Chapter 2

Gradient Methods

Convex Optimization for Computer Vision

SS 2018

Michael Moeller
Visual Scene Analysis
Department of Computer Science
and Electrical Engineering
University of Siegen

Visual
Scene
Analysis

Gradient Descent

Unconstrained and smooth optimization

Recall what the lecture is all about:

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

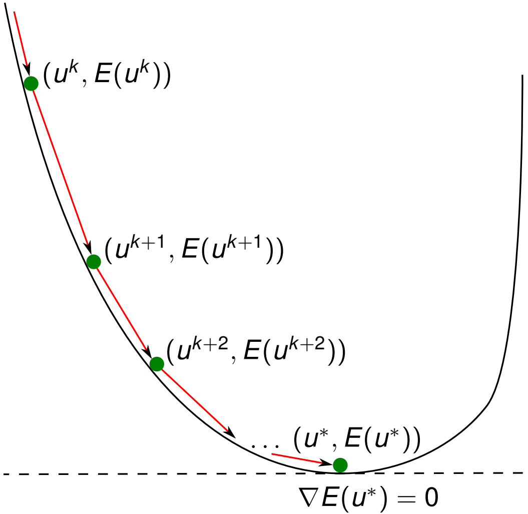
for $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ proper, closed, convex.

We start making our life easier:

- $\text{dom } E = \mathbb{R}^n$
- $E \in \mathcal{C}^1(\mathbb{R}^n)$
- Even more assumptions later :-)

Descent methods

$$\min E(u), \qquad u \in \mathbb{R}^n$$



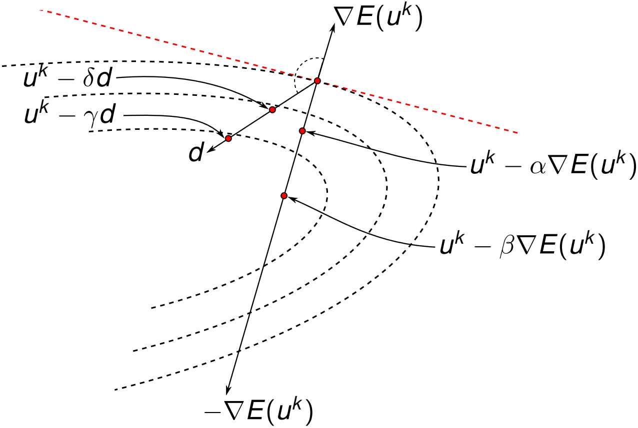
Descent methods

- 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 $d \in \mathbb{R}^n$
- Taylor expansion for E along ray

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

- The term $\tau \langle \nabla E(u^k), d \rangle$ dominates $o(\tau)$ for suff. small τ
- Pick d such that $\langle \nabla E(u^k), d \rangle < 0$, *descent direction*
- Then $E(u(\tau)) < E(u)$ for suff. small τ

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 $\gamma : I \to \mathbb{R}^n$

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

- Possible 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:

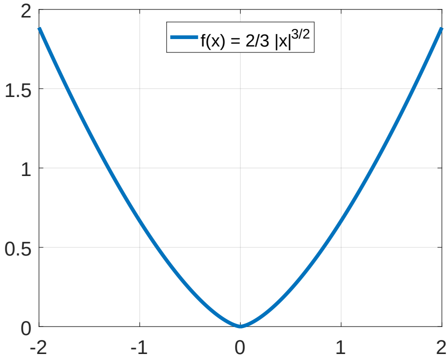
- Generate (decreasing?) sequence $\{E(u^k)\}_{k=0}^{\infty}$
- Each iteration is cheap, easy to code

Choice of τ_k:

- $\tau_k = \tau$ for some constant $\tau \in \mathbb{R}$ (this lecture)
- Exact line search $\tau_k = \arg\min_\tau \; E\left(u^k - \tau \nabla E(u^k)\right)$
- Inexact line search (more later)

Constant step size

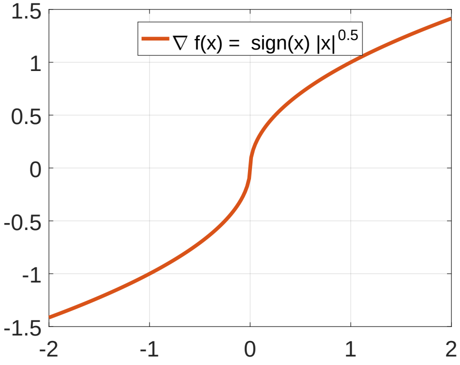
Let us first consider a constant step size $\tau^k = \tau$. Will gradient descent work for any convex function *E*? NO!



Board: For any $\tau > 0$, the starting point $u^0 = \left(\frac{\tau}{2}\right)^2$ leads to the gradient descent sequence $u^0, -u^0, u^0, -u^0 \ldots$. In fact, gradient descent will fail to converge for almost any starting point!

Constant step size

Let us first consider a constant step size $\tau^k = \tau$. Will gradient descent work for any convex function *E*? NO!



Board: For any $\tau > 0$, the starting point $u^0 = \left(\frac{\tau}{2}\right)^2$ leads to the gradient descent sequence $u^0, -u^0, u^0, -u^0 \ldots$. In fact, gradient descent will fail to converge for almost any starting point!

Why can it fail?

Intuitively, an "infinitely quickly changing gradient", ∇E, seems to cause the problems!

We already know a stronger version of continuity which prevents "infinitely quick changes"!

Reminder

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

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

Is there a (possibly easier) characterization of Lipschitz continuous functions?

Lipschitz continuity

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 $x \in \mathbb{R}^n$.

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 (in some literature L-strongly smooth). We denote the set

- of all L-smooth functions by $\mathcal{C}_L^{1,1}(\mathbb{R}^n)$.
- of all convex L-smooth functions by $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$.

How to analyze the convergence?

Conjecture

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

But how do we proceed in proving the assertion?

If this was a research project: Using the assumptions, try to write down smart estimates until you have an inequality from which you can conclude the convergence.

Since this is a lecture: General convergence framework that applies to many convex optimization algorithms!

Fixed-point iterations

A form many algorithms can be written into:

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

for an update function G, i.e. a **fixed-point iteration**!

Example:

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

If the iteration converges, i.e. $\hat{u} = \lim_{k\to\infty} u^k$, then

$$\hat{u} = \hat{u} - \tau \nabla E(\hat{u}),$$

i.e. $\nabla E(\hat{u}) = 0$ (where we assumed ∇E to be continuous).

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

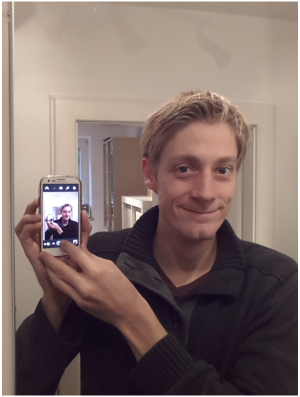
If the update rule $G : \mathbb{R}^n \to \mathbb{R}^n$ is a **contraction**, i.e. if there exists a $L < 1$ such that

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

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,

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

Examples for fixed-point iterations with contractions



The function $G(u) = \frac{u+\frac{1}{2}}{u+1}$ is a contraction on $[0, \infty[$. Therefore, the fixed point iteration converges to $\frac{1}{\sqrt{2}}$.

Later: The gradient descent update is a contraction for specific energies E.

Fixed-point iterations with averaged operators

As we will see, the assumption of G being a **contraction** is **too restrictive** in many cases!

One thing that often holds easily, is that G is **non-expansive**, i.e. Lipschitz continuous with constant $L = 1$.

Example: Any rotation G is non-expansive, any rotation has a fixed point (zero), but the iteration $u^{k+1} = G(u^k)$ does not converge!

$\rightarrow$ We need more!

Averaged operator

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

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

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: Board

Criteria for being averaged

We now have two loose ends: A conjecture about the conergence of the gradient descent iteration, and a theorem that states the convergence of a fixed-point iteration for averaged operators.

We need a better understanding of averaged operators!

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: Board

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: Board

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

Let us use the previous results for approaching our gradient descent convergence problem:

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.

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

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

Proof: Some parts on the board. Otherwise see Nesterov, *Introductory Lectures on Convex Optimization*, Theorem 2.1.5.

Tools for the proof

Theorem: characterization of convex functions

For a continuously differentiable E the following are equivalent:

1. E is convex,
2. $E(v) - E(u) - \langle \nabla E(u), v - u \rangle \geq 0 \ \ \forall u, v$,
3. $\langle \nabla E(u) - \nabla E(v), u - v \rangle \geq 0 \ \ \forall u, v$,
4. $\nabla^2 E(u) \succeq 0 \ \ \forall u$, if $E \in \mathcal{C}^2(\mathbb{R}^n)$

Proof: E.g. Ryu, Boyd, A Primer on Monotone Operator Methods, Appendix A.

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

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

- 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?

Reminder: $\mathcal{O}$-notation

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

Convergence speed of gradient descent

One can show that

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

Linear convergence ($\mathcal{O}(c^k)$ for $c < 1$) would be faster. Is there no way to get a contraction?

Quick answer: Impossible in this generality! A contraction would imply the existence of a unique fixed-point!

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 [a]

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

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 $E \in \mathcal{C}^2(\mathbb{R}^n)$

L-smoothness

If a continuously differentiable function $E : \mathbb{R}^n \to \overline{\mathbb{R}}$ is L-smooth then R, $R(u) = \frac{L}{2}\|u\|^2 - E(u)$, is convex.

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 $\|u^k - u^*\| \leq c^k \|u^0 - u^*\|$.

Partial proof on the board.

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?

Optimal convergence rates

Famous analysis by Nesterov, e.g. *Introductory Lectures on Convex Optimization*, Theorem 2.1.7 and Theorem 2.1.13:

First order method:

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

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

Obtaining optimal convergence rates

Nesterov's Accelerated Gradient Descent

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

1. Compute
$$u^{k+1} = v^k - \frac{1}{L}\nabla E(v^k)$$
2. Set
$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$$
3. Compute the extrapolation of u^{k+1} via
$$v^{k+1} = u^{k+1} + \frac{t_k - 1}{t_{k+1}}(u^{k+1} - u^k)$$

[1]

[1] Also see "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems" by Beck and Teboulle

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), or *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems* by Beck and Teboulle, page 194.

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

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

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

$$\hat{u} \in \arg\min_u \quad \underbrace{\frac{\lambda}{2}\|u-f\|_2^2}_{=H_f(u) \text{stay close to input}} \quad + \underbrace{\sum_{x\in\Omega} \sqrt{(D_1 u(x))^2 + (D_2 u(x))^2}}_{=TV(u) \text{ suppress noise}}$$

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

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

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!*

Convergence, backtracking line search

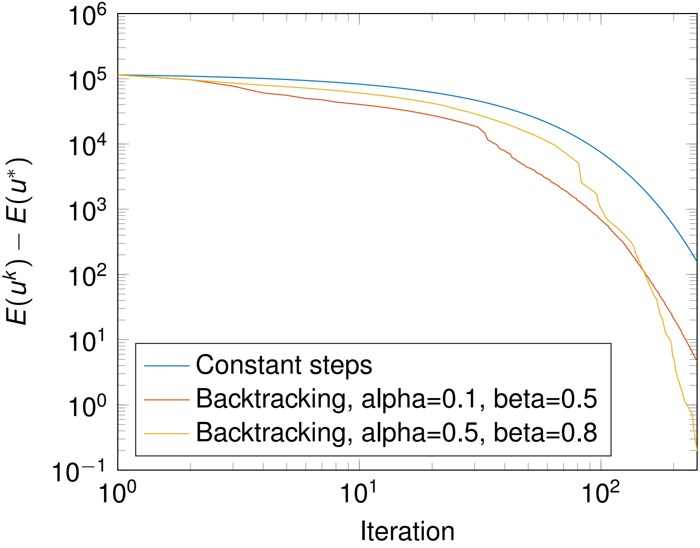
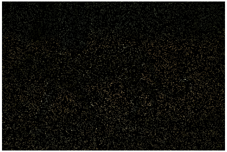
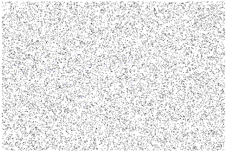


Image inpainting



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



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

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

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

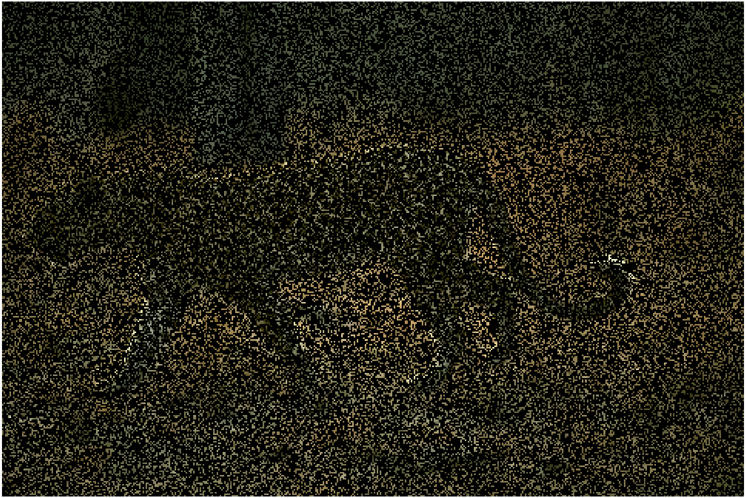
- Energy is not strongly convex, but L-smooth
- Sublinear upper bound on convergence speed

Image Inpainting

50% missing pixels

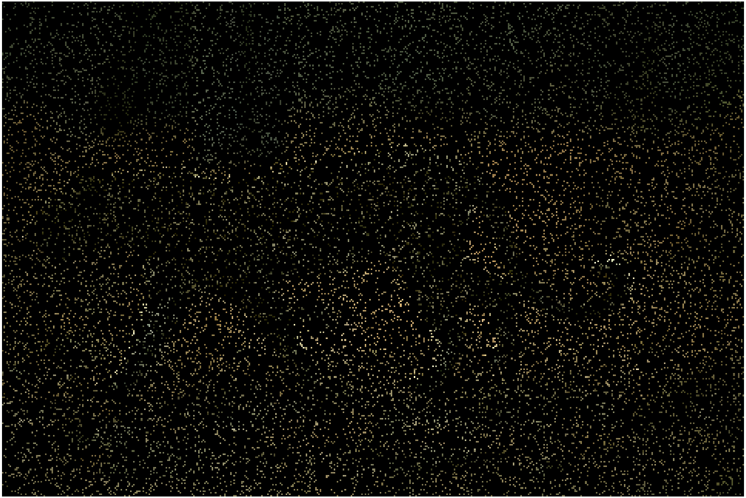
50% missing pixels

70% missing pixels



70% missing pixels

90% missing pixels



90% missing pixels

Fast optimization challenge I

- Minimize the inpainting energy

$$E(u) = \frac{\lambda}{2}\|m \cdot (u - f)\|^2 + \sum_{i=1}^{2N} h_\varepsilon((Du)_i) + \beta\|u\|^2$$

- Huber penalty $h_\varepsilon(x) = \begin{cases} \frac{x^2}{2\varepsilon} & \text{if } |x| \leq \varepsilon, \\ |x| - \frac{\varepsilon}{2} & \text{otherwise.} \end{cases}$
- Given all the parameters, return the solution once

$$\frac{E(u^k) - E(u^*)}{E(u^*)} < \delta$$

- See template `challenge_huber_inpainting.m`
- Live leaderboard on homepage
- Fastest solution at end of semester receives a prize!

Handwritten digit recognition

- MNIST dataset[2], handwritten digit recognition
- $K = 10$ digits, 28×28 grayscale images
- $n = 60000$ training images $X \in \mathbb{R}^{n \times 768}$, with ground-truth labels $Y \in \{1, \dots, 10\}^n$
- Learn simple *linear* model $W \in \mathbb{R}^{10 \times 768}$ on raw pixel data
- Softmax regression (multinomial logistic regression)

$$p(y_i = k | x_i, W) = \frac{\exp(\langle w_k, x_i \rangle)}{\sum_{j=1}^{K} \exp(\langle w_j, x_i \rangle)}$$

[2] http://yann.lecun.com/exdb/mnist/

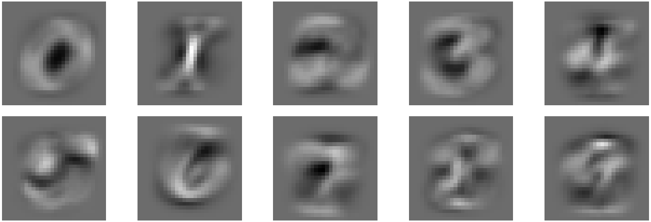
Multinomial logistic regression

- Minimize negative log-likelihood

$$
\begin{aligned}
E(W) &= -\log \frac{1}{n}\prod_{i=1}^{n} p(y_i = k|x_i, W)p(W) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \log p(y_i = k|x_i, W) + \lambda \|W\|_F^2
\end{aligned}
$$

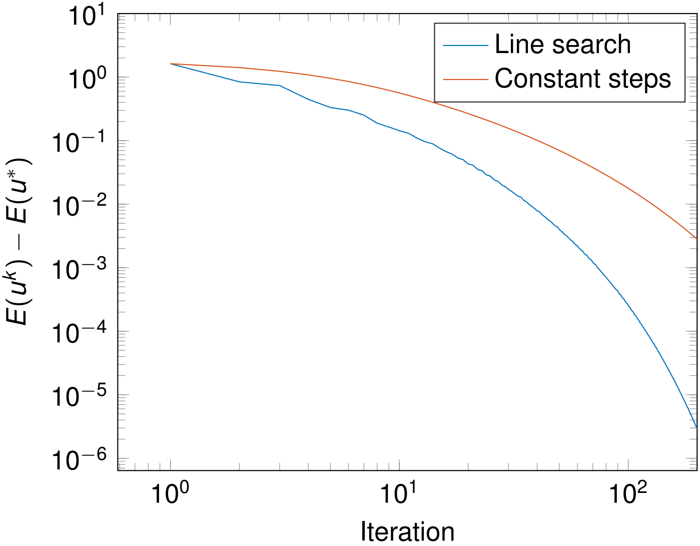
- It can be shown that $E(W)$ is λ-strongly convex
- $E(W)$ is also L-smooth (bound: $\lambda + \frac{\|X\|^2}{4n}$)
- Minimize using gradient descent with $\tau = \frac{2}{2\lambda + \|X\|^2/4n}$
- Gradient computation expensive $\rightarrow$ *stochastic* methods! (we won't cover them)

Multinomial logistic regression



- Classifier gives around 10% error on test set
- Current best: 0.21% (convolutional neural networks)

Multinomial logistic regression



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.

Up next: **Gradient projection!** Generalizes gradient descent to arbitrary (nonempty, closed, convex) dom(E).

Gradient Projection

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.

What is the *projection* onto the 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

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: Board.

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

Example projections

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

- $C = \{u \in \mathbb{R}^n \mid \|u\|_2 \leq 1\}$?
- $C = \{u \in \mathbb{R}^n \mid \|u\|_\infty := \max_i |u_i| \leq 1\}$?
- $C = \{u \in \mathbb{R}^n \mid u_i \in [a, b]\}$?
- $C = \{u \in \mathbb{R}^n \mid u_i \geq a\}$?

Idea of gradient projection

Consider a problem

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

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

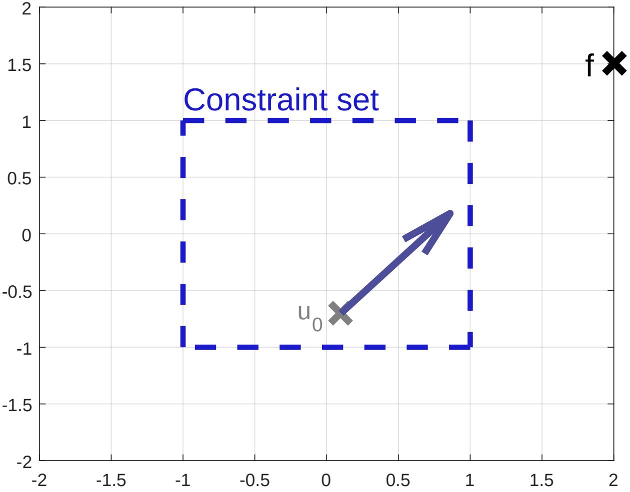
We know how gradient descent works, but updating $u^{k+1} = u^k - \tau^k \nabla E(u^k)$ may lead to $u^{k+1} \notin C$.

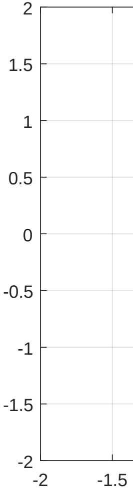
Idea: **Project every iteration back to the feasible set**, i.e.

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

Idea of gradient projection

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





Gradient projection algorithm

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.

When, how, why, and for which E and τ does it work?

As usual in this lecture: Analyze the **fixed-point iteration** of

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

Projected GD as a fixed-point iteration

We already know that ...

1. ... for $\tau \in]0, \frac{2}{L}[$ the following operator is averaged

$$G_1(u) = u - \tau \nabla E(u)$$

2. ... compositions of averaged operators are averaged.

All we have to do is showing that π_C is averaged!

Properties of the projection

Firm Nonexpansiveness

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

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

Proof: Board

This makes π_C averaged and we can immediately conclude:

Conclusion

For an L-smooth energy E that has a minimizer and a choice $\tau \in]0, \frac{2}{L}[$ the gradient projection converges!

Similar to the gradient descent case, the convergence rate is $\mathcal{O}(1/k)$ and suboptimal. We will discuss accelerations to $\mathcal{O}(1/k^2)$ of a generalized version later.

Convergence of the projected gradient descent

A simple calculation (done in the exercises) shows:

Compositions can yield contractions

The composition of a non-expansive operator with a contraction is a contraction.

The above means our gradient descent result carries over:

Conclusion

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 $E(u^k) - E(u^*) \in \mathcal{O}(c^k)$ for $c < 1$.

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	

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

How can we do this with convex optimization?

Example Application: Solving a SUDOKU

In the 4×4 case we look for a matrix $u \in \{1,2,3,4\}^{4\times 4}$ such that $u_{i,j} = f_{i,j}$ for those entries $f_{i,j}$ which are given.

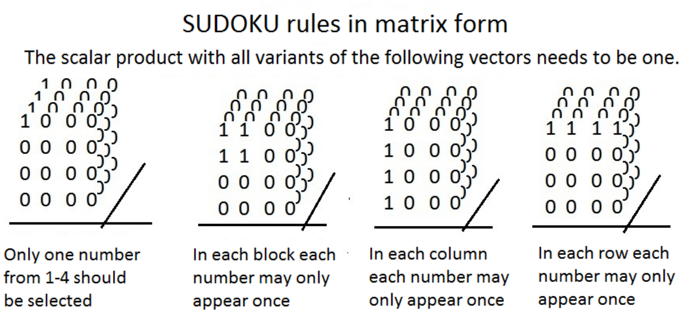
Reformulation: We look for a matrix $\boldsymbol{u} \in \{0,1\}^{4\times 4\times 4}$, where $\boldsymbol{u}_{i,j,k} = 1$ means $u_{i,j} = k$.

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	$\sum_{(i,j)\in B_l} \boldsymbol{u}_{i,j,k} = 1$	$\forall B_l, k$

Find $\boldsymbol{u}$ with $\boldsymbol{u}_{i,j,k} \in \{0,1\}$ subject to the above constraints!

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!

Convex relaxation: Use the smallest convex set that contains the nonconvex one, $\boldsymbol{u}_{i,j,k} \in [0,1]$.

If the result meets $\boldsymbol{u}_{i,j,k} \in \{0,1\}$, we solved the nonconvex problem.

Example Application: Solving a SUDOKU

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

This means we may solve

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

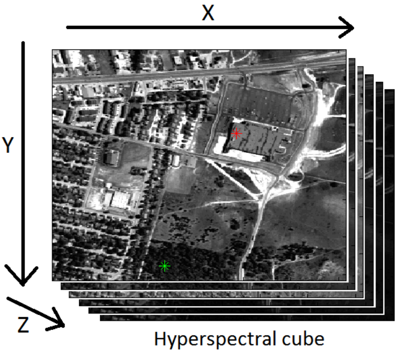
Hope that $\hat{\boldsymbol{u}}_{i,j,k} \in \{0,1\}$ in which case we solved the SUDOKU!

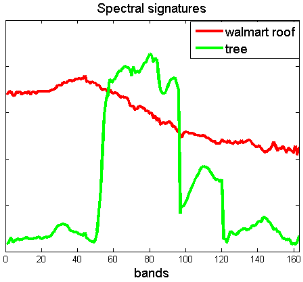
Remarks:

- 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 problems! Look for *labeling problems*, *segmentation*, *graph cuts*, or *functional lifting*.

Example application: Unmixing and sparse recovery

Hyperspectral imagery





z-direction: Material specific reflected energy depending on the wavelength of the incoming light

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

General setup: 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:

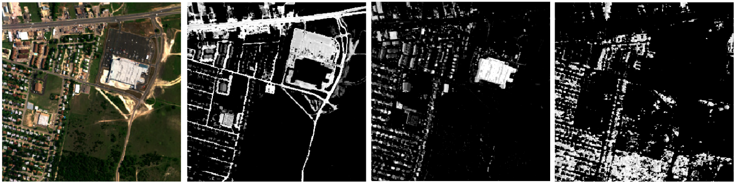
$$\min_u H_f(Au) + \alpha \|u\|_1.$$

Can we apply gradient descent/ gradient projection?

Not directly, but the problem is equivalent to

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

Example application: Unmixing and sparse recovery



color image illustration | endmember "road" | endmember "roof" | endmember "trees"

The reformulation of

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

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

possibly is a little unsatisfying. In particular, it doubles the size of our unknowns. Any other way?

Proximal Gradient

From proj to prox

Remember the theorem

Firm Nonexpansiveness

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

and its proof?

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

for $p_u \in \partial\delta_C(\pi_C(u)),\ p_v \in \partial\delta_C(\pi_C(v))$ denoting the subgradients.

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 operators

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

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

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

- **Existence:** $E(u) + (1/2)\|u - v\|^2$ is closed and has bounded sublevel sets
- **Uniqueness:** $E(u) + (1/2)\|u - v\|^2$ is strongly convex
- **Generalization of the projection**: Choose $E = \delta_C$.

We have just seen

Firm Nonexpansiveness

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

Consider minimizing an energy

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

for proper, closed, convex F and G 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 proximity operator, which we will call *simple*.

The we can take gradient descent steps on F and proximal steps on G! This is the proximal gradient algorithm!

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

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

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*

For us (=super-duper experts on fixed point iterations) the convergence analysis is easy!

Convergence analysis

We have already seen that the prox-operator is firmly nonexpansive, i.e., averaged with $\alpha = 1/2$.

Conclusion

For F being L-smooth, $\tau \in]0, \frac{2}{L}[$, and the overall energy having a minimizer, the proximal gradient method converges.

Contractive prox-operators

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.

Conclusion

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

Sanity check + examples

Sanity check: The algorithm converges, but to what?

Board: To a minimizer of $E = G + F$!

Examples of functions whose prox has a closed form:

- Quadratic functions

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

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

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

- Euclidean norm

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

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 one can do better and reach $E(u^k) - E(u^*) \in \mathcal{O}(1/k^2)$.

Similar to gradient descent finding the Lipschitz constant L can be annoying, and one can define line search schemes.

Gradient projection: *Introductory lectures on convex optimization* by Nesterov.
Proximal gradient: *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, Beck, Teboulle, 2009.

Accelerated proximal gradient

FISTA with constant step size

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

1. Compute

$$u^{k+1} = \text{prox}_{\frac{1}{L}G}\left(v^k - \frac{1}{L}\nabla F(v^k)\right)$$

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

FISTA with backtracking line search

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

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

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

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

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

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 minimization problem itself already is a proximal operator and not easy-to-evaluate.

Not solvable with any algorithm we did? Or maybe it is?
$\rightarrow$ Let us develop some ideas on the board!