System Identification – Practical Assignment 4
Linear Regression for Function Approximaton

Logistics

- This practical assignment is a compulsory part of the course "System identification".
- The assignment is preferably carried out in groups of two, but can also be done alone if necessary. Note that groups of three or more students are strongly discouraged.
- The assignment solution consists of Matlab code. Develop this code in a single Matlab script. This code will be checked and run by the teacher in order to validate your attendance to the lab; the teacher will strive to do this as far as possible during the lab, together with you. Nevertheless, please write your code in a self-explanatory fashion (adding comments where necessary), so as to make it understandable on its own. At the end of the lab, please email the code as an m-file or ZIP file to the teacher of the lab (either Lucian Busoniu at `lucian@busoniu.net` or Zoltán Nagy at `zoltan.nagy@aut.utcluj.ro`), using the following filename template:
  `sysid_lab2_indexINDEX_STUDENTNAME1_STUDENTNAME2`
  where INDEX stands for your dataset index, see below; STUDENTNAME1 and 2 stand for the last names of the two students in the group. Please *include this file name also in the subject line of your email* (for automatic email filing purposes).
- Discussing ideas amongst the students is encouraged; however, directly sharing and borrowing pieces of code is forbidden, and any violation of this rule will lead to disqualification of the solution.

Assignment description

In this assignment we will perform function approximation with linear models, see *Linear Regression* in the course material *Part 3 – Mathematical Background*. We will work with the same datasets as for the first part of your project assignment.

A data set of input-output pairs is given, where the outputs are generated by an unknown function g. The function has two input variables and one output variable, and the output measurements are affected by noise. Your will develop an approximator of this function, using a linear model with radial, Gaussian basis functions. The parameters of the model will be found using the identification data set. A second data set is provided for validating the developed model. The two data sets are given in a MATLAB data file, containing one structure for each set. The training data set is named `id` and the validation data set `val`. Each of these structures contains the following fields:

- A set of grid coordinates `X` for the inputs, where `X` is a cell array of two vectors, each vector `X{1}` and `X{2}` containing N' grid points for input dimension 1 and 2, respectively.
- A set of corresponding outputs `Y`, a matrix of size $N' \times N'$, where `Y(i,j)` is equal to the value of g at point `(X{1}(i),X{2}(j))`.
- The same data is provided in a different, 'flattened' format: the inputs `Xflat`, a wide matrix containing the N'^2 input points, one on each column, and the corresponding row vector of outputs `Yflat`.

The ‘structured’ variants X and Y are easier to use for creating graphs, while the ‘flat’ variants are more suitable to use in fitting and validation.

Each student group is assigned an index number. Then, the group downloads the Matlab programs and data file that form the basis of the assignment from the course webpage:

`http://busoniu.net/teaching/sysid2015`

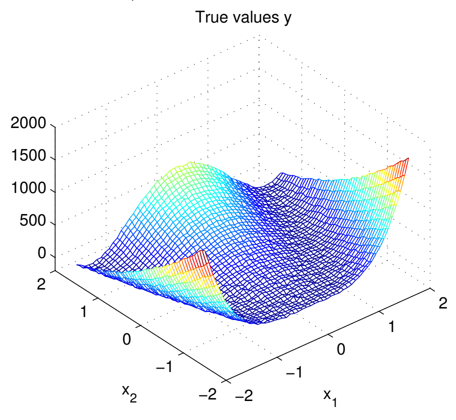
Since the data sets are the same as for the first part of the project, preference is given to using the same index (and therefore datafile) as for your project.

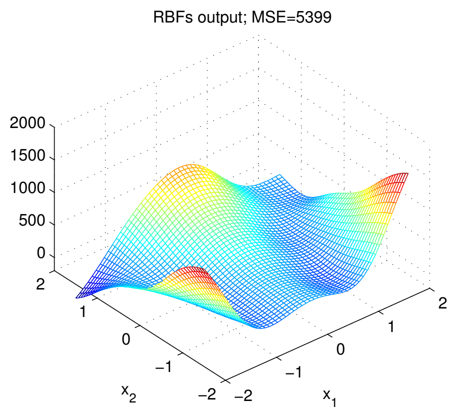
In addition to the data file, the website provides a function `rbfapprox` that implements an RBF approximator, and an example script `lab4_template` to get you started with the assignment. The template script exemplifies how the RBF approximator can be created and used. Summarizing, `rbfapprox` is called to create a grid of equidistant RBFs, taking as arguments the limits of the function input domain and the number of RBFs on each dimension of the domain. It returns a structure containing two functions, `phi`, which computes the regressor vector for any x; and `eval`, which evaluates the output $\hat{y}$ of the approximator at any x, given the parameter vector θ.

Requirements:

- Plot the training (identification) data to get an idea of the function shape (use `mesh`).
- Create a system of linear equations for linear regression, using the identification data. Use the matrix representation explained in the lecture. Solve this system using matrix left division, operator $\backslash$ in Matlab (or alternatively with `linsolve`). Report the MSE (or the objective function $V(\hat{\theta})$) on the identification data.
- Validate the model on the different, validation data set: compute the approximated outputs and from those the MSE on the validation data (or the squared error $\frac{1}{2}\sum_k \varepsilon(k)^2$, which corresponds to the objective function $V(\hat{\theta})$ but applied to the validation data). Show a plot of the approximated function on the validation data set, using `mesh`.
- Try a few values for the number of basis functions on each dimension and see how the approximation quality on the training and validation set evolves.

Your plots will look similar to those exemplified in the next figure (except your data and fit quality will be different, of course).


True values y
x2
x1


RBFs output; MSE=5399
x2
x1