use some standard calculus tricks)

Our goal

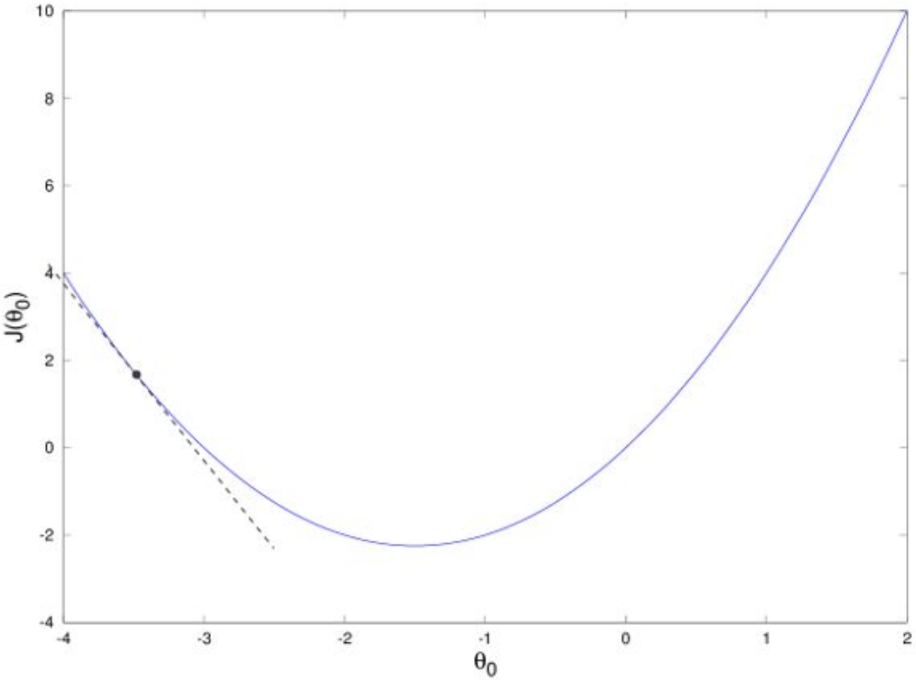
Goal: $\min_{x \in \mathcal{R}^d} f(x)$

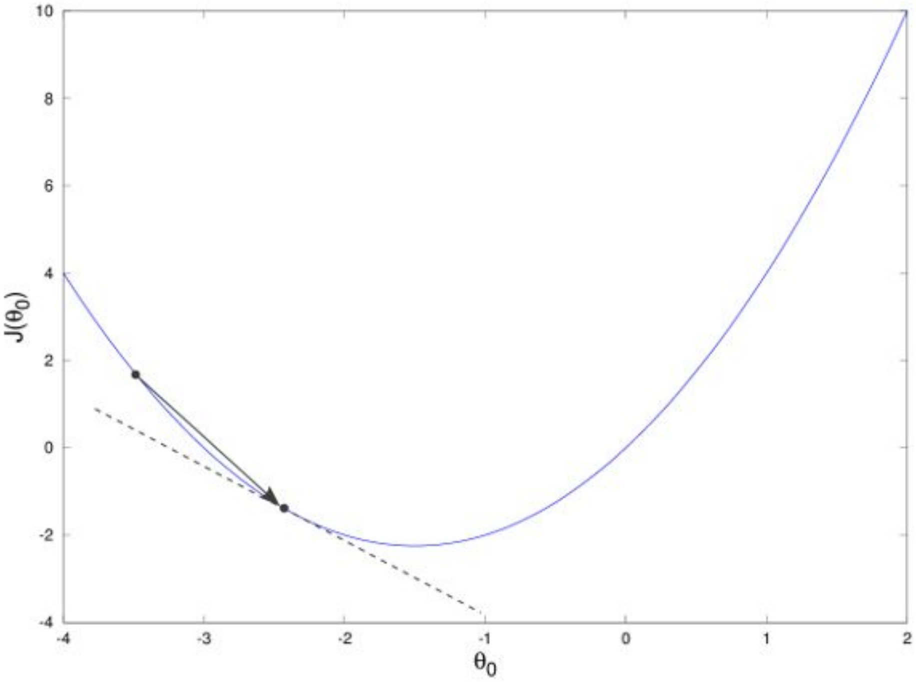




Available local information:

$$x, f(x), \nabla f(x), \nabla^2 f(x), \nabla^3 f(x), \ldots$$





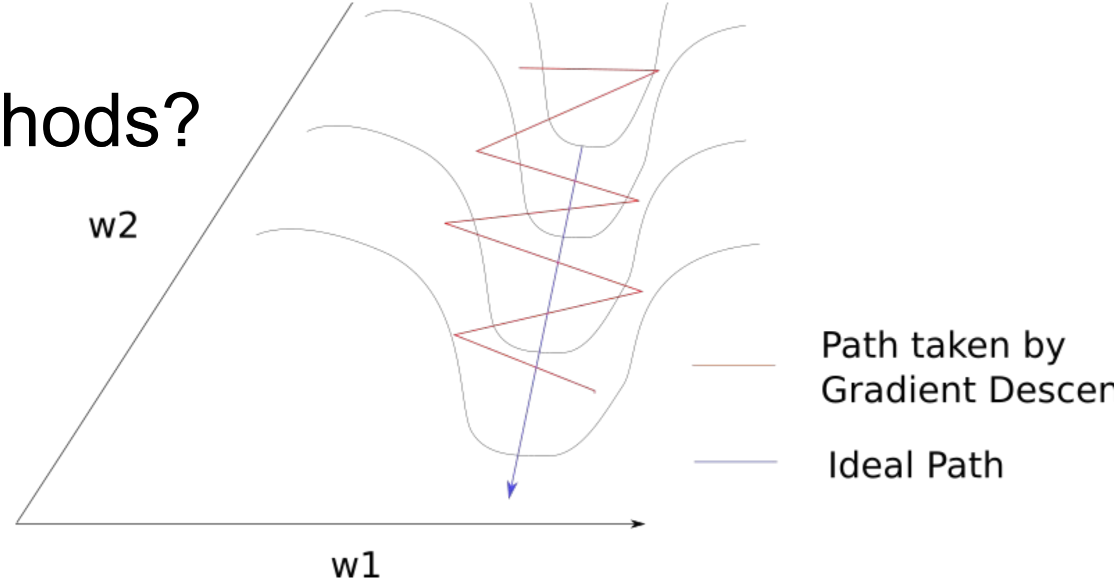
Common an optimizer: gradient descent $\text{GD}: \; x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$

Using local information: $x_t, \nabla f(x_t), f(x_t)$

Why higher-order methods?

Higher-order methods:

(1) find a fast path
(2) design efficient methods



Efficient methods (inspired from Newton's (2nd-order) method):

Quasi-Newton methods (LBFGS) e.g., Mark's minFunc package

Adaptive gradient methods (Adam)

Basics

Gradient Descent $\quad \text{GD}: \; x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$

Alternative formulation of GD:

$$h(y) := f(x_t) + < \nabla_x f(x_t), y - x_t > + \frac{1}{\alpha} \frac{1}{2!} \|y - x_t\|^2$$

$$x_{t+1} \leftarrow \arg\min_y h(y) \qquad \nabla_y h(y)|_{y=x_{t+1}} = 0$$

$$\nabla_y h(y)|_{y=x_{t+1}} = \nabla_x f(x_t) + \frac{1}{\alpha}(y - x_t) = \nabla_x f(x_t) + \frac{1}{\alpha}(x_{t+1} - x_t)$$

$$= \nabla_x f(x_t) + \frac{1}{\alpha}(x_t - \alpha \nabla_x f(x_t) - x_t) = 0$$

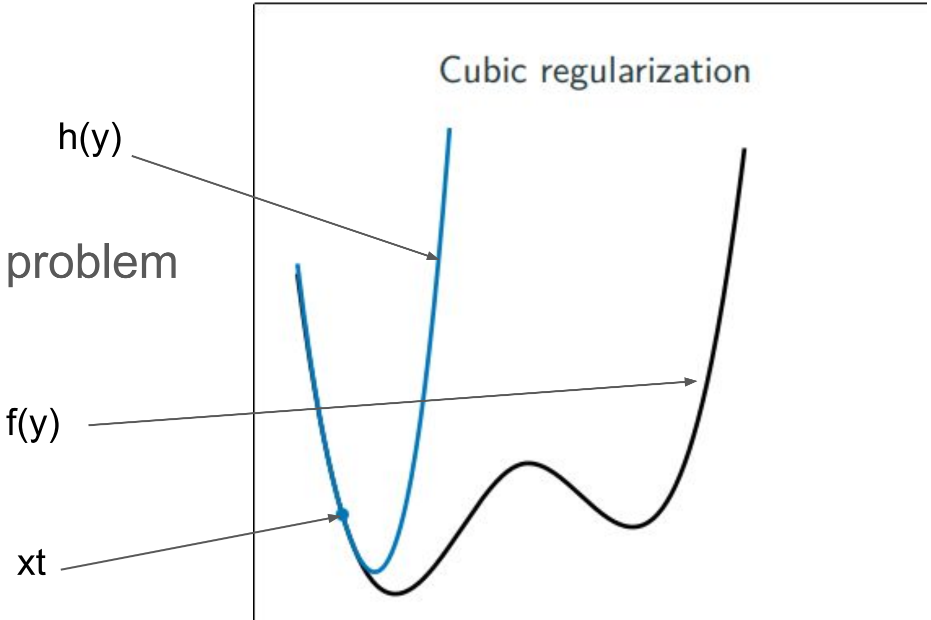
Basics

Newton's method with a cubic regularization (from last week's meeting)

$$h(y) := f(x_t) + < \nabla_x f(x_t), y - x_t > + \frac{1}{2} < \nabla_x^2 f(x_t)(y - x_t), y - x_t > + \frac{1}{\alpha}\frac{1}{3!}\|y - x_t\|^3$$

$$x_{t+1} \leftarrow \arg\min_y h(y)$$

Issue: difficult to solve this minimization problem



Hessian-vector product $\nabla^2 f(x_t)[d]$

Today's topic

Tensor-vector product $\nabla^3 f(x_t)[d]^2$

3rd-order methods (with a regularization)

$$\Gamma_{i,j,k} = \nabla^3 f(x_t); \nabla^3 f(x_t)[d]^2 := \sum_i \sum_j \Gamma_{i,j,k} d_i d_j$$

$$d := y - x_t$$

Order does not matter (e.g., hessian is symmetric)

$$h(y) := f(x_t) + < \nabla_x f(x_t), d > + \frac{1}{2} < \nabla_x^2 f(x_t) d, d > + \frac{1}{3!} < \nabla_x^3 f(x_t)[d]^2, d > + \frac{1}{\alpha} \frac{1}{4!} ||d||^4$$

$$x_{t+1} \leftarrow \arg\min_y h(y)$$

Issues:

(1) hard to compute exact 3rd derivatives/tensors $\nabla_x^3 f(x_t)$
(2) difficult to solve this auxiliary minimization problem

Assumptions in [1]: f(x) is convex and is Lipschitz at all its 3rd-order derivatives with a positive constant L

The meaning of a tensor

Tensor Algebra: Tensor decomposition under a coordinate system

A tensor is a coordinate component

In [1,2]: a tensor is defined by a (Euclidean) derivative. It is a coordinate component under the base.

A calculus trick

A similar identity for the hessian-vector product

$\nabla^2 f(x_t)[d]$

The tensor-vector product:

$$\nabla^3 f(x_t)[d]^2 := \lim_{\tau \to 0} \frac{1}{\tau^2}[\nabla f(x + \tau d) + \nabla f(x - \tau d) - 2\nabla f(x)]$$

Apply L'Hospital's rule twice

Numerator: $[\nabla f(x + \tau d) + \nabla f(x - \tau d) - 2\nabla f(x)] \to 0 \text{ as } \tau \to 0$

Denominator: $\tau^2 \to 0 \text{ as } \tau \to 0$

$$\nabla^3 f(x_t)[d]^2 := \lim_{\tau \to 0} \frac{1}{\tau^2}[\nabla f(x + \tau d) + \nabla f(x - \tau d) - 2\nabla f(x)]$$

$$\nabla_\tau[\nabla f(x + \tau d)] = \nabla_\tau[\nabla_{x+\tau d} f(x + \tau d)]$$

Apply L'Hospital's rule once (w.r.t. τ)

$$\nabla_\tau[\nabla_{x+\tau d} f(x + \tau d)] = [\nabla^2_{x+\tau d} f(x + \tau d)]\nabla_\tau(x + \tau d) = \nabla^2 f(x + \tau d)[d]$$

Numerator: $[\nabla^2 f(x + \tau d)[d] - \nabla^2 f(x - \tau d)[d]] \to 0 \text{ as } \tau \to 0$

Denominator: $2\tau \to 0 \text{ as } \tau \to 0$

Apply L'Hospital's rule twice (w.r.t. τ)

Numerator: $\nabla^3 f(x + \tau d)[d]^2 + \nabla^3 f(x - \tau d)[d]^2 \to 2\nabla^3 f(x)[d]^2 \text{ as } \tau \to 0$

Denominator: 2

$$\begin{aligned}\nabla_\tau[-\nabla^2_{x-\tau d} f(x - \tau d)[d]] &= -[\nabla^3_{x-\tau d} f(x - \tau d)[d]][\nabla_\tau(x - \tau d)] \\ &= -[\nabla^3_{x-\tau d} f(x - \tau d)[d]][-d] = [\nabla^3_{x-\tau d} f(x - \tau d)[d]^2]\end{aligned}$$

Approximate derivatives

Approximate the product using finite difference

$$\nabla^3 f(x)[d]^2 \approx \frac{1}{\tau^2}[\nabla f(x+\tau d) + \nabla f(x-\tau d) - 2\nabla f(x)]$$

If f(x) is Lipschitz at all its 3rd-order derivatives, we can bound the error between the exact product and the approximation. (see Eq 1.4-1.5 and Lemma 5 of [1])

Lipschitz at all its 3rd-order derivatives with a positive constant L

$$\|\nabla^3 f(x) - \nabla^3 f(y)\| \leq L\|x - y\|, \quad \text{for any } x, y$$

How to solve the auxiliary problem?

$d := y - x_t$

$h(y) := f(x_t) + < \nabla_x f(x_t), d > + \frac{1}{2} < \nabla_x^2 f(x_t) d, d > + \frac{1}{3!} < \nabla_x^3 f(x_t)[d]^2, d > + \frac{1}{\alpha} \frac{1}{4!} \|d\|^4$

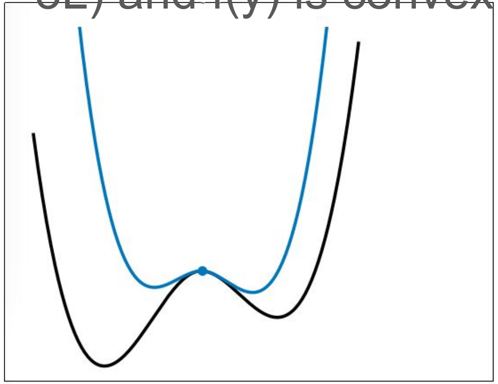
$$x_{t+1} \leftarrow \arg\min_y h(y)$$

Key results in [1] (assuming f(y) is Lipschitz of 3rd-order with a constant L):

(1) f(y) is bounded above by h(y) when $\frac{1}{\alpha}$ is large enough (>=L)

(2) If $\frac{1}{\alpha}$ is large enough (>=3L) and f(y) is convex, h(y) is also convex.

Convexity of f(y) is needed.



Implications of the results:

(1) h(y) is a valid upper bound for any x_t and y since we want to minimize f(y)
(2) Inexactly solve h(y) with convergence guarantee

$$h(y) := f(x_t) + < \nabla_x f(x_t), d > + \frac{1}{2} < \nabla_x^2 f(x_t) d, d > + \frac{1}{3!} < \nabla_x^3 f(x_t)[d]^2, d > + \frac{1}{\alpha} \frac{1}{4!} ||d||^4$$

$$d := y - x_t$$

In [1], a gradient-based method is used to solve h(y) (see Eq 4.8,4.19 of [1])

solving the auxiliary problem h(y) & solving the original problem f(x) In [2], this approach is called a bi-level minimization approach.

Summary

The algorithm proposed in [1]:

(1) Construct a 3rd-order approximation with a regularizer at a current point
(2) Approximate the tensor-vector product using finite difference
(3) Inexactly solve the auxiliary function (an upper bound and convexity)
(4) Update the current point using an inexact solution

Some results from [1]:

(1) A (theoretical) superfast convergence rate
(2) Implementing a 3rd-order method using the 2nd-order information (the trick)

Show f(y) <= h(y)

$$d := y - x_t$$

$$h(y) := f(x_t) + < \nabla_x f(x_t), d > + \frac{1}{2} < \nabla_x^2 f(x_t) d, d > + \frac{1}{3!} < \nabla_x^3 f(x_t)[d]^2, d > + \frac{1}{\alpha} \frac{1}{4!} \|d\|^4$$

Taylor truncation error for directional derivatives

$$f(y) = h(y) - \frac{1}{\alpha 4!} \|y - x_t\|^4 + \frac{1}{3!} \int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y - x_t))[y - x_t]^4 d\tau \le h(y)$$

When $L \le \frac{1}{\alpha}$, we have $\frac{L}{4!} \|y - x_t\|^4 \le \frac{1}{\alpha 4!} \|y - x_t\|^4$

$$-\frac{1}{\alpha 4!} \|y - x_t\|^4 + \frac{1}{3!} \int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y - x_t)) d\tau \le 0$$

$$\frac{1}{3!} \int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y - x_t)) d\tau \le \frac{1}{\alpha 4!} \|y - x_t\|^4$$

Our goal is to show

$$\frac{1}{3!} \int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y - x_t))[y - x_t]^4 d\tau \le \frac{L}{4!} \|y - x_t\|^4 \le \frac{1}{\alpha 4!} \|y - x_t\|^4$$

Recall: f(y) is Lipschitz at the 3rd-order (for simplicity, we assume x,y are scalars)

$\|\nabla^3 f(x) - \nabla^3 f(y)\| \leq L\|x - y\|, \;\; \text{for any } x, y$ implies $\|\nabla_x^4 f(x)\| \leq L$

$$\nabla^4 f(x) = \lim_{y \to x} \frac{\nabla^3 f(y) - \nabla^3 f(x)}{y - x}$$

Proof of $\frac{1}{3!}\int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y-x_t))[y-x_t]^4 d\tau \le \frac{L}{4!}\|y-x_t\|^4$

$$\|\nabla_x^4 f(x)\| \le L$$

$$\nabla^4 f(x_t + \tau(y-x_t))[y-x_t]^4 \le L\|y-x_t\|^4$$

$$\frac{1}{3!}\int_0^1 (1-\tau)^3 \nabla^4 f(x_t + \tau(y-x_t))[y-x_t]^4 d\tau \le \frac{1}{3!}\int_0^1 (1-\tau)^3 L[y-x_t]^4 d\tau = [y-x_t]^4 \frac{L}{3!}\int_0^1 (1-\tau)^3 d\tau$$

$$\int_0^1 (1-\tau)^3 d\tau = -\frac{1}{4}(1-\tau)^4\Big|_{\tau=0}^{\tau=1} = \frac{1}{4}$$

$$\frac{1}{3!}\int_0^1 (1-\tau)^3 \nabla_x^4 f(x_t + \tau(y-x_t))[y-x_t]^4 d\tau \le \frac{L}{4!}\|y-x_t\|^4$$

Thanks