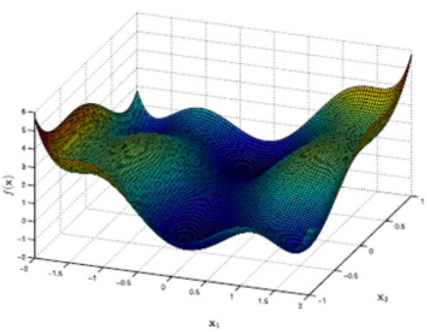
Prof. Inneke Van Nieuwenhuyse
Advanced Topics in Data Science

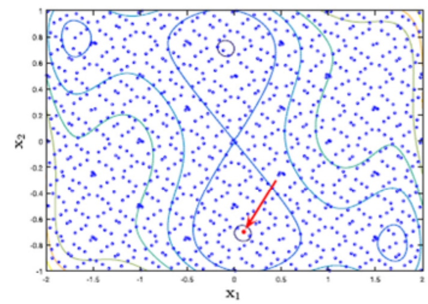
Assignment: GPR and Bayesian Optimization

Suppose that you need to minimize the six-humped camelback function:

Function	Description
Six-hump Camelback	$f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4$ $-2 \leqslant x_1 \leqslant 2$ and $-1 \leqslant x_2 \leqslant 1$ $x^* = (0.0977, -0.6973)$ $f(\mathbf{x}^*) = -1.0294, R_f \simeq 7.3$



(a) Camelback function



(b) Candidate points and contour plot of the Camelback function

You can only observe the function outputs through (deterministic) simulation. **Evidently, we are going to treat the function as a black box.**

First, generate an initial latin hypercube sample containing 20 two-dimensional input vectors (X1,X2). Next, generate a "candidate point" set containing a latin hypercube sample of 100 candidate points (X1,X2). To ensure that we all obtain similar initial designs and candidate sets, please set the seed of the (default) random number generator in Matlab using this command:

rng(4)

BEFORE giving the commands to generate these points.

Write code that allows you to search for the minimum of the function, using the EI infill criterion proposed in Jones et al. (1998), and *considering only the candidate points as potential infill points*. For fitting the Gaussian Process, you may assume a constant mean function and a squared exponential kernel (@corrgauss in DACE). Allow for 10 infill points (so: run 10 extra iterations on top of the initial design). *Hint:* write separate function files to calculate the EI and to evaluate true function values in arbitrary (X1,X2) combinations, such that you can "call" these functions upon request.

Keep track of the (X1,X2) location of the infill point chosen in each iteration, and the corresponding value of max(EI) at each iteration.

a) Plot the X locations of the initial design points (markers = green circles) and the candidate points (markers = red diamonds) in a 2D plot (*hint*: use the command "plot").

b) Add the X locations of the 10 infill points selected by the algorithm to the same 2D plot (markers = black circles).

c) What is the result of your optimization (optimal X location and optimal goal function value)? Compare this with the true optimum of the function over *all* points considered in your optimization setup (initial and candidate points). Do they match? Add the X locations of both points in the 2D plot that you obtained in (a) (indicate the true minimum by a blue square, and the minimum found by the algorithm by a blue star).

d) Plot the max(EI) values observed in terms of the iteration number. Can you explain why these values do not go down monotonically as the iterations proceed?

e) Plot the EI values for **all** points (20 initial design points + 100 candidate points), using a Gaussian Process fitted to **all** data available at the end of the 10^{th} iteration (i.e., 20 initial design points and 10 infill points), in a 3D scatter plot (*hint*: use the command "scatter3"). What do you learn/conclude: would it be useful to increase the simulation budget and continue the Bayesian optimization procedure?