Task 1 (70 points, programming)

In this task you will implement linear regression, as described in Sections 3.1.1 and 3.1.4 of the PRLM textbook.

Arguments

You must implement a Matlab function or a Python executable file called `linear_regression` that uses linear regression to fit a polynomial function to the data. Your function should be invoked as follows:

```
linear_regression(<training_file>, <degree>, <lambda>, <test_file>)
```

If you use Python, just convert the Matlab function arguments shown above to command-line arguments. The arguments provide to the function the following information:

- The first argument, <training_file>, is the path name of the training file, where the training data is stored. The path name can specify any file stored on the local computer.
- The second argument, <degree> is an integer between 1 and 10. We will not test your code with any other values. The degree specifies what function φ you should use. Suppose that you have an input vector $x = (x_1, x_2, ..., x_D)^{T,}$.
  - If the degree is 1, then $φ(x) = (1, x_1, x_2, ..., x_D)^T$.
  - If the degree is 2, then $φ(x) = (1, x_1, (x_1)^2, x_2, (x_2)^2..., x_D, (x_D)^2)^T$.
  - If the degree is 3, then $φ(x) = (1, x_1, (x_1)^2, (x_1)^3, x_2, (x_2)^2, (x_2)^3, ..., x_D, (x_D)^2, (x_D)^3)^T$.
- The third argument, <λ>, is a non-negative real number (it can be zero or greater than zero). This is the value of λ that you should use for regularization. If $λ = 0$, then no regularization is used.
- The fourth argument, <test_file>, is the path name of the test file, where the test data is stored. The path name can specify any file stored on the local computer.

The training and test files will follow the same format as the text files in the UCI datasets directory. A description of the datasets and the file format can be found on this link. For each dataset, a training file and a test file are provided. The name of each file indicates what dataset the file belongs to, and whether the file contains training or test data. Your code should also work with ANY OTHER training and test files using the same format as the files in the UCI datasets directory.

As the description states, **do NOT use data from the last column (i.e., the class labels) as features**. In these files, all columns except for the last one contain example inputs. The last column contains the target output.

Training Stage

Remember to use the pseudo-inverse function in Python (numpy.linalg.pinv) or in Matlab (pinv) when you need to invert matrices.

At the end of the training stage, your program should print out the values of the weights that you have estimated. The output of the training phase should be a sequence of lines like this:

```
w0=%.4f
w1=%.4f
w2=%.4f
...
```

Test Stage

After the training stage, you should apply the function that you have learned on the test data. For each test object (following the order in which each test object appears in the test file), you should print a line containing the following info:

- object ID. This is the line number where that object occurs in the test file. Start with 1 in numbering the objects, not with 0.
- output of the function that you have learned for that object.
- target value (the last column on the line where the object occurs).
- squared error. This is simply the squared difference between the output that your function produces for the test object and the target output for that object.

The output of the test stage should be a sequence of lines like this:

```
ID=%5d, output=%14.4f, target value = %10.4f, squared error = %.4f
```

Object IDs should be numbered starting from 1, not 0. Lines should appear sorted in increasing order of object ID.

In your answers.pdf document, provide the full output of the training stage, and ONLY THE LAST LINE (the line printing the result on the last test object) of the output by the test stage of your program, when given pendigits_training.txt as the training file, and pendigits_test.txt as the test file. Provide this output for all four combinations where degree=1 or degree=2 (where degree is the second argument) and λ=0 or λ=1.

Task 2 (15 points, written)

We are given these training examples for a linear regression problem:

$x_1 = 5.3$, $t_1 = 9.6$
$x_2 = 7.1$, $t_2 = 4.2$
$x_3 = 6.4$, $t_3 = 2.2$

We just want to fit a line to this data, and we to find the 2-dimensional vector **w** that minimizes $\tilde{E}_D(\mathbf{w})$ as defined in slide 56 of the linear regression slides. What is the value of **w** in the limit where λ approaches positive infinity? Justify your answer. Correct answers with insufficient justification will not receive credit.

Task 3 (15 points, written)

We are given these training examples for a linear regression problem:
$x_1 = 5.3$, $t_1 = 9.6$
$x_2 = 7.1$, $t_2 = 4.2$
$x_3 = 6.4$, $t_3 = 2.2$

We are also given these two lines as possible solutions:

- `f(x) = 3.1x + 4.2`
- `f(x) = 2.4x - 1.5`

Which of these lines is a better solution according to the sum-of-squares criterion? This criterion is defined as function $E_D(\mathbf{w})$ in slide 25 of the linear regression slides. Justify your answer. Correct answers with insufficient justification will not receive credit.