R. Beinert
M. Hasannasab

Exercise Sheet 2

due date: April 25, 2021, 23:59

Exercise 2.1.

Prove part (iii) of Theorem 5.2, that is: Let $f \in L_p(\mathbb{T})$ and $g \in L_q(\mathbb{T})$, where $1 \leq p, q, r \leq \infty$ and $\frac{1}{p} + \frac{1}{q} = \frac{1}{r} + 1$. Then $f * g \in L_r(\mathbb{T})$ and

$$\|f * g\|_{L_r(\mathbb{T})} \leq \|f\|_{L_p(\mathbb{T})} \|g\|_{L_q(\mathbb{T})}.$$

Exercise 2.2.

Let D_N be the Nth Dirichlet kernel. Show that $D_N^2 \in L_2(\mathbb{T})$ and compute the Fourier coefficients $c_k(D_N^2)$. The Fejér kernel is defined by

$$F_n(x) \coloneqq \frac{1}{n+1} \sum_{j=0}^{n} D_j(x) = \sum_{k=-n}^{n} \left(1 - \frac{|k|}{n+1}\right) \exp(2\pi i k x) = \frac{1}{n+1} \left(\frac{\sin(\pi(n+1)x)}{\sin(\pi x)}\right)^2.$$

What is the connection of $c_k(D_N^2)$ to the Fejér kernel?

Exercise 2.3.

The *discrete convolution* of two sequences f and w is defined by the sequence $u = f * w$, where

$$u(k) = \sum_{j \in \mathbb{Z}} f(j) w(k-j) = \sum_{j \in \mathbb{Z}} f(k-j) w(j).$$

(i) Consider the two sequences $f = (\ldots, 0, f(0), f(1), f(2), 0, \ldots)$ and $w = (\ldots, 0, w(0), w(1), w(2), 0, \ldots)$. Determine the convolution $u = f * w$ of f and w.

(ii) When dealing with finite sequences f and w (represented as vectors), there are several ways to continue f in order to convolve f and w, for instance zero padding, mirroring or periodization. In these cases, the convolution can be written as a matrix-vector product: Given a signal $f = \big(f(k)\big)_{k=0}^{n-1}$, we have $u = w * f = Wf$ for some matrix W. Determine this matrix for a filter $w = \big(w(k)\big)_{k=-(n-1)}^{n-1}$ in case of zero padding.

Exercise 2.4.

This exercise deals with Python programming. Make sure that you have the Python packages Numpy, Scipy and Matplotlib installed. Using Numpy (imported as np), the convolution of two vectors f and w can be computed with the command `u=np.convolve(f,w)`. Further, the python script conv_signal.py provided on the website contains a method `convmtx(w,n)`, which generates the convolution matrix of w, where n is the length of f. Make yourself familiar with the two commands, in particular determine which continuation for f is applied.
In the following, we examine the convolution between a signal f and different filters w. Test the following filters and report their effect:

(i) first and second order difference filters:

$$w(k) = \begin{cases} -1 & k = 0, \\ 1 & k = 1, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{respective} \qquad w(k) = \begin{cases} 1 & k = 0, \\ -2 & k = 1, \\ 1 & k = 2, \\ 0 & \text{otherwise.} \end{cases}$$

(ii) sampled Gaussian: The weights are given by the Gaussian function

$$\phi_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \, \mathrm{e}^{-\frac{x^2}{2\sigma^2}},$$

uniformly sampled at n points in $[-3\sigma, 3\sigma]$ (and set to 0 otherwise), followed by a normalization such that $\sum_{k\in\mathbb{Z}} w(k) = 1$.

Apply the convolution to the signal f and its noisy version g, where f and g can be generated by the command `f,g=load_signals()` from the script conv_signal.py provided on the homepage. Concerning part (ii), try different values for σ and n, for instance $\sigma \in \{1, 3, 5, 10\}$ and $n \in \{20, 50, 100\}$, and report your results.