University of Toronto Scarborough
Department of Computer and Mathematical Sciences

Introduction to Machine Learning and Data Mining
CSCC11H3, Fall 2021

Assignment 1

Programming Questions

Due October 26 at 11:49 pm

Q1.

Consider the following nonlinear programming model

$$\min 2x_1^2 + x_2^2 + x_3^4$$
$$\text{s.t. } x_1, x_2, x_3 \in \mathbb{R}$$

(I) Gradient Descent Method

Implement the gradient descent algorithm whose updating rule is given as follows:

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - \alpha_t \nabla f\left(\mathbf{x}^{(t-1)}\right) \quad \text{for } t = 1, 2, \ldots$$

Assuming $\alpha_t = 0.1$, `MAX_ITERATION` $= 20$ or `REL_ERROR` < 0.001 termination conditions, and the initial solution given by

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ -4 \\ 2 \end{bmatrix},$$

1. Solve the optimization problem given above using gradient descent algorithm and report the optimal value and the optimal solution.
2. Do gradient descent steps always decrease the loss? why? (You may plot the objective values against the iteration numbers of the method given different $\alpha_t = 1, 0.1, 0.01, 0.001$ values to support your rationale.)

(II) Newton's Method

Implement Newton's algorithm whose updating rule is given as follows:

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - \left[\nabla^2 f\left(\mathbf{x}^{(t-1)}\right)\right]^{-1} \nabla f\left(\mathbf{x}^{(t-1)}\right) \quad \text{for } t = 1, 2, \ldots$$

Assuming $\texttt{MAX\_ITERATION} = 20$ or $\texttt{REL\_ERROR} < 0.001$ termination conditions, and the initial solution given by

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ -4 \\ 2 \end{bmatrix},$$

1. Solve the optimization problem given above using Newton's method and report the optimal value and the optimal solution.
2. How does the gradient descent algorithm relate to Newton's method?
   Hint: You may first establish a relation between α_t and $\left[\nabla^2 f\left(\mathbf{x}^{(t-1)}\right)\right]^{-1}$ and then interpret that.
3. What are the main issues with Newton's method?

Q2.

We aim to find Least Squares (LS) polynomial models for this problem. Each polynomial is expressed as a weighted sum of monomials, where the largest monomial degree is denoted K.

$$y = f(x) = w_0 + \sum_{k=1}^{K} w_k x^k$$

Your goals are to find the LS parameter estimates for $\mathbf{w}_K = [w_0, w_1, \ldots, w_K]^T$, for each K from 1 to 10 . Then, you will select a single model (i.e., find a good value for K). You want a model that will give good predictions on future (unseen) test data. What follows are more detailed instructions. Note: w_0 is the bias term and is the first element of the vector $\mathbf{w}_K$.

Now, please download and untar the file containing the datasets. The data files are stored as a dictionary in Pickle format. To read them from file,
- Run one of Python Shell, Spyder, Jupyter Notebook, etc.
- Import the Pickle module (Note the underscore): `import _pickle as pickle`
- Open the Pickle file and read the content (replacing **data.pkl** below with the actual file name)

```
with open("data.pkl", "rb") as f:
      data = pickle.load(f)
```

- You may access the training inputs and outputs with data `data{"X"}` and `data{"Y"}`, respectively. They are stored as numpy arrays with shape $(N, 1)$. These are the N-dimensional column vectors comprising the N training inputs and outputs for the regression problem. Each dataset is generated from a polynomial function of a scalar input x, to which noise has been added. Furthermore, each dataset comes with a small training set, a large training set, and a test set. Among other things this will allow you to see how different amounts of data affect model fitting.

Once you are confident with your implementation of the polynomial regressor, answer the following questions:

1. In general, if I increase the size of the training set, what can we expect about the model's training error? What about the test error?
2. In general, if I increase the size of the test set, what can we expect the about the model's training error? What about the test error?
3. How much data should I try to obtain if I want to build a good model?

4. In general, if the model is too simple, what can we conclude about the training and test errors?
5. In general, if the model is too complex, what can we conclude about the training and test errors?
6. For each dataset, which (degree) model gives the best performance? How did you find out?
7. For each dataset, what degree of polynomial do you think was used to generate the data?