Homework Assignment 2: Linear Regression

Due Monday, January 27th, 2020 at 11:59pm

Description

In class we discussed linear regression and how to solve for model parameters using gradient descent and normal equation. In this problem set, you will implement such approaches and evaluate it on data.

What to submit

Download and unzip the template ps2_matlab_template.zip. Rename it to ps2_LastName_FirstName, and add in your solutions:

ps2_LastName_FirstName /

- input/ - input data, images, videos or other data supplied with the problem set
- output/ - directory containing output images and other generated files
- ps2.m - your Matlab code for this problem set
- ps2_report.pdf - A PDF file that shows all your output for the problem set, including images labeled appropriately (by filename, e.g. ps0-1-a-1.png) so it is clear which section they are for and the small number of written responses necessary to answer some of the questions (as indicated). Also, for each main section, if it is not obvious how to run your code please provide brief but clear instructions (no need to include your entire code in the report).
- *.m - Any other supporting files, including Matlab function files, etc.
- ps2_LastName_FirstName_debugging.m – one m-file that has all of your codes from all the files you wrote for this assignment. It should be a concatenation of your main script and all of your functions in one file (simply copy all the codes and pate them in this file). In fact, this file in itself can be executed and you can regenerate all of your outputs using it.

Zip it as `ps2_LastName_FirstName.zip`, and submit on canvas.

Guidelines

1. Include all the required images in the report to avoid penalty.
2. Include all the textual responses, outputs and data structure values (if asked) in the report.
3. Make sure you submit the correct (and working) version of the code.
4. Include your name and ID on the report.

5. Comment your code appropriately.
6. Please avoid late submission. Late submission is not acceptable.
7. Plagiarism is prohibited as outlined in the Pitt Guidelines on Academic Integrity.

Questions

1- **Cost function**: As you perform gradient descent to learn minimize the cost function $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$ $(you\ can\ also\ use\ the\ vectorized\ form\ defined\ in\ the\ class)$, it is helpful to monitor the convergence by computing the cost. Write a function, `function J = computeCost(X, y, theta)` that computes the cost given an estimate of the parameter vector θ. As you are doing this, remember that the variables X and y are not scalar values, but matrices whose rows represent the examples from the training set.

2- **Gradient descent**: write a function, `[theta, cost] = gradientDescent(X_train, y_train, alpha, iters)` that computes the gradient descent solution to linear regression.

inputs:

- `X_train` is an mxn feature matrix with m samples and n feature dimensions. m is the number of samples in the training set.
- `y_train` is an mx1 vector containing the labels for the training set. The i-th sample in `y_train` should correspond to the i-th row in `X_train`
- `alpha`, the learning rate to use in the weight update.
- `iters`, the number of iterations to run gradient descent for,

outputs:

- `theta` is a nx1 vector of weights (one per feature dimension).
- `cost` is a `iters` x1 vector of cost values (one per each iteration).

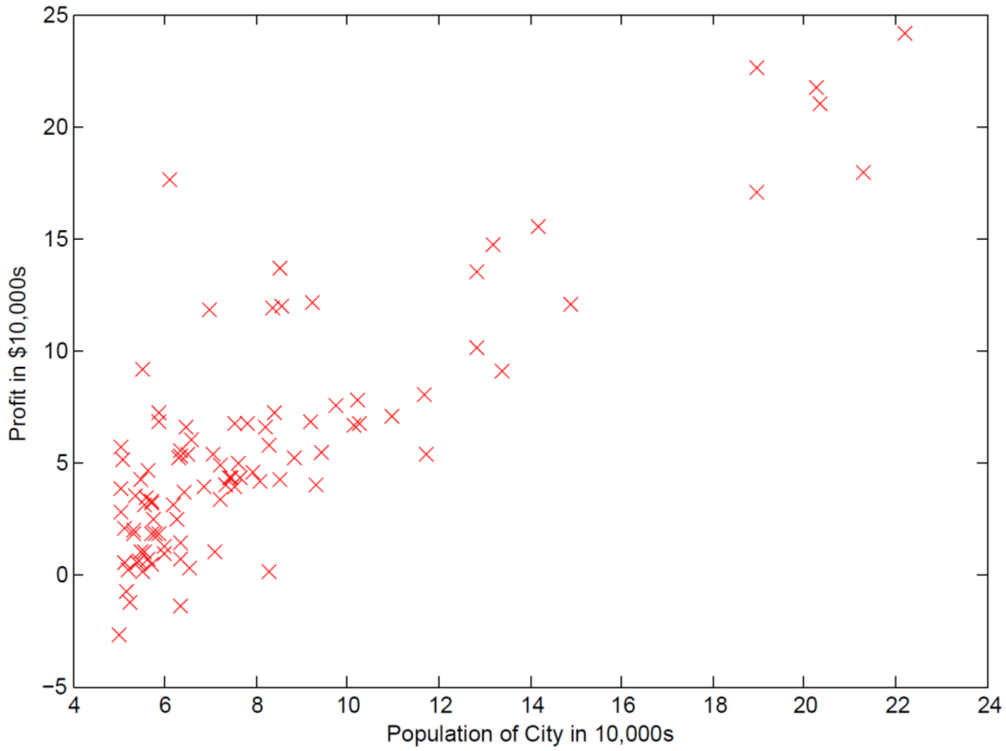
Your function should use ZERO initialization for theta, and then update theta `iters` times using the gradient descent algorithm we studied in the class.

3- **Normal equation**: write a function, `[theta] = normalEqn(X_train, y_train)` that computes the closed-form solution to linear regression using normal equation. The inputs and outputs are as defined in question 2.

4- **Linear regression with one variable**: The data in the file 'hw2_data1.txt' contains the profit of a food chain (in $10,000s) vs the population in each city (in 10,000). The food chain wants to expand in new cities. They want to get an estimate of the expected profit in each target city (based on the population) to priorities the expansion plan.

a. Load this data in MATLAB. The first column is the population of a city, and the second column is the profit in that city. A negative value for profit indicates a loss.

b. Plot the date to visualize the problem. Your output should look like the following figure.
Output: store the scatter plot of the data as ps2-4-b.png



c. We store each example as a row in the X matrix in MATLAB. To take into account the intercept term (θ_0), we add an additional first column to X and set it to all ones. This allows us to treat θ_0 as simply another `feature'. Define X and according to the description in this part and part a.
Text output: the size of the feature matrix X and the size of the label vector y.

d. Test you cost function by passing in the feature vector X (population), the output vector y (profit), and parameter vector $\theta = [0\ 0]^T$.
Text output: the cost associated with zero model parameters.

e. Use a learning rate of 0.01 and 1500 iterations, compute the gradient descent solution of the model parameters θ. Plot the vector cost that shows the cost function for each iteration.
Output: a plot of cost vs iteration# saved as ps2-4-e.png
Text output: the computed model parameters θ.

f. Use the obtained model parameters from e to make predictions on profits in cites of 35,000 and 70,000 people.
Text output: your predictions

g. Use the normalEqn function, make predictions on profits in cites of 35,000 and 70,000 people.

Text output: your predictions, compare these predictions to the one you obtained in f. Any comments?

h. In this part we want to study the effect of the learning rate. Use 250 iterations, solve for theta using the following learning rates $\alpha = [0.0001\ 0.001\ 0.03\ 0.1\ 1.0]$. This means that you have to run you function 5 times. In each time you would get a different theta and cost vectors. Plot the cost vs iteration# for each alpha on one figure. Use legend to identify different lines.
Output: one figure that showing the progression of cost vs iteration# for 5 different values of alpha, ps2-4-h.png
Text output: comment on the figure.

5- **Linear regression with multiple variables**: The data in the file 'hw2_data2.txt' contains a training set of housing prices in one city. The first column is the size of the house (in square feet), the second column is the number of bedrooms, and the third column is the price of the house.

a. Load the data into MATLAB. Then, the data, by computing the mean and standard deviation for each feature dimension, then subtracting the mean and dividing by the stdev for each feature and each sample. Append a 1 for each feature vector, which will correspond to the bias (θ_0) that our model learns.
Text output:
   - mean and standard deviation of each vector
   - the size of the feature matrix X and the size of the label vector y.

b. Use a learning rate of 0.01 and 1500 iterations, compute the gradient descent solution of the model parameters θ. Plot the vector cost that shows the cost function for each iteration.
Output: a plot of cost vs iteration# saved as ps2-5-b.png
Text output: the computed model parameters θ.

c. Predict the price of a house with 1650 square feet and 3 bedrooms. Note that you need to normalize this feature vector using the mean and standard deviation from a.
Text output: your predictions