Convex Optimization in Machine Learning and Computational Finance Lecture 9: Black-box Methods 1

Dr. Michel Baes, Pr. Patrick Cheridito

RiskLab / ETH Zürich

Quick recall of last week's lecture

- Approximation problems can often be casted as convex optimization problems.
- Multi-objective problems and Pareto curve. When $||x||_1$ is involved, the optimal x is often sparse.
- Convex games can yield profitable coalitions
- Convexity often appears in maximum likelihood problems (if you take logs).

I. In general, optimization problems cannot be solved

Two ways of doing Optimization

1. Write one universal optimization algorithm,
capable of solving all problems.

(Although pure Global Optimization is somewhat an ill-posed problem)

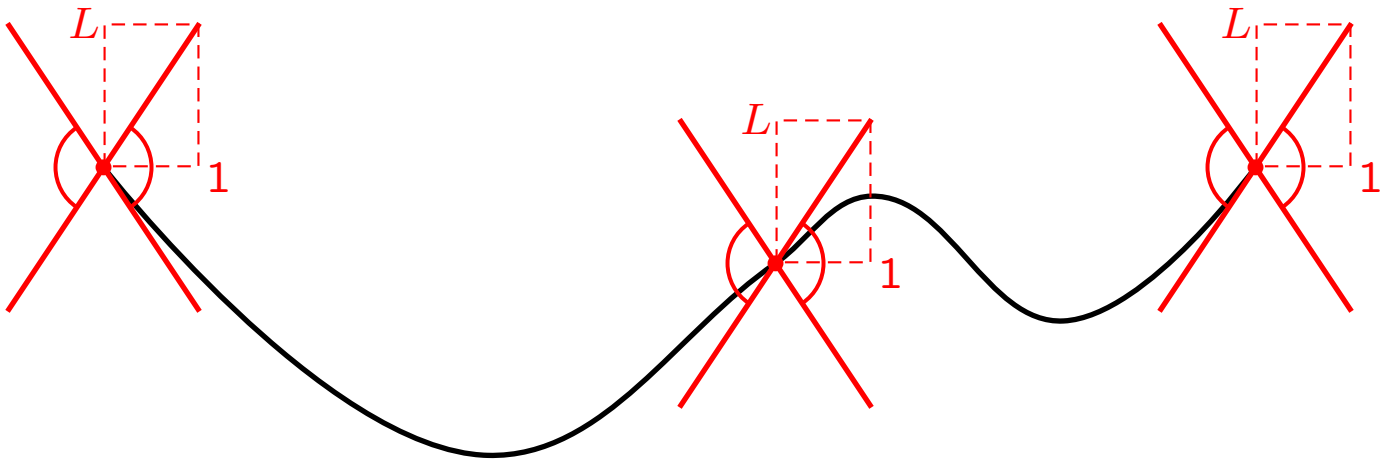
2. Write myriads of optimization algorithms,
each of them being dedicated to a
very particular class of problems.

Let us explore the first option for a minute.

How ill-defined is your problem?

Let $f : [0,1]^n \to \mathbb{R}$ be a Lipschitz continuous function:

For all $x, y \in [0,1]^n$: $|f(x) - f(y)| \leq L||x - y||_2$.



L measures how ill-defined the optimization problem is.

Towards a universal optimization algorithm

Let $f : [0,1]^n \to \mathbb{R}$ be a Lipschitz continuous function:

$$\text{For all } x, y \in [0,1]^n \ : \ |f(x) - f(y)| \leq L||x - y||_2.$$

Find a point $\widehat{x} \in [0,1]^n$ such that $f(\widehat{x}) - f^* < \epsilon$, where

$$\begin{aligned} f^* := \min \quad & f(x) \\ \text{s.t.} \quad & x \in [0,1]^n. \end{aligned}$$

The Grid Method: evaluate f on each point

$$(i_1, \ldots, i_n)/p,$$

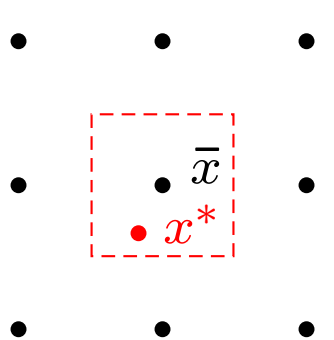
where $0 \leq i_1, \ldots, i_n \leq p$, and take for $\widehat{x}$ the one that minimizes f.

Complexity: $(p+1)^n$ evaluations of f.

Of course, it performs poorly...

Lemma 1
$f(\widehat{x}) - f^* \leq L\sqrt{n}/(2p)$.

Proof:



Let $\bar{x}$ be the grid point closest to the optimal x^*. As $||\bar{x} - x^*||_2 \leq \sqrt{n}/(2p)$ **(Why?)**, we have
$f(\widehat{x}) - f^* \leq f(\bar{x}) - f(x^*) \leq L\sqrt{n}/(2p)$.

Corollary 1 *The Grid Method needs* $\left(\left\lceil \frac{L\sqrt{n}}{2\epsilon} \right\rceil + 1\right)^n$ *evaluations of* f *for finding* $\widehat{x}$ *such that* $f(\widehat{x}) - f^* < \epsilon$.

...and we can't even hope for better

Theorem 1 *An optimization algorithm for this class requires at least* $\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$ *evaluations of* f.

The proof is based on the concept of *resisting oracle* (intuitively, the most annoying function for the method)

Proof: Consider a grid with $p = \lfloor L/(2\epsilon) \rfloor - 1$ (the grid size is larger than $2\epsilon/L$). Suppose we compute f on $(p+1)^n - 1$ points,

and we get the value 0 each time.

There must be a voxel of the grid with no computed point in its interior. Let $\tilde{x}$ be its center. Then $f(\tilde{x})$ can be as low as $-L \cdot (2\epsilon/L)/2 = -\epsilon$, so we cannot avoid an evaluation of f inside the missed voxel to get the desired accuracy.

What it means in numbers

Complexity: $\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$

If $n = 16$, $\epsilon := 0.01$, and $L = 1$, we need

$$1,525,878,906,250,000,000,000,000$$

evaluations of f.

If the computer can perform 4 billions of them a second, we need about 12,088,000,000 years (a bit less than the estimated age of the Universe).

Heuristics can be tried (genetic algorithms, ant colony, neural networks, simulated annealing, tabu search, ...) without any **absolute** guarantee on the computation time.

We are left with the second option

1. Write one universal optimization algorithm,
capable of solving all problems.

(Although pure Global Optimization is somewhat an ill-posed problem)

2. Write myriads of optimization algorithms,
each of them being dedicated to a
very particular class of problems.

We need to focus

on some more specific class of problem

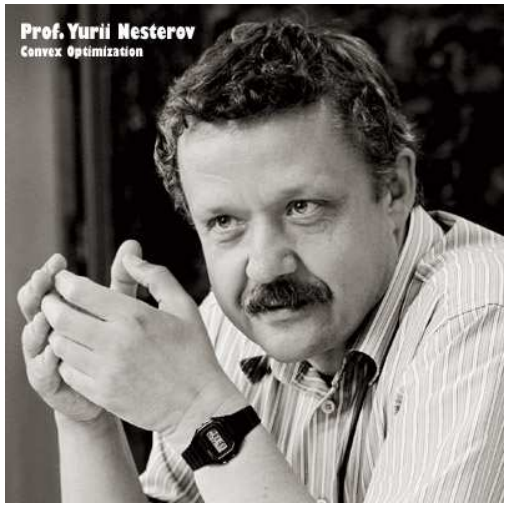
$$\min_{x \in Q} f(x)$$

Every provably efficient optimization method

- should be applied to a specific problem class
  (linear, convex, differentiable, Lipschitz continuous,. . .)
- is characterized by the way it extracts information
  on the specific instance of the problem

II. Simple methods and how good they can be

Arkadi Nemirovski



Yurii Nesterov

The Black-Box Model (Nemirovski, Yudin)

A device to study algorithms,

a tool to construct hard problems

$\boxed{\min_{x \in Q} f(x)}$ $(\mathcal{P})$

- $Q \subseteq \mathbb{R}^n$ is a convex set;
- $f : Q \to \mathbb{R}$ is convex;
- desired accuracy on objective's value is ϵ.

The Black-Box Model

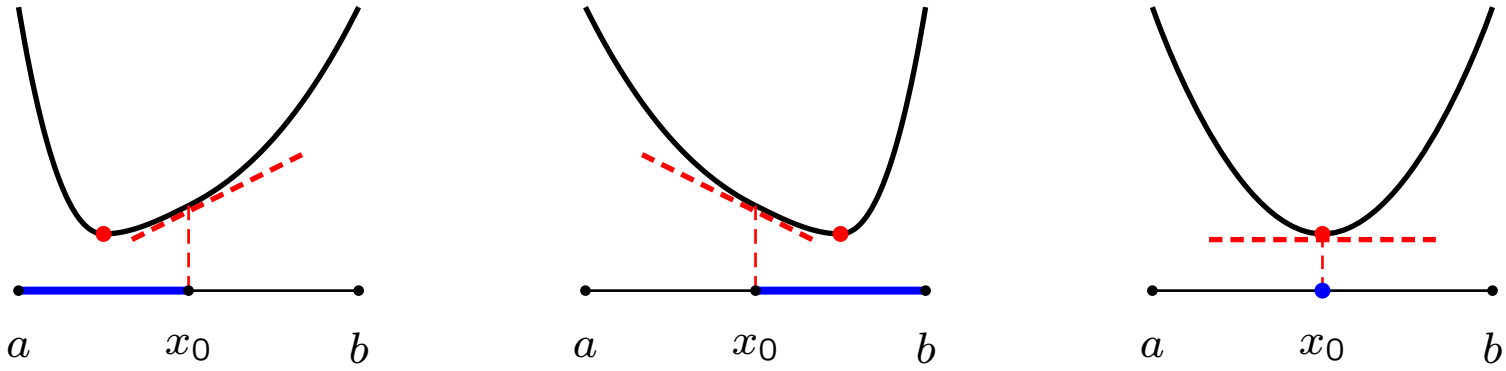
- Initially, we don't know anything on $(\mathcal{P})$, but its convexity
- At every iteration, we can ask a local question on $(\mathcal{P})$
  e.g: given a point x_k, what are $f(x_k)$ and a $g_k \in \partial f(x_k)$, what is the projection of a point on Q, ...?
- The device answering these questions is a black-box: we can't see what's inside (e.g. compiled code)

The black-box model: an illustration 1

$\min_{x \in [a,b]} f(x)$

- $f : [a, b] \to \mathbb{R}$ is convex;
- $\max\{f(x) - f(y) : x, y \in [a, b]\} \leq V$;
- oracle: $x \mapsto (f(x), g(x))$, with $g(x) \in \partial f(x)$;
- absolute accuracy ϵ.

1. The bisection method



The black-box model: an illustration 1

$\min_{x \in [a,b]} f(x)$

- $f : [a, b] \to \mathbb{R}$ is convex;
- $\max\{f(x) - f(y) : x, y \in [a, b]\} \leq V$;
- oracle: $x \mapsto (f(x), g(x))$;
- absolute accuracy ϵ.

1. The bisection method

Init.: $l_0 := a$, $u_0 := b$, $d_0 := 1$, $k := 0$

while $d_k \geq \epsilon/V$ **do**: $x_k := [u_k - l_k]/2$, $d_{k+1} := d_k/2$.

if $g(x_k) > 0$ **then** $l_{k+1} := l_k$, $u_{k+1} := x_k$.

if $g(x_k) < 0$ **then** $l_{k+1} := x_k$, $u_{k+1} := u_k$.

if $g(x_k) = 0$ **then** $d_{k+1} := 0$.

end;

Output the best x_k seen so far.

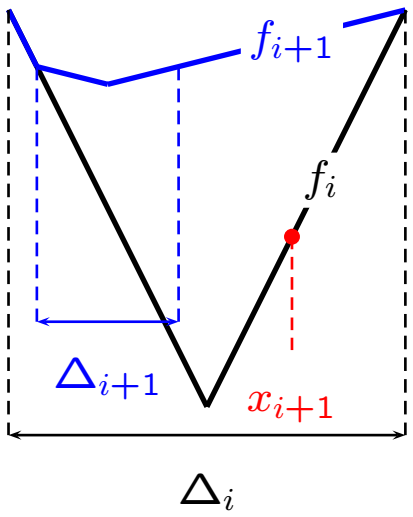
Complexity: at most $\lceil \log_2(V/\epsilon) \rceil$ calls of the oracle.

The black-box model: an illustration 2

$\min_{x\in[a,b]} f(x)$

- $f : [a, b] \to \mathbb{R}$ is convex;
- $\max\{f(x) - f(y) : x, y \in [a, b]\} \le V$;
- oracle: $x \mapsto (f(x), g(x))$;
- absolute accuracy ϵ.

2. A revoltingly hard function (Nemirovski)



Recursive construction

$f_0(x) = |x|$, $\Delta_0 = [-1, 1]$.

- No point of $\{x_1, \ldots, x_i\}$ is in Δ_i.
- length$(\Delta_i) \ge 2/4^i$.
- Outside Δ_i, $f_i = f_{i+1} = \cdots$
  Thus, convexity is preserved.
- $x_{i+1} \notin \Delta_{i+1} \ni \arg\min f_{i+1}(x)$.
- The slope of $f_i(x)$ on Δ_i is $1/8^i$.
- $f(x_i) - f_i^* > 1/32^i$.

In summary, we know exactly what to do in dimension 1

- Finding an ϵ-solution to $\min_{x\in[-1,1]} f(x)$,

where $\max\{f(x) - f(y) : x, y \in [-1,1]\} \leq 1$ takes:

$$\left\lfloor \frac{1}{5} \log_2(1/\epsilon) \right\rfloor \leq N \leq \lceil \log_2(1/\epsilon) \rceil \quad \text{iterations.}$$

- Nothing essentially beats the bisection method.
- Can be generalized to any dimension

(*cutting-plane methods*): complexity is bounded from above and below by multiples of $n \log(1/\epsilon)$.

However, these methods lack robustness (**blackboard**).

The gradient method

for unconstrained convex problems

$\min_{x\in\mathbb{R}^n} f(x)$

- $\star$ $f:\mathbb{R}^n\to\mathbb{R}$ is convex differentiable;
- $\star$ Oracle: $x\mapsto(f(x),f'(x))$

▶ **Gradient and steepest descent direction**

The *local decrease* of $f(x)$ along s is

$$\Delta(s):=\lim_{t\downarrow 0}\frac{f(x+ts)-f(x)}{t}=\langle f'(x),s\rangle,$$

minimized in $\{s:||s||=1\}$ for $s^*:=-f'(x)/||f'(x)||$.

▶ **Gradient method goes in the steepest descent**

Start from x_1, choose step-sizes $(h_k)_{k\geq 1}\subset\mathbb{R}_{++}$.

For $k\geq 1$ compute $x_{k+1}=x_k-h_kf'(x_k)$.

The gradient method converges

Idea: check how some well-chosen *potential* evolves between iterations.

Let x^* be a minimizer. We take $x \mapsto ||x - x^*||_2^2$ as potential.
As $x_{k+1} = x_k - h_k f'(x_k)$, we have:

$$||x_{k+1} - x^*||_2^2 = ||x_k - x^*||_2^2 - 2h_k\langle f'(x_k), x_k - x^*\rangle + h_k^2||f'(x_k)||_2^2.$$

By convexity of f, $\langle f'(x_k), x^* - x_k\rangle \leq f(x^*) - f(x_k)$. Thus:

$$2h_k[f(x_k) - f(x^*)] \leq ||x_k - x^*||_2^2 - ||x_{k+1} - x^*||_2^2 + h_k^2||f'(x_k)||_2^2.$$

Summing up from $k = 1$ to $k = N$,

$$\sum_{k=1}^{N} 2h_k[f(x_k) - f(x^*)] \leq ||x_0 - x^*||_2^2 - ||x_{N+1} - x^*||_2^2 + \sum_{k=1}^{N} h_k^2||f'(x_k)||_2^2,$$

If $||f'(x)||_2 \leq L$, $D := ||x_0 - x^*||_2$, $\bar{x}_N = \sum_{k=1}^N h_k x_k / \sum_{k=1}^N h_k$:

$$f(\bar{x}_N) - f(x^*) \leq \frac{D^2 + L^2 \sum_{k=1}^N h_k^2}{2\sum_{k=1}^N h_k}.$$

If $\sum_{k=1}^N h_k \to +\infty$ and $\sum_{k=1}^N h_k^2 < c$ as $N \to \infty$, the rhs goes to 0.

How fast can it converge?

$$f(\bar{x}_N) - f(x^*) \leq \frac{D^2 + L^2 \sum_{k=1}^N h_k^2}{2 \sum_{k=1}^N h_k}.$$

Fix N, and find h_k that make the rhs the smallest.

The rhs is convex in $h_k > 0$ for all k (**exercise**).

The minimum is attained in $h_k := D/(L\sqrt{N})$ for all k.

As $f(\bar{x}_N) - f(x^*) \leq LD/\sqrt{N}$, we need $N = \lceil (LD/\epsilon)^2 \rceil$ it.

Note: for f non-differentiable, we can use $g(x_k) \in \partial f(x_k)$ instead of $f'(x_k)$. The convergence analysis and the results are **completely the same**.

This method is the *subgradient algorithm*.

Comparing with the universal method, we solve $\min_{x \in [0,1]^{16}} f(x)$ with $\epsilon = 0.01$ in 16,000 oracle calls (vs. $1.5 \cdot 10^{27}$).

What happens for constrained problems?

An extremely fruitful reformulation

Observe that $x_{k+1} := x_k - h_k f'(x_k)$ is also:

$$x_{k+1} := \arg\min_{x \in \mathbb{R}^n} f(x_k) + \langle f'(x_k), x - x_k \rangle + \frac{1}{2h_k} ||x - x_k||_2^2.$$

(Just look at the optimality condition).

For $\min\{f(x) : x \in Q\}$, where $Q \subset \mathbb{R}^n$ is convex, we define:

$$x_{k+1} := \arg\min_{x \in Q} f(x_k) + \langle f'(x_k), x - x_k \rangle + \frac{1}{2h_k} ||x - x_k||_2^2.$$

Note: We have $x_{k+1} = \pi_Q(x_k - h_k f'(x_k))$.

Proof: The KKT optimality conditions give:

$$\langle f'(x_k) + (x_{k+1} - x_k)/h_k, y - x_{k+1} \rangle \geq 0 \quad \forall y \in Q.$$

For the problem $\min_{x \in Q} ||x - [x_k - h_k f'(x_k)]||_2^2$,

the optimality conditions are the same.

Both problems have a unique solution, therefore these coincide.

The convexity analysis gives the same result

$\min_{x \in Q} f(x)$

- $Q \subseteq \mathbb{R}^n$ is convex.
- Oracle: $x \mapsto (f(x), f'(x))$, and $y \mapsto \pi_Q(y)$.

We use the same reasoning as for the unconstrained gradient method (same potential function), except we need:

$$||x_{k+1} - x^*||_2^2 = ||\pi_Q(x_k - h_k f'(x_k)) - x^*||_2^2 \leq ||x_k - h_k f'(x_k) - x^*||_2^2.$$

(**exercise**: prove this inequality)

Again, we need $\mathcal{O}((LD/\epsilon)^2)$ oracle calls.

Can we hope for a faster convergence?

Theorem 2 (Nemirovski, Yudin) *Let* $D, L > 0$.
There exists a convex function $f : \mathbb{R}^n \to \mathbb{R}$ *such that* $||f'(x)||_2 \leq L$ *and for which* $\min\{f(x) : ||x||_2 \leq D\}$ *takes at least* $\min\{n-1, (LD/4\epsilon)^2\}$ *oracle calls to be minimized up to an* ϵ *absolute accuracy.*

In other words, with first-order information only,
you **cannot hope** beating the subgradient algorithm
for general convex problems.

Note: Some first-order black-box algorithms
outperform the subgradient algorithm in practice,
but it is impossible to improve their guaranteed
worst-case convergence.

The proof of Nesterov

Let $x[1], \ldots$ be the points generated by the optimization method. The starting point $x[0]$ is 0. We denote $g_i \in \partial f(x[i])$ the oracle answer.

1. We assume that $x[i+1]$ is a linear combination of $g_0, \ldots, g_i$.

2. Let $f_k(x) := \gamma \max_{1 \leq i \leq k} x_i + \frac{\mu}{2}||x||^2$ and $x^*[k] := \arg\min f_k(x)$.
We have $\partial f_k(x) = \mu x + \gamma \text{conv}\{e_j : j \in I(x) := \arg\max_{1 \leq i \leq k} x_i\}$.
We assume that $g_i = \mu x + e_{i^*}$ with $i^* = \min\{i \in I(x)\}$.
Note that $x_i^*[k] = -\gamma/\mu k$ if $1 \leq i \leq k$, and $x_i^*[k] = 0$ otherwise.
Thus $||x_i^*[k]||_2 = \gamma/(\mu\sqrt{k})$ and $\min f_k(x) = -\gamma^2/(2\mu k)$.
Finally, f_k is Lipschitz continuous on $B[0, R]$ with constant $\mu R^2 + \gamma$.

3. Let us fix $k < n$ and try to solve $\min f_k(x)$.
Observe that only the p first components of g_p are nonzero when $p < k$, thus $f_k(x[p]) \geq \gamma \max_{1 \leq i \leq k} x_i[p] = 0$, and $f_k(x[p]) - f_k(x^*[k]) = -\gamma^2/(2\mu k)$.
Let $\gamma := \frac{L\sqrt{k+1}}{1+\sqrt{k+1}}$ and $\mu := \frac{L}{D(1+\sqrt{k+1})}$; f_k is L-Lipschitz continuous, the iterates are all in $B[0, D]$, and

$$f_k(x[p]) - f_k(x^*[k]) \geq LD/2(1 + \sqrt{k+1}) \geq LD/4\sqrt{k+1}.$$

Computing gradients approximately without harming convergence too much?

A peek inside the box.

1. Suppose that $f(x) = \sum_{i=1}^{N} f_i(x)$ for a very large N.

Example: Approximation problem with a huge data set:

$f(x) = \sum_{i=1}^{N} \phi(r_i(x))$ (see previous lecture).

Computing the gradient is costly: $f'(x) = \sum_{i=1}^{N} f_i'(x)$.

Idea: Select randomly $I \subseteq \{1, \ldots, N\}$, with $|I| \ll N$.

Use $g_I(x) := \sum_{i \in I} f_i'(x)$ instead of $f'(x)$.

Can we guarantee the convergence of the algorithm?

Computing gradients approximately without harming convergence too much?

A peek inside the box.

2. Suppose that $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
Let $F : \mathbb{R}^N \times \Omega \to \mathbb{R}$, $(x, \omega) \mapsto F(x, \omega)$ convex in x,
in $L^1(\mathbb{P})$ for all x, and let $f(x) := \mathbb{E}[F(x, \omega)]$.
Fact: $\mathbb{E}[\partial_x F(x, \omega)] \subseteq \partial f(x)$ for $x \in \operatorname{int}\operatorname{dom}(f)$
(so, $f'(x)$ can be very hard to compute).
Idea: Select randomly $\{\omega_i : i \in I\}$, with $|I|$ small.
Use $g_I(x) := \sum_{i \in I} \partial_x F(x, \omega_i)/|I|$ instead of $f'(x)$.

Can we guarantee the convergence of the algorithm?

Stochastic gradient methods work if I is selected appropriately

- **Stochastic gradient method for** $\min\{f(x) : x \in Q\}$
  Start from x_1, choose step-sizes $(h_k)_{k\geq 1} \subset \mathbb{R}_{++}$.
  For $k \geq 1$, select $g_k = g_I(x_k)$,
  and compute $x_{k+1} = \pi_Q(x_k - h_k g_k)$.
- As g_k is a random variable for every k, x_k is random too
- Define the filtration $\mathcal{F}_k := \sigma(g_1, \ldots, g_k)$, $\mathcal{F}_0 = \{\emptyset, \Omega\}$.
- **Assumptions:** for all $k \geq 1$:
  $\mathbb{E}[g_I(x_k) \,|\, \mathcal{F}_{k-1}] \in \partial f(x_k)$,
  $\mathbb{E}\left[||g_I(x_k)||_2^2 \,|\, \mathcal{F}_{k-1}\right] \leq M^2$ for an $M > 0$.

Stochastic gradient methods work if I is selected appropriately

- **Assumptions:** for all $k \geq 0$:
  $\mathbb{E}[g_k \,|\, \mathcal{F}_{k-1}] \in \partial f(x_k)$,
  $\mathbb{E}\left[||g_k||_2^2 \,|\, \mathcal{F}_{k-1}\right] \leq M^2$ for an $M > 0$.
- Taking again $x \mapsto ||x - x^*||_2^2$ as potential,

  $$||x_{k+1} - x^*||_2^2 \leq ||x_k - x^*||_2^2 - 2h_k\langle g_k, x_k - x^*\rangle + h_k^2||g_k||_2^2.$$

  Taking conditional expectations under $\mathcal{F}_{k-1}$,

  $$\begin{aligned} & \mathbb{E}\left[||x_{k+1} - x^*||_2^2 \,|\, \mathcal{F}_{k-1}\right] \\ \leq\ & \mathbb{E}\left[||x_k - x^*||_2^2 \,|\, \mathcal{F}_{k-1}\right] - 2h_k\mathbb{E}\left[\langle g_k, x_k - x^*\rangle \,|\, \mathcal{F}_{k-1}\right] + h_k^2 M^2 \\ =\ & ||x_k - x^*||_2^2 - 2h_k\langle f'(x_k), x_k - x^*\rangle + h_k^2 M^2. \end{aligned}$$

- Using the tower property for conditional expectations, we continue **exactly** as for the standard gradient method.

For next week

- **Some reasons to hope**:
  (Much) faster methods for *smooth* convex problems.

- **What if we use gradient's method on non-convex functions?**

- **What happens if the oracle also gives $f''(x)$**:
  Newton's method

- **Gradient methods are intrinsically ill-conceived**:
  Mirror descent methods as a powerful remedy
  (Application: some boosting methods)