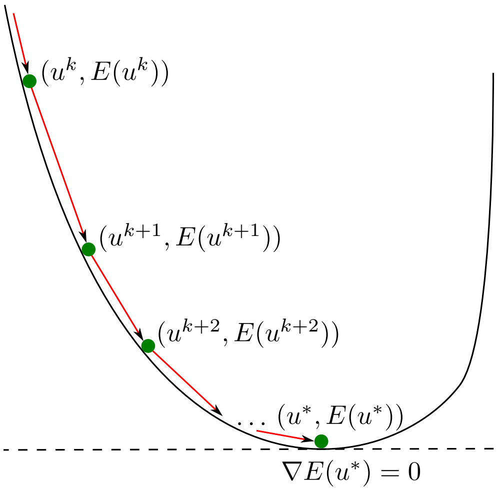
Gradient Methods

V. Estellers

WS 2017

Gradient Methods



Outline

Gradient Descent

Consider the unconstrained and smooth optimization problem

$$u^* \in \arg\min_{u \in \mathbb{R}^n} E(u),$$

for $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ proper, closed, and convex

Gradient descent is an optimization technique for the "simple" case

- dom $E = \mathbb{R}^n$
- $E \in \mathcal{C}^1(\mathbb{R}^n)$

Descent methods

Suppose we are at a point $u^k \in \mathbb{R}^n$ where $\nabla E(u^k) \neq 0$

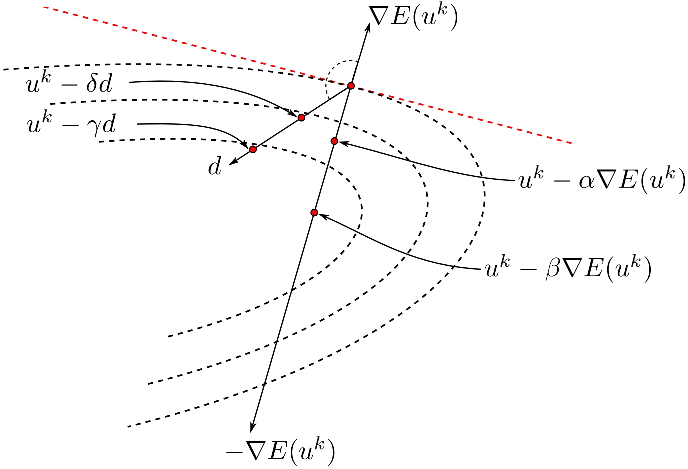
Consider the ray $u(\tau) = u^k + \tau d$ for some direction $d \in \mathbb{R}^n$

$$E(u(\tau)) = E(u^k + \tau d) = E(u^k) + \tau\langle \nabla E(u^k), d\rangle + o(\tau)$$

- $\tau\langle \nabla E(u^k), d\rangle$ dominates $o(\tau)$ for sufficiently small τ
- If $\langle \nabla E(u^k), d\rangle < 0$, d is a *descent direction* as, for suff. small τ,

$$E(u(\tau)) < E(u)$$

Descent methods



Descent methods

The negative gradient is the *steepest* descent direction

$$\underset{\|d\|=1}{\operatorname{argmin}} \;\; \left\{ \langle d, \nabla E(u^k) \rangle \right\} = -\frac{\nabla E(u^k)}{\|\nabla E(u^k)\|}$$

The gradient is orthogonal to the iso-contours $\gamma : I \to \mathbb{R}^n$

$$\nabla E(\gamma(t)) \perp \dot{\gamma}(t), \qquad t \in I$$

Common choices of descent directions

- Scaled gradient: $d^k = -D^k \nabla E(u^k)$, $D^k \succeq 0$
- Newton: $D^k = [\nabla^2 E(u^k)]^{-1}$
- Quasi-Newton: $D^k \approx [\nabla^2 E(u^k)]^{-1}$
- Steepest descent: $D^k = I$

Gradient descent

Definition

Given a function $E \in \mathcal{C}^1(\mathbb{R}^n)$, an initial point $u^0 \in \mathbb{R}^n$ and a sequence $(\tau_k) \subset \mathbb{R}$ of step sizes, the iteration

$$u^{k+1} = u^k - \tau_k \nabla E(u^k), \qquad k = 0, 1, 2, \dots,$$

is called *gradient descent*.

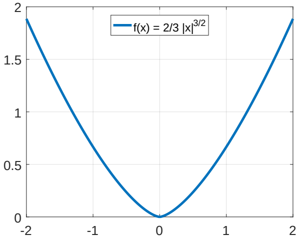
Philosophy:

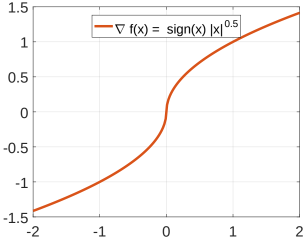
- Generate a decreasing sequence $\{E(u^k)\}_{k=0}^{\infty}$
- Each iteration is cheap, easy to code
- Choosing τ_k to guarantee convergence is not trivial

Constant step size

Consider a constant step size $\tau^k = \tau$

Will gradient descent work for any convex function?





For any constant time step $\tau > 0$, the starting point $u^0 = \left(\frac{\tau}{2}\right)^2$ results in a gradient descent sequence $u^0, -u^0, u^0, \ldots$

Intuition and requirements for constant step-size

Intuitively, an "infinitely quickly changing gradient" leads to "infinitely quickly changing" gradient descent updates

$$u^{k+1} = u^k - \tau_k \nabla E(u^k), \qquad k = 0, 1, 2, \ldots,$$

Need a stronger version of differentiability to prevent inf. quick changes

Definition: L-smooth function

If $E : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its first derivative is Liptschitz continuous, i.e. there exists an $L \geq 0$ such that

$$\|\nabla E(u) - \nabla E(v)\| \leq L \|u - v\|, \forall u, v \in \mathbb{R}^n,$$

then E is called L-smooth

Lipschitz continuity

Reminder

$f : \mathbb{R}^n \to \mathbb{R}^m$ is called Lipschitz continuous if for some $L \geq 0$

$$\|f(x) - f(y)\| \leq L \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

If the function is differentiable, we can characterize Lipschitz continuous functions by the size of its gradient.

Theorem: Lipschitz continuity for differentiable functions

A differentiable function $E : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz with parameter L if and only if $\|\nabla E(x)\|_{S^\infty} \leq L$ for all $x \in \mathbb{R}^n$.

Convergence Analysis

Conjecture

For any L-smooth proper convex function E (with a minimizer) there exists a step size τ such that the gradient descent algorithm converges

To prove this conjecture, we will use a general **fixed-point Iteration** for algorithms of the form

$$u^{k+1} = G(u^k)$$

Example:

$$G(u) = u - \tau \nabla E(u).$$

If the iteration converges to $\hat{u}$ and ∇E is continuous, then $\nabla E(\hat{u}) = 0$.

Outline

Convergence of Fixed-Point Iterations

References:

Ryu and Boyd, *Primer on Monotone Operator Methods*, 2016.

Burger, Sawatzky, and Steidl, *First Order Algorithms in Variational Image Processing*, 2017.

Bauschke, and Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2011.

Fixed-point iterations with contractions

When does the fixed-point iteration

$$u^{k+1} = G(u^k) \tag{1}$$

converge?

Banach fixed-point theorem

If the update rule $G : \mathbb{R}^n \to \mathbb{R}^n$ is a **contraction**, i.e. if there exists a $L < 1$ such that

$$\|G(u) - G(v)\|_2 \leq L\|u - v\|_2$$

holds for all $u,\ v \in \mathbb{R}^n$, then the iteration (1) converges to the unique fixed-point $\hat{u}$ of G. More precisely,

$$\|u^k - \hat{u}\|_2 \leq L^k\|u^0 - \hat{u}\|_2.$$

Fixed-point iterations with averaged operators

G being a **contraction** is **too restrictive** in many cases

G being **non-expansive**, i.e. Lipschitz continuous with constant $L = 1$, is commonly true.

- any rotation G is non-expansive and has a fixed point (0)
- the iteration $u^{k+1} = G(u^k)$ does not converge

Averaged operator

An operator $G : \mathbb{R}^n \to \mathbb{R}^n$ is called **averaged** if there exists a non-expansive mapping $H : \mathbb{R}^n \to \mathbb{R}^n$ and a constant $\alpha \in (0, 1)$ such that

$$G = \alpha I + (1 - \alpha)H.$$

Criteria for being averaged

Lemma about nonexpansive operators

Convex combinations as well as compositions of nonexpansive operators are nonexpansive.

Being averaged for smaller α

If a function $G : \mathbb{R}^n \to \mathbb{R}^n$ is averaged with respect to $\alpha \in]0, 1[$, then it is also averaged with respect to any other parameter $\tilde{\alpha} \in]0, \alpha[$.

Composition of averaged operators

If $G_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $G_2 : \mathbb{R}^n \to \mathbb{R}^n$ are averaged, then $G_2 \circ G_1$ is also averaged.

Proofs: Notes

Criteria for being averaged

Firmly non-expansive

A function $G : \mathbb{R}^n \to \mathbb{R}^n$ is called **firmly nonexpansive**, if for all $u, v \in \mathbb{R}^n$ it holds that

$$\|G(u) - G(v)\|_2^2 \leq \langle G(u) - G(v), u - v \rangle.$$

Firmly nonexpansive operators are averaged

A function $G : \mathbb{R}^n \to \mathbb{R}^n$ is firmly nonexpansive if and only if G is averaged with $\alpha = \frac{1}{2}$.

Proof: Notes

Convergence for averaged operators

Krasnosel'skii-Mann Theorem

If the operator $G : \mathbb{R}^n \to \mathbb{R}^n$ is averaged and has a fixed-point, then the iteration

$$u^{k+1} = G(u^k)$$

converges to a fixed point of G for any starting point $u^0 \in \mathbb{R}^n$.

Proof: Notes

Short summary

We have seen:

- An operator G is called a **contraction** if it is Lipschitz continuous with $L < 1$.
- Contractions have a unique fixed-point and their fixed-point iteration converges with $\mathcal{O}(L^k)$.
- An operator R is called a **nonexpansive** if it is Lipschitz continuous with $L = 1$.
- An operator G is called a **averaged** if $G = \alpha I + (1 - \alpha)R$ for some nonexpansive operator R and $\alpha \in (0, 1)$.
- If an averaged operator has a fixed-point, then the fixed-point iteration converges. The convergence rate states that $\sum_{k=1}^{n} \|G(u^k) - u^k\|_2 \leq C$ for some constant C.
- **Firmly nonexpansive** operators are the same as averaged operators with $\alpha = \frac{1}{2}$.

Relation to gradient descent

We now have two loose ends:

- a conjecture about the convergence of the gradient descent iteration
- theorem that states the convergence of a fixed-point iteration for averaged operators.

we need to write gradient descent as an averaged operator

Baillon-Haddad theorem

A continuously differentiable convex function $E : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if and only if $\frac{1}{L}\nabla E$ is firmly nonexpansive, i.e.

$$\langle \nabla E(u) - \nabla E(v), u - v \rangle \geq \frac{1}{L} \left\| \nabla E(u) - \nabla E(v) \right\|_2^2$$

for all $u, v \in \mathbb{R}^n$.

Proof: See Nesterov, *Introductory Lectures on Convex Optimization*, Theorem 2.1.5.

Outline

Convergence of gradient descent

Gradient descent as an averaged operator

If $E : \mathbb{R}^n \to \mathbb{R}$ has a minimizer, is convex and L-smooth, and $\tau \in]0, \frac{2}{L}[$, then the gradient descent iteration converges to a minimizer.

Sufficient: $G(u) = u - \tau\nabla E(u)$ is averaged.

We know $\frac{1}{L}\nabla E$ is averaged with $\alpha = 1/2$, i.e., $\frac{1}{L}\nabla E = \frac{1}{2}(I + T)$ for a non-expansive T.

It hold that

$$G(u) = u - \tau L \frac{1}{L}\nabla E(u) = \left(1 - \frac{L\tau}{2}\right) I + \frac{L\tau}{2}(-T)$$

If T is non-expansive, $(-T)$ is non-expansive, too.
$\Rightarrow$ For $\tau \in]0, \frac{2}{L}[$, G is averaged.

Convergence rate

How fast does gradient descent converge?

Theory of averaged operators shows $\sum_k \|\nabla E(u^k)\|_2^2$ is bounded.

Careful analysis shows that for L-smooth functions with $\tau \in (0, \frac{2}{L})$:

$$E(u^{k+1}) \leq E(u^k) \qquad E(u^k) - E(u^*) \in \mathcal{O}(1/k)$$

.

It is not possible to get a contraction to speed up convergence because a contraction would imply the existence of a unique fixed-point.

Reminder

$\mathcal{O}(g) = \{f \mid \exists C \geq 0, \exists n_0 \in \mathbb{N}_0, \forall n \geq n_0 : |f(n)| \leq C|g(n)|\}$

Back to GD

Strongly-convex + L-smooth

Gradient descent as an averaged operator

If $E : \mathbb{R}^n \to \mathbb{R}$ is m-strongly convex and L-smooth, and $\tau \in]0, \frac{2}{m+L}[$, then the gradient descent iteration converges to the unique minimizer u^* of E with $\|u^k - u^*\| \leq c^k \|u^0 - u^*\|$.

Proof on the Notes.

Strong convexity

Definition: strong convexity

A function $E : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called *strongly convex* with constant m or m-strongly convex if $E(u) - \frac{m}{2}\|u\|_2^2$ is still convex.

Theorem: characterization of m-strongly convex functions [1]

For $E \in \mathcal{C}^1(\mathbb{R}^n)$ the following are equivalent:

1. $E(u) - \frac{m}{2}\|u\|^2$ is convex
2. $E(v) \geq E(u) + \langle \nabla E(u), v - u \rangle + \frac{m}{2}\|v - u\|^2$
3. $\langle \nabla E(u) - \nabla E(v), u - v \rangle \geq m\|u - v\|^2$
4. $\nabla^2 E(u) \succeq m \cdot I$, if $E \in \mathcal{C}^2(\mathbb{R}^n)$

[1] Ryu, Boyd, A Primer on Monotone Operator Methods, Appendix A

Back to GD

Optimal convergence rates

In computer vision, m-strongly convex L-smooth energies are very rare! Can one do better than the $\mathcal{O}(1/k)$ in the L-smooth case?

Famous analysis by Nesterov, (Th 2.1.7 and Th2.1.13) for first order methods of the form:

$$u^{k+1} \in u^0 + \mathsf{span}\{\nabla E(u^0), \ldots, \nabla E(u^k)\}$$

If E can be any convex L-smooth function

> then no first order method can have a worst-case complexity less than $\mathcal{O}(1/k^2)$.
>
> and E is m-strongly convex, then no first order method can have a worst-case complexity less than $\mathcal{O}((\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^{2k})$ for $\kappa = L/m$.

Back to GD

Obtaining optimal convergence rates

Nesterov's Accelerated Gradient Descent

Pick some starting point $v^0 = u^0$, and iterate

1. Compute
$$u^{k+1} = v^k - \frac{1}{L}\nabla E(v^k)$$
2. Find the next $\alpha \in]0,1[$ by solving
$$\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \frac{m}{L}\alpha_{k+1}$$
3. Compute the extrapolation of u^{k+1} via
$$\beta_k = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$$
$$v^{k+1} = u^{k+1} + \beta_k(u^{k+1} - u^k)$$

Backtracking line search

Sometimes Lipschitz constant L not known

The convergence analysis shows that one really only needs

$$E(u^{k+1}) \leq E(u^k) - \beta_k \|\nabla E(u^k)\|^2$$

for some $\beta_k \geq \beta > 0$.

Idea: Pick $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$

Then determine τ_k each iteration by:

$\tau_k \leftarrow 1$

while $E\left(u^k - \tau_k \nabla E(u^k)\right) > E(u^k) - \alpha \tau_k \left\|\nabla E(u^k)\right\|^2$

$\quad \tau_k \leftarrow \beta \tau_k$

end

Backtracking line search

Line search...

... often leads to improved convergence in practice

... has a (slight) overhead each iteration

... has the same convergence rate as with constant steps

For a backtracking line search scheme for Nesterov's accelerated gradient method please see *Introductory Lectures on Convex Optimization*, page 76, scheme (2.2.6).

Remark: Other strategies for linear search exists, e.g.

$$\tau_k = \arg\min_{\tau} E(u^k - \tau \nabla E(u^k))$$

Back to GD

Application: TV image denoising

Lets consider the applications of image denoising:

Via energy minimization: Let D_1 and D_2 be finite difference operators for the partial derivatives. Determine

$$\hat{u} \in \arg\min_u \quad \underbrace{\frac{\lambda}{2}\|u-f\|_2^2}_{=H_f(u)\text{stay close to input}} \quad + \underbrace{\sum_{x\in\Omega}\sqrt{(D_1u(x))^2+(D_2u(x))^2}}_{=TV(u)\text{ suppress noise}}$$

Application: TV image denoising

Problem: The so called *total variation regularization*

$$TV(u) = \sum_{x \in \Omega} \sqrt{(D_1 u(x))^2 + (D_2 u(x))^2}$$

is not differentiable!

Idea: Approximate it with a differentiable function

$$TV_\epsilon(u) = \sum_{x \in \Omega} \phi \sqrt{(D_1 u(x))^2 + (D_2 u(x))^2 + \epsilon^2}$$

Exercises: Our denoising model is L-smooth for

$$L = \lambda + \frac{\|D\|_{S^\infty}}{\epsilon}$$

where $\|D\|_{S^\infty}$ is the spectral norm of a matrix. It is defined as the square root of largest eigenvalue of $D^T D$.

We expect the convergence to be better for large ϵ, but we expect $TV(u) \approx TV_\epsilon(u)$ only for small ϵ...

Image denoising

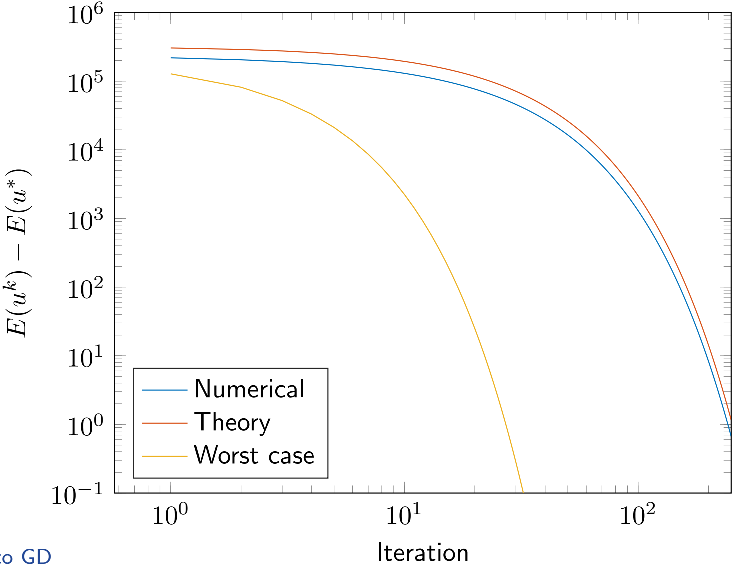
$$\varepsilon = 0.1$$

$$\varepsilon = 0.01$$

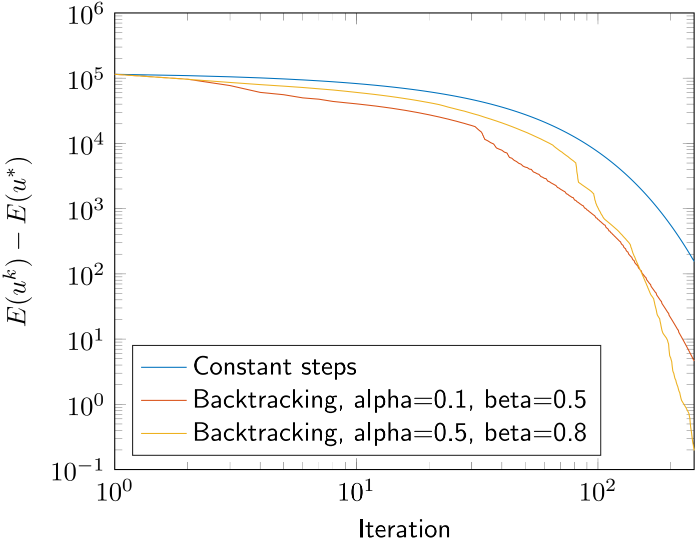
$\rightarrow$ *Motivation for non-smooth optimization!*

Back to GD

Convergence, $\tau = 2/(m+L)$

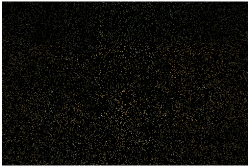


Convergence, backtracking line search

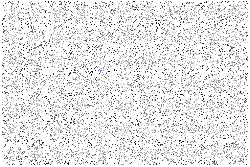


Back to GD

Image inpainting



$f \in \mathbb{R}^N$



$1 - m \in \mathbb{R}^N$

$u^* \in \mathbb{R}^N$

$$u^* \in \operatorname*{argmin}_u \frac{\lambda}{2} \|m \cdot (u - f)\|^2 + TV_\epsilon(u)$$

Energy is not strongly convex, but L-smooth

Sublinear upper bound on convergence speed

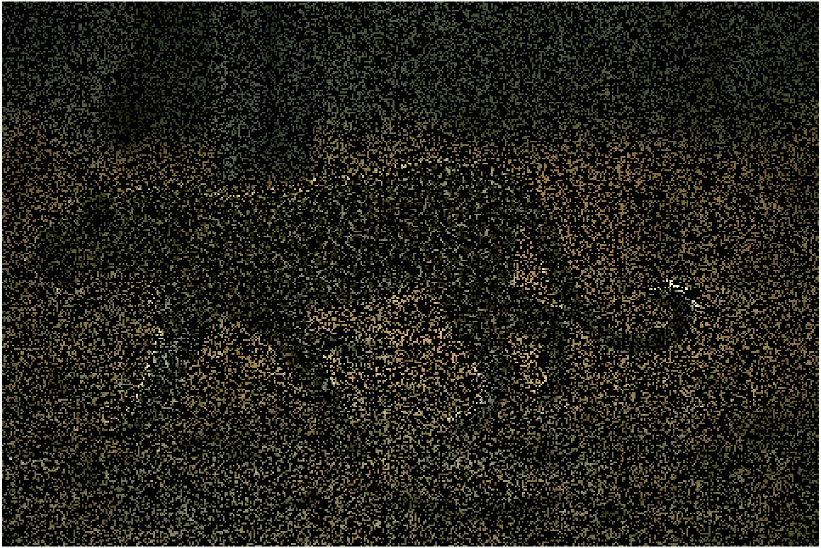
Back to GD

Image Inpainting

50% missing pixels

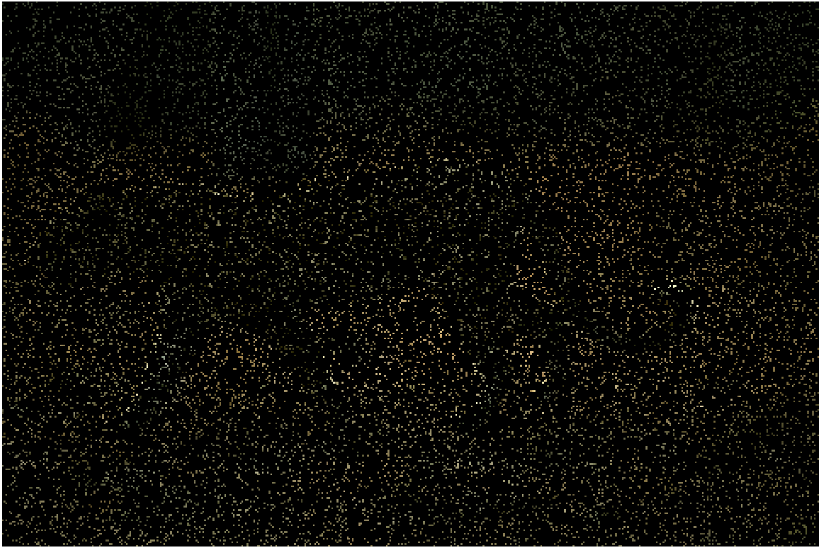
50% missing pixels

70% missing pixels



70% missing pixels

90% missing pixels



90% missing pixels

Concluding remarks and outlook

GD is still popular to date due to its simplicity and flexibility

Various theoretically optimal extensions (Heavy-ball acceleration, Nesterov momentum) exist

Envelope approach: many advanced algorithms for non-smooth optimization are just gradient descent on a particular (albeit complicated) energy

Endless of variants and modifications of descent methods

conjugate, accelerated, preconditioned, projected, conditional, mirrored, stochastic, coordinate, continuous, online, variable metric, subgradient, proximal, ...

Subgradient descent in one slide

We have seen in the exercises, that even for functions that are not L-smooth, gradient descent with a small step size reduces the energy up to some point where it starts oscillating.

Possible convergent variant: **Subgradient descent**

$$u^{k+1} = u^k - \tau_k p^k, \qquad \text{for any } p^k \in \partial E(u^k).$$

If it holds that

- E has a minimizer
- E is Lipschitz continuous
- $\tau_k \to 0$, but $\sum_{k=1}^{n} \tau_k \to \infty$, e.g. $\tau_k = 1/k$

then the subgradient descent iteration converges with

$$E(u^k) - E(u^*) \in \mathcal{O}(1/\sqrt{k})$$

Summary

This lecture is about

$$u^* \in \arg\min_{u \in \mathbb{R}^n} E(u),$$

for $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ proper, closed, convex.

Gradient descent:

dom $E = \mathbb{R}^n$

For L-smooth E (that has a minimizer)

- energy convergence in $\mathcal{O}(1/k)$ for constant step sizes
- energy convergence in $\mathcal{O}(1/k^2)$ for Nesterov's method.

For L-smooth m-strongly convex E: energy and iterate convergence in $\mathcal{O}(c^k)$

Line search strategies for unknown Lipschitz constant L.

Back to GD

Outline

Gradient projection

Type of problem:

$$u^* \in \arg\min_{u \in C} E(u), \tag{2}$$

for an L-smooth E, and a nonempty, closed, convex set C.

Definition

Projection For a (nonempty) closed convex set $C \subset \mathbb{R}^n$,

$$\pi_C(v) = \operatorname*{argmin}_{u \in C} \|u - v\|_2^2$$

is called the projection of v onto the set C.

Projections

Theorem

Existence and Uniqueness of the Projection For any (nonempty) closed convex set $C \subset \mathbb{R}^n$ and any v the projection $\pi_C(v)$ exists and is single valued.

Proof: Notes.

Abuse of notation: Although $\pi_C(v)$ is, by definition, a set, we usually identify $\pi_C(v)$ with the single element in the set.

Example projections

What is the projection of $v \in \mathbb{R}^n$ onto

$C = \{u \in \mathbb{R}^n \mid \|u\|_2 \leq 1\}$?

$C = \{u \in \mathbb{R}^n \mid \|u\|_\infty := \max_i |u_i| \leq 1\}$?

$C = \{u \in \mathbb{R}^n \mid u_i \in [a, b]\}$?

$C = \{u \in \mathbb{R}^n \mid u_i \geq a\}$?

$C = \{u \in \mathbb{R}^n \mid \|u\|_1 = \sum_i |u_i|\}$?

Intuition on gradient projection

Let E be L-smooth convex function and C a nonempty, closed, convex set. Consider a problem

$$u^* \in \arg\min_{u \in C} E(u), \tag{3}$$

We know that, without the constraint $u \in C$, gradient descent works and looks like:

$$u^{k+1} = u^k - \tau^k \nabla E(u^k)$$

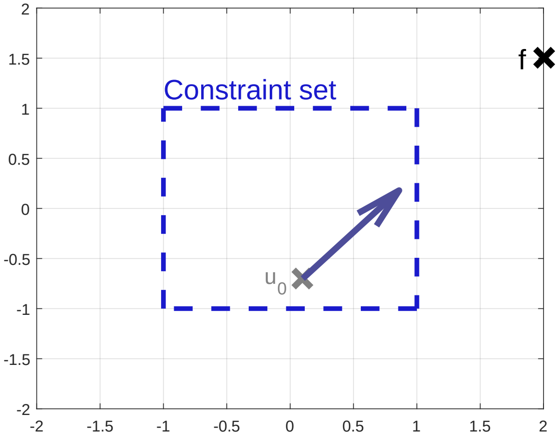
The problem with GD is that the update might violate $u^{k+1} \in C$

Gradient projection solves this by **projecting every iteration back to the feasible set**

$$u^{k+1} = \pi_C(u^k - \tau^k \nabla E(u^k))$$

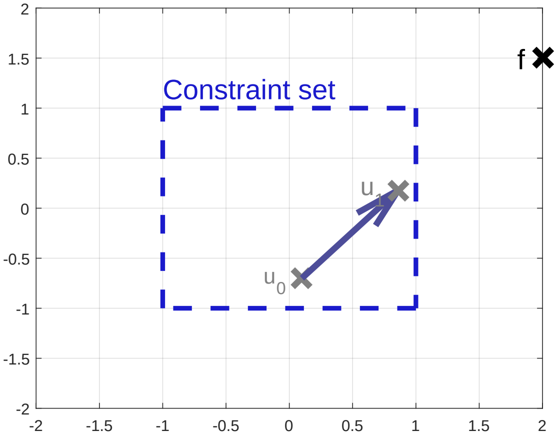
Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u - f\|_2^2$



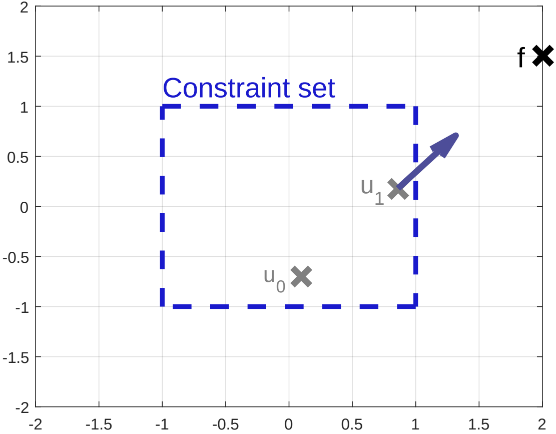
Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u - f\|_2^2$



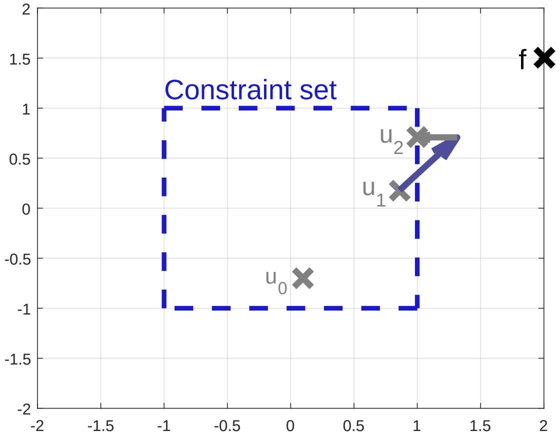
Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u - f\|_2^2$



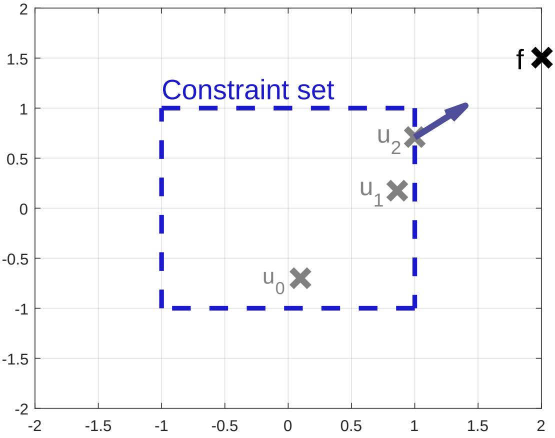
Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u-f\|_2^2$



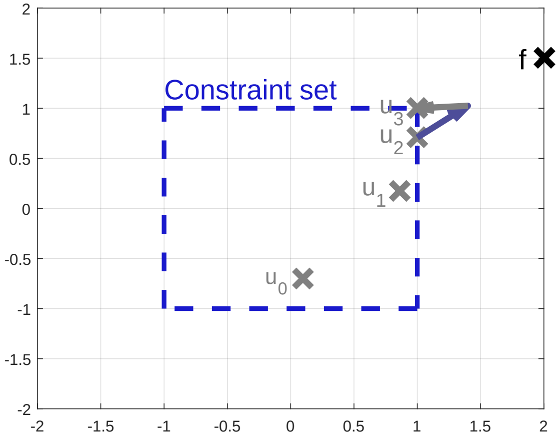
Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u - f\|_2^2$



Intuition on gradient projection

Toy problem $\min_{|u_i|\leq 1} \|u - f\|_2^2$



Gradient Projection Algorithm

Definition

Gradient Projection Algorithm Let $C \subset \mathbb{R}^n$ be a nonempty closed convex set and let $E : \mathbb{R}^n \to \mathbb{R} \in C^1(\mathbb{R}^n)$. Then, for $u^0 \in C$

$$u^{k+1} = \pi_C(u^k - \tau \nabla E(u^k))$$

is called the *gradient projection* algorithm.

Before we spend time implementing it, we need to know *when, how, and why* it works, i.e., for which E and τ the **fixed-point iteration**

$$G(u) = \pi_C(u - \tau \nabla E(u))$$

converges

Projected GD as a fixed-point iteration

Strategy: show that the fixed point iteration

$$G(u) = \pi_C(u - \tau \nabla E(u))$$

converges because G is an averaged operator

From the analysis of gradient descent, we know:

1. for $\tau \in (0, \frac{2}{L})$ the operator $G_1(u) = u - \tau \nabla E(u)$ is averaged
2. the composition of averaged operators is averaged

If we can show that π_C is averaged, we are done

Properties of the projection

Theorem

Firm Nonexpansiveness The projection π_C onto a nonempty closed convex set $C \subset \mathbb{R}^n$ is firmly nonexpansive, i.e. it meets

$$\langle u - v, \pi_C(u) - \pi_C(v) \rangle \geq \|\pi_C(u) - \pi_C(v)\|^2 \qquad \forall u, v \in \mathbb{R}^n.$$

Remember that a firmly non-expansive operator is averaged with $\alpha = \frac{1}{2}$

Corollary

For an L-smooth energy E that has a minimizer and a choice $\tau \in]0, \frac{2}{L}[$ the gradient projection converges with rate rate is $\mathcal{O}(1/k)$

$\mathcal{O}(1/k)$ is suboptimal, a generalized version with $\mathcal{O}(1/k^2)$ comes later

Convergence of the projected gradient descent

Recall: *The composition of a non-expansive operator with a contraction is a contraction*

This means that our gradient descent result carries over:

Theorem

For E being L-smooth and m-strongly convex and $\tau \in (0, \frac{2}{L})$ the gradient projection algorithm converges to the (unique) global minimizer u^* with $E(u^k) - E(u^*) \in \mathcal{O}(c^k)$ with $c < 1$

Example Application: Solving a SUDOKU

Find the missing numbers such that each block, each row, and each column contains each number 1– 4 only once

2			3
1	3		
		3	2
	2	4	

Example Application: Solving a SUDOKU

Find the missing numbers such that each block, each row, and each column contains each number 1– 4 only once

2	4	1	3
1	3	2	4
4	1	3	2
3	2	4	1

Example Application: Solving a SUDOKU

Find the missing numbers such that each block, each row, and each column contains each number 1– 4 only once

2	4	1	3
1	3	2	4
4	1	3	2
3	2	4	1

We can do this with convex optimization?

Example Application: Solving a SUDOKU

Find the missing numbers such that each block, each row, and each column contains each number 1– 4 only once

2	4	1	3
1	3	2	4
4	1	3	2
3	2	4	1

We can do this with convex optimization?

Idea: Identify the number i with

$$e_i = (0, \cdots, 0, \underbrace{1}_{i^{th}\text{ position}}, 0, \cdots, 0)^T.$$

Example Application: Solving a SUDOKU

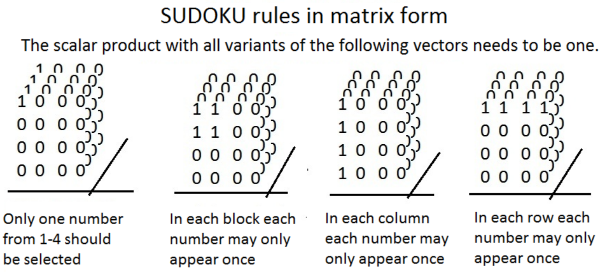
For the 4×4 case, look for a matrix $u \in \{1, 2, 3, 4\}^{4\times4}$ such that $u_{i,j} = f_{i,j}$ for the entries $f_{i,j}$ that are given

Reformulation: find $\boldsymbol{u} \in \{0, 1\}^{4\times4\times4}$, where $\boldsymbol{u}_{i,j,k} = 1$ means $u_{i,j} = k$, subject to the constraints

Rule	**Implication**	
One number for each blank spot	$\sum_k \boldsymbol{u}_{i,j,k} = 1$	$\forall i, j$
Respect given entries	$\boldsymbol{u}_{i,j,k} = 1$ if $f_{i,j} = k$	
Numbers occur in a row once	$\sum_j \boldsymbol{u}_{i,j,k} = 1$	$\forall i, k$
Numbers occur in a column once	$\sum_i \boldsymbol{u}_{i,j,k} = 1$	$\forall j, k$
Numbers occur in a block once	$\sum_{(i,j)\in B_l} \boldsymbol{u}_{i,j,k} = 1$	$\forall B_l, k$

Example Application: Solving a SUDOKU

All constraints are linear, i.e. can be expressed as $A\vec{u} = \vec{1}$.



Find $\boldsymbol{u}$ with $\boldsymbol{u}_{i,j,k} \in \{0,1\}$ is a nonconvex constraint, so we *relax* it.

Convex relaxation: use the smallest convex set that contains the nonconvex one, $\boldsymbol{u}_{i,j,k} \in [0,1]$. Solve the convex problem and if the result meets $\boldsymbol{u}_{i,j,k} \in \{0,1\}$, it also solves the nonconvex problem

Example Application: Solving a SUDOKU

Nice thing for SUDOKU: There exists a solution to $A\vec{\boldsymbol{u}} = \vec{\mathbf{1}}$

This means we may solve

$$\hat{\boldsymbol{u}} \in \underset{\boldsymbol{u}_{i,j,k} \in [0,1]}{\operatorname{argmin}} \|A\vec{\boldsymbol{u}} - \vec{\mathbf{1}}\|_2^2$$

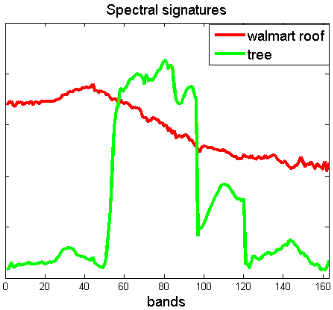
Hope that $\hat{\boldsymbol{u}}_{i,j,k} \in \{0,1\}$, in which case we solved the SUDOKU

Remarks:

- Exact recovery guarantees (when is $\hat{\boldsymbol{u}}_{i,j,k} \in \{0,1\}$) are an active field of research.
- Similar constructions can be done for many computer vision and machine learning problems (*labeling problems*, *segmentation*, *graph cuts*, or *functional lifting*)

Example application: Unmixing and sparse recovery

Hyperspectral imagery



z-direction: reflected energy depending on the wavelength of the incoming light. It is material specific.

Example application: Unmixing and sparse recovery

Measured signals f

Find decomposition $f = Au + n$

Dictionary of materials A, mixing coefficients u (sparse) and noise n

Example application: Unmixing and sparse recovery

Sparse recovery: Minimize a data fidelity term $H_f(v)$ which is L-smooth, such that v can be represented in a dictionary A, i.e. $v = Au$, and the representing coefficients u are sparse.

Energy minimization approach:

$$\min_u H_f(Au) + \alpha \|u\|_1.$$

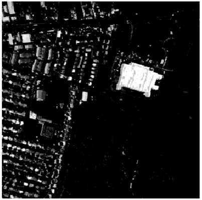
To apply gradient descent or projection algorithms, we need to reformulate the problem

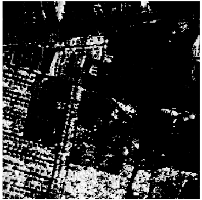
$$\min_u H_f(A(u_1 - u_2)) + \alpha\langle u_1, \mathbf{1}\rangle + \alpha\langle u_2, \mathbf{1}\rangle, \quad u_1 \geq 0, u_2 \geq 0$$

Example application: Unmixing and sparse recovery


color image illustration


endmember "road"


endmember "roof"


endmember "trees"

The reformulation

$$\min_{u} H_f(Au) + \alpha\|u\|_1,$$

$$\min_{u_1,u_2} H_f(A(u_1 - u_2)) + \alpha\langle u_1, \mathbf{1}\rangle + \alpha\langle u_2, \mathbf{1}\rangle, \quad u_1 \geq 0, u_2 \geq 0$$

is unsatisfying because it doubles the size of the unknowns. Another way?

Outline

From Proj to Prox

Remember the proof of

Theorem

Firm Nonexpansiveness The projection π_C onto a nonempty closed convex set $C \subset \mathbb{R}^n$ is firmly nonexpansive.

Let $p_u \in \partial\delta_C(\pi_C(u)),\ p_v \in \partial\delta_C(\pi_C(v))$ be subgradients

$$\begin{aligned}
\langle u - v, \pi_C(u) - \pi_C(v)\rangle &= \langle \pi_C(u) - \pi_C(v) + p_u - p_v, \pi_C(u) - \pi_C(v)\rangle \\
&= \|\pi_C(u) - \pi_C(v)\|^2 + \langle p_u - p_v, \pi_C(u) - \pi_C(v)\rangle \\
&\geq \|\pi_C(u) - \pi_C(v)\|^2
\end{aligned}$$

We did not use that p_u and p_v were subgradients of an indicator function. The proof still works after replacing δ_C with an arbitrary convex function.

Proximal Operator

Definition

Given a closed, proper, convex function $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, the mapping $\text{prox}_E : \mathbb{R}^n \to \mathbb{R}^n$ defined as

$$\text{prox}_E(v) := \underset{u \in \mathbb{R}^n}{\operatorname{argmin}} \; E(u) + \frac{1}{2} \left\| u - v \right\|^2$$

is called the *proximal operator* or *proximal mapping* of E.

Existence: $E(u) + \frac{1}{2} \left\| u - v \right\|^2$ is closed, it has bounded sublevel sets

Uniqueness: $E(u) + (1/2) \left\| u - v \right\|^2$ is strongly convex

Generalization of the projection: Choose $E = \delta_C$.

Proximal Operator

Theorem

The proximal operator prox_E *for a closed, proper, convex function* E *is firmly nonexpansive.*

Course notes.

Consider minimizing an energy

$$E(u) = F(u) + G(u),$$

for proper, closed, convex E_1 and E_2 such that

$F : \mathbb{R}^n \to \mathbb{R}$ is L-smooth.

$G : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ has an easy-to-evaluate proximal operator

Intuition: we can generalize projected gradient by taking gradient descent steps on F and proximal steps on G

Proximal gradient algorithm

Definition

For a closed, proper, convex function $G : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and a function $F \in \mathcal{C}^1(\mathbb{R}^n)$, given an initial point $u^0 \in \mathbb{R}^n$ and a step size τ, the algorithm

$$u^{k+1} = \mathsf{prox}_{\tau G}\left(u^k - \tau \nabla F(u^k)\right), \qquad k = 0, 1, 2, \ldots,$$

is called the *proximal gradient method*.

- Often referred to as *forward-backward splitting* or ISTA
- For constant G, it reduces to *gradient descent*
- For constant F, it is called *proximal point algorithm*
- For $G = \delta_C$, it reduces to *projected gradient descent*

Easy convergence analysis as fixed-point iteration of averaged operator

Convergence analysis

Theorem

If F is L-smooth and $\tau \in (0, \frac{2}{L})$, the proximal gradient method converges.

We have seen: prox-operator is firmly nonexpansive (averaged $\alpha = \frac{1}{2}$)

Theorem

If the proper, closed function G is m-strongly convex, then $\text{prox}_{\tau G} : \mathbb{R}^n \to \mathbb{R}^n$ is a contraction.

Corollary

If F is L-smooth, $\tau \in (0, \frac{2}{L})$, and either G or F is strongly convex, then the proximal gradient method converges linearly, i.e., $\|u^k - u^*\|_2^2 \in \mathcal{O}(c^k)$ for some $c < 1$.

Sanity check and Examples

Sanity check: the algorithm converges to what? minimizer of $E = G + F$

Examples of functions whose prox has a closed form:

Quadratic functions

$$f(x) = \frac{1}{2}\|Au - b\|^2, \quad \mathrm{prox}_{\tau f}(v) = (I + \tau A^T A)^{-1}(v - \tau b)$$

Euclidean norm

$$f(x) = \|x\|, \quad \mathrm{prox}_{\tau f}(v) = \begin{cases} (1 - \tau / \|v\|)v & \text{if } \|v\| \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

ℓ_1-norm (cf. exercise sheet 3), "soft thresholding"

$$f(x) = \|x\|_1, \quad \big(\mathrm{prox}_{\tau f}(v)\big)_i = \begin{cases} v_i + \tau & \text{if } v_i < -\tau \\ 0 & \text{if } |v_i| \leq \tau \\ v_i - \tau & \text{if } v_i > \tau. \end{cases}$$

Application sparse recovery

We can now solve

$$\min_u \|Au - f\|_2^2 + \alpha \|u\|_1$$

without smoothing and without the introduction of additional variables

Convergence Rates and Extensions

Similar to gradient descent the proximal gradient method on

$$E = F + G$$

for L-smooth F, E having a minimizer, and choosing the step size τ to be constant converges with $E(u^k) - E(u^*) \in \mathcal{O}(1/k)$.
Similar to gradient descent

accelerated to $E(u^k) - E(u^*) \in \mathcal{O}(1/k^2)$ with Nesterov's scheme

line search: if we cannot find the Lipschitz constant for acceleration

For gradient projection, the analysis is in *Introductory lectures on convex optimization* by Nesterov. For proximal gradient, in *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, Beck, Teboulle, 2009.

Accelerated proximal gradient

Pick some starting point $v^0 = u^0$, set $t_0 = 1$, and iterate

1. Compute
$$u^{k+1} = \mathrm{prox}_{\frac{1}{L}G}\left(v^k - \frac{1}{L}\nabla F(v^k)\right)$$
2. Determine
$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2},$$
3. Compute the extrapolation of u^{k+1} via
$$v^{k+1} = u^{k+1} + \frac{t_k - 1}{t_{k+1}}(u^{k+1} - u^k)$$

See Chambolle, Dossal, *On the Convergence of the Iterates of the "Fast Iterative Shrinkage/Thresholding Algorithm"*, 2015, for more general algorithms.

Accelerated gradient projection with line search

Let $Q_\tau(u,v) = F(v) + \langle u - v, \nabla F(v)\rangle + \frac{1}{2\tau}\|u-v\|^2 + G(u)$ Pick $v^0 = u^0$, $\beta < 1$, $\tau_0 > 0$, set $t_0 = 1$ and iterate

1. Find a suitable step size $\tau_k \leq \tau_{k-1}$ via

$$\tau_k = \tau_{k-1}, \quad u^{k+1} = \mathrm{prox}_{\tau_k G}\left(v^k - \tau_k \nabla F(v^k)\right)$$

while $E(u^{k+1}) > Q_\tau(u^{k+1}, v^k)$

$$\tau_k \leftarrow \beta\tau_k, \quad u^{k+1} \leftarrow \mathrm{prox}_{\tau_k G}\left(v^k - \tau_k \nabla F(v^k)\right)$$

end

2. Determine

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2},$$

3. Compute the extrapolation of u^{k+1} via

$$v^{k+1} = u^{k+1} + \frac{t_k - 1}{t_{k+1}}(u^{k+1} - u^k)$$

What we can and cannot do yet

As we have seen

$$\min_u \frac{1}{2}\|Au - f\|^2 + \alpha\|u\|_1$$

does not pose a problem anymore.

But what about our TV-denoising model:

$$\min_u \frac{1}{2}\|u - f\|^2 + \alpha\|Du\|_1?$$

The problem itself is a proximal operator but not easy-to-evaluate. We will see how to solve it next week.