In the Name of God

Machine Learning (Spring 2020)

Assignment #4

Regression Algorithms

Due date: 4th of Tir

Experiments on regression problems use five different sets of synthetic data with n data points $\{x_i\}_i^n$ with p dimensions. Stack these data points in a $p \times n$ matrix of X. The univariate response variable $\boldsymbol{y}$ depends only on a particular set of features. The jth feature is denoted by $X_{j:}$ (jth row of X) and the ith data point is denoted by $X_{:i}$ (ith column of X).

Regression A: This regression model is defined as:

$$\boldsymbol{y} = \frac{X_{1:}}{0.5 + (X_{2:} + 1.5)^2} + (1 + X_{2:})^2 + 0.5\varepsilon$$

where $X_{:i} \sim N(0, I_4)$ is a four-dimensional input vector and $\varepsilon \sim N(0,1)$ is the normal additive Gaussian noise. In this model only, the first two features are related to the response variable $\boldsymbol{y}$**.**

Regression B: The second regression model is as follows:

$$y = \frac{1}{2}(X_{1:}^2)\varepsilon$$

where $X_{:i} \sim N(0, I_{10})\ \$$ and $\varepsilon \sim N(0,1)$ is the independent noise. The dimension of the true space is 1 and the noise is multiplicative rather than additive in this case.

Regression C: The third regression model is defined as follows:

$$\boldsymbol{y} = X_{1:}^2 + X_{2:} + 0.5\varepsilon$$

where $X \sim N(0, I_4)$ is as it is defined in regression (A), and $\varepsilon \sim N(0,1)$.

Regression D: The final regression model is as follows:

$$\boldsymbol{y} = \cos\left(\frac{3X_{1:}}{2}\right) + \frac{X_{2:}^3}{2} + 0.5\varepsilon$$

in which $X = (X_1, \ldots, X_{10})^T \sim N(0, I_{10})$, and $\varepsilon \sim N(0,1)$.

Samples of size $n = 100$ should be drawn out of each regression model and in each set 70%, of the data is used for training and the remaining 30% is used as testing data. The average root mean square error (RMSE) of estimating the test data response variable for different regression models should be reported in each case for fifty times.

After constructing these regression models, you are required to implement the following four algorithms.

In the Name of God
Machine Learning (Spring 2020)
Assignment #4
Regression Algorithms

I. Linear Regression

In linear regression we have a set of data points $X \in R^{m\times d}$, $\{x_i\}_{i=1}^{m}$ $x_i \in R^d, and\ \{y_i\}_{i=1}^{m}$ and the following objective function:

$$\widehat{w} = \underset{w\in R^d}{\operatorname{argmin}} \sum_{i=1}^{m} (w^T x_i - y_i)^2$$

which can be solved in closed form as:

$$W = (X^T X)^{-1} X^T Y$$

II. Ridge Regression (Regression with L_2-regularization)

As you know, when a regularization penalty (scaled by λ) is added to the objective function of linear regression, we call it ridge regression, and its objective function is:

$$\widehat{w} = \underset{w\in R^d}{\operatorname{argmin}} \sum_{i=1}^{m} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

which can be solved in closed-form as:

$$W = (X^T X + \lambda I)^{-1} X^T Y$$

III. Coordinate Descent for Lasso

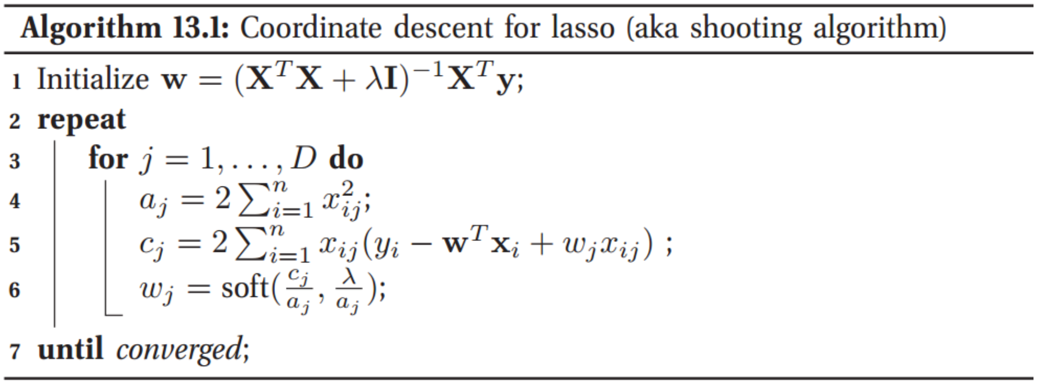
The Lasso optimization problem can be formulated as:

$$\widehat{w} = \underset{w\in R^d}{\operatorname{argmin}} \sum_{i=1}^{m} (w^T x_i - y_i)^2 + \lambda \|w\|_1$$

in which $\|w\|_1 = \sum_{j=1}^{d} |w_j|$.

Since the L_1-regularization term in the objective function is non-differentiable, it is not clear how gradient descent or SGD could be used to solve this optimization problem, directly. Another approach to solve an optimization problem is coordinate descent, in which at each step we optimize over one component of the unknown parameter vector and fix all other unknown components. The descent path is a sequence of steps, each of which is parallel to a coordinate axis in R^d.

This gives us the following algorithm, known as the shooting algorithm:



Algorithm 13.1: Coordinate descent for lasso (aka shooting algorithm)

```
Initialize $\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$;
repeat
    for $j = 1, \ldots, D$ do
        $a_j = 2\sum_{i=1}^{n} x_{ij}^2$;
        $c_j = 2\sum_{i=1}^{n} x_{ij}(y_i - \mathbf{w}^T\mathbf{x}_i + w_j x_{ij})$ ;
        $w_j = \text{soft}(\frac{c_j}{a_j}, \frac{\lambda}{a_j})$;
until converged;
```

(Source: Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.)

The "soft thresholding" function is defined as:

$$soft(a, \delta) = sign(a)(|a| - \delta)_+$$

where $(|a| - \delta)_+ = \max((|a| - \delta), 0)$ is the positive part of $(|a| - \delta)$.

IV. Kernelized Ridge Regression

We replace all data points with their feature vector: $x_i \rightarrow \Phi_i = \Phi(x_i)$. In this case, the number of dimensions can be much higher, or even infinitely higher, than the number of data points. There is a neat trick that allows us to perform ridge regression in high-dimensional space as follows:

$$\hat{w} = \underset{w \in R^d}{\operatorname{argmin}} \sum_{i=1}^{m} (w^T \Phi(x_i) - y_i)^2 + \lambda \|w\|_2^2$$

$$\alpha = (K + \lambda I)^{-1} y \,; \qquad \hat{w} = \sum_{i=1}^{m} \alpha_i \, \Phi(x_i); \qquad g(x) = \ w^T \Phi(x) = \sum_{i=1}^{m} \alpha_i K(x, x_i)$$

where $K(x, x_i) = \Phi(x)\Phi(x_i)^T$.

You should apply four mentioned methods on four generated data sets and report the RMSE for different values of the regularization parameter $\lambda = \{0.5,\ 1,\ 10, 100, 1000\}$.

For kernel ridge regression, report RMSE for the Gaussian kernel: $K(x, x_i) = e^{-\frac{\|x_i - x\|_2^{\,2}}{2}}$ and polynomial kernel: $k(x, x_i) = (x\, x_i^T + 1)^d$ where $d = \{2, 5, 10\}$. For lasso, set the convergence condition to 200 iterations.

Questions:

- Explain advantages and disadvantages of kernel ridge regression? Under what circumstances is the kernel regression better than the other methods?
- What if λ is set to an extremely large value? Explain the role of the regularization parameter λ on training phase.
- In Kernel ridge regression, which kernel is better than the other, and why?
- When is lasso worse than ridge regression?
- Compare four mentioned methods, in terms of, noise, outlier, the linearly separable data, the non-linearly separable data, number of samples, and number of features. What is the effect of ε on each regression algorithm?

Important Notes:

- Pay extra attention to the due date. It is fixed and **will not be extended**.
- Be advised that no submissions after the deadline would **be graded**.
- Be sure to comment your code. Also, include instructions on how to run your code (if necessary).
- Prepare a complete report for your assignment and answer all the questions.
- The name of the uploading file should be your **Lastname_Firstname.**

Good Luck