COMP 6321 Machine Learning

Assignment 2 Answers

Name : Parsa Kamalipour , *StudentID* : 40310734

Exercise 1: Regression Implementation (9 pts)

Recall that ridge regression refers to

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \overbrace{\underbrace{\frac{1}{2n}\|X\mathbf{w}+b\mathbf{1}-\mathbf{y}\|_2^2}_{\text{error}}+\lambda\|\mathbf{w}\|_2^2}^{\text{loss}}, \tag{1}$$

where $X \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$ are the given dataset and $\lambda \geq 0$ is the regularization hyperparameter. If $\lambda = 0$, then this is the standard linear regression problem. Observe the distinction between the error (which does not include the regularization term) and the loss (which does).

1. (1.75 pts) Show that ridge regression can be rewritten as a non-regularized linear regression problem with data augmentation. That is, prove 1 is equivalent to

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2n}\left\|\begin{bmatrix} X & \mathbf{1}_n \\ \sqrt{2\lambda n} I_d & \mathbf{0}_d \end{bmatrix}\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix} - \begin{bmatrix}\mathbf{y}\\ \mathbf{0}_d\end{bmatrix}\right\|_2^2, \tag{2}$$

   where I_d is the d-dimensional identity matrix, and $\mathbf{0}_k$ and $\mathbf{1}_k$ are zero and one column vectors in k dimensions.

2. (2.25 pts) Implement the Ridge regression algorithm using the closed form solution for linear regression. Do not use any library like Scikit Learn that already has linear regression implemented. Feel free to use general libraries for array and matrix operations such as numpy. You may find the function `numpy.linalg.solve` useful. Test Ridge regression implementation on the Boston housing dataset (to predict the median house price, i.e., y). Use the train and test splits provided on Moodle. Try $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ and report your training error, and test error for each.

3. (2.25 pts) Repeat step 2 but solve Ridge regression using the gradient descent algorithm. Try $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ and report your training error, and test error for each.

4. (2.25 pts) Repeat step 3, but this time solve Lasso regression using gradient descent. For Lasso regression, the regularization term in equation (1) is $\lambda||w||_2$ i.e. you only have the norm and not the square of the norm. Try $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ and report your training error, and test error for each.

5. (0.25 pts) Do you think gradient descent is better than the closed form solution of Ridge regression? Explain why.

6. (0.25 pts) Print the θ values found using gradient descent for both Ridge and Lasso regression. Are these two values different? If so, can you explain how the Lasso and Ridge regression algorithms affected the θ values?

Answers to Exercise 1

Part 1

Given the Ridge Regression objective function:

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \left(\frac{1}{2n}\|Xw+b\mathbf{1}_n-y\|_2^2+\lambda\|w\|_2^2\right)$$

we need to show that it can be rewritten as a non-regularized linear regression problem using data augmentation, specifically:

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2n} \left\| \begin{bmatrix} X \\ \sqrt{2\lambda} I_d \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} - \begin{bmatrix} y \\ 0_d \end{bmatrix} \right\|_2^2$$

where I_d is the identity matrix of size $d \times d$ and 0_d is a zero vector of length d.

Step 1: Breakdown of the Original Objective Function

The given Ridge Regression objective function consists of two parts:

- The error term: $\frac{1}{2n}\|Xw + b\mathbf{1}_n - y\|_2^2$, which measures the difference between the predicted and actual outputs.

- The regularization term: $\lambda\|w\|_2^2$, which penalizes the magnitude of the weights.

We need to rewrite this objective in a form that includes the regularization term as part of the least squares problem by augmenting the data.

Step 2: Regularization and Augmentation

To incorporate the regularization term $\lambda\|w\|_2^2$ into the least squares framework, we augment both the input matrix X and the target vector y.

Augmenting the input matrix: We append $\sqrt{2\lambda} \cdot I_d$ to the original data matrix X, ensuring that the regularization term is treated like a squared error in the new least squares formulation. The augmented input matrix $\tilde{X}$ becomes:

$$\tilde{X} = \begin{bmatrix} X \\ \sqrt{2\lambda} I_d \end{bmatrix}$$

Augmenting the target vector: We also augment the target vector y by appending a zero vector 0_d (of dimension d) to ensure that the regularization term does not affect the bias term b. The augmented target vector $\tilde{y}$ is:

$$\tilde{y} = \begin{bmatrix} y \\ 0_d \end{bmatrix}$$

Step 3: Substituting the Augmented Data

Now, we substitute the augmented input matrix $\tilde{X}$ and augmented target vector $\tilde{y}$ into the Ridge Regression objective function:

$$L(w, b) = \frac{1}{2n} \left\| \tilde{X} \begin{bmatrix} w \\ b \end{bmatrix} - \tilde{y} \right\|_2^2$$

Expanding this:

$$L(w, b) = \frac{1}{2n} \left\| \begin{bmatrix} X \\ \sqrt{2\lambda} \cdot I_d \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} - \begin{bmatrix} y \\ 0_d \end{bmatrix} \right\|_2^2$$

Step 4: Replacing X and y in the Formula

We can further break this down into two components: - $Xw+b\mathbf{1}_n-y$ represents the least squares error between the predicted values and the target y. - $\sqrt{2\lambda} \cdot I_d \cdot w - 0_d$ represents the regularization term $\lambda\|w\|_2^2$, ensuring that the weights w are penalized without affecting the bias b.

Thus, the augmented least squares problem now accounts for both the error term and the regularization term.

Other questions

Please look at the ParsaKamalipour_COMP6321_A2.ipynb file to read the answer to rest of the questions.

Exercise 2: Decision Trees (7 pts)

In this exercise, you will implement decision trees for binary classification. Use the provided stub files for training and test data.

Recall: decision trees are constructed by repeatedly splitting of nodes. We split a node by measuring the loss of splitting with respect to each possible feature and threshold, and split based on the feature and threshold that minimizes this loss. Mathematically:

$$X_L = \{x \ : \ x \in X \wedge x[i] \leq j\},$$

$$X_R = \{x \ : \ x \in X \wedge x[i] > j\},$$

where $x[i]$ is the ith coordinate of point x. The vector of labels y is split into vectors y_L and y_R using the same indices.

The loss of splitting a training dataset into a left and right half is computed as

$$\ell(X, y, i, j) = \frac{|y_L|}{|y|}\ell(y_L) + \frac{|y_R|}{|y|}\ell(y_R)$$

We will consider the following loss functions ℓ, specialized for binary classification.[a] Define $\hat{p}$ for a vector of labels y to be $\frac{|\{y_j=1:y_j\in y\}|}{|y|}$, that is, the fraction of labels which are 1. We have the following three loss functions.

Misclassification error:

$$\min\{\hat{p}, 1-\hat{p}\}$$

Gini coefficient:

$$\hat{p}(1-\hat{p})$$

Entropy:

$$-\hat{p}\log_2(\hat{p}) - (1-\hat{p})\log_2(1-\hat{p})$$

We do not split a node if it is pure (i.e., consists entirely of either 0's or 1's), or if a split would exceed a maximum depth hyperparameter provided to the decision tree (recall that the depth of a single-node tree is 0).

1. (6 pts) Implement and train decision trees on the provided dataset. Create a different plot for each of the three loss functions (misclassification error, Gini index, and entropy). The x-axis of each plot should show the maximum depth of the tree (starting from 0), and the y-axis should indicate the accuracy. Include two trend lines, one for the training accuracy and test accuracy.

2. (1 pt) Observe and comment on how the different loss functions perform, and how train and test accuracy change as a function of the maximum depth.

[a] Note that these are slightly different from what we discussed in class, since we are only focusing on binary classification. Additionally, there was some implicit rescaling which we omit here.

Answers to Exercise 2

Please look at the ParsaKamalipour_COMP6321_A2.ipynb file to read the answer to rest of the questions.