Industrial Big Data Analytics and Machine Learning

Assignment #1

Problem 1 (*Please feel free to use any machine learning software package for this problem*)
Consider the data set below. Please apply kernel density estimation to plot the estimated densities for the following cases.

(a) Consider the 10 data points of *x* values only (i.e. single dimension data), and plot the estimated densities using the bandwidth as 0.1, 0.3, and 1, respectively. Please put all the three density distributions in one figure, together with the *x* values. Report the estimated *cumulative distribution function* value or the estimated *probability density* value at *x*=0.7 for the three cases using different bandwidths.

(b) Consider the 10 data points with both *x* and *y* values (i.e. bivariate data), and plot the estimated 2-D density using the bandwidth of 0.3.

x	*y*
0.1678	0.6097
0.1743	0.6100
0.3867	-0.6372
0.4555	-1.4296
0.5976	-2.9039
0.6332	-3.1160
0.8128	-2.4052
0.8401	-2.0088
0.9805	0.9767
0.9904	1.2236
0.7	*?*

Problem 2
Given the above data set on input *x* and output *y* of 10 data examples. Hypothesize *y* is a polynomial function of *x*, i.e. *y* = *f*(x). Consider the following five hypotheses:

- 2nd order polynomial
- 3rd order polynomial
- 4th order polynomial
- 5th order polynomial
- 6th order polynomial

(a) For each hypothesis, please define appropriate nonlinear basis function, and then use multivariate linear regression method to determine the coefficients of the polynomial function. Please use the analytic method (i.e. the one with matrix inversion) to solve each of the multivariate linear regression problems directly. Next, use Ridge Regression (set $\delta^2 = 0.005$ for the weight of the L2 regularizer). Predict the value of y for x=0.7 in each case, and make comparisons between different models obtained.

(b) Perform leave-one out cross validation (LOOCV) for the hypothesis of 4th order polynomial. Use Ridge Regression for each problem. Calculate the cross validation error for $\delta^2 = 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}$ (seven cases). Choose the best δ^2 according to cross-validation and report the final prediction at x=0.7.

(c) What conclusions can you draw from the results?

You can use any computational tool (e.g. MATLAB) to calculate the matrix inverse. Please clearly present the nonlinear basis functions, the mapping process, and the resulting polynomial function forms (with coefficients) for f(x). Please present your derivation results **step-by-step**.

Note: *In this problem, you are expected to build regressors using the training data based on the algorithms covered in the lectures. Your calculation process should be presented in sufficient details, ideally step-by-step. You could use any basic calculation/computing tools (e.g. NumPy, MATLAB without toolboxes, etc.) to facilitate your calculation process. If only the final regression model and/or results are presented without necessary intermediate steps and results, you will* ***not*** *get most of the credit.*

Problem 3 (*Please feel free to use any machine learning software package for this problem*)
Consider the data set of Problem 2 (on regression). Please develop the following regression models to predict the value of y for x=0.7.

(a) Standard linear regression (using the hypothesis of 4th order polynomial).
(b) Gaussian process.

Problem 4 [*Note*: *This is a bonus problem, and it is not required. You may work on it to earn an extra credit of at most 5 points that might offset any possible loss of points from previous problems*]

A dataset of housing prices in Portland, Oregon is given in the table below. The inputs $\{x_1, x_2\}$ are the living area and the number of bedrooms, and the output y to be predicted is the price. There are m=47 input-output data sample pairs $\{\boldsymbol{x}(k), y(k)\}_{k=1}^{m}$ in total. Please first scale two inputs by their standard deviations and set their means to zero. Next, add a constant term $x_0 = 1$ to the input. The prediction function is then given by:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=0}^{2} \theta_i x_i \triangleq \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}.$$

The cost function to be minimized is given by,

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{k=1}^{m}\left(y(k) - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}(k)\right)^2,$$

and thus the gradient updating rule is given by:

$$\theta_i := \theta_i - \alpha\frac{1}{m}\sum_{k=1}^{m}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}(k)\right) - y(k)\right)x_i(k),\ i = 0,1,2$$

where α denotes the learning rate / step size. Initialize the parameter to $\boldsymbol{\theta} = \mathbf{0}$, and carry out gradient descent for about 50 iterations with learning rates set as 0.01, 0.03, 0.1, 0.3, 1, 1.2, respectively. Plot the values of $J(\boldsymbol{\theta})$ during iterations for different learning rates on the SAME graph. Please implement the gradient descent algorithm, instead of solving the quadratic program directly.

x_1	x_2	y
2104	3	399900
1600	3	329900
2400	3	369000
1416	2	232000
3000	4	539900
1985	4	299900
1534	3	314900
1427	3	198999
1380	3	212000
1494	3	242500
1940	4	239999
2000	3	347000
1890	3	329999
4478	5	699900
1268	3	259900
2300	4	449900
1320	2	299900
1236	3	199900
2609	4	499998
3031	4	599000
1767	3	252900
1888	2	255000
1604	3	242900
1962	4	259900
3890	3	573900
1100	3	249900
1458	3	464500
2526	3	469000
2200	3	475000
2637	3	299900
1839	2	349900

1000	1	169900
2040	4	314900
3137	3	579900
1811	4	285900
1437	3	249900
1239	3	229900
2132	4	345000
4215	4	549000
2162	4	287000
1664	2	368500
2238	3	329900
2567	4	314000
1200	3	299000
852	2	179900
1852	4	299900
1203	3	239500