Problem Set 1

Submission instructions and due date:

The due date for this homework assignment is as listed on Blackboard as part of the "Problem Set ..." file name on Blackboard. The completed assignment must be submitted in class (on paper, stapled per instructions below), at the beginning of the class session on the above due date, **unless** a different due date is announced in class *or* on Blackboard.

What to submit (print and staple in this order):

1) hw1scores.txt (complete and print out).
2) solutions to written problems.
3) solutions to programming problems.

For item 2 *of the list above* (solutions to written problems) – solutions to written problems can be handwritten; however, they should be written out clearly. It is required that you show all relevant steps leading to the solutions.

For all other items (i.e., items that require programming): complete and print out code, include screenshots of the output (related to this assignment) and the results' plots.

Grading:

This problem set is self-graded. It is strongly recommended that you solve and complete all of its tasks (written and programming), since – in addition to contributing towards the homework component of the final course-grade – they should help you prepare for quizzes/exams and the final project. You must also submit your self-assigned grades in "hw1scores.txt" (see also instructions within that file). To verify self-grading, for each homework assignment we will randomly choose a subset of submissions for grade checking and re-grade them if needed.
Reminders: Late-homework policy is as listed in the course syllabus. Copying solutions from **any** source – e.g., online, solutions manuals, assignment solutions from prior semesters, or solutions from any other sources – is prohibited.

1 Written Problems

1.1 Localized linear regression [10 points]

Suppose we want to estimate localized linear regression by weighting the contribution of the data points by their distance to the query point $x^{(q)}$, i.e. using the cost

$$E(x^{(q)}) = \frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))^2(x^{(i)} - x^{(q)})^{-2}$$

where $(x^{(i)} - x^{(q)})^{-2} = (w^{(i)})^2$ is the inverse Euclidean distance between the training point $x^{(i)}$ and query (test) point $x^{(q)}$.

Derive the modified normal equations for the above cost function $E(x^{(q)})$.

Hint: first, rewrite the cost function in matrix/vector notation, using a diagonal matrix to represent the weights $w^{(i)}$.

1.2 Maximum Likelihood Estimate for Coin Toss [10 points]

The probability distribution of a single binary variable $x \in \{0, 1\}$ that takes value 1 with probability μ is given by the *Bernoulli* distribution

$$Bern(x|\mu) = \mu^x(1-\mu)^{1-x}$$

For example, we can use it to model the probability of seeing 'heads' (x = 1) or "tails" (x = 0) after tossing a coin, with μ being the probability of seeing "heads".

Suppose we have a dataset of independent coin flips $D = \left\{x^{(1)}, x^{(2)}, ..., x^{(m)}\right\}$ and we would like to estimate μ using Maximum Likelihood (ML). Recall that we can write down the likelihood function as

$$p(D|\mu) = \prod_{i=1}^{m} p(x^{(i)}|\mu) = \prod_{i=1}^{m} \mu^{x^{(i)}}(1-\mu)^{1-x^{(i)}}$$

The log of the likelihood function is

$$\ln p(D|\mu) = \sum_{i=1}^{m} x^{(i)} \ln \mu + (1 - x^{(i)}) \ln (1-\mu)$$

Show that the ML solution for μ is given by $\mu_{ML} = \dfrac{h}{m'}$, where h is the total number of "heads" in the dataset. Show all of your steps.

1.3 Maximum likelihood for Logistic Regression [10 points]

Showing all steps, derive the Logistic Regression cost function using maximum likelihood. Assume that the probability of y given x is described by

$$P(y = 1|x; \theta) = h_\theta(x)$$
$$P(y = 0|x; \theta) = 1 - h_\theta(x)$$

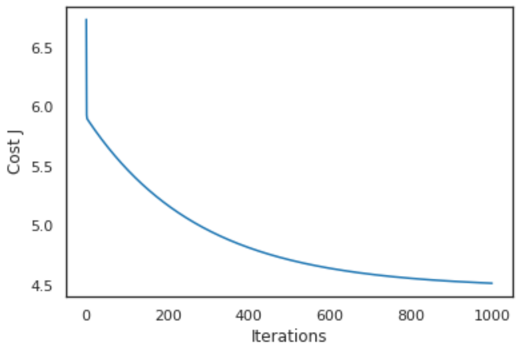
2 Programming problems

2.1 Linear Regression

2.1.1 Cost function [22 points]

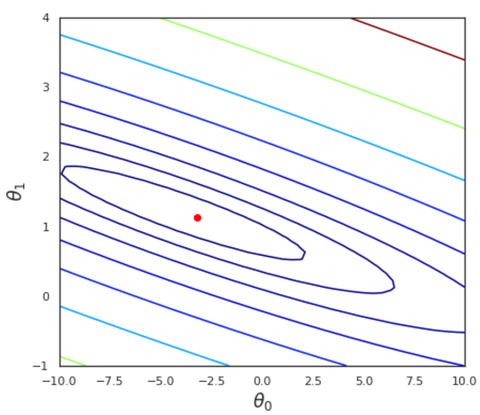
Complete the linear_regression.py file using python. The necessary information has been provided within that file. You need to find the linear regression coefficients (θ) using both normal equation and iterative approaches.Your result for θ to be approximately [-3.89578088 , 1.19303364].

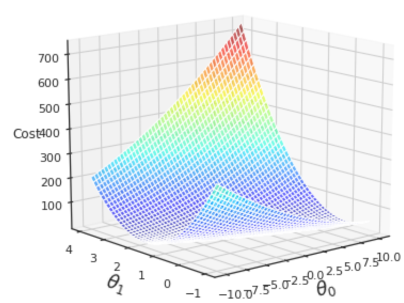
For the iterative approach, plot the cost function over number of iterations. Your plot should be similar to the following:



2.1.2 Gradient descent [9 points]

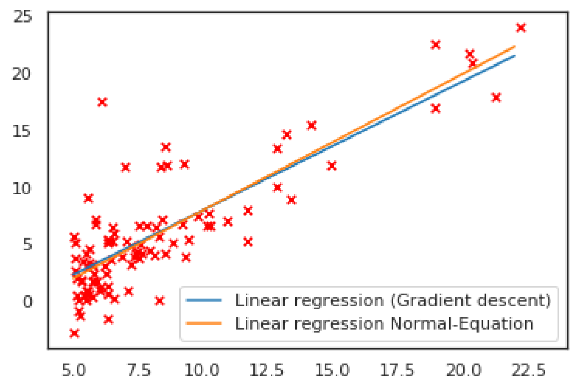
Plot the gradient descent in both 2d and 3d version. Your plots should be similar to:





2.1.3 Linear regression lines – gradient descent and normal equations [9 points]

Plot the linear regression lines for both the normal equation and gradient descent. Your plot should be similar to:



2.2 Logistic Regression [30 points]

Complete the logisitc_regression.py file using python. The necessary information has been provided within the file. You need to find the logistic regression coefficients (θ), using the iterative approach.

Your result for θ to be approximately: [2.23519281, -0.48565523, -0.20155501, -0.4060702 , 0.85979324, 1.30147915, -0.86998486, - 0.67611493, -0.42953018, -0.38405549, -0.5310473 , 0.64110397, -1.50052861, -0.79182939, 0.53264779, -0.49718156, 1.67300408].