CS 419M :Assignment 1

February 2021

This assignment is entirely programming-based. The task is hosted on Kaggle here.

1. Go to the Kaggle site.
2. Create a new login using your roll number. It is important that you use your roll number and nothing else.
3. Please contact the TAs if you need any help setting this up.

1 Many faces of linear regression

Predicting prices of pre-owned automobiles can be useful for websites like Cars24 and Olx. Given just the model and manufacture year, one could come up with an reasonable price. Can one do better with additional features? In this assignment, you will be given a set of data points with different features characterizing pre-owned cars in India. You will need to find the best linear regression model that most accurately predicts car prices for a set of test cars.

The data sets train.csv, dev.csv and test.csv are available for you in the provided file. Within lr.py, you will find a function closed_soln(phi, y) that computes the closed-form solution for an unregularized least squares linear regression model. You will also see placeholders for functions with self-explanatory names (e.g. compute_RMSE(phi, w, y), generate_output(phi_test, w)) that need to be filled in. The function main() has been expanded to show the sequence of steps we will run to evaluate tasks 1 to 3. Please use **Python 3.5** to run your script and minimize version discrepancies during grading.

Submission deadline of this assignment is **11:55 PM,** 7^{th} **March, '21** on Moodle.

1. Implement the gradient descent algorithm to estimate the model parameters for an unregularized least squares linear regression model. Add your implementation to the gradient_descent(phi, y) function. Compute the weights using gradient descent and find predicted car prices for the development set samples in dev.csv. Report the root mean squared error you obtain on these samples in submission/report.pdf. Set the learning rate in gradient_descent(phi, y) to the value we should use during grading. (Your weight vector should start from an all-zeros vector or a random initialization.) **Tip**: You will find it useful to apply some form of feature normalization on your inputs before optimizing your loss function. [**10 points**]

   Report the following details in submission/report.pdf:

   - What is the absolute difference between the following two calls, compute_RMSE(phi, w1, y) and compute_RMSE(phi, w2, y) on dev.csv, where w1 and w2 were obtained using the closed-form solution and gradient descent, respectively? [**2 points**]
   - What stopping criterion did you use to determine when gradient descent converged? [**1 point**]
   - Instead of batch gradient descent, implement stochastic gradient descent in sgd(phi, y). Let w1 and w2 be the respective weight vectors. Report the absolute difference between the two calls compute_RMSE(phi_train, w1, y) and compute_RMSE(phi_train, w2, y) on dev.csv. (Tune and set the learning rate such that this difference is minimized.) [**5 points**]

2. Within pnorm(phi, y, p), optimize a p-norm regularized least squares objective function (where p is 2 or 4) using either gradient descent or stochastic gradient descent. Your regularization term now becomes $\lambda(||w||_p)^p$ where p-norm of a d-dimensional w is defined as $||w||^p = (\Sigma_{i=1}^{d}|w_i|^p)^{1/p}$ and λ is a non-negative value. This term is added to the standard least squares objective to form the p-norm regularized objective. Report root mean squared error on all the development set samples when p=2 and p=4 in submission/report.pdf. Set λ in pnorm(phi, y, p) to the value we should use during grading. [**8 points**]

3. Implement a basis function that will be applied to your input features with the L2-regularized model and optimized using gradient descent. Add your implementations of the basis functions to get_features_basis(file_path). Describe your choice of basis functions in submission/report.pdf. Also report the root mean squared error on the development set samples using this basis function. [**4 points**]

4. Create subsets of the training data containing the first 2000, 2500, 3000 instances in train.csv. Create a plot where the X-axis is the size of the training set (2000, 2500, 3000, full) and the Y-axis is the root mean-square error on the dev.csv instances. You can create a new function within lr.py and use existing libraries (like matplotlib) to create this plot. Add it to submission/report.pdf. [**3 points**]

5. Of the 12 features, which are the two least useful ones? Describe how you identified these two features in submission/report.pdf. [**2 points**]

6. The objective for this last part is to build the best possible linear regression model using any enhancements you like. Submit the output from generate_output(phi_test, w) on test.csv to Kaggle so that your roll number appears on both the "Public Leaderboard" and "Private Leaderboard". (In this output file, do make sure that you maintain the same order of the samples that appear in test.csv.) Top-scoring performers on the "Private Leaderboard" (with a suitable threshold determined after the submission date) will be awarded extra credit points. Describe your innovations in submission/report.pdf. Also, submit your code for this part within lr.py. (You can define any new functions for this last part and run this within main().) [**5 points**]

1.1 Submission

The directory submission will contain 2 files, lr.py and report.pdf. The directory submission should be submitted on Moodle as a .tgz file using the command:

tar -czf submission.tgz submission