Project 2: Penalized Regression

Isaiah Chen

March 25, 2019

Question 1

We consider a multivariate version of ridge regression where $X : \Omega \to \mathbb{R}^d$, $Y : \Omega \to \mathbb{R}^q$, $\beta_0 \in \mathbb{R}^q$, and b is a $d \times q$ matrix, with predictor $f : \mathbb{R}^d \to \mathbb{R}^q$ given by:

$$f(x) = \beta_0 + b^T X$$

Introducing a $d \times d$ positive semi-definite symmetric matrix D, we consider penalized least-square estimators $\hat{\beta}_0$ and $\hat{b}$ given by minimizers of:

$$F(\beta_0, b) = \sum_{k=1}^{N} |y_k - \beta_0 - b^T x_k|^2 + \lambda \text{ trace}(b^T D b)$$

where (x_1, y_1),...,(x_N, y_N) are the training data.

(1.1) Let $\bar{x}$ and $\bar{y}$ denote the average of x_1,...,x_N and y_1,...,y_N respectively, and denote by $\mathcal{Y}_c$ and $\mathcal{X}_c$ the matrices:

$$\mathcal{Y}_c = \begin{pmatrix} (y_1 - \bar{y})^T \\ \vdots \\ (y_N - \bar{y})^T \end{pmatrix}, \mathcal{X}_c = \begin{pmatrix} (x_1 - \bar{x})^T \\ \vdots \\ (x_N - \bar{x})^T \end{pmatrix}$$

To find $\hat{b}$ and $\hat{\beta}_0$ such that $F(\beta_0, b)$ is minimized, we take the partial derivative of F with respect to β_0 and set it equal to 0 as such:

$$0 = -2 \sum_{k=1}^{N} |y_k - \hat{\beta}_0 - \hat{b}^T x_k|$$

We can solve this equation for $\hat{\beta}_0$ as follows:

$$0 = \sum_{k=1}^{N} y_k - N\hat{\beta}_0 - \sum_{k=1}^{N} \hat{b}^T x_k$$

$$
\begin{gathered}
0 = N\hat{\beta}_0 - \sum_{k=1}^{N}(y_k - \hat{b}^T x_k) \\
N\hat{\beta}_0 = \sum_{k=1}^{N}(y_k - \hat{b}^T x_k) \\
\hat{\beta}_0 = \tfrac{1}{N}\sum_{k=1}^{N}(y_k - \hat{b}^T x_k) \\
\hat{\beta}_0 = \tfrac{1}{N}\sum_{k=1}^{N} y_k - \hat{b}^T \tfrac{1}{N}\sum_{k=1}^{N} x_k \\
\hat{\beta}_0 = \bar{y} - \hat{b}^T \bar{x}
\end{gathered}
$$

Plug $\hat{\beta}_0$ into the original function to be minimized:

$$\sum_{k=1}^{N} |(y_k - \bar{y}) - (x_k - \bar{x})\hat{b}^T|^2 + \lambda \operatorname{trace}(\hat{b}^T D \hat{b})$$

Plug in $\mathcal{X}_c$ and $\mathcal{Y}_c$:

$$|\mathcal{Y}_c - \mathcal{X}_c \hat{b}^T|^2 + \lambda \operatorname{trace}(\hat{b}^T D \hat{b})$$

Take the derivative of this expression with respect to $\hat{b}$, set it equal to 0, and solve for $\hat{b}$:

$$
\begin{gathered}
0 = -2\mathcal{X}_c^T(\mathcal{Y}_c - \hat{b}\mathcal{X}_c) + 2\lambda D\hat{b} \\
0 = \lambda D\hat{b} - \mathcal{X}_c^T(\mathcal{Y}_c - \hat{b}\mathcal{X}_c) \\
0 = \lambda D\hat{b} - \mathcal{X}_c^T\mathcal{Y}_c + \hat{b}\mathcal{X}_c^T\mathcal{X}_c \\
\hat{b}(\mathcal{X}_c^T\mathcal{X}_c + \lambda D) = \mathcal{X}_c^T\mathcal{Y}_c \\
\hat{b} = (\mathcal{X}_c^T\mathcal{X}_c + \lambda D)^{-1}\mathcal{X}_c^T\mathcal{Y}_c
\end{gathered}
$$

(1.2) This multivariate problem can be seen as solving independently q separate univariate linear problems. The objective of the regression is to use the predictor $f : \mathbb{R}^d \to \mathbb{R}^q$ to predict values for $Y(\Omega \to \mathbb{R}^q)$. The dimensions of the X, b, and β_0 matrices are $d \times 1, d \times q$, and $q \times 1$, respectively. Consequently, the quantity $b^T X$ will be a $q \times 1$ matrix and the predictor will also result in a $q \times 1$ matrix, which is consistent with the dimension of Y. Each dimension in q (each row of the final matrices, in this case) can be reduced to its own individual linear problem to be solved:

$$f(x_i) = \beta_0 + b^T x_i \;\; \forall i \in q$$

(1.3) A program is used to fit a multivariate regression model on the provided set of training data for values of λ ranging from 0.01 to 2.00. The prediction error is evaluated on the provided set of testing data. The error as a function of λ is shown in Figure 1.

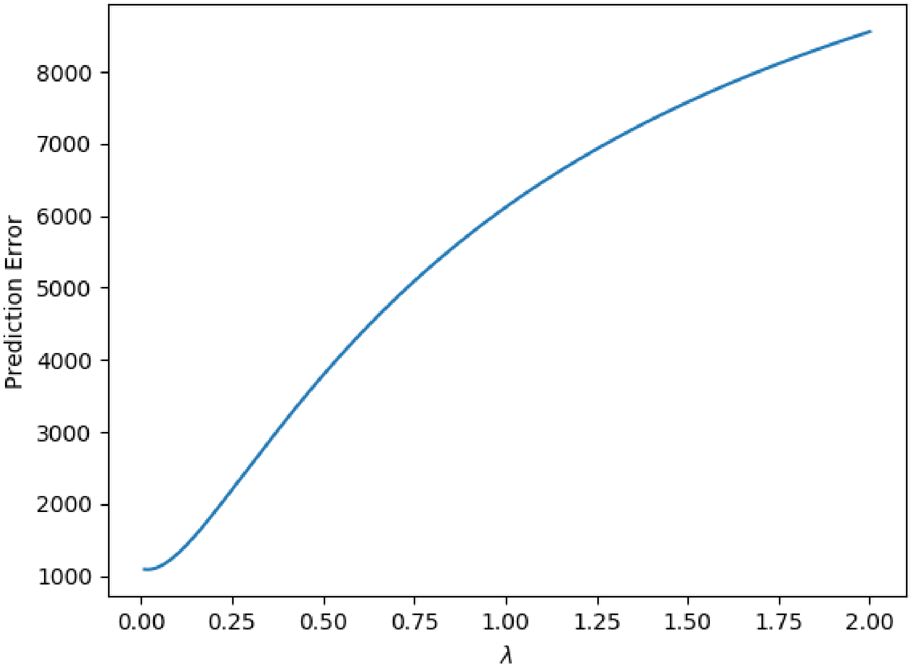

Figure 1: Prediction error as a function of λ for question (1.3)

For $\lambda = 1$, the estimated values of β_0 and b are:

$$\beta_0 = \begin{pmatrix} -0.1714 \\ -0.3714 \\ 0.1460 \\ -0.1330 \end{pmatrix}$$

$$b = \begin{pmatrix} 0.3226 & 0.7312 & -0.0157 & 0.0241 \\ 0.4002 & -0.0274 & 0.0560 & 0.0945 \\ 0.0332 & 0.0519 & 0.3572 & 0.7226 \\ -0.0685 & -0.0826 & 0.3265 & -0.0234 \end{pmatrix}$$

(1.4) Now, if we want to minimize a slightly modified version of the original function:

$$F(\beta_0, b) = \sum_{k=1}^{N} |y_k - \beta_0 - b^T x_k|^2 + \lambda \,\mathrm{trace}(\beta D \beta^T)$$

where D is now a positive semi-definite $q \times q$ symmetric matrix and $\beta = (\beta_0, b^T)^T$. We can define the matrices $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathcal{X} = \begin{pmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_N^T \end{pmatrix}, \mathcal{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \text{ where } \tilde{x} = \begin{pmatrix} 1 \\ x(1) \\ \vdots \\ x(d) \end{pmatrix}$$

Take the partial derivative of $F(\beta_0, b)$ with respect to β_0, set it equal to 0 and plug in variables:

$$0 = -2\mathcal{X}^T(\mathcal{Y} - \mathcal{X}b) + 2\lambda bD$$
$$0 = -\mathcal{X}^T(\mathcal{Y} - \mathcal{X}b) + \lambda bD$$
$$0 = \mathcal{X}^T\mathcal{X}b + \lambda bD - \mathcal{X}^T\mathcal{Y}$$

The optimal $\hat{\beta}$ satisfies the "Sylvester Equation":

$$\mathcal{X}^T\mathcal{X}b + \lambda bD = \mathcal{X}^T\mathcal{Y}$$

(1.5) The program used to return optimal parameters for β_0 and b for the multivariate problem in question (1.4) is tested using the given dataset for two cases:

1. $\lambda = 10$, $D = \text{Id}$
2. $\lambda = 1000$, D is a tridiagonal matrix with -1 above and below the diagonal, and 2 on the diagonal, except $D(1,\,1) = D(q, q) = 1$.

The values of $\beta_0(j), b(1, j), b(2, j)$, and $b(3, j)$ as functions of j for cases 1 and 2 are shown below in Figures 2 and 3, respectively. While the overall trends for each vector do not change significantly for different values of λ and D, the amount of noise between the two cases tends to change slightly.

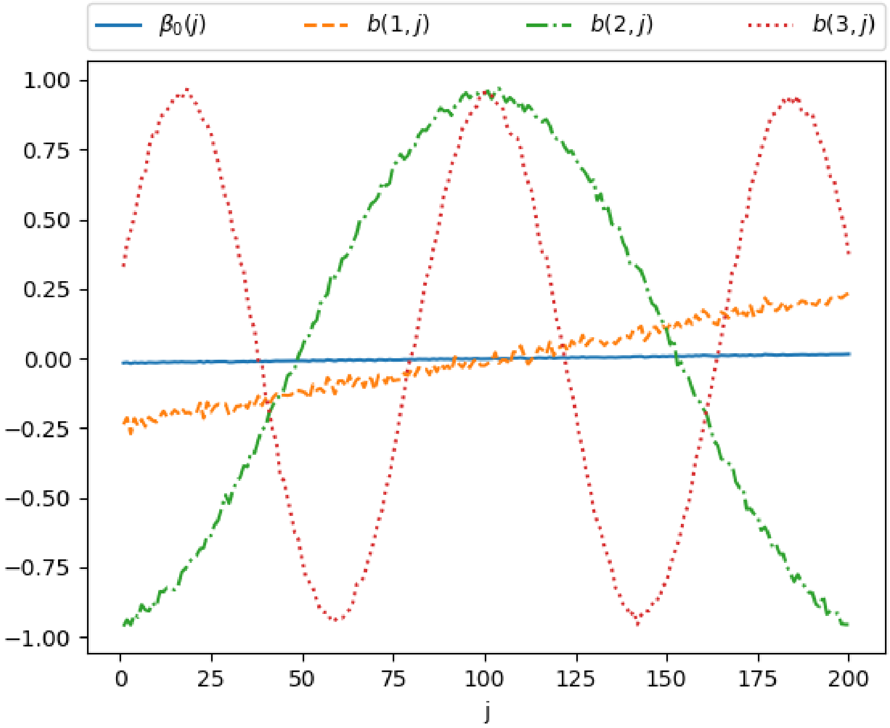

Figure 2: Optimal parameters as a function of j for case 1

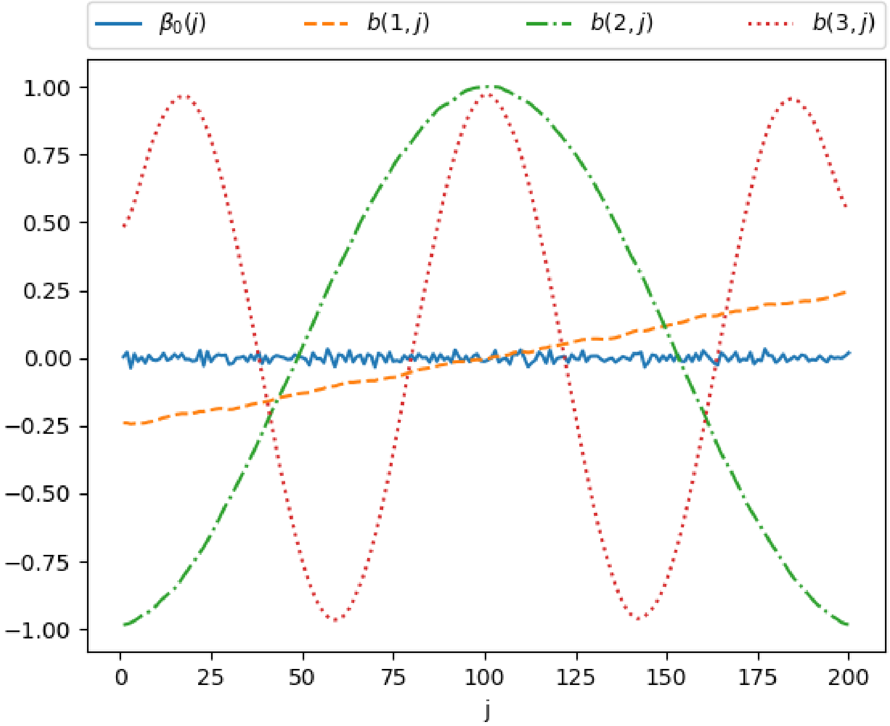

Figure 3: Optimal parameters as a function of j for case 2

Question 2

(2.1) The ridge regression program written for question (1.3) is used to evaluate regression parameters for $\lambda = 1, 2, ..., 100$. The prediction error computed on the test set is shown in Figure 4. The minimum value of error is approximately 209833 and is obtained when $\lambda = 2$

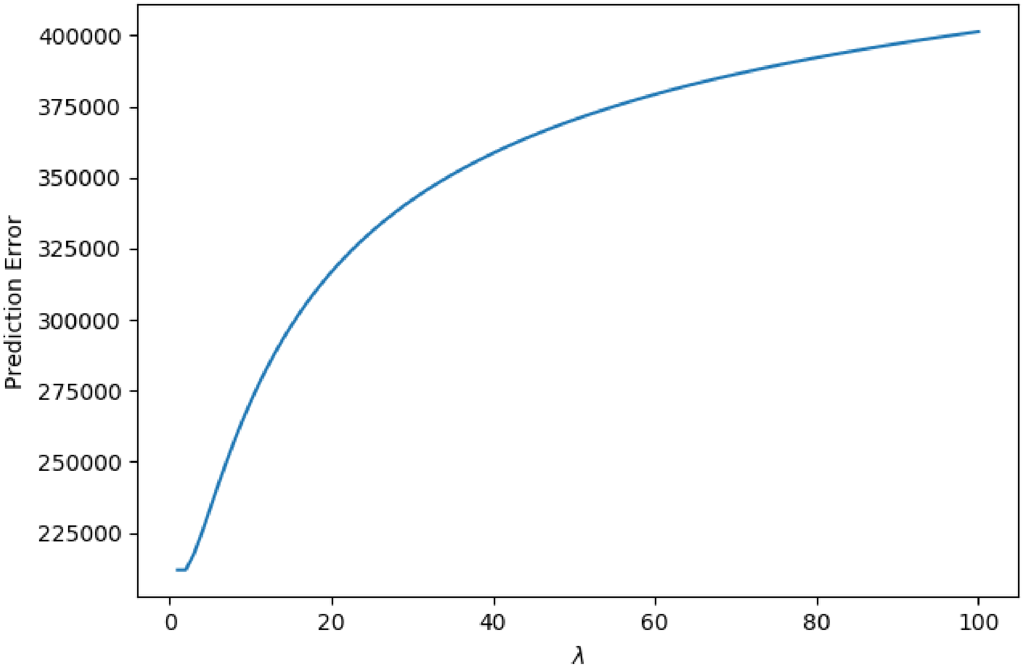

Figure 4: Prediction error as a function of λ for question (2.1)

(2.2) The program used to perform the kernel version of ridge regression is done for two cases:

1. Gaussian kernel: $K(x, y) = \exp(-|x - y|^2/2\sigma^2)$
2. Polynomial kernel of order h: $K(x, y) = \sum_{k=1}^{h} (x^T y)^k$

The prediction error for both cases is calculated using the following formula:

$$\sum_{l=1}^{M} |y_l - (\beta_0 + \sum_{k=1}^{N} \alpha_k K(x_l, x_k))|^2$$

For the Gaussian kernel case and the polynomial kernel case with orders ranging from 1 to 4, the minimum values of the prediction error and the corresponding λ values for are shown below in Table 1. For cases where the minimum error occurs over a range of values for λ, the range is listed below. Figures 5 - 9 show the prediction error as a function of λ for all 5 cases.

Case	λ	Error
Gaussian	0.0025 - 0.1	495260
Polynomial ($h = 1$)	1 - 100	490324
Polynomial ($h = 2$)	100	509628
Polynomial ($h = 3$)	100	913104
Polynomial ($h = 4$)	100	2800327

Table 1: Estimates of prediction error and corresponding values of λ

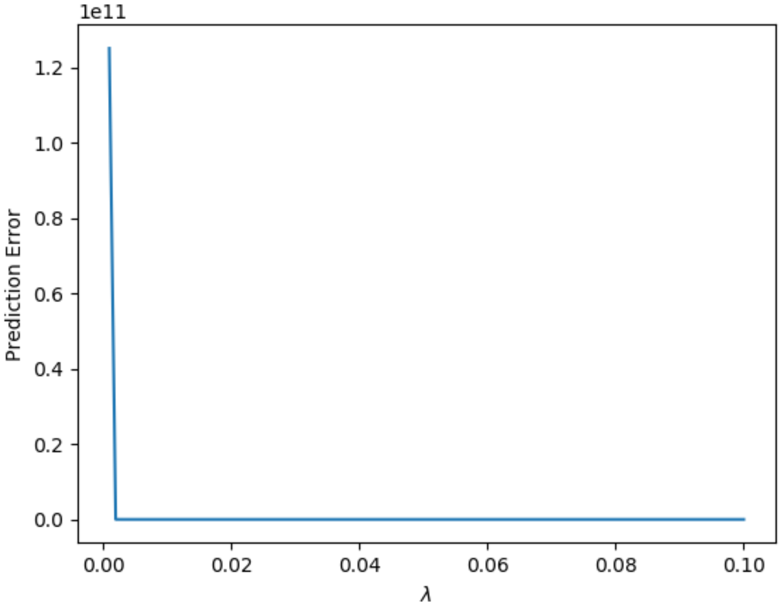

Figure 5: Prediction error as a function of λ for question (2.2) for the Gaussian kernel case

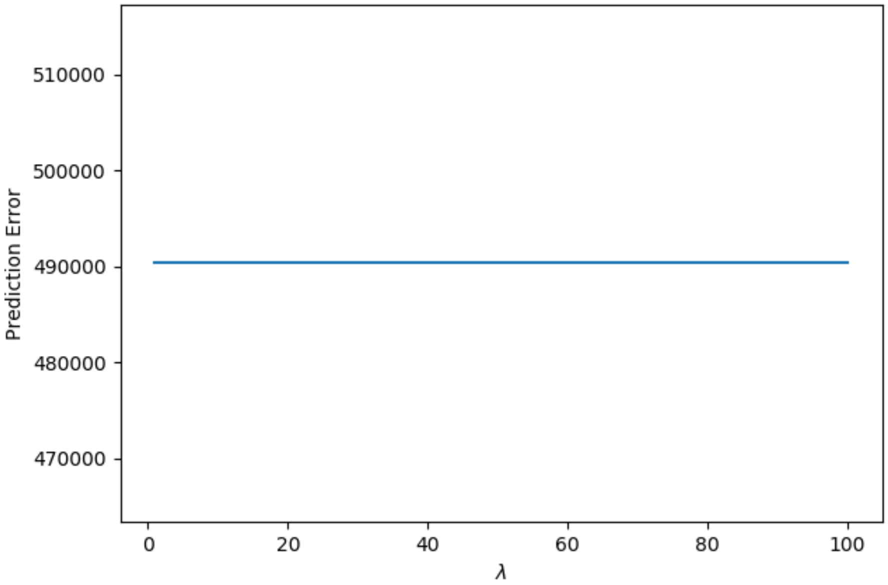

Figure 6: Prediction error as a function of λ for question (2.2) for the polynomial kernel case where $h = 1$

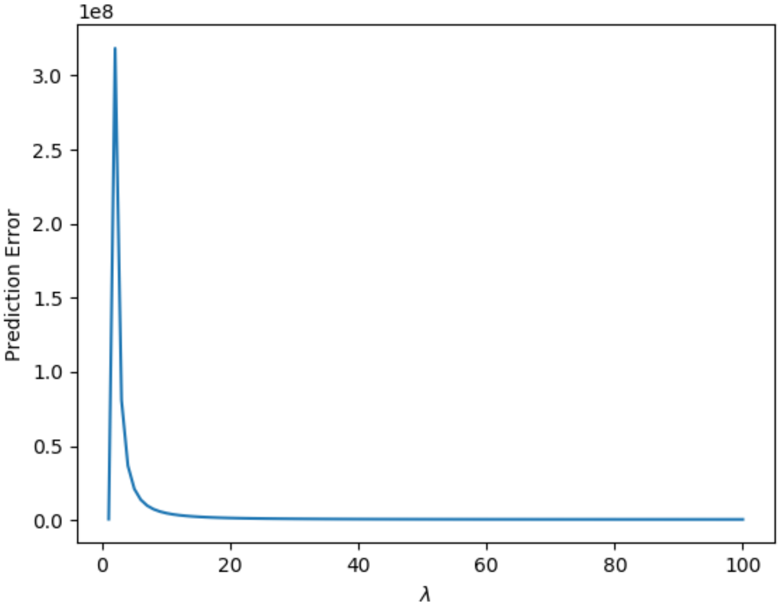

Figure 7: Prediction error as a function of λ for question (2.2) for the polynomial kernel case where $h = 2$

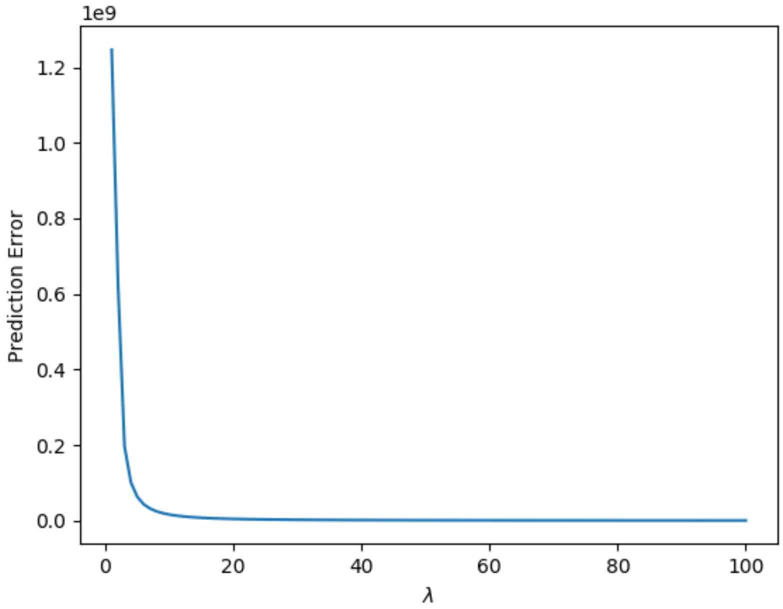

Figure 8: Prediction error as a function of λ for question (2.2) for the polynomial kernel case where $h = 3$

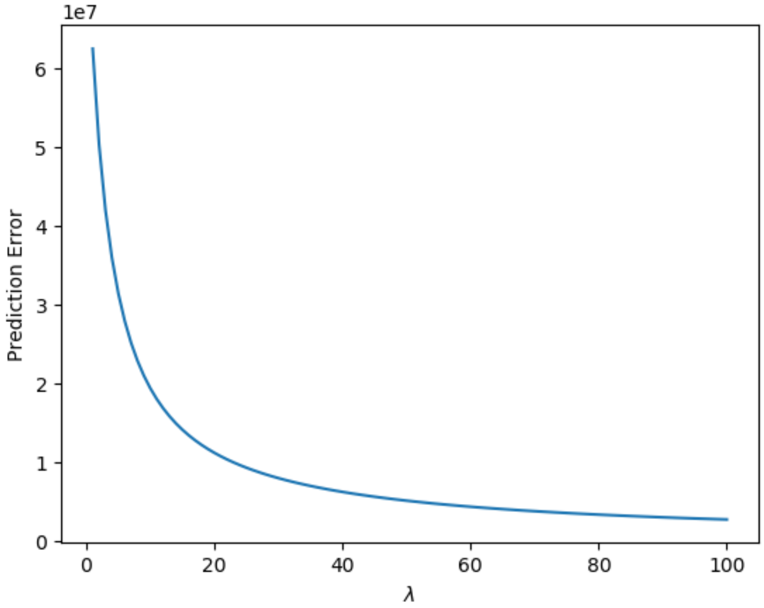

Figure 9: Prediction error as a function of λ for question (2.2) for the polynomial kernel case where $h = 4$

(2.3) The results from the polynomial kernel case ($h = 2$) should be identical to the linear one. The kernel for the second-degree polynomial is equivalent to the original expression as shown in the original error function to be minimized. In the actual results from the previous problem, the results do not match due to programming errors.