Exercise 6. Variational Methods and the Primal Dual Algorithm

Dongho Kang

16-948-598

May 28, 2017

MATLAB R2016b version was used for coding and testing:

MathWorks, MATLAB R2016b (9.1.0.441655)
64-bit (maci64)

The ***code*** directory contains the followings:

part2.m script .m file for exercise part 2.

part3_1.m script .m file for exercise part 3, task 1.

part3_2.m script .m file for exercise part 3, task 2.

functions directory which contains functions by me.

Part 2 - Interactive Segmentation provided directory with modified code.

Part 3 - Inpainting provided directory with modified code.

IMPORTANT: Code implementation should be run by those scripts above only. Every custom function paths and parameters are only set in the script ***part2.m***, ***part3_1.m*** and ***part3_2.m***.

In order to run each task program, go to the code directory and run script by MATLAB command *part2*, *part3_1* or *part3_2*

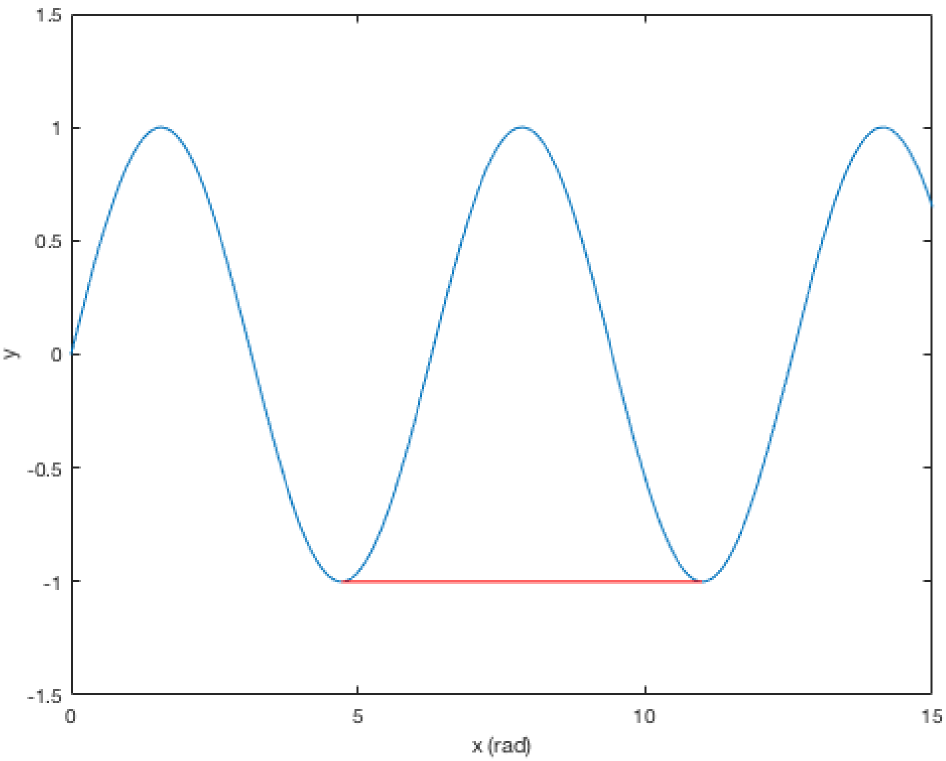
1 Exercise part 1: Convexity of the Rudin Osher Fatemi Functional

1.1 Task 1: Proof of the convexity

In this part, the following functions were checked if they are convex or not.

- $x \to \sin(x)$

Figure 1.1: $f(x) = \sin(x)$



$f(x) = \sin(x)$ **is not a convex function.** Recall the definition of convex function:

Definition 1. *A function $f : \mathcal{V} \to \mathbb{R}$ is called convex if the function domain $\mathcal{V}$ is a convex set and if*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \qquad \forall x, y \in \mathcal{V} \quad and \quad 0 \leq \lambda \leq 1.$$

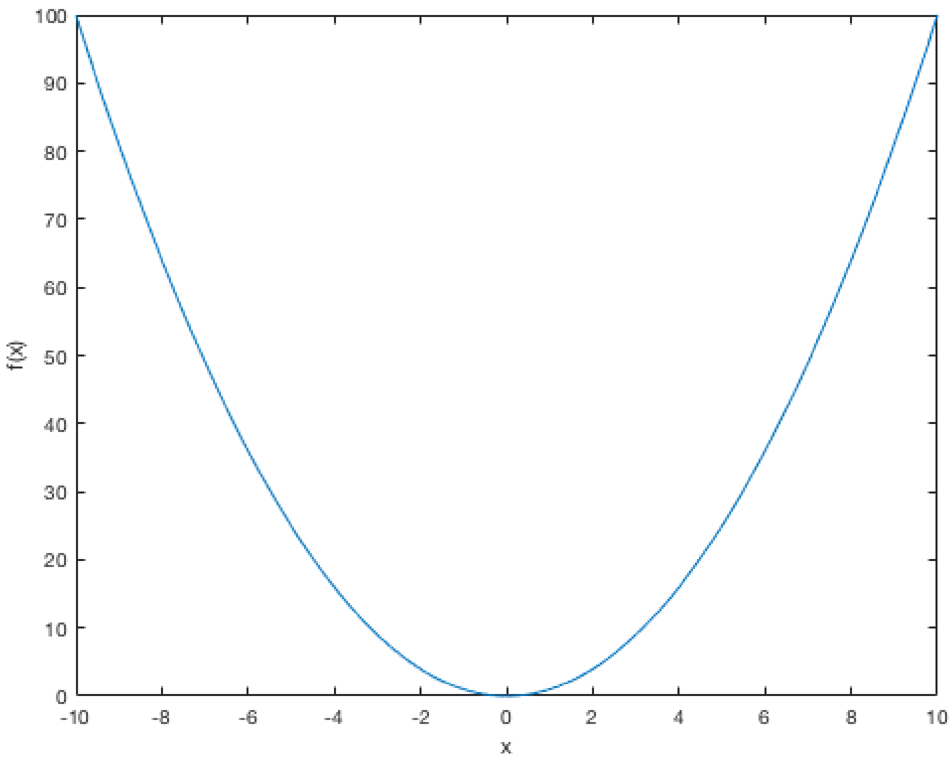
For $\lambda = \frac{1}{2}$, $x = \frac{3}{2}\pi$ and $y = \frac{7}{2}\pi$, (as Figure 1.1)

$$f(\frac{3}{4}\pi + \frac{7}{4}\pi) = \sin(\frac{10}{4}\pi) = 1 \nleq \frac{1}{2} f(\frac{3}{4}\pi) + \frac{1}{2} f(\frac{7}{4}\pi) = \frac{1}{2}\sin(\frac{3}{4}\pi) + \frac{1}{2}\sin(\frac{7}{4}\pi) = 0 \qquad (1.1)$$

thus $f(x) = \sin(x)$ is not a convex function.

- $x \to x^2$

Figure 1.2: $f(x) = x^2$



$f(x) = x^2$ **is a convex function.**

Theorem 1. *First order condition: A differentiable function $f : \mathcal{V} \to \mathbb{R}$ is convex iff*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \qquad \forall x, y \in \mathcal{V} \tag{1.2}$$

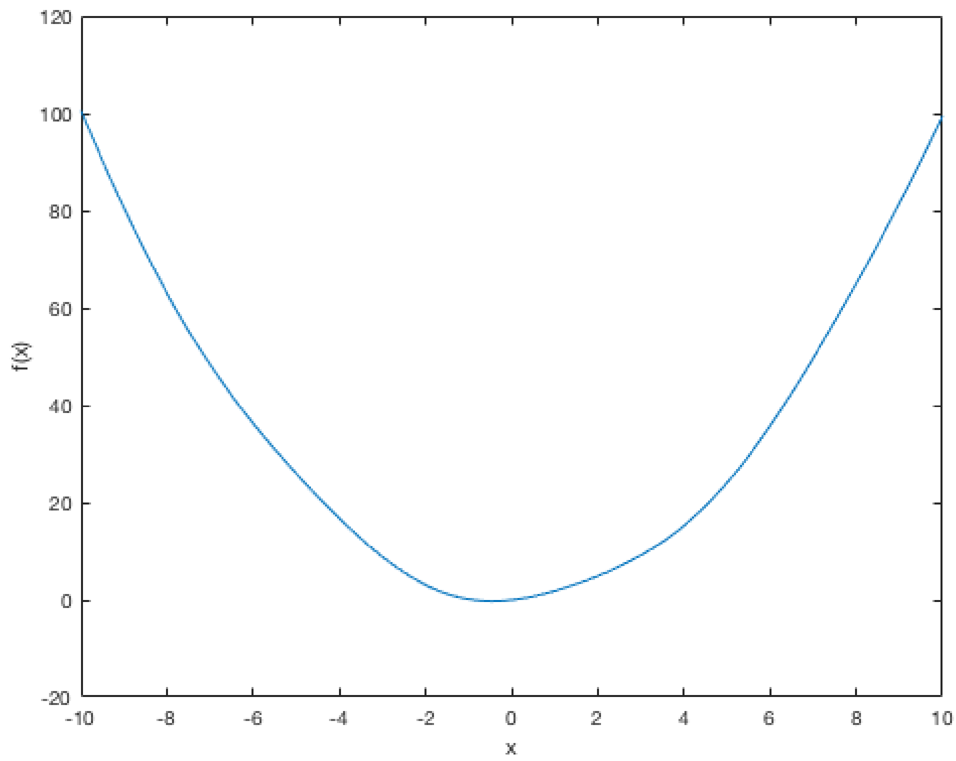
By the theorem, for $f(x) = x^2$

$$\begin{align} f(x) + \nabla f(x)^T (y - x) &= x^2 + 2x(y - x) \tag{1.3} \\ &= x^2 + 2xy - 2x^2 \tag{1.4} \\ &= 2xy - x^2 \tag{1.5} \\ f(y) - f(x) - \nabla f(x)^T (y - x) &= y^2 - 2xy + x^2 \tag{1.6} \\ &= (y - x)^2 \geq 0 \tag{1.7} \\ \therefore f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{1.8} \end{align}$$

By (1.8) and Theorem 1, $f(x) = x^2$ is a convex function.

- $x \to \sin(x) + x^2$

Figure 1.3: $f(x) = \sin(x) + x^2$



$f(x) = \sin(x) + x^2$ **is a convex function.** Here is a proof:

Theorem 2. *Second order condition: A twice differentiable function $f : \mathcal{V} \to \mathbb{R}$ is convex iff*

$$\nabla^2 f(x) \geq 0 \qquad \forall x \in \mathcal{V} \tag{1.9}$$

$$\begin{aligned} \nabla^2 f(x) &= \nabla^2 \left[\sin(x) + x^2 \right] && (1.10) \\ &= \frac{\partial^2}{\partial x^2} \left[\sin(x) + x^2 \right] && (1.11) \\ &= -\cos(x) + 2 && (1.12) \end{aligned}$$

Since $1 \geq \cos(x) \geq -1$, $\nabla^2 f(x) = -\cos(x) + 2 \geq 1 \geq 0$.

Thus, by Theorem2, $f(x) = \sin(x) + x^2$ is a convex function.

1.2 Task 2: Proof of the convexity of the Rudin Osher Fatemi functional

Here, convexity of the following Rudin Osher Fatemi functional is proved.

$$E_{ROF}(I_u) = \int_\Omega \left[\left| \nabla I_u(\mathbf{x}) \right| + \left\| I_u(\mathbf{x}) - I_0(\mathbf{x}) \right\|_2^2 \right] d\mathbf{x} \tag{1.13}$$

Let f_1, f_2 are defined as follows, thus $E_{ROF}(I_u) = f_1(I_u) + f_2(I_u)$:

$$f_1(I_u) = \int_\Omega \left| \nabla I_u(\mathbf{x}) \right| d\mathbf{x} \tag{1.14}$$

$$f_2(I_u) = \int_\Omega \left\| I_u(\mathbf{x}) - I_0(\mathbf{x}) \right\|_2^2 d\mathbf{x} \tag{1.15}$$

For f_1, and certain images $I_1(\mathbf{x}), I_2(\mathbf{x})$:

$$f_1(\lambda I_1 + (1-\lambda) I_2) = \int_\Omega \left| \nabla \big(\lambda I_1 + (1-\lambda) I_2 \big) \right| d\mathbf{x} \tag{1.16}$$

$$= \int_\Omega \left| \lambda \nabla I_1 + (1-\lambda) \nabla I_2 \right| d\mathbf{x} \tag{1.17}$$

by triangular inequality

$$\leq \int_\Omega \left[\left| \lambda \nabla I_1 \right| + \left| (1-\lambda) \nabla I_2 \right| \right] d\mathbf{x} \tag{1.18}$$

$$\leq \int_\Omega \left[|\lambda| \left| \nabla I_1 \right| + |1-\lambda| \left| \nabla I_2 \right| \right] d\mathbf{x} \tag{1.19}$$

$\forall \lambda, 0 \leq \lambda \leq 1$

$$= \int_\Omega \left[\lambda \left| \nabla I_1 \right| + (1-\lambda) \left| \nabla I_2 \right| \right] d\mathbf{x} \tag{1.20}$$

$$= \lambda \int_\Omega \left| \nabla I_1 \right| d\mathbf{x} + (1-\lambda) \int_\Omega \left| \nabla I_2 \right| d\mathbf{x} \tag{1.21}$$

$$= \lambda f_1(I_1) + (1-\lambda) f_1(I_2) \tag{1.22}$$

$$\therefore f_1(\lambda I_1 + (1-\lambda) I_2) \leq \lambda f_1(I_1) + (1-\lambda) f_1(I_2) \tag{1.23}$$

By (1.23) and Theorem 1, f_1 **is a convex function.**

For f_2,

$$\nabla^2_{I_u} f_2 = \nabla^2_{I_u} \left[\int_\Omega \left\| I_u(\mathbf{x}) - I_0(\mathbf{x}) \right\|_2^2 d\mathbf{x} \right] \tag{1.24}$$

$$= \int_\Omega \nabla^2_{I_u} \left\| I_u(\mathbf{x}) - I_0(\mathbf{x}) \right\|_2^2 d\mathbf{x} \tag{1.25}$$

$$= \int_\Omega 2 d\mathbf{x} \geq 0 \tag{1.26}$$

By Theorem 2, f_2 **is a convex function.** Since sum of two convex functions is convex, $E_{ROF}(I_u)$ **is convex.**

2 EXERCISE PART 2: SEGMENTATION REVISITED

2.1 DESCRIPTION

For segmentation problem, cost function has the following form:

$$G(x) = \langle x, f \rangle + \delta_{[0,1]}(x) \tag{2.1}$$

f and $\delta_{[0,1]}(x)$ is defined as follows:

$$f_i = \log H_{bg}(I_i) - \log H_{fg}(I_i) \qquad \text{for all } i \in \mathcal{D}_{\mathcal{I}} \tag{2.2}$$

$$\delta_{[0,1]}(x) = \begin{cases} 0 & \text{if } x \in [0,1] \\ \infty & \text{if } x \notin [0,1] \end{cases} \tag{2.3}$$

where H_{fg} and H_{bg} are the color histogram.

Thus, the primal dual algorithm is used to solve following equation:

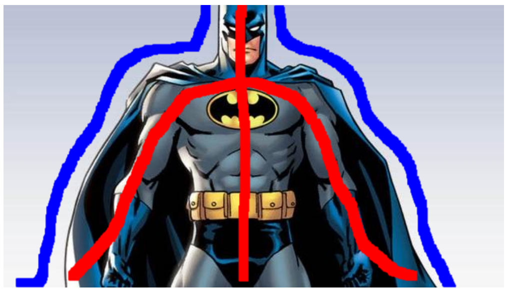
$$\begin{aligned} &\min_{x\in X}\max_{y\in D_X} \langle \nabla x, y \rangle + \lambda G(x) - \delta_Y(y) \\ &= \min_{x\in X}\max_{y\in D_X} \langle \nabla x, y \rangle + \lambda \left[\langle x, f \rangle + \delta_{[0,1]}(x) \right] - \delta_Y(y) && (2.4) \\ &= \min_{x\in X}\max_{y\in D_X} \langle \nabla x, y \rangle + \lambda \langle x, f \rangle + \delta_{[0,1]}(x) - \delta_Y(y) && (2.5) \\ &= \min_{x\in X}\max_{y\in D_X} \langle \nabla x, y \rangle + \langle x, \lambda f \rangle + \delta_{[0,1]}(x) - \delta_Y(y) && (2.6) \end{aligned}$$

2.2 RESULTS

The original image and scribbles for getting the color histogram is as Figure 2.1.

Figure 2.1: batman.jpg

(a) batman.jpg



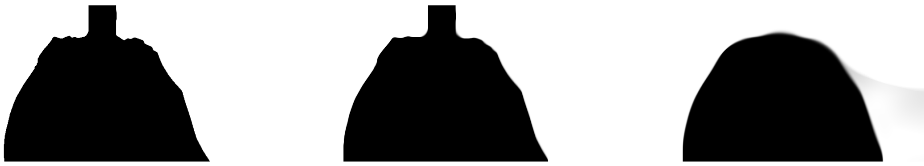
(b) Scribbles for the color histogram

The segmentation result with $\lambda = 0.0075$ is as Figure 2.3 (after 2000 iterations).

Figure 2.3: Segmentation result($\lambda = 0.0075$)

The segmentation result with different value of λ is as follows:

Figure 2.4: Function x with different λ



(a) $\lambda = 0.075$ (b) $\lambda = 0.0075$ (c) $\lambda = 0.00075$

2.3 Discussion

- Primal dual TV segmentation vs Graph-cuts
  - Primal dual TV segmentation is easy to parallelize thus faster on a multi cores.
  - Graph-cuts is faster on a single core.
  - Primal dual TV treats image as continuous function, thus there's no metrication error. (discretization errors originated from the graph neighborhood)
  - Primal dual TV is more memory efficient: it does not require graph structure.
  - Graph-cuts is more intuitive: each nodes represent each pixels of a image.
  - Primal dual TV has mathematically clear formulation but more complex.
  - For graph-cuts, non-metric smoothness terms can be applied.

3 Task 3: Applications of Inpainting

3.1 Description

For recovering(inpainting) problem, cost function has the following form:

$$G(x) = \frac{1}{2} \sum_{i,j \in \mathcal{D}_I \setminus \mathcal{I}} \frac{1}{2}(I_{i,j} - x_{i,j}) \tag{3.1}$$

where, $\mathcal{D}_I$ is the domain of 2D image I, and $\mathcal{I}$ is the set of all the missing pixels.

The primal dual algorithm is used to solve following equation:

$$\begin{aligned} &\min_{x \in X} \max_{y \in D_X} \langle \nabla x, y \rangle + \lambda G(x) - \delta_Y(y) \\ &= \min_{x \in X} \max_{y \in D_X} \langle \nabla x, y \rangle + \frac{\lambda}{2} \sum_{i,j \in \mathcal{D}_I \setminus \mathcal{I}} \frac{1}{2}(I_{i,j} - x_{i,j}) - \delta_Y(y) \end{aligned} \tag{3.2}$$

3.2 Task 1: Recovering an image with a high number of missing pixels

3.2.1 Results

For this task, primal dual algorithm used for recovering damaged image which 90 % of the pixels were removed.

Figure 3.1: Original image and damaged image

(a) Original image oz2.jpg

(b) Damaged image

As iteration goes on, damaged image is getting recovered. Figure 3.3 shows the evolution of image with $\lambda = 5$.

Figure 3.3: Image evolution by inpainting

(a) iter = 1 (b) iter = 50 (c) iter = 200

(d) iter = 500 (e) iter = 1000 (f) iter = 2000

Figure 3.5: Original image and final result

(a) Original image oz2.jpg (b) result of iter = 2000

The result with different λ is as Figure 3.7.

Figure 3.7: Result with different λ

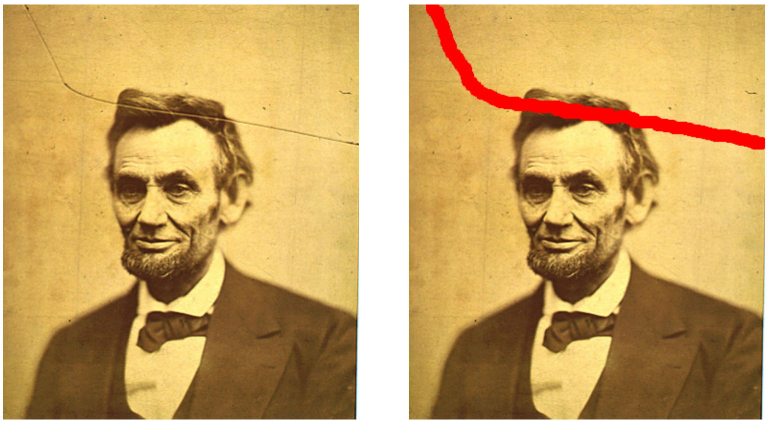
(a) $\lambda = 0.5$ (b) $\lambda = 5$ (c) $\lambda = 50$

3.3 Task 2: Removing an object from an image

3.3.1 Results

In this task, artifacts of the original image were removed and inpainted. Figure 3.13 is original image with artifacts and scrabbles which indicates the region to be removed.

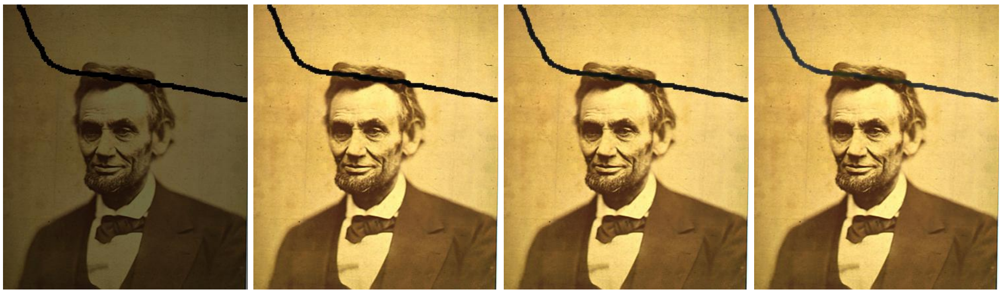
Figure 3.9: Original image and damaged image



(a) iter = 1 (b) iter = 50

After removing artifacts regions, the image is getting recovered as Figure 3.11.

Figure 3.11: Image eveolution by inpainting



(a) iter = 1 (b) iter = 50 (c) iter = 100 (d) iter = 200

(e) iter = 500 (f) iter = 750 (g) iter = 1000 (h) iter = 2000

Figure 3.13: Original image and recovered image

(a) lincoln.jpg

(b) result of iter = 2000