Variational Methods for Computer Vision: Exercise Sheet 1

Exercise: October 23, 2014

Part I: Theory

The following exercises should be **solved at home**. You do not have to hand in your solutions, however, writing it down will help you present your answer during the tutorials.

1. Let X be an arbitrary set. A metric on X is a function

$$d \colon X \times X \to \mathbb{R}$$

satisfying for all $x, y, z \in X$ the following conditions:

- $d(x, y) \geq 0,$ (non-negativity)
- $d(x, y) = 0 \Leftrightarrow x = y,$ (identity of indiscernibles)
- $d(x, y) = d(y, x),$ (symmetry)
- $d(x, y) \leq d(x, z) + d(z, y).$ (subadditivity / triangle inequality)

Verify whether or not the following functions are metrics:

(a) For $X = \mathbb{R}^n$, the Manhattan distance:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|.$$

(b) For $X = \mathbb{R}^n$, the Mahalanobis distance:

$$d(x, y) = \langle x - y, Q(x - y) \rangle^{\frac{1}{2}} ,$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix.

(c) The Kullback-Leibler (KL) divergence between two probability distributions on $\mathbb{R}$:

$$d(x, y) = \mathrm{KL}(x \| y) = - \int_{\mathbb{R}} x(t) \ln \frac{y(t)}{x(t)} \, \mathrm{d}t.$$

2. Let $f, g, h \in L^1(\mathbb{R})$ be absolutely integrable functions. Consider the convolution of the functions f and g:

$$(f * g)(x) = \int_{\mathbb{R}} f(y) \, g(x - y) \, \mathrm{d}y.$$

(a) Show that:

$$(f * g) * h = f * (g * h) \qquad \text{and} \qquad f * (g + h) = f * g + f * h.$$

(b) Let $\mathcal{F}$ denote the Fourier transform operator:

$$\mathcal{F}\{f\} := \hat{f}(\nu) = \int_{\mathbb{R}} f(x) e^{-2\pi i x \nu} \, \mathrm{d}x.$$

Prove that the Fourier transform of the convolution of two functions is the same as the the pointwise multiplication of the respective Fourier transforms hence:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}.$$

What implications does this have for computing the convolution?

(c) Additionally let f and g be continuously differentiable. Show that:

$$\frac{d}{dx}(f * g) = \frac{df}{dx} * g = \frac{dg}{dx} * f.$$

Part II: Practical Exercises

This exercise is to be solved **during the tutorial**.

1. Download the archive `vmcv_ex01.zip` and unzip it on your home folder. In there should be a file named `coins.png`.
2. Start Matlab and load the unzipped image using the following command:

   ```
   f=double(imread('coins.png'));
   ```

   Show the image using Matlab's command:

   ```
   figure; imshow(uint8(f));
   ```

   Also familiarize yourself with the following commands:

   ```
   for, size, zeros, ones, matrix indexing
   ```

   using Matlab's great documentation or the Tutorials listed below.
3. Compute the convolution of the image with a Gaussian kernel. In theory, the Gaussian distribution is nonzero everywhere, however in practice we restrict ourself to truncated kernels. Set the radius of the kernel to $r = \text{ceil}(3 \times \sigma)$. The discrete convolution is given as:

   $$g(i,j) = (w * f)(i,j) := \sum_{m=-r}^{r} \sum_{n=-r}^{r} w(m,n)\, f(i-m, j-n).$$

   The discrete truncated Gaussian kernel can be written as follows:

   $$w(m,n) \propto \exp\left(-\frac{m^2+n^2}{2\sigma^2}\right)$$

   In order to stay consistent with the continuous formulation of the Gaussian distribution make sure to normalize the kernel function such that the following holds:

   $$\sum_{m=-r}^{r} \sum_{n=-r}^{r} w(m,n) = 1.$$

   For simplicity ignore pixels where the convolution filter goes beyond the edge of the image.
4. Let W and H denote respectively the width and height of the input image f. Compute the the gradient $\nabla f = (\partial_x^+ f, \partial_y^+ f)^\mathsf{T}$ of the image using the discretization scheme of forward differences:

   $$(\partial_x^+ f)_{i,j} = \begin{cases} f_{i+1,j} - f_{i,j} & \text{if } i < W \\ 0 & i = W. \end{cases}$$

   $$(\partial_y^+ f)_{i,j} = \begin{cases} f_{i,j+1} - f_{i,j} & \text{if } j < H \\ 0 & j = H. \end{cases}$$

   Notice that the boundary values of the gradient are set to zero.

5. Try solving exercise 4 by avoiding using any for loops this time. Can you tell the difference?
6. Let f_σ be the input image convolved with a Gaussian kernel of standard deviation σ. Compute the magnitude of the Gradient $|\nabla f_\sigma|$ for different values for σ. What do you observe?

Matlab-Tutorials:

`http://www.math.utah.edu/lab/ms/matlab/matlab.html`
`http://www.math.ufl.edu/help/matlab-tutorial/`
`http://www.glue.umd.edu/~nsw/ench250/matlab.htm`