Artificial Intelligence (CO3519): Tutorial – Calendar Week 42

Optimization

M. T. Horsch, School of Psychology and Computer Science

1.2.1. 2D visualization with colour codes

The `multivar` Jupyter Notebook example[1] defines a function with two parameters,

$$f(x,z) = x^4 + z^4 + 2x^2z^2 - 10x^3 - 3z^3 - 5x^2z - 17xz^2 + 32x^2 + 36z^2 + 34xz - 39x - 63z + 68. \quad (1)$$

That function is called `objective()` in the notebook. It is shown how `objective()` can be evaluated over a grid of x, z value pairs by using a *two-dimensional array* from the numpy library as a data structure. The 2D array is generated using `np.meshgrid()`:

```
import numpy as np

# create a two-dimensional array
#
x_param_values = np.arange(xlow, xhigh, step)
z_param_values = np.arange(zhigh, zlow, -step)
gridx, gridz = np.meshgrid(x_param_values, z_param_values)

# evaluate the objective function for all x, z combinations
#
objective_values = objective(gridx, gridz)
```

Choosing a negative step size for z, the second parameter, helps obtain an output, *e.g.*, from `print(objective_values)`, that arranges the values in the same way as we would expect them in a diagram. Two different ways of visualizing this function in a colour-coded diagram are shown, once using `matplotlib` directly, once using `seaborn` on top of `matplotlib`, yielding the output from Fig. 1.

a) Reproduce the visualization from Fig. 1 and see whether you can make the diagrams more beautiful or intelligible based on the documentation.[2]

[1] See `https://home.bawue.de/~horsch/teaching/co3519/material/multivar.ipynb`.

[2] See `https://matplotlib.org/stable/contents.html` for `matplotlib` and `https://seaborn.pydata.org/` for `seaborn`, whichever you prefer. Any other tools or libraries are also welcome.

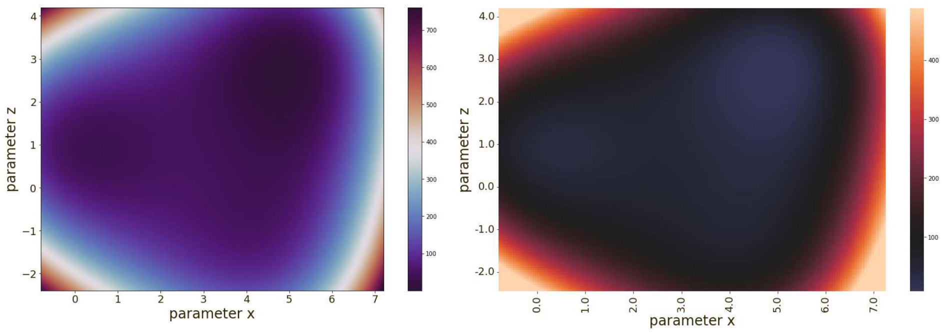

Figure 1. Colour-coded diagrams showing the values of $f(x,z)$; left, using `matplotlib.pyplot.imshow()`, right, using `seaborn.heatmap()`.

b) Apply your preferred technique to visualize the function

$$g(x,z) \;=\; (x-3)^2+(z-2)^2. \tag{2}$$

1.2.2. Local minimization over two parameters

In the `multivar` notebook, the `scipy` library is used to minimize $f(x,z)$ using the Nelder-Mead simplex method[3] as follows:

```
import scipy.optimize as opt
opt.minimize(objlistf, initial_point, \
             method='nelder-mead', options={'xatol': 0.0001})
```

There, `opt.minimize()` requires a version of the cost function where the two arguments are passed *as a list*; in the example notebook, that is `objlistf()`, a wrapper around `objective()`. As usual in local optimization, it also requires an initial value (in list form), *e.g.*, `[3, 3]`; since $f(x,z)$ has two local minima, the outcome depends on the initial value – starting from `[1, 1]`, the other local minimum will be found. The `'xatol'` option passes the termination threshold, *i.e.*, the variable called ε in the lecture.

Apply the same method to minimize $g(x,z)$ from Eq. (2); note that for this to work, the function needs to accept the two parameters `[x, z]` *as a list*, just like `objlistf()`.

1.2.3. Cost functions with more than two parameters

The `multivar` notebook defines a cost function for a hypothetical industrial operation; at planning and design stage, you have direct control over the following parameters:

- The investment i, done a single time, in units of £.
- The amount of goods p to be produced, in units of £/year.
- The depreciation period d (how long it is meant to operate), in units of years.

There is a single minimization objective, *i.e.*, a cost function, expressing:

- The deficit of the industrial operation, in units of £/year.

[3] This method is chosen here because it is very simple to use; *cf.* `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html` for the many options that `scipy` provides.

The cost function is given by `yearly_deficit()`, and a wrapper for accepting list input, as required by `opt.minimize()`, is given as `yearly_deficit_list_wrapper()`.

a) Minimize the deficit using `opt.minimize()` as above; the outcome should be a profit, *i.e.*, a negative deficit.
b) How would you ***visualize the optimization problem***, the cost function, or the optimal solution if you were asked to present the optimization outcome to decision makers? ***Creative solutions are welcome.***

Submission deadline: 6th November 2021; discussion planned for 18th November 2021. Group work by up to four people is welcome.