703642, Advanced Machine Learning course 29th October, 2018

Assignment 1

Programming Project: Ridge Regression and Model Selection

Kamal Zakieldin, Yusuf Ipek

Overview:

In this report we are describing our implementation for the ridge regression model and our experiments we have made, first we introduce the used technology and the libraries, then we discuss our tasks illustrating what we have made then showing the results and the values we have got.

Technologies:

Python as the programming language.

Libraries:

This section will give a short overview of the libraries we used.

- **numpy** provide help working with matrices.
- **matplotlib** for all ploting and drawing the figures to visualize our results.

Discussion:

We start our implementation by generating the data set of N points, choosing a quadratic function f(x) using the following equation:

$$t_n = f(x_n) + \epsilon_i \quad \text{where} \quad \epsilon_i \sim^{iid} N(0, \sigma^2)$$

We have used epsilon to add a normal distributed uniform noise calculated using the mean and the variance of points.

in the beginning we have created dataset with high X values that create a very high Y values, and depending on high Y values were really difficult and increased the error, that's why we used a smaller X values that subsequently produced better Y values, generating the data set was one of the most underestimated step that affect the performance of our model.

Then we built our model of the linear regression model using matrix formulation.

Using the **Erms** to reduce the error between the actual values and the predicted values, one way to compute the minimum of a function is to set the partial derivatives to zero

$$\frac{\partial}{\partial \theta_i}\left[\frac{1}{2N}\parallel \boldsymbol{y} - Q\boldsymbol{\theta} \parallel^2\right] = 0$$

We then use this derivation for all the weights.

We then can use them to solve our matrices to get the **f(x) =** $\sum_{j=0} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$

We then get the design matrix;

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}$$

and finally we get the $\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{w} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{t}.$ as **AX = b**

and we can then calculate the weights using this equation and adding the regularization term to penalize the parameters.

$$\mathbf{w} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$