

---

## HPC for numerical methods and data analysis

*Fall Semester 2023*

*Prof. Laura Grigori*

*Assistant: Mariana Martinez*

**Session 9 – November 14, 2023**

---

### Randomized low rank approximation (pt 2)

The Radial Basis Function (RBF) applications can be found in neural networks, data visualization, surface reconstruction, etc. These techniques are based on collocation in a set of scattered nodes, the **computational cost of these techniques increase** with the number of points in the given dataset with the dimensionality of the data.

For RBF approximation we assume that we have an unordered dataset  $\{x_i\}_{i=1}^n$ , each point associated with a given  $f_i \in \mathbb{R}^p$ . We are going to consider  $f_i \in \mathbb{R}$  (meaning that each point in the dataset is associated with a label). The approximation scheme can be written as follows:

$$s(x) = \sum_{i=1}^n \lambda_i \phi(\|x - x_i\|),$$

where:

- $x_i$  are the data points
- $x$  is a free variable at which we wish to evaluate the approximation
- $\phi$  is the RBF
- $\lambda_i$  are the scalar parameters

The  $\lambda_i$ 's are chosen so that  $s$  approximates  $f$  in a desired way. One of the simplest ways of computing these parameters is by forcing the interpolation to be exact at  $x_i$  i.e.  $s(x_i) = f(x_i) = f_i$ . Define a matrix  $A \in \mathbb{R}^{n \times n}$  such that  $A_{ij} = \phi(\|x_i - x_j\|)$ , let  $\lambda = [\lambda_1, \dots, \lambda_n] \in \mathbb{R}^n$  and  $f = [f_1, \dots, f_n] \in \mathbb{R}^n$  (both column vectors). Then in order to compute the scalar parameters we need to solve the following linear system:

$$A\lambda = f. \tag{1}$$

Before computing  $A$ , answer the following questions:

- a) How does 1 scale in both the number of data points and the dimension of such points?

- b) What would it mean if  $A$  is nearly singular?
- c) What would be the effect on  $A$  if  $\phi$  has compact support? What would be the disadvantage of using such RBF?

The *MNIST data set* contains pictures of handwritten digits. It contains 60'000 training images and 10'000 testing images. You can download this database from here: <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>. You can also download the labels for the training and testing images (these are going to be our  $f_i$ 's. We are going to use the following RBF:

$$\phi(\|x_i - x_j\|) = e^{-\|x_i - x_j\|/c},$$

with  $c > 0$ .

- d) We are going to start by taking a relatively small sample of the training set (i.e.  $n$  being "small"). Download the data set (both the test and training sets). Then from the training set (and the labels) pick the  $n$  top rows.
- e) Write a Python scrip that computes  $A$  using the subsampled data set and optionally saves it to memory. In this section you are going to determine the value of  $c$  to use. You can test different values of  $c$  to solve 1. (Optional: write a parallel implementation of the function to build  $A$ )
- f) Explain Nyström approximation and why it would be useful in this setting.
- g) Given a sketch matrix  $\Omega$  and using your code from last week and for different values of  $l$  compute  $A_{\text{Nyst}} = (A\Omega)(\Omega^\top A\Omega)^\dagger(\Omega^\top A)$ .
- h) Test the accuracy of the previously computed Nyström approximation. Provide graphs that show the error of the approximation using the nuclear norm.
- i) (Optional) Try solving 1 using  $A_{\text{Nyst}}$