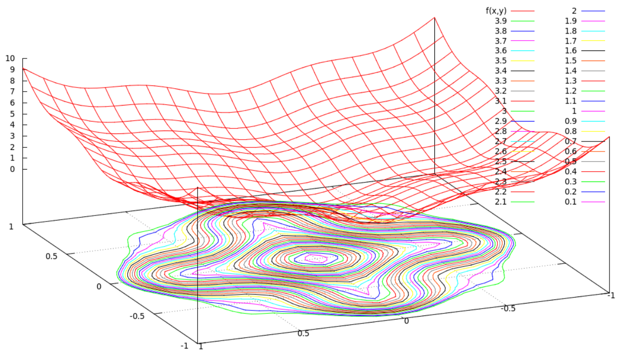
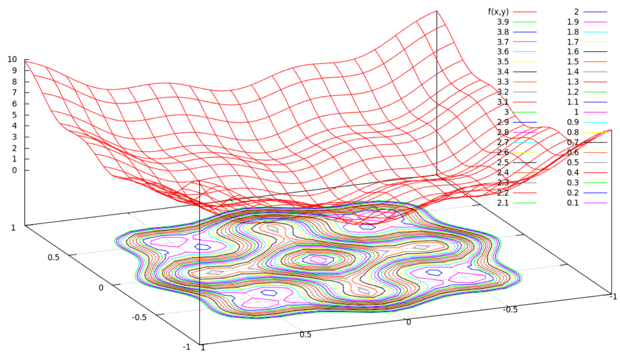
Maths for Intelligent Systems
Exercise 11

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

January 28, 2019

1 Restarts of Local Optima





The following function is essentially the Rastrigin function, but written slightly differently. It can be tuned to become uni-modal and is a sum-of-squares problem. For $x \in \mathbb{R}^2$ we define

$$f(x) = \phi(x)^\top \phi(x) \ , \quad \phi(x) = \begin{pmatrix} \sin(ax_1) \\ \sin(acx_2) \\ 2x_1 \\ 2cx_2 \end{pmatrix}$$

The function is plotted above for $a = 4$ (left) and $a = 5$ (right, having local minima), and conditioning $c = 1$. The function is non-convex.

a) Choose $a = 6$ or larger and implement a random restart method: Repeat initializing $x \sim \mathcal{U}([-2, 2]^2)$ uniformlly, followed by a gradient descent (with backtracking line search and monotone convergence).

Restart the method at least 100 times. Count how often the method converges to which local optimum.

2 UCB Bayesian Optimization

Find an implementation of Gaussian Processes for your language of choice (e.g. python: scikit-learn, or Sheffield/Gpy; octave/matlab: gpml). and implement UCB. Test your implementation with different hyperparameters (Find the best combination of kernel and its parameters in the GP) on the 2D function defined above.

3 No Free Lunch Theorems

Broadly speaking, the No Free Lunch Theorems state that all algorithms perform "in average" exactly the same, if no restrictions or assumptions are made w.r.t. the problem. Algorithms outperform each other only w.r.t. specific classes of problems.

a) Read the publication "No Free Lunch Theorems for Optimization" by Wolpert and Macready and get a better feel for what the statements are about.

b) You are given an optimization problem where the search space is a discrete set X of integers $\{1, \dots, 100\}$, and the cost space Y is the set of integers $\{1, \dots, 100\}$. Come up with three different algorithms, and three different assumptions about the problem-space such that each algorithm outperforms the others in one of the assumptions.