Homework 2

2-AIN-150, Winter 2025

Deadline: 27.10.2025, 23:59, via Classroom

Before you start solving the homework, please read the general instruction at the end of the document. Submitted solutions should be your own. Do not copy and do not tried to find solution in literature or over the internet.

Selection of relevant attributes

Read carefully what you should submit.

Consider linear regression, where we have m attributes, n training samples, but output only depends on only few input attributes.

Your task is to find for given data (i.e. set of inputs $\vec{x}^{(i)}$ and expected outputs $y^{(i)}$), on which attributes y really depends (or data shows us so).

You do not need to program general method. It is enough, if you use machine learning methods and halfmanually determine the output. You should submit following things:

- the result (i.e. list of relevant attributes)
- short commentary about your methods and source code you used to determine the result

There are two input files in the package. You should determine the results from `input.txt`. The `sample.txt` file is there for you to check your methods. You should find that first 10 attributes are relevant. But you can also ignore it.

You can use any libraries, which do basic math, matrix operations, matrix inversions, linear system solving, calculate derivatives. You can also use libraries for linear regression (and its regularized variations) and cross-validation (e.g. relevant scikit-learn parts).

Do not use library functions, which do relevant attribute selection (i.e. they solve your homework in one line).

Tips

- L1 regularization pushes weight to zero for attributes, which seem to be notrelevant. But you need to set the α parameter to right value.
- Cross validation can help you to compare quality of your selected attributes.

Python instructions There is `template.py` in package, where you can fill in the `select_features(X, y)` function. There is data loading and demos on using $L1$ regression and cross validation.

You can run the program with `python template.py <input file>`. You need to have scikit-learn installed.

Other languages instructions Write similar function as in Python. Input file contains numbers n, m on the first line. There are n more lines with training samples. Each line (training sample) has following format: First there is m numbers (the attributes from vector x) and then one number – expected output y..