**IE 7374 Machine Learning in Engineering**

**Team 8 Lab 1 Report**

**Team Member: Yaodong Chen, Yuxiao Ma, Lin Zhu**

**Gradient Descent for Linear Regression**

**Dataset description**

* Housing: This is a regression dataset where the task is to predict the value of houses

in the suburbs of Boston based on thirteen features that describe different aspects that

are relevant to determining the value of a house, such as the number of rooms, levels

of pollution in the area, etc. There are 14 columns total, the first 13 columns are input variables, and last column is the output: the value of houses.

* Yacht: This is a regression dataset where the task is to predict the resistance of a

sailing yacht's structure based on six different features that describe structural and

buoyancy properties. There are 7 columns total, first 6 columns are input variables, last column is the output: the resistance.

* Concrete: This is a regression dataset where the task is to predict the compressive

strength of concrete on eight different features. There are a total of 1030 instances and

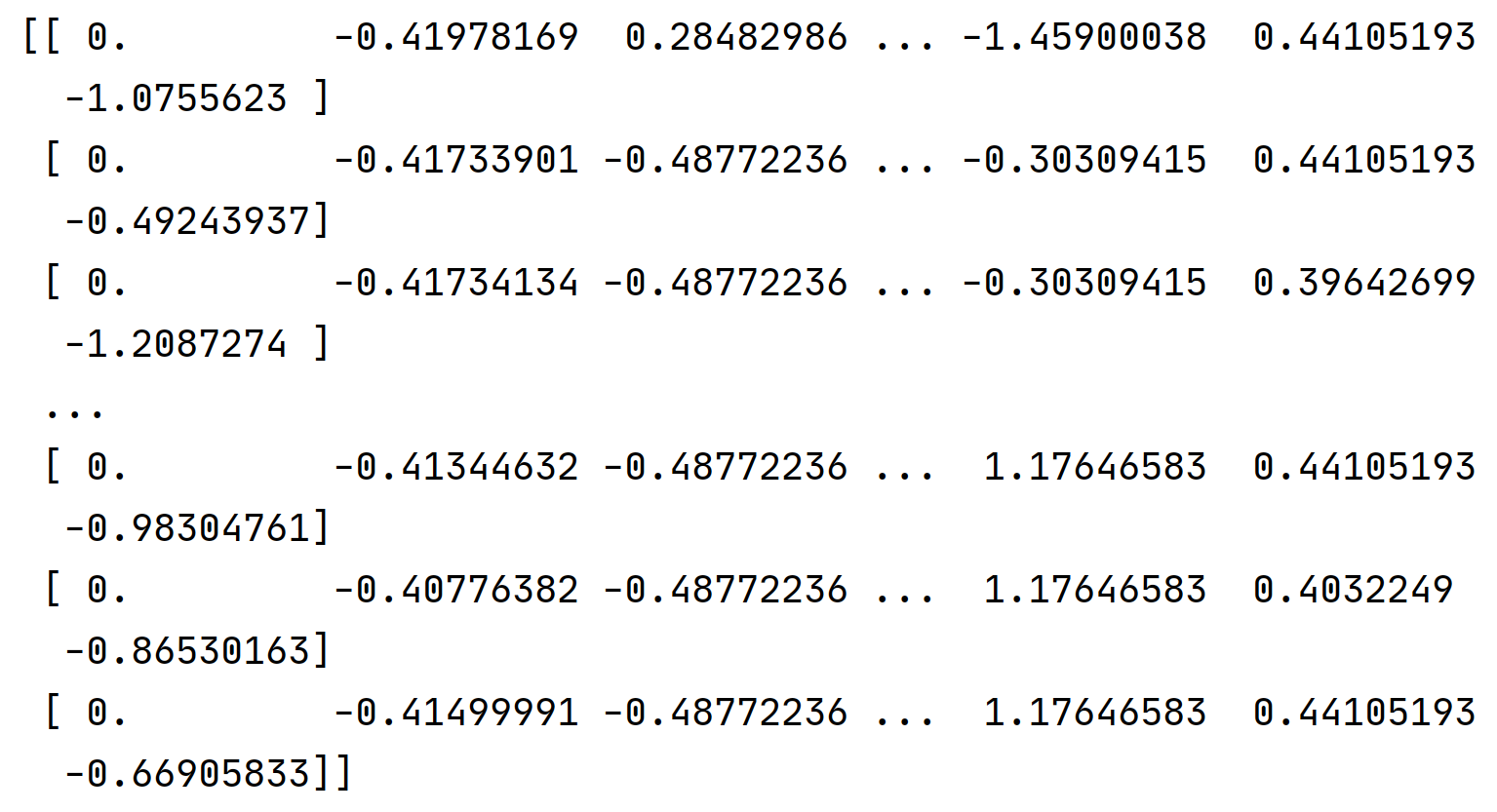
all the features are numeric. There are 9 columns total. The first 8 are input variables, the last column is the output: the strength.

**Major steps**

* Normalize the features in the training data using z-score normalization.
* Initialize the weights for the gradient descent algorithm to all zeros.
* Calculate the risk function and its gradient with respect to the parameters.
* Update the estimates of regression coefficients using the following set of parameters:
  + Housing: learning rate=, tolerance=
  + Yacht: learning rate=, tolerance=
  + Concrete: learning rate=, tolerance=
* The tolerance is defined based on the RMSE measured on the training set between iterations.
* We tested with other learning rate and tolerance to our three datasets.
* We set the maximum number of iterations to recommended level of 50000.

**Result For housing dataset**

1. **housing dataset training set normalization**



1. **Housing training parameters and estimate regression weight**

* When *iteration=50000，alpha = 0.0004，tolerance = 0.005*, we have

[ 0.9996 -0.91577908 1.06002207 0.07618625 0.6909578 -2.041962

2.68671408 0.00903157 -3.1060596 2.49853143 -1.89216495 -2.05385865

0.84831267 -3.73726057]

* Normal Equation

[ 3.64593681e+01 -1.08010875e-01 4.64207413e-02 2.05600424e-02

2.68673440e+00 -1.77665128e+01 3.80986285e+00 6.92502811e-04

-1.47555701e+00 3.06050777e-01 -1.23347228e-02 -9.52744421e-01

9.31168870e-03 -5.24759100e-01]

* RMSE

23.013613998951612

* When *iteration=50000，alpha = 0.05，tolerance = 0.1*, we have:

[ 0.95 -0.92814163 1.08157522 0.1409097 0.68173987 -2.05670687

2.67422851 0.01947389 -3.10402925 2.66222893 -2.07680339 -2.06060058

0.84926891 -3.74363228]

* Normal Equation

[ 3.64593681e+01 -1.08010875e-01 4.64207413e-02 2.05600424e-02

2.68673440e+00 -1.77665128e+01 3.80986285e+00 6.92502811e-04

-1.47555701e+00 3.06050777e-01 -1.23347228e-02 -9.52744421e-01

9.31168870e-03 -5.24759100e-01]

* RMSE

23.01352239064961

* When *iteration=50000，alpha = 0.01，tolerance = 0.1*, we have:

[ 0.99 -0.92814163 1.08157522 0.1409097 0.68173987 -2.05670687

2.67422851 0.01947389 -3.10402925 2.66222893 -2.07680339 -2.06060058

0.84926891 -3.74363228]

* Normal Equation

[ 3.64593681e+01 -1.08010875e-01 4.64207413e-02 2.05600424e-02

2.68673440e+00 -1.77665128e+01 3.80986285e+00 6.92502811e-04

-1.47555701e+00 3.06050777e-01 -1.23347228e-02 -9.52744421e-01

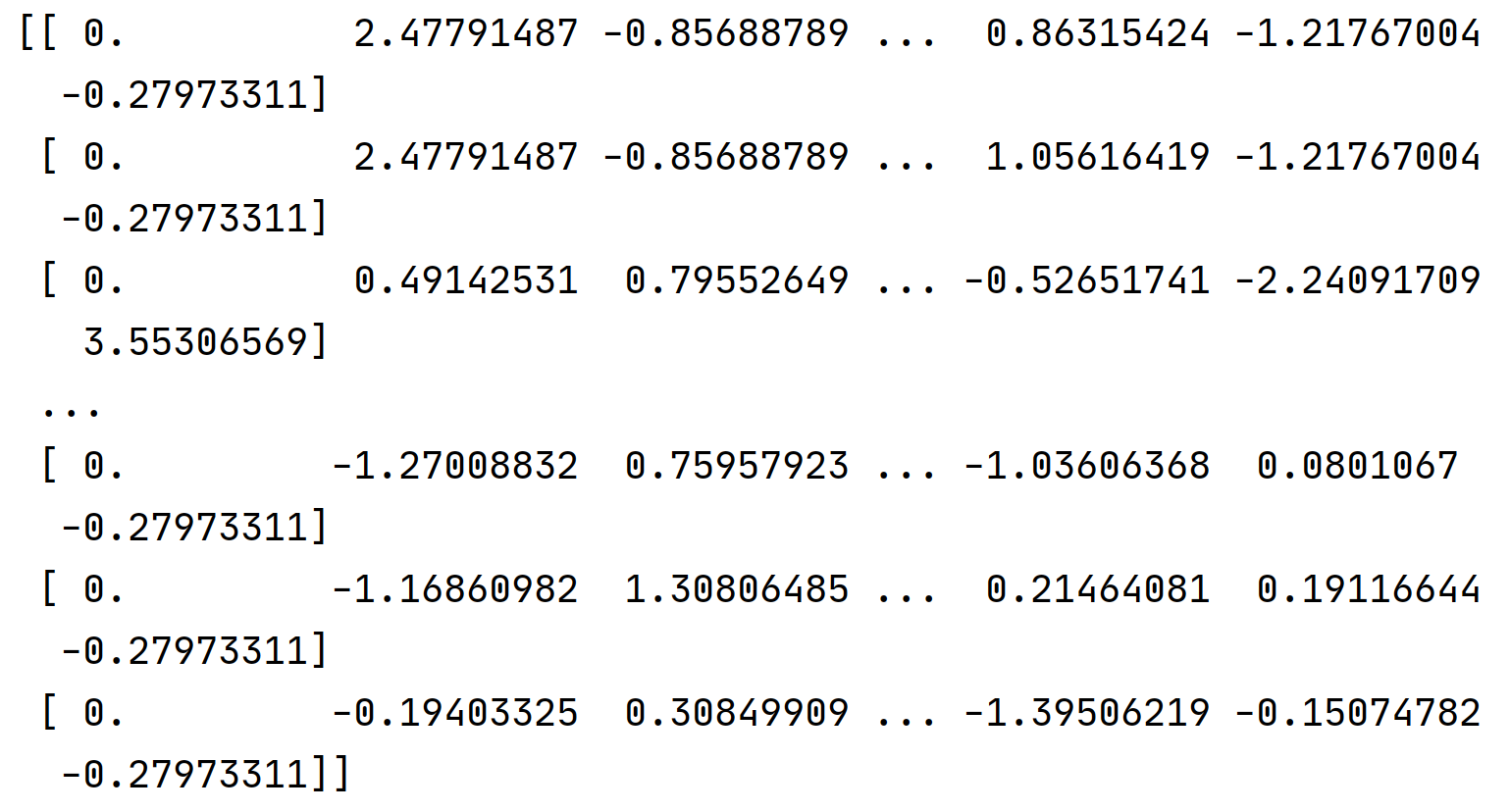
9.31168870e-03 -5.24759100e-01]

* RMSE

23.013522390649612

**Result For concrete dataset**

1. **concrete dataset training set normalization**



1. **concrete training parameters and estimate regression weight**

* When *iteration=50000，alpha = 0.0007，tolerance = 0.0001*, we have

[ 0.9993 12.03933543 8.49156576 5.21831346 -3.61360127 1.68978748

1.03378048 1.15671672 7.19818247]

* When *iteration=50000，alpha = 0.05，tolerance = 0.01*, we have

[ 0.95 12.51423615 8.95712234 5.62480067 -3.19983624 1.74485575

1.40559189 1.61799518 7.21189448]

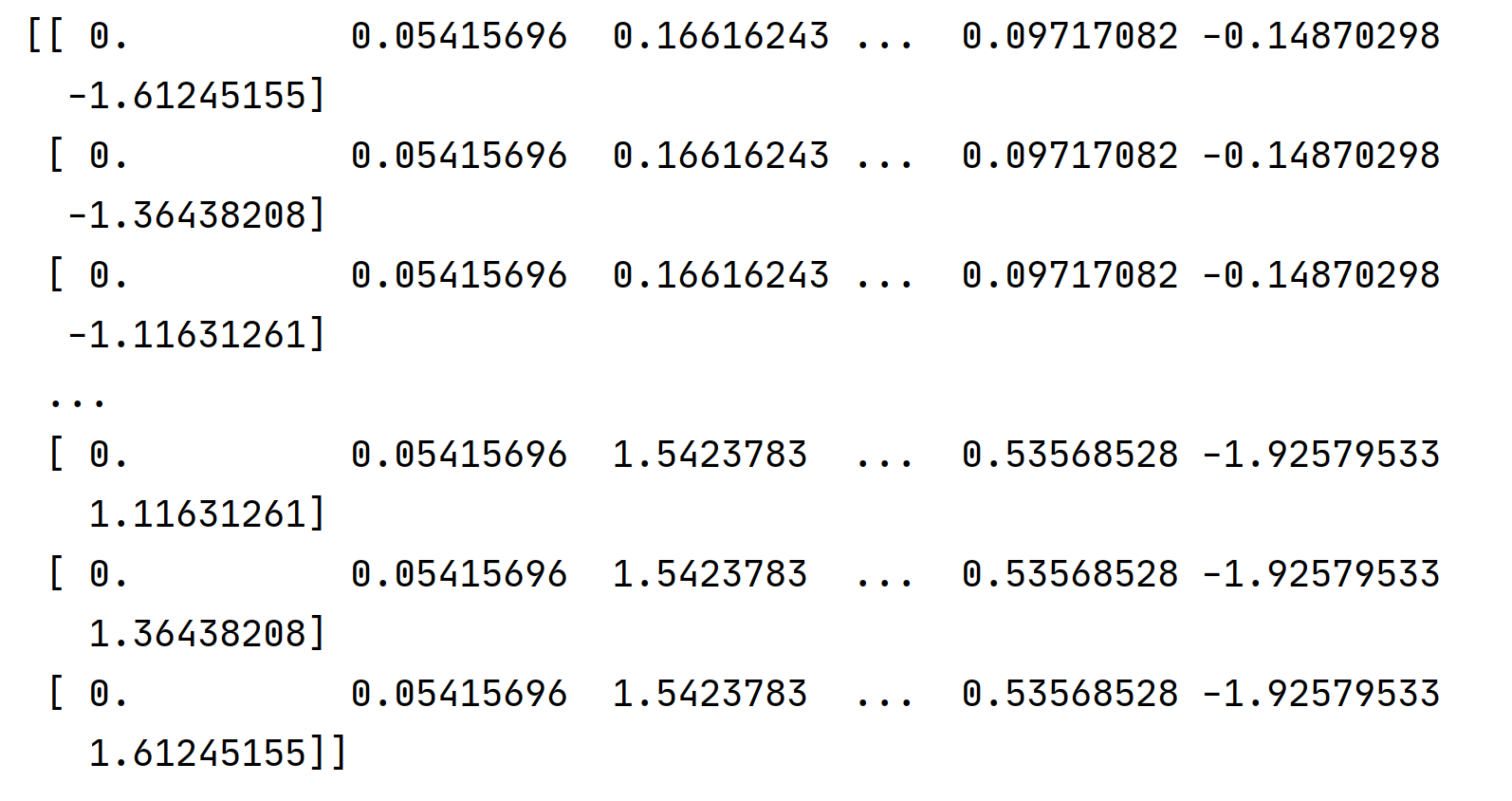
* When *iteration=50000，alpha = 0.1，tolerance = 0.01*, we have:

[ 0.9 12.51423615 8.95712234 5.62480067 -3.19983624 1.74485575

1.40559189 1.61799518 7.21189448]

**Result For yacht dataset**

1. **yacht dataset training set normalization**



1. **yacht training parameters and estimate regression weight**

* When *iteration=50000，alpha = 0.001，tolerance = 0.001*, we have:

[ 0.999 0.29045296 -0.29840626 0.46935347 -0.46177636 -0.5290195

12.26144161]

* + Normal Equation

[-19.23666079 0.19384434 -6.41937593 4.23299863 -1.76569481

-4.51643177 121.66757243]

* + RMSE value

13.733537822143006

* When *iteration=50000，alpha = 0.01，tolerance = 0.05*, we have

[ 0.99 0.29284953 -0.1494382 1.0687505 -0.96578092 -1.11756135

12.26144161]

* + Normal Equation

[-19.23666079 0.19384434 -6.41937593 4.23299863 -1.76569481

-4.51643177 121.66757243]

* + RMSE

13.73326872907699

* When *iteration=50000，alpha = 0.1，tolerance = 0.05*

[ 0.9 0.29285233 -0.14926439 1.06944982 -0.96636894 -1.11824801

12.26144161]

* + Normal Equation

[-19.23666079 0.19384434 -6.41937593 4.23299863 -1.76569481

-4.51643177 121.66757243]

* + RMSE

13.733268728711552

**Conclusion**

The performance of the gradient descent linear regression is better than the normal equation. By continued iteration, the gradient descent method keeps updating the weight and return the optimal regression weight and give us a more precise predicted value. The RMSE value of housing dataset is about 23, and the RMSE value of yacht is about 13.7. By looking at the relation between these values and their original dataset dimension, we can estimate that the RMSE value of the housing dataset is a little bit high. Well, the RMSE value of the yacht dataset is a little bit better.