Homework 3 report

Luka Utješinović, 63210495

I. Ridge Regression implementation

Let $X \in \mathbb{R}^{N\times p}$ be the target dataset and let $y \in \mathbb{R}^N$ be the response. For a regularization weight $\lambda > 0$ our objective function is:

$$J(\beta_0, \beta) = \sum_{i=1}^{N} (y_i - \beta_0 - \beta^T x_i)^2 + \lambda ||\beta||_2^2$$

where x_i denotes the i-th row of X.Differentiating J with respect to β_0, and assuming X is centered (i.e. every column of X is centered) we can obtain that, we can obtain that $\beta_0^* = \frac{1}{N}\sum_{i=1}^{N} y_i = \hat{y}$ nullifies $\frac{\partial J}{\partial \beta_0}$. Additionally, $\frac{\partial^2 J}{\partial \beta_0^2} = 2$, which makes $\beta_0 = \hat{y}$ the best choice for the intercept regardless of the value of other weights. Choosing $\hat{y}$ as the intercept gives us the following cost function in vectorized form:

$$J(\beta) = ||\overline{X}\beta - \overline{Y}||_2^2 + \lambda ||\beta||_2^2$$

where $\overline{X}, \overline{y}$ are the centered X and y respectively. Differentiating J with respect to vector β we obtain that $\beta^* = (\overline{X}^T\overline{X} + \lambda I_p)^{-1}\overline{X}^T\overline{y}$ is the stationary point, and furthermore, $\frac{\partial^2 J}{\partial \beta^2} = X^T X + \lambda I_p$ is the Hessian of J. $X^T X$ is the Gram matrix which is always positive semidefinite, and aditionally $spectrum(X^T X + \lambda I_p) = \{\sigma + \lambda | \sigma \in spectrum(X^T X)\}$. Since $\lambda > 0$, $X^T X + \lambda I_p$ is positive definite (which implies that J is strictly convex) and that $X^T X + \lambda I_p$ is always invertible (which means that β^* will always be uniquely determined, which was not always the case with ordinary linear regression). With this, we obtain closed form solution for ridge regression. In addition to centering X, one should also scale columns of X to unit variance (i.e. standardize X), since the scale of our input variables affects how much regularization will be applied to a specific weight.

Another approach (which seems to be the one sci-kit Python library is using) is to start from the same objective function, but instead of assuming X is centered, one can derive (through elementary differentiation) that the stationary point of the objective is:

$$[\beta_0^*, \beta_1^*, \dots \beta_p^*] = (X'^T X' + \lambda A_{p+1})^{-1} X'^T y$$

where $X' \in \mathbb{R}^{N\times(p+1)}$ is the feature matrix with a padded columns of 1s, and $A_{p+1} = \begin{bmatrix} 0 & 0_p^T \\ 0_p & I_p \end{bmatrix}$. In this case, the Hessian of the objective is $X'^T X' + \lambda A_{p+1}$.

Let us show that this matrix is positive definite. Let $z \in \mathbb{R}^{p+1} \setminus \{0_{p+1}\}$ be an arbitrary vector. We need to show that $z^T(X'^T X' + \lambda A_{p+1})z > 0 = z^T(X'^T X')z + z^T \lambda A_{p+1} z > 0$. If at least one $z_j \neq 0$ for some $j \in \{2, \dots p+1\}$ then $z^T(X'^T X')z \geq 0$ because $X'^T X'$ is the Gram matrix which is always positive semidefinite, and $z^T \lambda A_{p+1} z \geq z_j^2 > 0$, which implies that $z^T(X'^T X' + \lambda A_{p+1})z > 0$. If $z_j = 0, \forall j \in \{2, \dots, p+1\}$, then $z_1 \neq 0$ (because we exclude the zero vector from positive definiteness definition). In that case $z^T \lambda A_{p+1} z = 0$, but since $X'^T X' = \begin{bmatrix} 1 & 1_N^T X \\ X^T 1_N & X \end{bmatrix}$, it is clear that $z^T(X'^T X')z = z_1^2 > 0$, and hence $X'^T X + \lambda A_{p+1}$ is positive definite. This implies that J is again a strictly convex function, and β^* is the global minimum. We again emphasize that this approach does not assume that X is centered, and this will lead to $\beta_0 \neq \hat{y}$ in general.

II. Lasso Regression Implementation

The setup is the same as in the previous section. We have the following objective function:

$$J(\beta_0, \beta) = \sum_{i=1}^{N} (y_i - \beta_0 - \beta^T x_i)^2 + \lambda ||\beta||_1$$

The fact that L_1 norm is not differentiable implies that our objective function is also not differentaible, which means that we cannot find a closed form solution using the stationary point method. Furthermore all first and higher order optimization algorithms are not applicable because of this fact. For this homework we used Powell's method, which is an optimization method which does not require differentiability of the objective function. The algorithm requires an initial approximation of the solution. We opted to sample the weights and the intercept from 0 mean unit variance Laplace distribution, which is justified by the Bayesian interpretation of Lasso regression. The same standardization argument holds for Lasso regression.

III. Ridge regression for the superconductor dataset

This dataset contains information about different superconductors, and the goal is to predict the critical temperature of a superconductor based on other features. There are 300 rows and 28 columns, and it is justified to try Ridge regression in this case to reduce the chances of overfiting. As per homework instructions, first 200 rows were used for training, and remaining 100 rows were used for testing.

First, we **standardized** $X_{training}$ using column-wise means and standard deviations $\mu_{training}, \sigma_{training}$, and then we **scaled** X_{test} using $\mu_{training}$ and $\sigma_{training}$. Notice that we used the word scaled for X_{test} instead of standardized, because applying the transformation $\overline{X_{test}^{i,j}} = \frac{X_{test}^{i,j} - \mu_{training}^j}{\sigma_{training}^j}, i = \overline{1,N}, \overline{j=1p}$ does not in general lead to having 0 mean and unit variance columns of X_{test}, but this is the only correct way to scale the test data with respect to the training data. $\overline{X_{test}^{i,j}} = \frac{X_{test}^{i,j} - \mu_{test}^j}{\sigma_{test}^j}, i = \overline{1,N}, \overline{j=1p}$ would lead to having standardized columns of X_{test} but this would not be correct since our model was trained with $X_{training}$, and the transformation of the feature space was

done using $\mu_{training}, \sigma_{training}$. $y_{training}$ and y_{test} have been left untouched (centered version of $y_{training}$ is used during training and is discarded after that). An alternative way would be to standardize $y_{training}$ and scale y_{test} in the same manner as for X, which would lead to having $\beta_0 = 0$, but then one would have to re-scale the predictions of the model.

To infer the best regularization weight λ we used grid search with leave one out cross validation - **LOOCV**, using **RMSE** as the evaluation metric and restricting $0.1 \leq \lambda \leq 2$ with increments of 0.01. Results are shown on the figure below:

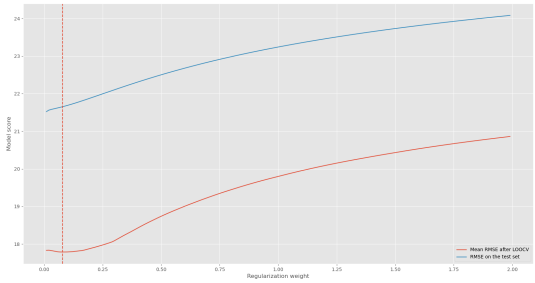

Fig. 1. In addition to LOOCV score, we also showed RMSE on the test set for a particular regularization weight.

Through grid searching, we obtained that $\lambda = 0.08$ was the best regularization weight, with LOOCV score being **17.79**, and RMSE on the test set for this regularization weight was **21.65**. Having domain knowledge about the response y, we can say that a RMSE of 21.65 on the test set is very high (we are predicting critical temperature of a superconductor), which suggests that Ridge regression might not be the most appropriate model for this problem. Alternatively, one could try using a kernelized variant of ridge regression (using a polynomial or a RBF kernel) but this was not demanded for this homework. These approaches would introduce more hyper-parameters (polynomial degree for a polynomial kernel and bandwith for a RBF kernel), which would substantially slow down grid searching even for a dataset of this size.