Problem 7.1 (Normal Equations) **(20 P.)**

Given a dataset that was generated by a function $f : \mathbb{R} \to \mathbb{R}$ which is known to be linear, i.e. $f(x) = w_0 + w_1 x$ with unknown $w = (w_0, w_1)$. The following data was sampled from f and distorted by normally distributed, zero-mean noise: $x = (1, 2, 4, 5, 8)$ and $y = (3.5, 2.5, 5.5, 9.5, 14.5)$. Determine an estimate $\hat{w}$ of w using the normal equations (manually, i.e. "per Hand"). Give the following intermediate results:

a) X and y (2 P.)

b) $X^t X$ (2 P.)

c) $(X^t X)^{-1}$ (4 P.)

d) $(X^t X)^{-1} X^t$ (2 P.)

e) $\hat{w} = (X^t X)^{-1} X^t y$ (the values of $\hat{w}$ are not necessarily nice (provide **simple** fractions)) (5 P.)

f) Plot informatively the sampled data and your estimated hypothesis $h(x) = \hat{w_0} + \hat{w_1} x$. (5 P.)

Note: The inverse of a 2×2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is given by $A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$.

Problem 7.2 (Model Selection using Cross-Validation) **(40 P.)**

In this exercise, the task is to automatically learn a hypothesis h that fits some given data well. The data stems from a function $f : \mathbb{R} \to \mathbb{R}$ which is a polynomial with a degree not larger than 10 and was corrupted by normally distributed, zero-mean noise. Before you can fit the hypothesis, you have to determine an appropriate degree for your hypothesis class using a 10-fold cross-validation.

For this task download the file *regression.py* from StudIP and complete the function stubs.
The dataset "`regression_dataset_2.txt`" is also found on StudIP and is two-dimensional (containing 30 points). The first column corresponds to x and the second to y (i.e. the goal is to learn a hypothesis h such that $h(x)$ is "as close as possible" to y). The dataset can be loaded in Python with:

```
x, y = np.loadtxt('regression_dataset_2.txt')
```

a) Complete the function "normal_equations(x, y, d)" that computes $\hat{w}$ for the data $x = \{x_1, \dots, x_n\}$ and $y = \{y_1, \dots, y_n\}$ with $x_i, y_i \in \mathbb{R}$ and d (degree of the polynomial) (i.e. $\phi(x_i) = (x_i^0, \dots, x_i^d)$). (10 P.)

b) Write a program that does the following: (30 P.)

   1) Perform a standard cross-validation with $num\_folds = 10$.
   2) **For each** degree $d \in \{0, 1, \dots, 10\}$ learn a hypothesis using your function "normal_equations" **for each** of the folds training sets.
   3) Complete the regressor that computes the SSE of these hypotheses on the corresponding validation sets.
   4) For each degree, compute the **average SSE** over the num_folds folds.
   5) Plot the averaged SSEs for all degrees and select the degree that minimizes the average SSE.
   6) Train a final hypothesis with the chosen degree on the **whole available data** and make a plot informative plot showing the data and your hypothesis.
   7) Explain why we can use linear regression to fit non-linear data.

On the hand-in date, **21.12.2016**, you must hand-in the following: [1]

a) a text file stating how much time you (all together) used to complete this exercise sheet

b) your solutions / answers / code

for problem 7.1 and 7.2.

[1] upload via StudIP (if there are problems with the upload contact me **beforehand**: krell@uni-bremen.de)