DS-GA.1013 Mathematical Tools for Data Science
Homework 5
Yves Greatti - yg390

1. (Augmented dataset) Ridge regression is equivalent to applying OLS on an expanded dataset that has additional examples. Describe these additional examples in detail. Intuitively, what effect do these additional examples have?

   The ridge regression estimate is defined as $\beta_{\text{RR}} = (XX^T + \lambda I)^{-1} X y$, where $X \in \mathbb{R}^{p \times n}, y \in \mathbb{R}^n, \beta \in \mathbb{R}^p$, which can be reformulated as a modified least-square problem

$$\beta_{\text{RR}} = \arg\min_{\beta} \| \begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X^T \\ \sqrt{\lambda} I_{n \times p} \end{bmatrix} \beta \|_2^2$$

   It is like we have added p vectors to the original n datapoints $[x_i, \ldots, x_n], x_i \in \mathbb{R}^p, i = 1, \ldots, n$. By doing so, Ridge regression embedded the original problem in $\mathbb{R}^n$ into a larger space $\mathbb{R}^{n+p}$ by moving into p different directions with a small amount $\sqrt{\lambda}$ which could decrease any collinearity present in the original data points X. As we have seen in the notes of linear regression, when the number of training data is small, λ neutralizes the large variance of the errors between the true β coefficients and the ridge regression coefficients due to small singular values of the sample covariance matrix.

2. (Correlated features) Consider a regression problem where the response only depends on one feature, but we don't know it, so we incorporate an additional feature into the model that happens to be very correlated with the first feature. More specifically, let $y \in \mathbb{R}^n$ be defined by

$$y := \beta_{\text{true}} w_1 + z, \tag{1}$$

where $\beta_{\text{true}} \in \mathbb{R}$ is the true coefficient, $w_1 \in \mathbb{R}^n$ is the first feature vector, and $z \in \mathbb{R}^n$ is additive noise. The second feature vector is given by $w_2 \in \mathbb{R}^n$ and can be decomposed into

$$w_2 = \alpha w_1 + \sqrt{1-\alpha^2} w_\perp, \tag{2}$$

where $w_\perp$ is orthogonal to w_1. The vectors w_1, w_2, $w_\perp$ and z all have unit ℓ_2 norm. In addition, we assume

$$w_1^T z = 0.1, \tag{3}$$
$$w_\perp^T z = 0.1. \tag{4}$$

We fit a linear regression model to y using the feature matrix

$$X = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix}. \tag{5}$$

(a) What does the OLS estimator of the coefficients β_{OLS} equal to when $\alpha \to 1$? Explain what is happening.
Hint: Use the fact that for any a, b, c, and d such that $ad \neq bc$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \tag{6}$$

The OLS estimator of the coefficients of β_{OLS} is given by $\beta_{\text{OLS}} = (XX^T)^{-1}Xy$. Expanding each term, we have:

$$
\begin{aligned}
(XX^T) &= \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} [w_1 \; w_2] \\
&= \begin{bmatrix} w_1^T w_1 & w_1^T w_2 \\ w_2^T w_1 & w_2^T w_2 \end{bmatrix} \\
&= \begin{bmatrix} \|w_1\|_2^2 & w_1^T w_2 \\ w_2^T w_1 & \|w_2\|_2^2 \end{bmatrix} \\
w_1^T w_2 &= w_1^T(\alpha w_1 + \sqrt{1-\alpha^2} w_\perp) \\
&= \alpha \|w_1\|_2^2 + \sqrt{1-\alpha^2} w_1^T w_\perp \\
&= \alpha \quad \text{since } \|w_1\|_2 = 1 \,, w_1 \perp w_\perp \\
w_2^T w_2 &= (\alpha w_1 + \sqrt{1-\alpha^2} w_\perp)^T (\alpha w_1 + \sqrt{1-\alpha^2} w_\perp) \\
&= \alpha^2 w_1^T w_1 + 2\alpha\sqrt{1-\alpha^2} w_1^T w_\perp + (1-\alpha^2) w_\perp^T w_\perp \\
&= \alpha^2 \|w_1\|_2^2 + (1-\alpha^2)\|w_\perp\|_2^2 \quad \text{knowing that } w_\perp \perp w_1 \\
&= \alpha^2 + (1-\alpha^2) = 1 \text{ by assumptions } \|w_1\|_2 = \|w_\perp\|_2 = 1 \\
\Rightarrow (XX^T) &= \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} \\
\Rightarrow (XX^T)^{-1} &= \frac{1}{1-\alpha^2} \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix} \\
Xy &= \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} (\beta_{\text{true}} w_1 + z) \\
&= \begin{bmatrix} \beta_{\text{true}} w_1^T w_1 + w_1^T z \\ \beta_{\text{true}} w_2^T w_1 + w_2^T z \end{bmatrix} \\
&= \begin{bmatrix} \beta_{\text{true}} + 0.1 \\ \alpha\beta_{\text{true}} + 0.1 \end{bmatrix}
\end{aligned}
$$

Substituting each of the previous terms back into the expression of β_{OLS}, we find that:

$$
\begin{aligned}
\beta_{\text{OLS}} &= \frac{1}{1-\alpha^2} \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix} \begin{bmatrix} \beta_{\text{true}} + 0.1 \\ \alpha\beta_{\text{true}} + 0.1 \end{bmatrix} \\
&= \frac{1}{1-\alpha^2} \begin{bmatrix} \beta_{\text{true}} + 0.1 - \alpha^2\beta_{\text{true}} - 0.1\alpha \\ -\alpha\beta_{\text{true}} - 0.1\alpha + \alpha\beta_{\text{true}} + 0.1 \end{bmatrix} \\
&= \frac{1}{1-\alpha^2} \begin{bmatrix} (1-\alpha^2)\beta_{\text{true}} + 0.1(1-\alpha) \\ 0.1(1-\alpha) \end{bmatrix} \\
&= \begin{bmatrix} \beta_{\text{true}} + \frac{0.1}{1+\alpha} \\ \frac{0.1}{1+\alpha} \end{bmatrix}
\end{aligned}
$$

When $\alpha \to 1$, $\beta_{\text{OLS}} \to \begin{bmatrix} \beta_{\text{true}} + 0.05 \\ 0.05 \end{bmatrix}$. Depending of the value of β_{true} compared to 0.05, the OLS estimator could consider only the feature w_1 and not the correlated feature w_2. It also adds a fixed constant 0.05 to the true β_{true} coefficient which could be significant compared to β_{true}.However this could be inferred and corrected during the validation step. Notice also that in this case XX^T is rank 1 and not invertible, and in some cases, the algorithm used to find the OLS estimator may be unable to find a solution.

(b) What does the corresponding estimate of the response $y_{\text{OLS}} := X^T\beta_{\text{OLS}}$ equal to when $\alpha \to 1$? Is it collinear with the true feature w_1 when $\alpha \to 1$? Explain what is happening.

Taking $\alpha \to 1$, $w_2 \to w_1$ and we have

$$\begin{aligned} y_{\text{OLS}} := X^T\beta_{\text{OLS}} &\to [w_1\ w_2] \begin{bmatrix} \beta_{\text{true}} + 0.05 \\ 0.05 \end{bmatrix} \\ &\to [w_1\ w_1] \begin{bmatrix} \beta_{\text{true}} + 0.05 \\ 0.05 \end{bmatrix} \\ &\to (\beta_{\text{true}} + 0.1)w_1 \end{aligned}$$

When $\alpha \to 1$, the correlated feature $w_2 \to w_1$, and the response variable is collinear with the true feature w_1, the OLS estimator estimates y_{OLS} as a linear scaling of w_1 up to a factor of 0.1 which could lead to an important error depending the magnitude of the linear coefficient β_{true} compared to 0.1.

(c) What does the ridge regression estimator of the coefficients β_{RR} equal to when $\alpha \to 1$ and the regularization parameter $\lambda > 0$ is fixed? Describe the difference with the OLS estimate.

By definition the ridge regression estimator of the coefficients β_{RR} is:

$$\begin{aligned} \beta_{\text{RR}} &= (XX^T + \lambda I)^{-1}Xy,\ \lambda > 0 \\ XX^T + \lambda I &= \begin{bmatrix} 1+\lambda & \alpha \\ \alpha & 1+\lambda \end{bmatrix} \\ (XX^T + \lambda I)^{-1} &= \frac{1}{(1+\lambda)^2 - \alpha^2} \begin{bmatrix} 1+\lambda & -\alpha \\ -\alpha & 1+\lambda \end{bmatrix} \\ \Rightarrow \beta_{\text{RR}} &= \frac{1}{(1+\lambda)^2 - \alpha^2} \begin{bmatrix} 1+\lambda & -\alpha \\ -\alpha & 1+\lambda \end{bmatrix} \begin{bmatrix} \beta_{\text{true}} + 0.1 \\ \alpha\beta_{\text{true}} + 0.1 \end{bmatrix} \\ &= \frac{1}{(1+\lambda)^2 - \alpha^2} \begin{bmatrix} (1+\lambda)(\beta_{\text{true}} + 0.1) - \alpha^2\beta_{\text{true}} - 0.1\alpha \\ -\alpha\beta_{\text{true}} - 0.1\alpha + \alpha(1+\lambda)\beta_{\text{true}} + 0.1(1+\lambda) \end{bmatrix} \\ &= \frac{1}{(1+\lambda)^2 - \alpha^2} \begin{bmatrix} (1+\lambda-\alpha^2)\beta_{\text{true}} + 0.1(1+\lambda-\alpha) \\ \alpha\lambda\beta_{\text{true}} + 0.1(1+\lambda-\alpha) \end{bmatrix} \\ &= \begin{bmatrix} \frac{(1+\lambda-\alpha^2)\beta_{\text{true}}}{(1+\lambda)^2-\alpha^2} + \frac{0.1}{1+\lambda+\alpha} \\ \frac{\lambda\alpha\beta_{\text{true}}}{(1+\lambda)^2-\alpha^2} + \frac{0.1}{1+\lambda+\alpha} \end{bmatrix} \\ {}_{\alpha\to 1} &\to \frac{\beta_{\text{true}} + 0.1}{2+\lambda} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned}$$

If we compare to β_{OLS} estimator, the ridge regression estimator has coefficients on both w_1 and w_2 and this coefficient can be regularized using λ. When the data gives no reason to choose between different linear combinations of colinear features (we did not that know they are correlated) , ridge estimator chooses equal weighting. Also even if XX^T is singular (rank 1), the L_2 regularization using λ makes the matrix XX^T non singular and we are still able to find a solution which includes both features. In addition we have a new set of coefficients each time we tune λ: when $\lambda \to 0, \beta_{\text{RR}} \to (\frac{\beta_{\text{OLS}}}{2} + 0.05)[1\ 1]^T$ and when $\lambda \to \infty, \beta_{\text{RR}} \to [0\ 0]^T$.

(d) What does the corresponding estimate of the response $y_{\text{RR}} := X^T\beta_{\text{RR}}$ equal to when $\alpha \to 1$? Is it collinear with the true feature w_1?
When $\alpha \to 1, w_2 \to w_1$, which leads to

$$\begin{aligned} y_{\text{RR}} := X^T\beta_{\text{RR}} &\to \frac{\beta_{\text{true}} + 0.1}{2+\lambda}[w_1 w_2]\begin{bmatrix}1\\1\end{bmatrix} \\ &\to \frac{\beta_{\text{true}} + 0.1}{2+\lambda}[w_1 w_1]\begin{bmatrix}1\\1\end{bmatrix} \\ &\to 2\,\frac{\beta_{\text{true}} + 0.1}{2+\lambda}\, w_1 \end{aligned}$$

y_{RR} is collinear with the true feature w_1. Compared to y_{OLS}, the parameter λ allows to control the amount of linearity between the response and control variable (when $\lambda = 0, y_{\text{RR}} = y_{\text{OLS}}$), which might be desirable if the test data points, not known in advance, are not totally dependent on the feature w_1.

3. (Prior knowledge) Consider a linear regression problem where we have prior information indicating that the coefficients should be close to a certain value β_{prior}.

   (a) How can you incorporate this prior knowledge if you are using ridge regression? Write the corresponding optimization problem.

   If we want to include that the coefficients should be close to β_{prior}, the ridge-regression estimator is the minimizer of the optimization problem:

$$\beta_{\text{RR}} := \arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta - \beta_{\text{prior}}\|_2^2$$

   where $\lambda > 0$ is a fixed regularization parameter.

   (b) Assume that the data are generated according to a linear model $\tilde{y} := X^T\beta_{\text{true}} + \tilde{z}$, where $\beta_{true} \in \mathbb{R}^p$ and $X \in \mathbb{R}^{p\times n}$ are fixed and $\tilde{z}$ is an iid Gaussian random vector with zero mean and variance σ^2. Does the modification change the mean or the covariance matrix of the estimator? If so, report the new value.

$$\begin{aligned}
\|y - X^T\beta\|_2^2 &= (y - X^T\beta)^T(y - X^T\beta) + \lambda(\beta - \beta_{\text{prior}})^T(\beta - \beta_{\text{prior}}) \\
&= (y^T - \beta^T X)(y - X^T\beta) + \lambda(\beta^T - \beta_{\text{prior}}^T)(\beta - \beta_{\text{prior}}) \\
&= \beta^T XX^T\beta - 2(Xy)^T\beta + y^Ty + \lambda\big(\beta^T\beta - 2\beta_{\text{prior}}^T\beta + \beta_{\text{prior}}^T\beta_{\text{prior}}\big) \\
&= \beta^T XX^T\beta - 2(Xy)^T\beta + \lambda\big(\beta^T\beta - 2\beta_{\text{prior}}^T\beta\big) + y^Ty + \lambda\beta_{\text{prior}}^T\beta_{\text{prior}} := f(\beta)
\end{aligned}$$

   f is a quadratic form in β and its gradient and Hessian equal:

$$\begin{aligned}
\nabla_\beta f(\beta) &= 2(XX^T\beta - Xy + \lambda(\beta - \beta_{\text{prior}})) \\
\nabla_\beta^2 f(\beta) &= 2(XX^T + \lambda I)
\end{aligned}$$

   $v^T(XX^T + \lambda I)v = v^TXX^Tv + \lambda v^Tv = \|X^Tv\|_2^2 + \lambda\|v\|_2^2 \geq 0$ and it is zero only when v is the zero vector (by property of the norm operator), thus the matrix $XX^T + \lambda I$ is positive definite and invertible. The unique minimum can be found by setting the gradient to zero, which gives that the ridge regression estimator is:

$$\beta_{\text{RR}} = (XX^T + \lambda I)^{-1}(Xy + \lambda\beta_{\text{prior}})$$

   with mean:

$$\begin{aligned}
\text{E}[\beta_{\text{RR}}] &= \text{E}[(XX^T + \lambda I)^{-1}(XX^T\beta_{\text{true}} + X\tilde{z} + \lambda\beta_{\text{prior}})] \\
&= (XX^T + \lambda I)^{-1}(XX^T\beta_{\text{true}} + \lambda\beta_{\text{prior}}) \\
&\quad \text{by linearity of the expectation and } \tilde{z} \text{ has zero mean}
\end{aligned}$$

Let $X = USV^T$ the SVD of X, if X is full rank, and $p \leq n$, U is square, $UU^T = U^TU = I$,we can then expand the previous expression:

$$\begin{aligned}
\mathrm{E}[\beta_{\mathrm{RR}}] &= (US^2U^T + \lambda U\ U^T)^{-1}(US^2U^T\beta_{\mathrm{true}} + \lambda\beta_{\mathrm{prior}}) \\
&= U(S^2 + \lambda I)^{-1}S^2U^T\beta_{\mathrm{true}} + \lambda\beta_{\mathrm{prior}}U(S^2 + \lambda I)^{-1}U^T \\
&= \frac{S^2\beta_{\mathrm{true}} + \lambda\beta_{\mathrm{prior}}}{(S^2 + \lambda I)^{-1}}
\end{aligned}$$

and variance:

$$\begin{aligned}
\mathrm{Var}(\beta_{\mathrm{RR}}) &= \mathrm{Var}((XX^T + \lambda I)^{-1}(Xy + \lambda\beta_{\mathrm{prior}})) \\
&= \mathrm{Var}((XX^T + \lambda I)^{-1}Xy + \lambda\beta_{\mathrm{prior}}(XX^T + \lambda I)^{-1}) \\
&= \mathrm{Var}((XX^T + \lambda I)^{-1}Xy) \\
&= (XX^T + \lambda I)^{-1}X\mathrm{Var}(y)((XX^T + \lambda I)^{-1}X)^T \\
&= \sigma^2(XX^T + \lambda I)^{-1}XX^T(XX^T + \lambda I)^{-1} \\
&= \sigma^2(S^2 + \lambda I)^{-1}US^2U^T(S^2 + \lambda I)^{-1} \\
&= \sigma^2\mathrm{diag}_{i=1}^{p}\left(\frac{s_i^2}{(s_i^2 + \lambda)^2}\right)
\end{aligned}$$

Note that when $\lambda = 0$ then $\mathrm{E}[\beta_{\mathrm{RR}}] = \beta_{\mathrm{true}}$ and $\mathrm{Var}(\beta_{\mathrm{RR}}) = \sigma^2 I$, and when $\lambda \to \infty$, $\mathrm{E}[\beta_{\mathrm{RR}}] \to \beta_{\mathrm{prior}}$ and $\mathrm{Var}(\beta_{\mathrm{RR}}) \to 0_{p\times p}$.

(c) How can you incorporate this prior knowledge if you are using gradient descent with early stopping? Write the corresponding update equation as a function of β_{prior}.

For $X \in \mathbb{R}^{p\times n}$ and a response vector $y \in \mathbb{R}^n, \beta \in \mathbb{R}^p$, from part (b) we know that the gradient equals

$$\nabla_\beta f(\beta) = 2(XX^T\beta - Xy + \lambda(\beta - \beta_{\mathrm{prior}}))$$

The gradient-descent updates are:

$$\begin{aligned}
\beta^{(k+1)} &= \beta^{(k)} - \ \alpha_k\nabla_\beta f(\beta^{(k)}) \\
&= \beta^{(k)} - \alpha_k(XX^T\beta^{(k)} - Xy + \lambda(\beta^{(k)} - \beta_{\mathrm{prior}})) \\
&= ((1 - \lambda\alpha_k)I - \alpha_k XX^T)\beta^{(k)} + \alpha_k Xy + \lambda\alpha_k\beta_{\mathrm{prior}} \\
&= \beta^{(k)} + \alpha_k\big(\sum_{i=1}^{n}(y[i] - \langle x_i, \beta^{(k)}\rangle)x_i - \lambda(\beta^{(k)} - \beta_{\mathrm{prior}})\big)
\end{aligned}$$

where $\beta^{(k)} \in \mathbb{R}^p$ and $\alpha_k > 0$ which are the estimated step size at iteration k. Note that for the term corresponding to α_k, at step k, for the next $\beta^{(k+1)}$ we want to reduce the error with the response variable y and at the same time to be close to the prior β_{prior}, the last term being controlled by the regularization parameter λ.

(d) Assume that the data are generated according to the linear model described above. Does the modification change the mean or the covariance matrix of the estimator? If so, report the new value.

Assuming constant step size, we can express the previous expression as:

$$\begin{aligned}\beta^{(k+1)} &= \big((1-\lambda\alpha)I - \alpha XX^T\big)\beta^{(k)} + \alpha(Xy + \lambda\beta_{\text{prior}})\\ &= \big(I - \alpha(\lambda I + XX^T)\big)^{k+1}\beta^{(0)} + \sum_{i=0}^{k}\big(I - \alpha(\lambda I + XX^T)\big)^i \alpha(Xy + \lambda\beta_{\text{prior}})\end{aligned}$$

Assuming X full rank and p $\leq$ n, U is square and, $UU^T = U^TU = I$, let $X = USV^T$, we have then:

$$\begin{aligned}\beta^{(k+1)} &= U(I - \alpha(S^2 + \lambda I))^{k+1}U^T\beta^{(0)} + \alpha\sum_{i=0}^{k} U(I - \alpha(S^2+\lambda I))^i U^T(USV^Ty + \lambda\beta_{\text{prior}})\\ &= U(I - \alpha(S^2 + \lambda I))^{k+1}U^T\beta^{(0)} + \alpha\sum_{i=0}^{k} U(I - \alpha(S^2+\lambda I))^i (SV^Ty + \lambda\beta_{\text{prior}}U^T)\\ &= U\text{diag}_{j=1}^p\left((1-\alpha(s_j^2+\lambda))^{k+1}\right)U^T\beta^{(0)}\\ &+ \alpha\, U\ \text{diag}_{j=1}^p\left(\sum_{i=0}^{k}(1-\alpha(s_j^2+\lambda))^i\right)(SV^Ty + \lambda\beta_{\text{prior}}U^T)\\ &= U\text{diag}_{j=1}^p\left((1-\alpha(s_j^2+\lambda))^{k+1}\right)U^T\beta^{(0)}\\ &+ U\ \text{diag}_{j=1}^p\left(\frac{1-(1-\alpha(s_j^2+\lambda))^{k+1}}{s_j^2+\lambda}\right)(SV^Ty + \lambda\beta_{\text{prior}}U^T)\end{aligned}$$

If step size α is small enough $0 < \alpha < \frac{2}{\lambda + s_1^2} \leq \frac{2}{\lambda + s_j^2} \to |1 - \alpha(s_j^2+\lambda)| < 1 \to \lim_{k\to\infty}(1-\alpha(s_j^2+\lambda))^k = 0, j = 1,\dots,p$ then gradient descent converges to:

$$\lim_{k\to\infty}\beta^{(k+1)} = U\ \text{diag}_{j=1}^p\left(\frac{1}{s_j^2+\lambda}\right)(SV^Ty + \lambda\beta_{\text{prior}}U^T)$$

Let $\nu_j := 1 - \alpha(s_j^2+\lambda)$, then we now have

$$\begin{aligned}\tilde{\beta}^{(k)} &= U\ \text{diag}_{j=1}^p\left(\frac{1-\nu_j^k}{s_j^2+\lambda}\right)(SV^TX^T\beta_{\text{true}} + SV^T\tilde{z} + \lambda\beta_{\text{prior}}U^T)\\ &= U\ \text{diag}_{j=1}^p\left(\frac{1-\nu_j^k}{s_j^2+\lambda}\right)(S^2U^T\beta_{\text{true}} + \lambda\beta_{\text{prior}}U^T + SV^T\tilde{z})\\ &= U\ \text{diag}_{j=1}^p\left(\frac{s_j^2(1-\nu_j^k)}{s_j^2+\lambda}\right)U^T\beta_{\text{true}} + \lambda U\ \text{diag}_{j=1}^p\left(\frac{1-\nu_j^k}{s_j^2+\lambda}\right)\beta_{\text{prior}}U^T\\ &+ U\ \text{diag}_{j=1}^p\left(\frac{s_j(1-\nu_j^k)}{s_j^2+\lambda}\right)V^T\tilde{z}\end{aligned}$$

Then using Theorem 8.6 in the PCA lecture notes, we have a Gaussian random vector with mean:

$$\begin{aligned}\beta_{\text{GD}} &= \sum_{i=1}^{p} \frac{1-\nu_j^k}{s_i^2+\lambda} \left\langle u_i, s_i^2\beta_{\text{true}} + \lambda\beta_{\text{prior}} \right\rangle u_i \\ &= \sum_{i=1}^{p} \frac{1-(1-\alpha(s_i^2+\lambda))^k}{s_i^2+\lambda} \left\langle u_i, s_i^2\beta_{\text{true}} + \lambda\beta_{\text{prior}} \right\rangle u_i\end{aligned}$$

and variance:

$$\begin{aligned}\Sigma_{\text{GD}} &= \sigma^2 U \text{diag}_{j=1}^{p}\left(\frac{s_j(1-\nu_j^k)}{s_j^2+\lambda}\right) V^T V \text{diag}_{j=1}^{p}\left(\frac{s_j(1-\nu_j^k)}{s_j^2+\lambda}\right) U^T \\ &= \sigma^2 U \text{diag}_{j=1}^{p}\left(\frac{s_j^2(1-(1-\alpha(s_i^2+\lambda))^k)^2}{(s_j^2+\lambda)^2}\right) U^T\end{aligned}$$

Note that we find the same limiting behaviors with the regularization parameter λ (and number of iterations k), that we found for the ridge regression estimator in part (b).

4. The code you will implement in this question is located in the `regress.py` file in the time folder of `hw5.zip`. Define a sequence of random variables as follows:

$$\begin{aligned}\vec{\tilde{x}}[0] &= 1 \\ \vec{\tilde{x}}[1] &= \vec{\tilde{x}}[0] + \vec{\tilde{z}}[1] \\ \vec{\tilde{x}}[2] &= \vec{\tilde{x}}[1] + \vec{\tilde{z}}[2] \\ \vec{\tilde{x}}[3] &= \vec{\tilde{x}}[2] + \vec{\tilde{z}}[3] \\ \vec{\tilde{x}}[4] &= \vec{\tilde{x}}[3] + \vec{\tilde{z}}[4],\end{aligned}$$

where $\vec{\tilde{z}}[1], \vec{\tilde{z}}[2], \vec{\tilde{z}}[3], \vec{\tilde{z}}[4]$ are independent, $\vec{\tilde{z}}[1] \sim \mathcal{N}(0,1)$ and $\vec{\tilde{z}}[2], \vec{\tilde{z}}[3], \vec{\tilde{z}}[4] \sim \mathcal{N}(0, 0.01^2)$. There is a function $f : \mathbb{R}^5 \to \mathbb{R}$ of the form $f(x) = \vec{\beta}^T x$ where $\vec{\beta}$ is unknown. We are given a training sample of independent draws

$$(\vec{\tilde{x}}_1, f(\vec{\tilde{x}}_1) + \tilde{w}_1), \ldots, (\vec{\tilde{x}}_n, f(\vec{\tilde{x}}_n) + \tilde{w}_n) \in \mathbb{R}^5 \times \mathbb{R},$$

where $\tilde{w}_i$ are iid standard normal random variables corrupting our measurements of f. Using this training data, we will estimate $\vec{\beta}$ and test our performance on a validation set drawn from the same distribution. Below we refer to the square loss function $L : \mathbb{R}^5 \times \mathbb{R}^{n\times 5} \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$L(\hat{\beta}, X, y) = \sum_{i=1}^{n} (X[i,:]\hat{\beta} - y[i])^2$$

where $X \in \mathbb{R}^{n\times 5}$ denotes a matrix of data (training or validation; each row is a data point), and y is the corresponding vector of f-values.

(a) Using least squares (i.e., minimizing the square loss on the training set) compute an estimate for $\vec{\beta}$. Include your estimate for $\vec{\beta}$, your square loss on the training set, and your square loss on the validation set in your submission. [Hint: If computed correctly your training loss should be larger than 30 and your validation loss should be larger than 10.]

(b) Compute the singular values of the training data matrix $X \in \mathbb{R}^{n\times 5}$.

(c) The true value of $\vec{\beta}$ can be found at the top of `regress.py`. Give an explanation as to why the least squares estimates aren't close to the true $\vec{\beta}$-values.

(d) Use ridge regression to produce a new estimate of $\vec{\beta}$ and report the resulting estimate of $\vec{\beta}$, and your square loss on the training and validation sets. Here $\hat{\beta}$ should solve

$$\text{minimize}_{\vec{\eta}} \quad \|X\vec{\eta} - \vec{y}\|_2^2 + 0.5\|\vec{\eta}\|_2^2.$$

You're not required to include your code in your submission, but you are free to do so.