Assignment 1

Machine Learning COMS 4771

Spring 2017, Itsik Pe'er

Assigned: Jan 30th Due: Class time, Feb 6th

Submission: Courseworks

1. Edit the class `LinearRegressionSimulator`[1] to add a method `SimPoly(XInput)` where `XInput` is a single-column `pandas` vector of input datapoints, that returns a `numpy` array whose i-th entry is drawn from a normally distributed random variable with mean which the polynomial `Theta[0]`+$\Sigma_{d=1\ldots D}$`Theta[d]`× `XInput[i]^d` and standard deviation `StdDev`. Please submit `LinearRegressionSimulator.py` in a subfolder called `Assignment01_Problem01` of your zipped CourseWorks submission.
   [10 points]

2. A. Define a cubic polynomial with based on the digits in your UNI (mine would be $2x^3+x^2+6x+9$ as my uni is ip2169). Use `SimPoly` to simulate outputs with this polynomial and $\sigma = 0.1$. Simulate outputs for N training inputs and M testing inputs that are uniformly distributed in [0,1]. Perform polynomial curve fitting of degrees 0 to 10 by explicitly performing the matrix multiplications and (pseudo)inverse operations for the relevant matrices. To be clear, you may use standard library functions for computing pseudoinverses, matrix multiplications and such, but cannot use any functions that are specifically purposed to directly solve regression problems. Compare empirical risks on training and testing data by plotting them along the degree axis. Do all this three times: run #1 with N=10, M=10; run #2 with N=100, M=10; run #3 with N=10, M=100.
   Your code should save the plot files `RiskPlot.[Run].pdf` as well as files with the following information (as columns of numbers):
   `x.train.[Run].txt` - for `Run`=1,2,3: 3 training inputs for the corresponding run
   `x.test.[Run].txt` - for `Run`=1,2,3: 3 testing inputs for the corresponding run
   `y.train.[Run].txt` - for `Run`=1,2,3: 3 training outputs for the corresponding run
   `y.test.[Run].txt` - for `Run`=1,2,3: 3 testing outputs for the corresponding run
   `ThetaStar.[Run].[D].txt` - for `Run`=1,2,3, and `D`=0,…,10 : 3x11 files, each with the fit coefficients for the corresponding run and corresponding degree polynomial.
   `Risk.train.[Run].txt` - for `Run`=1,2,3: 3 training empirical risk values for the corresponding run
   `Risk.test.[Run].txt` - for `Run`=1,2,3: 3 testing outputs for the corresponding run
   The function to do all of this should be called should be called `FitCubic()` in a file `FitCubic.py` within a submitted folder called `Assignment01_Problem02`

[1] You may use your own version at your own risk, but are encouraged to use the provided solution

B. In the written section of your submission describe what parts of these results did you expect from the mathematical theory discussed in class, what are outcome of bad luck when drawing random numbers and what are due to inaccuracies of operations on the computer? Justify your claims.
[60 points]

3. You are considering data on terminal tumors that includes the tumor size upon admission into the hospital, and the time from the patient's admission till them passing away, as in the example below. You assume that given a specific tumor volume patients die at a constant rate λ. You further suspect a linear connection between the tumor volume measured e.g. in mm^3 and the rate of patients' death, measured in persons/year and would like to apply linear regression to quantify this connection. Define loss based on maximum likelihood and write the empirical risk function.
[20 points]

Good luck!