UNIVERSITE DE GENEVE Section de Mathématiques
Faculté des sciences

Optimization with Applications I
TP 4

We will implement a general cross validation scheme and test it on some data. Cross validation allows us not to do an explicit train-test split and instead make that splitting part of the training procedure.

To that end download the Postdata.txt. Here, again we want to find a linear model matching the "y" variable to the others.

Exercise 1. Implement a k-fold cross-validation function that receives as input a general learner l and a data set X, Y, the amount k of observations to split your dataset into, an error function err and returns the achieved errors.

Thus we are looking for a function

```
def cross_val(learner, X, Y, k):
    # learner is a function that finds an optimum given some data
    # We will shuffle and then split the data set into k equal parts and
    # perform learning on them. Return all the errors.
    return [errors]
```

Exercise 2. We assume that the test error function for ridge regression

$$R : \lambda \mapsto \frac{1}{N_{\text{test}}} ||\underline{y_{\text{test}}} - X_{\text{test}} \hat{\alpha}_\lambda||_2^2$$

is convex and that we know of an upper bound K for the optimal λ. Here $\hat{\alpha}_\lambda$ is the α solving the ridge regression problem with parameter λ. Apply an optimization method here in conjunction with your result from Exercise 1 to find the optimal λ. Use whatever method you want but make sure to keep the number of evaluations of the cross validation low to minimize the runtime.

Exercise 3. Test your function on the postdata.txt and plot relevant results. That includes a plot for the errors produced by different lambdas.

Important : Every function and exercise must be tested. Plug in some values for which you know the correct answers and compare the output of your function.