iPiano: Inertial Proximal Algorithm for Non-Convex Optimization

David Stutz

April 13, 2016

Table of Contents

Problem

Problem. Minimize composite function

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x))$$

where

Problem

Problem. Minimize composite function

$$\boxed{\min_{x\in\mathbb{R}^d} h(x) = \min_{n\in\mathbb{R}^d}(f(x) + g(x))} \tag{1}$$

where

- $f : \mathbb{R}^d \to \mathbb{R} \in C^1$ with L-Lipschitz continuous gradient;

Problem

Problem. Minimize composite function

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)) \quad (1)$$

where

- $f : \mathbb{R}^d \to \mathbb{R} \in C^1$ with L-Lipschitz continuous gradient;
- g : dom$(g) \subset \mathbb{R}^d \to \mathbb{R}_\infty$ is proper closed convex and lower semicontinuous;

Problem

Problem. Minimize composite function

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)) \tag{1}$$

where

- $f : \mathbb{R}^d \to \mathbb{R} \in C^1$ with L-Lipschitz continuous gradient;
- g : dom$(g) \subset \mathbb{R}^d \to \mathbb{R}_\infty$ is proper closed convex and lower semicontinuous;
- and h coercive and bounded below by

$$-\infty < h_{\min} := \inf_{x \in \mathbb{R}^d} h(x).$$

Problem

Problem. Minimize composite function

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)) \tag{1}$$

where

- $f : \mathbb{R}^d \to \mathbb{R} \in C^1$ with L-Lipschitz continuous gradient;
- g : dom$(g) \subset \mathbb{R}^d \to \mathbb{R}_\infty$ is proper closed convex and lower semicontinuous;
- and h coercive and bounded below by

$$-\infty < h_{\min} := \inf_{x \in \mathbb{R}^d} h(x).$$

Ochs et al. [**?**] combine forward-backward splitting with an inertial force/momentum term to solve Equation (**??**) iteratively.

Table of Contents

Related Work

Gradient descent for $h \in C^1$:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}).$$

Gradient descent with inertial force/momentum term:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}).$$

Proximal point for h being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n h}(x^{(n)}).$$

Forward-backward splitting for $h = f + g$ with $f \in C^1$ and f, g being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)})).$$

Related Work

Gradient descent for $h \in C^1$:

$$\boxed{x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}).}$$

Gradient descent with inertial force/momentum term:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}).$$

Proximal point for h being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n h}(x^{(n)}).$$

Forward-backward splitting for $h = f + g$ with $f \in C^1$ and f, g being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)})).$$

Related Work

Gradient descent for $h \in C^1$:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}).$$

Gradient descent with inertial force/momentum term:

$$\boxed{x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}).}$$

Proximal point for h being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n h}(x^{(n)}).$$

Forward-backward splitting for $h = f + g$ with $f \in C^1$ and f, g being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)})).$$

Related Work

Gradient descent for $h \in C^1$:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}).$$

Gradient descent with inertial force/momentum term:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}).$$

Proximal point for h being proper closed convex:

$$\boxed{x^{(n+1)} = \text{prox}_{\alpha_n h}(x^{(n)}).}$$

Forward-backward splitting for $h = f + g$ with $f \in C^1$ and f, g being proper closed convex:

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)})).$$

Related Work

Gradient descent for $h \in C^1$:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}).$$

Gradient descent with inertial force/momentum term:

$$x^{(n+1)} = x^{(n)} - \alpha_n \nabla h(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}).$$

Proximal point for h being proper closed convex:

$$x^{(n+1)} = \mathrm{prox}_{\alpha_n h}(x^{(n)}).$$

Forward-backward splitting for $h = f + g$ with $f \in C^1$ and f, g being proper closed convex:

$$\boxed{x^{(n+1)} = \mathrm{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)})).}$$

Table of Contents

Algorithm – Iterates and Backtracking

Ochs et al. [**?**] combine forward-backward splitting with an inertial force/momentum term.

Algorithm – Iterates and Backtracking

Ochs et al. [**?**] combine forward-backward splitting with an inertial force/momentum term:

$$\boxed{x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n(x^{(n)} - x^{(n-1)}))}$$

with step size parameters $(\alpha_n)_{n\in\mathbb{N}}$ and momentum parameters $(\beta_n)_{n\in\mathbb{N}}$.

Algorithm – Iterates and Backtracking

Ochs et al. [**?**] combine forward-backward splitting with an inertial force/momentum term

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}))$$

with step size parameters $(\alpha_n)_{n\in\mathbb{N}}$ and momentum parameters $(\beta_n)_{n\in\mathbb{N}}$.

Backtracking to estimate the local Lipschitz constant L_n such that

$$\begin{aligned} f(x^{(n+1)}) \leq f(x^{(n)}) &+ \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) \\ &+ \frac{L_n}{2} \|x^{(n+1)} - x^{(n)}\|_2^2 \end{aligned}$$

Algorithm – Iterates and Backtracking

Ochs et al. [**?**] combine forward-backward splitting with an inertial force/momentum term

$$x^{(n+1)} = \text{prox}_{\alpha_n g}(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)}))$$

with step size parameters $(\alpha_n)_{n\in\mathbb{N}}$ and momentum parameters $(\beta_n)_{n\in\mathbb{N}}$.

Backtracking to estimate the local Lipschitz constant L_n such that

$$\boxed{\begin{aligned} f(x^{(n+1)}) \leq f(x^{(n)}) &+ \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) \\ &+ \frac{L_n}{2} \|x^{(n+1)} - x^{(n)}\|_2^2 \end{aligned}}$$

Algorithm – iPiano

Algorithm iPiano.

choose $c_1, c_2 > 0$ close to zero, $L_{-1} > 0$, $\eta > 1$, $x^{(0)}$
$x^{(-1)} := x^{(0)}$
for $n = 1, \ldots$ **do**
    choose $\alpha_n \geq c_1$, $\beta_n \geq 0$
    $x^{(n+1)} = \text{prox}_{\alpha_n g}\left(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n(x^{(n)} - x^{(n-1)})\right)$
end for

Algorithm – iPiano

Algorithm iPiano.

choose $c_1, c_2 > 0$ close to zero, $L_{-1} > 0$, $\eta > 1$, $x^{(0)}$

$x^{(-1)} := x^{(0)}$

for $n = 1, \ldots$ **do**

  repeat

    choose $\alpha_n \geq c_1$, $\beta_n \geq 0$

  until $\delta_n := \frac{1}{\alpha_n} - \frac{L_n}{2} - \frac{\beta_n}{2\alpha_n} \geq \gamma_n := \frac{1}{\alpha_n} - \frac{L_n}{2} - \frac{\beta_n}{\alpha_n} \geq c_2$

  $x^{(n+1)} = \text{prox}_{\alpha_n g}\left(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n(x^{(n)} - x^{(n-1)})\right)$

end for

Algorithm – iPiano

Algorithm iPiano.

choose $c_1, c_2 > 0$ close to zero, $L_{-1} > 0$, $\eta > 1$, $x^{(0)}$

$x^{(-1)} := x^{(0)}$

for $n = 1, \ldots$ **do**

  $L_n := \frac{1}{\eta} L_{n-1}$

  repeat

    $L_n := \eta L_n$

    repeat

      choose $\alpha_n \geq c_1$, $\beta_n \geq 0$

    until $\delta_n := \frac{1}{\alpha_n} - \frac{L_n}{2} - \frac{\beta_n}{2\alpha_n} \geq \gamma_n := \frac{1}{\alpha_n} - \frac{L_n}{2} - \frac{\beta_n}{\alpha_n} \geq c_2$

    $x^{(n+1)} = \text{prox}_{\alpha_n g}\left(x^{(n)} - \alpha_n \nabla f(x^{(n)}) + \beta_n (x^{(n)} - x^{(n-1)})\right)$

  until (**??**) is satisifed for $x^{(n+1)}$

end for

Algorithm – Monotonically Decreasing δ_n

Lemma

For each $n \in \mathbb{N}$, given $L_n > 0$, there exist $\alpha_n < 2(1 - \beta_n)/L_n$ and $0 \leq \beta_n < 1$ as in iPiano such that $c_2 \leq \gamma_n \leq \delta_n$ and $(\delta_n)_{n\in\mathbb{N}}$ is monotonically decreasing.

Algorithm – Monotonically Decreasing δ_n

Lemma

For each $n \in \mathbb{N}$, given $L_n > 0$, there exist $\alpha_n < 2(1 - \beta_n)/L_n$ and $0 \leq \beta_n < 1$ as in iPiano such that $c_2 \leq \gamma_n \leq \delta_n$ and $(\delta_n)_{n\in\mathbb{N}}$ is monotonically decreasing.

Proof Sketch.

With $b_n := (\delta_{n-1} + \frac{L_n}{2})/(c_2 + \frac{L_n}{2})$:

$$\gamma_n \geq c_2 \;\Leftrightarrow\; \alpha_n \leq \frac{1 - \beta_n}{c_2 + \frac{L_n}{2}} < \frac{2(1 - \beta_n)}{L_n}$$

$$\delta_{n-1} \geq \delta_n \;\Leftrightarrow\; \frac{1 - \beta_n}{c_2 + \frac{L_n}{2}} \geq \alpha_n \geq \frac{1 - \frac{\beta_n}{2}}{\delta_{n-1} + \frac{L_n}{2}} \;\Rightarrow\; \beta_n \leq \frac{b_n - 1}{b_n - \frac{1}{2}}$$

□

Table of Contents

Convergence – Overview

Convergence analysis is based on **three** requirements regarding

$$H_{\delta_{n+1}}(x^{(n+1)}, x^{(n)}) := h(x^{(n+1)}) + \delta_{n+1} \underbrace{\|x^{(n)} - x^{(n-1)}\|_2^2}_{\Delta_{n+1}^2}$$
$$:= h(x^{(n+1)}) + \delta_{n+1}$$

and the sequence

$$(z^{(n+1)})_{n\in\mathbb{N}} := (x^{(n+1)}, x^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$$

generated by iPiano.

Convergence – Overview

Convergence analysis is based on **three** requirements regarding

$$\boxed{\begin{aligned} H_{\delta_{n+1}}(x^{(n+1)}, x^{(n)}) &:= h(x^{(n+1)}) + \delta_{n+1} \underbrace{\|x^{(n)} - x^{(n-1)}\|_2^2}_{\Delta_{n+1}^2} \\ &:= h(x^{(n+1)}) + \delta_{n+1} \qquad \Delta_{n+1}^2 \end{aligned}}$$

and the sequence

$$(z^{(n+1)})_{n\in\mathbb{N}} := (x^{(n+1)}, x^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$$

generated by iPiano.

Convergence – Overview

Convergence analysis is based on **three** requirements regarding

$$H_{\delta_{n+1}}(x^{(n+1)}, x^{(n)}) := h(x^{(n+1)}) + \delta_{n+1} \underbrace{\|x^{(n)} - x^{(n-1)}\|_2^2}_{\Delta_{n+1}^2}$$
$$:= h(x^{(n+1)}) + \delta_{n+1} \Delta_{n+1}^2$$

and the sequence

$$\boxed{(z^{(n+1)})_{n\in\mathbb{N}} := (x^{(n+1)}, x^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}}$$

generated by iPiano.

Convergence – Overview

Convergence analysis is based on **three** requirements regarding

$$H_{\delta_{n+1}}(x^{(n+1)}, x^{(n)}) := h(x^{(n+1)}) + \delta_{n+1} \underbrace{\|x^{(n)} - x^{(n-1)}\|_2^2}$$
$$:= h(x^{(n+1)}) + \delta_{n+1} \quad \Delta_{n+1}^2$$

and the sequence

$$(z^{(n+1)})_{n\in\mathbb{N}} := (x^{(n+1)}, x^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$$

generated by iPiano.

Furthermore, H_{δ_n} is required to satisfy the Kurdyka-Lojasiewicz property [**?**, **?**] at a critical point $\tilde{z}$ of H_{δ_n}.

Convergence – Requirements

Definition

Given a, $b > 0$. $H : \mathbb{R}^{2d} \rightarrow \mathbb{R}_\infty$ and a sequence $(z^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$ satisfy:

(H1) if for each $n \in \mathbb{N}$, it holds

$$H(z^{(n+1)}) + a\Delta_n^2 \leq H(z^{(n)});$$

Convergence – Requirements

Definition

Given a, $b > 0$. $H : \mathbb{R}^{2d} \to \mathbb{R}_\infty$ and a sequence $(z^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$ satisfy:

(H1) if for each $n \in \mathbb{N}$, it holds

$$H(z^{(n+1)}) + a\Delta_n^2 \leq H(z^{(n)});$$

(H2) if for each $n \in \mathbb{N}$, there exists $w^{(n+1)} \in \partial H(z^{(n+1)})$ with

$$\|w^{(n+1)}\|_2 \leq \frac{b}{2}(\Delta_n + \Delta_{n+1});$$

Convergence – Requirements

Definition

Given a, $b > 0$. $H : \mathbb{R}^{2d} \to \mathbb{R}_\infty$ and a sequence $(z^{(n)})_{n\in\mathbb{N}} \subset \mathbb{R}^{2d}$ satisfy:

(H1) if for each $n \in \mathbb{N}$, it holds

$$H(z^{(n+1)}) + a\Delta_n^2 \leq H(z^{(n)});$$

(H2) if for each $n \in \mathbb{N}$, there exists $w^{(n+1)} \in \partial H(z^{(n+1)})$ with

$$\|w^{(n+1)}\|_2 \leq \frac{b}{2}(\Delta_n + \Delta_{n+1});$$

(H3) if there exists a subsequence $(z^{(n_j)})_{j\in\mathbb{N}}$ with $z^{(n_j)} \to \tilde{z} = (\tilde{x}, \tilde{x})$ and $H(z^{(n_j)}) \to H(\tilde{z})$ for $j \to \infty$.

Convergence – Requirements, Condition **??**

Lemma

H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??***, in particular for each $n \in \mathbb{N}$ it holds*

$$H_{\delta_{n+1}}(z^{(n+1)}) + \gamma_n \Delta_n^2 \leq H_{\delta_n}(z^{(n)});$$

Convergence – Requirements, Condition **??**

Lemma

H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??**, *in particular for each $n \in \mathbb{N}$ it holds*

$$H_{\delta_{n+1}}(z^{(n+1)}) + \gamma_n \Delta_n^2 \leq H_{\delta_n}(z^{(n)});$$

Proof Sketch.

Iteration (Equation (**??**)) $\Rightarrow$

$$w := \frac{x^{(n)} - x^{(n+1)}}{\alpha_n} - \nabla f(x^{(n)}) + \frac{\beta_n}{\alpha_n}(x^{(n)} - x^{(n-1)}) \in \partial g(x^{(n+1)})$$

$\square$

Convergence – Requirements, Condition ??

Proof Sketch (cont'd).

With $w \in \partial g(x^{(n+1)})$, using the convexity of g,

$$g(x^{(n+1)}) \leq g(x^{(n)}) - w^T(x^{(n)} - x^{(n-1)}),$$

and the L_n-Lipschitz continuity of ∇f,

$$f(x^{(n+1)}) \leq f(x^{(n)}) - +\nabla f(x^{(n)})^T(x^{(n+1)} - x^{(n)}) + \frac{L_n}{2}\|x^{(n)} - x^{(n+1)}\|_2^2;$$

it can be shown

$$h(x^{(n+1)}) \leq h(x^{(n)}) - \delta_n \Delta_{n+1}^2 + \delta_n \Delta_n^2 - \gamma_n \Delta_n^2$$

which implies the claim as δ_n is monotonically decreasing. □

Convergence – Requirements, Condition **??**

Lemma

H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??***, i.e. for each $n \in \mathbb{N}$ there exists $w^{(n+1)} \in \partial H_{\delta_{n+1}}(z^{(n+1)})$ such that $\|w^{(n+1)}\|_2 \leq \frac{7}{c_1}(\Delta_n + \Delta_{n+1})$.*

Convergence – Requirements, Condition **??**

Lemma

*H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??**, i.e. for each $n \in \mathbb{N}$ there exists $w^{(n+1)} \in \partial H_{\delta_{n+1}}(z^{(n+1)})$ such that $\|w^{(n+1)}\|_2 \leq \frac{7}{c_1}(\Delta_n + \Delta_{n+1})$.*

Proof Sketch.

For $w^{(n+1)} \in \partial H_{\delta_{n+1}}(z^{(n+1)})$ it is $w^{(n+1)} = (w_1^{(n+1)}, w_2^{(n+1)})$ with

$$w_1^{(n+1)} \in \partial g(x^{(n+1)}) + \nabla f(x^{(n+1)}) + 2\delta_n(x^{(n+1)} - x^{(n)})$$
$$w_2^{(n+1)} = -2\delta_n(x^{(n+1)} - x^{(n)})$$

and

$$\|w^{(n+1)}\|_2 \leq \ldots \leq (\frac{1}{\alpha_n} + 4\delta_n + L_n)\Delta_{n+1} + \frac{\beta_n}{\alpha_n}\Delta_n \leq \frac{7}{c_1}(\Delta_{n+1} + \Delta_n)$$

□

Convergence – Requirements, Condition **??**

Lemma

H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??***, i.e. there exists a subsequence $(z^{(n_j)})_{j\in\mathbb{N}}$ with $z^{(n_j)} \to \tilde{z} = (\tilde{x}, \tilde{x})$ and $H_{\delta_{n_j}}(z^{(n_j)}) \to H_\delta(\tilde{z})$ for $j \to \infty$.*

Convergence – Requirements, Condition **??**

Lemma

*H_{δ_n} and $(z^{(n)})_{n\in\mathbb{N}}$ as generated by iPiano satisfy Condition **??**, i.e. there exists a subsequence $(z^{(n_j)})_{j\in\mathbb{N}}$ with $z^{(n_j)} \rightarrow \tilde{z} = (\tilde{x}, \tilde{x})$ and $H_{\delta_{n_j}}(z^{(n_j)}) \rightarrow H_\delta(\tilde{z})$ for $j \rightarrow \infty$.*

Proof Sketch.

Claim 1: by summing Condition **??** and deducing $\sum_{n=0}^\infty \Delta_n^2 < \infty$ it can be shown that $\lim_{n\rightarrow\infty} \Delta_n = 0$.
Claim 2: from the coercivity of h and the Bolzano-Weierstrass theorem it follows the existence of a subsequence $(x^{(n_j)})_{j\in\mathbb{N}}$ with.
Then:

$$\lim_{j\rightarrow\infty} H_{\delta_{n_j+1}}(x^{(n_j+1)}, x^{(n_j)}) = H_\delta(\tilde{x}, \tilde{x}) = h(\tilde{x}).$$

□

Convergence – Kurdyka-Lojasiewicz Property

The Kurdyka-Lojasiewicz property is intended to relate the behavior of the subdifferential ∂H to the function values.

Convergence – Kurdyka-Lojasiewicz Property

The Kurdyka-Lojasiewicz property is intended to relate the behavior of the subdifferential ∂H to the function values.

Definition (Informally)

For a point $\tilde{z} \in \text{dom}(\partial H)$, H is said to satisfy the Kurdyka-Lojasiewicz property if there exists a concave $\phi \in C^1$ with $\phi(0) = 0$ and $\phi' > 0$ such that

$$\phi'(H(z) - H(\tilde{z})) \inf_{\hat{z} \in \partial H(z)} \|\hat{z}\|_2 \geq 1$$

for all z in an appropriate neighborhood of $\tilde{z}$.

Intuitively, the inequality controls the difference in function values by the subdifferential.

Convergence – Kurdyka-Lojasiewicz Property

The Kurdyka-Lojasiewicz property is intended to relate the behavior of the subdifferential ∂H to the function values.

Definition (Informally)

For a point $\tilde{z} \in \text{dom}(\partial H)$, H is said to satisfy the Kurdyka-Lojasiewicz property if there exists a concave $\phi \in C^1$ with $\phi(0) = 0$ and $\phi' > 0$ such that

$$\boxed{\phi'(H(z) - H(\tilde{z})) \inf_{\hat{z} \in \partial H(z)} \|\hat{z}\|_2 \geq 1}$$

for all z in an appropriate neighborhood of $\tilde{z}$.

Intuitively, the inequality controls the difference in function values by the subdifferential.

Convergence – Kurdyka-Lojasiewicz Property

The Kurdyka-Lojasiewicz property is intended to relate the behavior of the subdifferential ∂H to the function values.

Definition (Informally)

For a point $\tilde{z} \in \text{dom}(\partial H)$, H is said to satisfy the Kurdyka-Lojasiewicz property if there exists a concave $\phi \in C^1$ with $\phi(0) = 0$ and $\phi' > 0$ such that

$$\phi'(H(z) - H(\tilde{z})) \inf_{\hat{z} \in \partial H(z)} \|\hat{z}\|_2 \geq 1$$

for all z in an **appropriate neighborhood** of $\tilde{z}$.

Intuitively, the inequality controls the difference in function values by the subdifferential.

Convergence – Convergence Theorem

Theorem

Let H be proper lower semicontinuous, satisfying the Kurdyka-Lojasiewicz property at $\tilde{z} = (\tilde{x}, \tilde{x})$ specified by Condition **??**, *and $(z^{(n)})_{n\in\mathbb{N}}$, satisfying Conditions* **??** - **??**. *Then $(x^{(n)})_{n\in\mathbb{N}}$ converges to $\tilde{x}$ such that $\tilde{z}$ is a critical point of H.*

Convergence – Convergence Theorem

Theorem

Let H be proper lower semicontinuous, satisfying the Kurdyka-Lojasiewicz property at $\tilde{z} = (\tilde{x}, \tilde{x})$ specified by Condition **??***, and $(z^{(n)})_{n\in\mathbb{N}}$, satisfying Conditions* **??** *-* **??***. Then $(x^{(n)})_{n\in\mathbb{N}}$ converges to $\tilde{x}$ such that $\tilde{z}$ is a critical point of H.*

It can further been shown that the convergence rate is $\mathcal{O}\left(1/\sqrt{n}\right)$ for the residual

$$r(x) := x - \text{prox}_g(x - \nabla f(x))$$

in L_2 norm.

Convergence – Convergence Theorem (cont'd)

Proof Sketch.

The proof is based on the following claim:

$$\sum_{i=1}^{n} \Delta_i \leq \frac{1}{2}(\Delta_0 - \Delta_n) + \frac{b}{a}\left[\phi(H(z^{(1)}) - H(\tilde{z})) - \phi(H(z^{(n+1)}) - H(\tilde{z}))\right]$$

which is shown by induction. Then, it follows $\sum_{n=0}^{\infty} \Delta_n < \infty$ and $x^{(n)} \rightarrow \tilde{x}$. Using the Kurdyka-Lojasiewicz property it can be shown that $H(z^{(n)}) \rightarrow H(\tilde{z})$. With Condition **??** it also follows that $\tilde{z}$ is a critical point of H. □

Table of Contents

Implementation – Initialization

Remember, derived bounds for α_0 and β_0:

$$\alpha_0 < \frac{2(1-\beta_0)}{L_0};$$
$$\beta_0 \leq \frac{b_0 - 1}{b_0 - \frac{1}{2}} \quad \text{with} \quad b_0 := \frac{\delta_{-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}}.$$

Guessing an appropriate β_0 is obviously easier than guessing δ_{-1}, so fix β_0 and estimate L_0 using

$$\frac{\|\nabla f(x^{(0)}) - \nabla f(\hat{x})\|_2}{\|x^{(0)} - \hat{x}\|_2} \leq L_0$$

for $\hat{x} = \text{prox}_g(x^{(0)} - \nabla f(x^{(0)}))$.

Implementation – Initialization

Remember, derived bounds for α_0 and β_0:

$$\boxed{\alpha_0 < \frac{2(1-\beta_0)}{L_0};}$$

$$\beta_0 \leq \frac{b_0 - 1}{b_0 - \frac{1}{2}} \quad \text{with} \quad b_0 := \frac{\delta_{-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}}.$$

Guessing an appropriate β_0 is obviously easier than guessing δ_{-1}, so fix β_0 and estimate L_0 using

$$\frac{\|\nabla f(x^{(0)}) - \nabla f(\hat{x})\|_2}{\|x^{(0)} - \hat{x}\|_2} \leq L_0$$

for $\hat{x} = \text{prox}_g(x^{(0)} - \nabla f(x^{(0)}))$.

Implementation – Initialization

Remember, derived bounds for α_0 and β_0:

$$\alpha_0 < \frac{2(1 - \beta_0)}{L_0};$$

$$\boxed{\beta_0 \leq \frac{b_0 - 1}{b_0 - \frac{1}{2}} \quad \text{with} \quad b_0 := \frac{\delta_{-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}}.}$$

Guessing an appropriate β_0 is obviously easier than guessing δ_{-1}, so fix β_0 and estimate L_0 using

$$\frac{\|\nabla f(x^{(0)}) - \nabla f(\hat{x})\|_2}{\|x^{(0)} - \hat{x}\|_2} \leq L_0$$

for $\hat{x} = \text{prox}_g(x^{(0)} - \nabla f(x^{(0)}))$.

Implementation – Initialization

Remember, derived bounds for α_0 and β_0:

$$\alpha_0 < \frac{2(1-\beta_0)}{L_0};$$

$$\beta_0 \leq \frac{b_0 - 1}{b_0 - \frac{1}{2}} \quad \text{with} \quad b_0 := \frac{\delta_{-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}}.$$

Guessing an appropriate β_0 is obviously easier than guessing δ_{-1}, so fix β_0 and estimate L_0 using

$$\boxed{\frac{\|\nabla f(x^{(0)}) - \nabla f(\hat{x})\|_2}{\|x^{(0)} - \hat{x}\|_2} \leq L_0}$$

for $\hat{x} = \text{prox}_g(x^{(0)} - \nabla f(x^{(0)}))$.

Implementation – Initialization (cont'd)

In practice, fix $K \gg 100$ and compute

$$\alpha_0^{(k)} := \alpha_0 - k\frac{a_0 - c_1}{K} \text{ with } a_0 := \frac{2(1-\beta_0)}{(L_0 + 2c_2)} \text{ and } k = 1, \ldots, K$$

until $\alpha_0^{(k)}$ satisfies

$$\delta_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{2\alpha_0^{(k)}} \geq \gamma_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{\alpha_0^{(k)}} \geq c_2.$$

Implementation – Initialization (cont'd)

In practice, fix $K \gg 100$ and compute

$$\boxed{\alpha_0^{(k)} := \alpha_0 - k\frac{a_0 - c_1}{K} \text{ with } a_0 := \frac{2(1-\beta_0)}{(L_0 + 2c_2)} \text{ and } k = 1, \ldots, K}$$

until $\alpha_0^{(k)}$ satisfies

$$\delta_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{2\alpha_0^{(k)}} \geq \gamma_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{\alpha_0^{(k)}} \geq c_2.$$

Implementation – Initialization (cont'd)

In practice, fix $K \gg 100$ and compute

$$\alpha_0^{(k)} := \alpha_0 - k\frac{a_0 - c_1}{K} \text{ with } a_0 := \frac{2(1-\beta_0)}{(L_0 + 2c_2)} \text{ and } k = 1, \ldots, K$$

until $\alpha_0^{(k)}$ satisfies

$$\boxed{\delta_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{2\alpha_0^{(k)}} \geq \gamma_0 := \frac{1}{\alpha_0^{(k)}} - \frac{L_0}{2} - \frac{\beta_0}{\alpha_0^{(k)}} \geq c_2.}$$

Implementation – Finding α_n and β_n

Given L_{n-1} and $\eta > 1$, find the smallest $l \in \mathbb{N}$ such that

$$L_n := \eta^l L_{n-1}$$

satisfies

$$f(x^{(n+1)}) \leq f(x^{(n)}) + \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) + \frac{L_n}{2} \|x^{(n+1)} - x^{(n)}\|_2^2.$$

Alternatively, instead of L_{n-1}, use

$$\frac{\|\nabla f(x^{(n-1)}) - \nabla f(\hat{x})\|_2}{\|x^{(n-1)} - \hat{x}\|_2} \leq L_n$$

with $\hat{x} = \text{prox}_g(x^{(n-1)} - \nabla f(x^{(n-1)}))$ as starting point for Equation (**??**).

Implementation – Finding α_n and β_n

Given L_{n-1} and $\eta > 1$, find the smallest $l \in \mathbb{N}$ such that

$$\boxed{L_n := \eta^l L_{n-1}} \quad (2)$$

satisfies

$$f(x^{(n+1)}) \leq f(x^{(n)}) + \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) + \frac{L_n}{2} \|x^{(n+1)} - x^{(n)}\|_2^2.$$

Alternatively, instead of L_{n-1}, use

$$\frac{\|\nabla f(x^{(n-1)}) - \nabla f(\hat{x})\|_2}{\|x^{(n-1)} - \hat{x}\|_2} \leq L_n$$

with $\hat{x} = \text{prox}_g(x^{(n-1)} - \nabla f(x^{(n-1)}))$ as starting point for Equation (**??**).

Implementation – Finding α_n and β_n

Given L_{n-1} and $\eta > 1$, find the smallest $l \in \mathbb{N}$ such that

$$L_n := \eta^l L_{n-1} \quad (2)$$

satisfies

$$\boxed{\begin{aligned} f(x^{(n+1)}) \leq f(x^{(n)}) &+ \nabla f(x^{(n)})^T(x^{(n+1)} - x^{(n)}) \\ &+ \frac{L_n}{2}\|x^{(n+1)} - x^{(n)}\|_2^2. \end{aligned}}$$

Alternatively, instead of L_{n-1}, use

$$\frac{\|\nabla f(x^{(n-1)}) - \nabla f(\hat{x})\|_2}{\|x^{(n-1)} - \hat{x}\|_2} \leq L_n$$

with $\hat{x} = \text{prox}_g(x^{(n-1)} - \nabla f(x^{(n-1)}))$ as starting point for Equation (**??**).

Implementation – Finding α_n and β_n

Given L_{n-1} and $\eta > 1$, find the smallest $l \in \mathbb{N}$ such that

$$L_n := \eta^l L_{n-1} \tag{2}$$

satisfies

$$f(x^{(n+1)}) \leq f(x^{(n)}) + \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) + \frac{L_n}{2} \|x^{(n+1)} - x^{(n)}\|_2^2.$$

Alternatively, instead of L_{n-1}, use

$$\boxed{\frac{\|\nabla f(x^{(n-1)}) - \nabla f(\hat{x})\|_2}{\|x^{(n-1)} - \hat{x}\|_2} \leq L_n}$$

with $\hat{x} = \text{prox}_g(x^{(n-1)} - \nabla f(x^{(n-1)}))$ as starting point for Equation (**??**).

Implementation – Finding α_n and β_n (cont'd)

Similar to initialization, fix $J, K \gg 100$ and compute

$$\beta_n^{(j)} := \frac{b_n - 1}{b_n - \frac{1}{2}} - \frac{j}{J}\frac{b_n - 1}{b_n - \frac{1}{2}} \quad \text{with} \quad b_n := \frac{\delta_{n-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}} \text{ and } j = 0, \ldots, J,$$

$$\alpha_n^{(k)} := a_n - k\frac{a_n - c_1}{K} \quad \text{with} \quad a_n := \frac{2(1 - \beta_n)}{(L_n + 2c_2)} \text{ and } k = 1, \ldots, K$$

until $\beta_n^{(j)}$ and $\alpha_n^{(k)}$ satisfy

$$\delta_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{2\alpha_n^{(k)}} \geq \gamma_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{\alpha_n^{(k)}} \geq c_2$$

and

$$\delta_n \leq \delta_{n-1}.$$

Implementation – Finding α_n and β_n (cont'd)

Similar to initialization, fix $J, K \gg 100$ and compute

$$\boxed{\begin{array}{l} \beta_n^{(j)} := \dfrac{b_n - 1}{b_n - \frac{1}{2}} - \dfrac{j}{J}\dfrac{b_n - 1}{b_n - \frac{1}{2}} \quad \text{with} \quad b_n := \dfrac{\delta_{n-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}} \text{ and } j = 0, \ldots, J, \\ \alpha_n^{(k)} := a_n - k\dfrac{a_n - c_1}{K} \quad \text{with} \quad a_n := \dfrac{2(1 - \beta_n)}{(L_n + 2c_2)} \text{ and } k = 1, \ldots, K \end{array}}$$

until $\beta_n^{(j)}$ and $\alpha_n^{(k)}$ satisfy

$$\delta_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{2\alpha_n^{(k)}} \geq \gamma_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{\alpha_n^{(k)}} \geq c_2$$

and

$$\delta_n \leq \delta_{n-1}.$$

Implementation – Finding α_n and β_n (cont'd)

Similar to initialization, fix $J, K \gg 100$ and compute

$$\beta_n^{(j)} := \frac{b_n - 1}{b_n - \frac{1}{2}} - \frac{j}{J}\frac{b_n - 1}{b_n - \frac{1}{2}} \quad \text{with} \quad b_n := \frac{\delta_{n-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}} \text{ and } j = 0, \ldots, J,$$

$$\alpha_n^{(k)} := a_n - k\frac{a_n - c_1}{K} \quad \text{with} \quad a_n := \frac{2(1 - \beta_n)}{(L_n + 2c_2)} \text{ and } k = 1, \ldots, K$$

until $\beta_n^{(j)}$ and $\alpha_n^{(k)}$ satisfy

$$\boxed{\delta_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{2\alpha_n^{(k)}} \geq \gamma_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{\alpha_n^{(k)}} \geq c_2}$$

and

$$\delta_n \leq \delta_{n-1}.$$

Implementation – Finding α_n and β_n (cont'd)

Similar to initialization, fix $J, K \gg 100$ and compute

$$\beta_n^{(j)} := \frac{b_n - 1}{b_n - \frac{1}{2}} - \frac{j}{J}\frac{b_n - 1}{b_n - \frac{1}{2}} \quad \text{with} \quad b_n := \frac{\delta_{n-1} + \frac{L_n}{2}}{c_2 + \frac{L_n}{2}} \text{ and } j = 0, \ldots, J,$$

$$\alpha_n^{(k)} := a_n - k\frac{a_n - c_1}{K} \quad \text{with} \quad a_n := \frac{2(1 - \beta_n)}{(L_n + 2c_2)} \text{ and } k = 1, \ldots, K$$

until $\beta_n^{(j)}$ and $\alpha_n^{(k)}$ satisfy

$$\delta_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{2\alpha_n^{(k)}} \geq \gamma_n := \frac{1}{\alpha_n^{(k)}} - \frac{L_n}{2} - \frac{\beta_n^{(j)}}{\alpha_n^{(k)}} \geq c_2$$

and

$$\boxed{\delta_n \leq \delta_{n-1}.}$$

Table of Contents

Denoising – Model

Given a noisy image $u^{(0)} : \Omega = [0,1]^2 \rightarrow [0,1]$, minimize

$$h(u; u^{(0)}, \lambda) = \int_\Omega \rho_1(u(x) - u^{(0)}(x))dx + \lambda \int_\Omega \rho_2(\|\nabla u(x)\|_2)dx$$

with

$$\rho_{1,\text{abs}} = |x| \text{ and } \rho_{1,\text{sqr}}(x) = x^2;$$
$$\rho_2(x) = \log\left(1 + \frac{x^2}{\sigma^2}\right).$$

Denoising – Model

Given a noisy image $u^{(0)} : \Omega = [0,1]^2 \rightarrow [0,1]$, minimize

$$h(u; u^{(0)}, \lambda) = \boxed{\int_\Omega \rho_1(u(x) - u^{(0)}(x))dx} + \lambda \int_\Omega \rho_2(\|\nabla u(x)\|_2)dx$$

with

$$\rho_{1,\text{abs}} = |x| \text{ and } \rho_{1,\text{sqr}}(x) = x^2;$$
$$\rho_2(x) = \log\left(1 + \frac{x^2}{\sigma^2}\right).$$

Denoising – Model

Given a noisy image $u^{(0)} : \Omega = [0,1]^2 \rightarrow [0,1]$, minimize

$$h(u; u^{(0)}, \lambda) = \int_\Omega \rho_1(u(x) - u^{(0)}(x))dx + \boxed{\lambda \int_\Omega \rho_2(\|\nabla u(x)\|_2)dx}$$

with

$$\rho_{1,\text{abs}} = |x| \text{ and } \rho_{1,\text{sqr}}(x) = x^2;$$
$$\rho_2(x) = \log\left(1 + \frac{x^2}{\sigma^2}\right).$$

Denoising – Model

Given a noisy image $u^{(0)} : \Omega = [0,1]^2 \rightarrow [0,1]$, minimize

$$h(u; u^{(0)}, \lambda) = \int_\Omega \rho_1(u(x) - u^{(0)}(x))dx + \lambda \int_\Omega \rho_2(\|\nabla u(x)\|_2)dx$$

with

$$\rho_{1,\text{abs}} = |x| \text{ and } \rho_{1,\text{sqr}}(x) = x^2;$$
$$\rho_2(x) = \log\left(1 + \frac{x^2}{\sigma^2}\right).$$

$\rho_{1,\text{sqr}}$ and ρ_2 are differentiable; the proximal mapping of $\rho_{1,\text{abs}}(x - x^{(0)})$ is

$$\text{prox}_{\alpha\rho_{1,\text{abs}}}(x) = \max(0, |x| - \alpha) \cdot \text{sign}(x) - x^{(0)}.$$

Denoising – Results

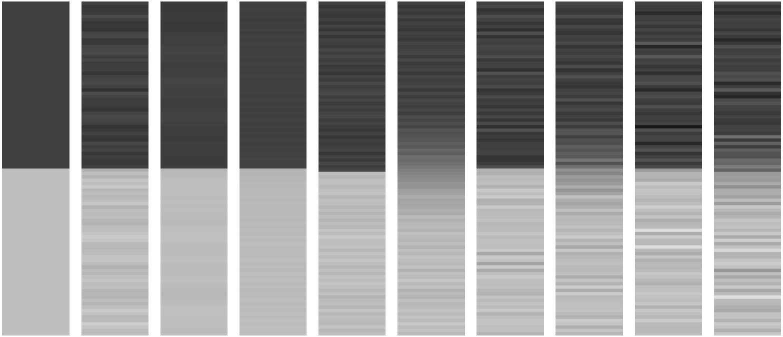

Figure : Signal denoising experiment; input signal shown on the left with the perturbed/noisy signal on its right. Results using $\rho_{1,\text{abs}}$ and $\rho_{1,\text{sqr}}$ with $\lambda \in \{0.2, 0.4, 0.6, 0.8\}$ are shown.

Denoising – Convergence

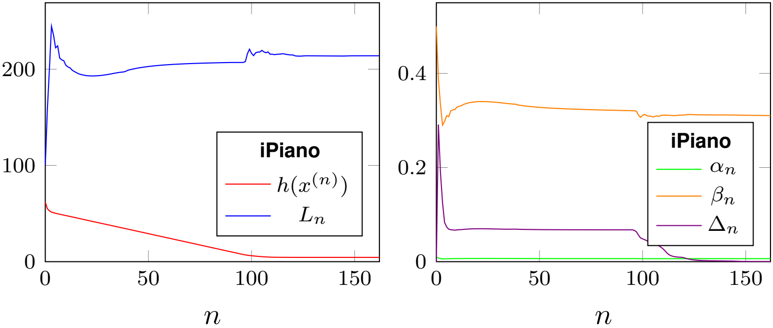

Figure : Convergence of iPiano. Shown is the value of the objective function $h(x(n))$ for each iterate $x(n)$, $n \geq 0$, as well as the corresponding parameters α_n, β_n and L_n. Furthermore, $\Delta_n := \|x^{(n)} - x^{(n-1)}\|_2$ is shown.

Denoising – Results (cont'd)

Figure : Image denoising experiment; noisy image in the top row, $\rho_{1,\text{abs}}$ in the middle row and $\rho_{1,\text{sqr}}$ in the bottom row.

Binary Segmentation – Model

Binary segmentation based on an approximation of the Mumford-Shah model [**?**, **?**]; $u : [0,1]^2 \to [-1,1]$:

$$h_\epsilon(u; c_+, c_-, u^{(0)}, \lambda) = \int_\Omega \left(9\epsilon\|\nabla u(x)\|_2^2 + \frac{(1-u(x)^2)^2}{64\epsilon}\right) dx$$
$$+ \lambda \int_\Omega \left(\frac{1+u(x)}{2}\right)^2 (u^{(0)}(x) - c_+)^2 dx$$
$$+ \lambda \int_\Omega \left(\frac{1-u(x)}{2}\right)^2 (u^{(0)}(x) - c_-)^2 dx.$$

Binary Segmentation – Model

Binary segmentation based on an approximation of the Mumford-Shah model [**?**, **?**]; $u : [0,1]^2 \to [-1,1]$:

$$h_\epsilon(u; c_+, c_-, u^{(0)}, \lambda) = \boxed{\int_\Omega \left(9\epsilon \|\nabla u(x)\|_2^2 + \frac{(1 - u(x)^2)^2}{64\epsilon}\right) dx}$$
$$+ \lambda \int_\Omega \left(\frac{1 + u(x)}{2}\right)^2 (u^{(0)}(x) - c_+)^2 dx$$
$$+ \lambda \int_\Omega \left(\frac{1 - u(x)}{2}\right)^2 (u^{(0)}(x) - c_-)^2 dx.$$

(It can be shown, that for $\epsilon \to 0$,

$$\int_\Omega \left(9\epsilon \|\nabla u(x)\|_2^2 + \frac{(1 - u(x)^2)^2}{64\epsilon}\right) dx$$

approximates $|u|_{BV}$.)

Binary Segmentation – Model

Binary segmentation based on an approximation of the Mumford-Shah model [**?**, **?**]; $u : [0,1]^2 \to [-1,1]$:

$$h_\epsilon(u; c_+, c_-, u^{(0)}, \lambda) = \int_\Omega \left(9\epsilon\|\nabla u(x)\|_2^2 + \frac{(1-u(x)^2)^2}{64\epsilon}\right) dx$$
$$\boxed{\begin{aligned} &+ \lambda \int_\Omega \left(\frac{1+u(x)}{2}\right)^2 (u^{(0)}(x) - c_+)^2 dx \\ &+ \lambda \int_\Omega \left(\frac{1-u(x)}{2}\right)^2 (u^{(0)}(x) - c_-)^2 dx. \end{aligned}}$$

(It can be shown, that for $\epsilon \to 0$,

$$\int_\Omega \left(9\epsilon\|\nabla u(x)\|_2^2 + \frac{(1-u(x)^2)^2}{64\epsilon}\right) dx$$

approximates $|u|_{BV}$.)

Binary Segmentation – Results (cont'd)

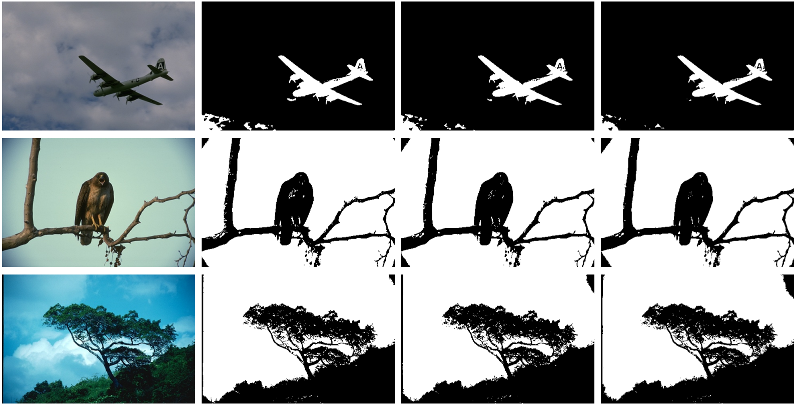

Figure : Segmentation results for thresholds $\tau = -0.2, 0.0, 0.2$ and using g_{sqr}; the foreground segment S_f is depicted in white.

Table of Contents

Conclusion

We discussed the minimization of composite functions of the form

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)).$$

Conclusion

We discussed the minimization of composite functions of the form

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)).$$

Ochs et al. [**?**] proposed the iPiano algorithm to solve this problem under to following requirements:

- g proper closed convex and lower semi continuous;
- $f \in C^1$ with L-Lipschitz continuous ∇f;
- h coercive and bounded below;
- and $H_{\delta_n}(x^{(n)}, x^{(n-1)}) = h(x^{(n)}) + \delta_n \Delta_n$ satisfying the Kurdyka-Lojasiewicz property [**?**, **?**] at a critical point.

Conclusion

We discussed the minimization of composite functions of the form

$$\min_{x \in \mathbb{R}^d} h(x) = \min_{n \in \mathbb{R}^d} (f(x) + g(x)).$$

Ochs et al. [**?**] proposed the iPiano algorithm to solve this problem under to following requirements:

- g proper closed convex and lower semi continuous;
- $f \in C^1$ with L-Lipschitz continuous ∇f;
- h coercive and bounded below;
- and $H_{\delta_n}(x^{(n)}, x^{(n-1)}) = h(x^{(n)}) + \delta_n \Delta_n$ satisfying the Kurdyka-Lojasiewicz property [**?**, **?**] at a critical point.

The algorithm can be implemented efficiently in C++ and used to solve image processing tasks.

Appendix – Kurdyka-Lojasiewicz Property

Definition

H has the Kurdyka-Lojasiewicz property at point $\tilde{z} \in \text{dom}(\partial H)$ there exist $\eta \in (0, \infty]$, a neighborhood U of $\tilde{z}$, and a continuous concave function $\phi : [0, \eta) \to \mathbb{R}_+$ such that

- $\phi \in C^1((0, \eta))$, $\phi(0) = 0$, and for all $s \in (0, \eta)$, $\phi'(s) > 0$;
- and for all $z \in U \cap \{z \in \mathbb{R}^{2d} | H(\tilde{z}) < H(z) < H(\tilde{z}) + \eta\}$ the Kurdyka-Lojasiewicz inequality holds:

$$\phi'(H(z) - H(\tilde{z})) \inf_{\hat{z} \in \partial H(z)} \|\hat{z}\|_2 \geq 1.$$

Appendix – Kurdyka-Lojasiewicz Property

Definition

H has the Kurdyka-Lojasiewicz property at point $\tilde{z} \in \text{dom}(\partial H)$ there exist $\eta \in (0, \infty]$, a neighborhood U of $\tilde{z}$, and a continuous concave function $\phi : [0, \eta) \to \mathbb{R}_+$ such that

- $\phi \in C^1((0, \eta))$, $\phi(0) = 0$, and for all $s \in (0, \eta)$, $\phi'(s) > 0$;
- and for all $z \in U \cap \{z \in \mathbb{R}^{2d} | H(\tilde{z}) < H(z) < H(\tilde{z}) + \eta\}$ the Kurdyka-Lojasiewicz inequality holds:

$$\boxed{\phi'(H(z) - H(\tilde{z})) \inf_{\hat{z} \in \partial H(z)} \|\hat{z}\|_2 \geq 1.}$$

Intuitively, for $H \in C^1$, this means that ϕ has to be steep around critical points $\tilde{z}$ of H where ∇H is flat.