Lab Course Machine Learning
Exercise Sheet 6

Prof. Dr. Dr. Lars Schmidt-Thieme, Shayan Jawed
Information Systems and Machine Learning Lab
University of Hildesheim

December 3rd, 2020
Submission on December 9th, 2020 at 12 noon, (on learnweb, course code 3116)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a jupyter notebook detailing your solution.
2. Please set the seed(s) to 3116.
3. Please explain your approach i.e. how you solved a given problem and present your results in form of graphs and tables.
4. Please submit your jupyter notebook to learnweb before the deadline. Please refrain from emailing the solutions except in case of emergencies.
5. Unless explicitly noted, you are not allowed to use scikit, sklearn or any other library for solving any part.
6. Please refrain from plagiarism.

Exercise 0: Dataset Preprocessing (0 Points)

Datasets

- 1. Regression Datasets

  (a) Generate a Sample dataset called D1 :
  i. Initialize matrix $x \in R^{100\times1}$ using Uniform distribution with $\mu = 1$ and $\sigma = 0.05$
  ii. Generate target $y \in R^{100\times1}$ using $y = 1.3x^2 + 4.8x + 8 + \psi$, where $\psi \in R^{100\times1}$ randomly initialized.
  (b) Wine Quality called D2: (use winequality-red.csv)http://archive.ics.uci.edu/ml/ datasets/Wine+Quality

You are required to pre-process given datasets when applicable using instructions communicated earlier.

Exercise 1: Generalized Linear Models with Scikit Learn (5 Points)

In previous labs you have implemented various optimization algorithms to solve linear or logistic regression problem. In this task you are required to use Scikit Learn to experiment with following linear models and Stochastic Gradient Descent (SGD) [Hint: use $SGDRegressor$]. You may use scikit learn for this question.

1. Ordinary Least Squares
2. Ridge Regression
3. LASSO

Following are required in this task

1. Split your data into Train and Test Splits according to the 80%:20% ratio. Use dataset D2
2. For each model, pick three sets of hyperparameters and learn each model (without cross validation). Measure Train and Test RMSE and plot it on one plot. Explain the plots and relate it to the theory studied in lectures i.e. influence of regularized vs non-regularized models. You have to compare the following models and argument should explain underfitting and overfitting.
3. Now tune the hyperparameters using scikit learn GridSearchCV and plot the results of cross validation for each model. [Hint: use GridSearchCV object method $cv\_results$ to see different metrics]
4. Using the optimal hyperparameter you have to evaluate each model on the Test Set. Report the results in a meaningful manner.

Exercise 2: Higher Order Polynomial Regression (5 Points)

In this task you are required to use dataset D1. So far we have only looked at 1st degree polynomial, i.e. linear polynomial and your D1 is also generated using linear polynomial. In this task you have to use higher degrees of polynomial feature for your data i.e. degrees 1, 2, 7, 10, 16 and 100. [Hint: use sklearn.preprocessing to generate polynomial features]. You may use scikit learn for this question. Your tasks are:

1. **Task A**: Prediction with high degree of polynomials
   a For each newly created dataset learn LinearRegression.
   b Plot the predicted curves for each dataset. Explain the phenomena you observed for different prediction curves.
2. **Task B**: Effect of Regularization
   a Fixed the degree of polynomial to 10
   b Pick Four values of λ (regularization constant) and learn Ridge Regression [Hint: use Ridge and your λ values should be far a part i.e. 0, $10^{-6}, 10^{-2}$, 1].
   c Plot the predicted curves for each dataset. Explain the phenomena you observed for different prediction curves.

Exercise 3: Implementing Coordinate Descent (10 Points)

So far we have looked at Gradient Descent, Stochastic Gradient Descent(Ascent). This week the main task is to implement Coordinate Descent that has been covered in the lecture. To make things a bit more interesting, we will be implementing Lasso Regression along with the Coordinate Descent. You may NOT use scikit-learn for this question . We will use the Wine Dataset for this question.

Fig. 1 Shows the implementation for the Coordinate Descent along with its minimization.

1. **Coordinate Descent.**
   a Implementing the Coordinate Descent algorithm.
   b Maintain a history of your β values. After training plot them against iterations [hint: If you have 10 features, you should have 11 βs (one bias, 10 features)]. Plot them all in a singel plot. This should show you the progression of your feature values as your train the model.
2. **Coordinate Descent with L1 Regularization**
   a Implement CD with L1 regularization (Fig. 2). Note that the update step is now including the L1 term.
   b Maintain a history of your β values. After training plot them against iterations.
3. **Task C**: Comparison
   a Compare the plots of the unregularized and regularized CD
   b Highlight the difference. What information can be inferred from these values.

Coordinate Descent

procedure MINIMIZE-CD($f : \mathbb{R}^N \to \mathbb{R}, g, x^{(0)} \in \mathbb{R}^N, i_{\max} \in \mathbb{N}, \epsilon \in \mathbb{R}^+$)
    for $i := 1, \ldots, i_{\max}$ **do**
        $x^{(i)} := x^{(i-1)}$
        for $n := 1, \ldots, N$ **do**
            $x_n^{(i)} := g_n(x_{-n}^{(i)})$
        if $f(x^{(i-1)}) - f(x^{(i)}) < \epsilon$ **then**
            return $x^{(i)}$
    error "not converged in $i_{\max}$ iterations"

procedure LEARN-LINREG-CD($\mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_N, y_N)\}, i_{\max} \in \mathbb{N}, \epsilon \in \mathbb{R}^+$)
    $X := (x_1, x_2, \ldots, x_N)^T$
    $y := (y_1, y_2, \ldots, y_N)^T$
    $\hat{\beta}_0 := (0, \ldots, 0)$
    $\hat{\beta} :=$ MINIMIZE-CD($f(\hat{\beta}) := (y - X\hat{\beta})^T(y - X\hat{\beta})$, $g(\hat{\beta}_m; \hat{\beta}_{-m}) := \frac{(y - X_{-m}\hat{\beta}_{-m})^T x_m}{x_m^T x_m}$, $\hat{\beta}_0, \alpha, i_{\max}, \epsilon$)
    return $\hat{\beta}$

Figure 1: Coordinate Descent Algorithm

procedure LEARN-LINREG-L1REG-CD($\mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_N, y_N)\}, \lambda \in \mathbb{R}^+, i_{\max} \in \mathbb{N}, \epsilon \in \mathbb{R}^+$)
    $X := (x_1, x_2, \ldots, x_N)^T$
    $y := (y_1, y_2, \ldots, y_N)^T$
    $\hat{\beta}_0 := (0, \ldots, 0)$
    $\hat{\beta} :=$ MINIMIZE-CD($f(\hat{\beta}) := (y - X\hat{\beta})^T(y - X\hat{\beta}) + \lambda||\beta||_1$, $g(\hat{\beta}_m; \hat{\beta}_{-m}) := \text{soft}\left(\frac{(y - X_{-m}\hat{\beta}_{-m})^T x_m}{x_m^T x_m}, \frac{\frac{1}{2}\lambda}{x_m^T x_m}\right)$, $\hat{\beta}_0, \alpha, i_{\max}, \epsilon$)
    return $\hat{\beta}$

$$\text{soft}(x, \epsilon) := \begin{cases} x - \epsilon, & \text{if } x > \epsilon \\ 0, & \text{if } |x| \leq \epsilon \\ x + \epsilon, & \text{if } x < -\epsilon \end{cases}$$

Figure 2: Coordinate Descent with Regularization

0.1 ANNEX

- Following lecture is relevant this exercise https://www.ismll.uni-hildesheim.de/lehre/ml-20w/script/ml-04-A3-regularization.pdf
- sklearn.model_selection, sklearn.metrics, sklearn.linear_model, sklearn.preprocessing
- Scikit Learn User Guide http://scikit-learn.org/stable/user_guide.html
- You can use matplotlib for plotting.
- sklearn.metrics http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics