Statistical Machine Learning

Aakarsh Nair
aakarsh.nair@student.uni-tuebingen.de
Matriculation Number :6546577

Due: 5 June 2024

Exercise 1: Ridge Regression

(a) **(2 points)** In the following you have to implement least squares and ridge regression (both L2-loss)

- **w = LeastSquares(Designmatrix,Y)**:
  - input: design matrix $\Phi \in R^{n \times d}$ and the outputs $Y \in \mathbb{R}^n$ (column vector)
  - output: weight vector w of least squares regression as column vector

  Answer: See *ridge.py* function *LeastSquares*.

- **w = RidgeRegression(Designmatrix,Y,Lambda)**:
  - input: the design matrix $\Phi \in \mathbb{R}^n \times d$, the outputs $Y \in \mathbb{R}^n$ (column vector), and the regularization parameter $\lambda \in \mathbb{R}^+ := \{x \in \mathbb{R} | x \geq 0\}$.
  - output: weight vector w of ridge regression as column vector. Use the non-normalized version $w = (\phi^T \phi + \lambda \mathbb{1}_d)^{-1} \phi^T Y$

  Note that that the regression with L1-loss is already provided in **L1LossRegression(Designmatrix,Y,Lambda)**
  Answer:
  See *ridge.py* function *RidgeRegression*.

(b) **(1 points)** Let us assume that $d = 1$. Write a function Basis(X, k) to generate the design matrix using the orthogonal Fourier basis functions, with

- input: the input data matrix $X \in \mathbb{R}^{n \times 1}$ and the maximal frequency k of the Fourier basis.
- output: the design matrix $\phi \in \mathbb{R}^{n \times (2k+1)}$ using the Fourier basis functions: $\phi_{i,0} = 1$ for all $i = 1, ..., n$ and $\phi_{i,2l-1} = \cos(2\pi l x_i)$ and $\phi_{i,2l} = \sin(2\pi l x_i)$ for all $i = 1, ..., n$ and $l = 1, ..., k$.

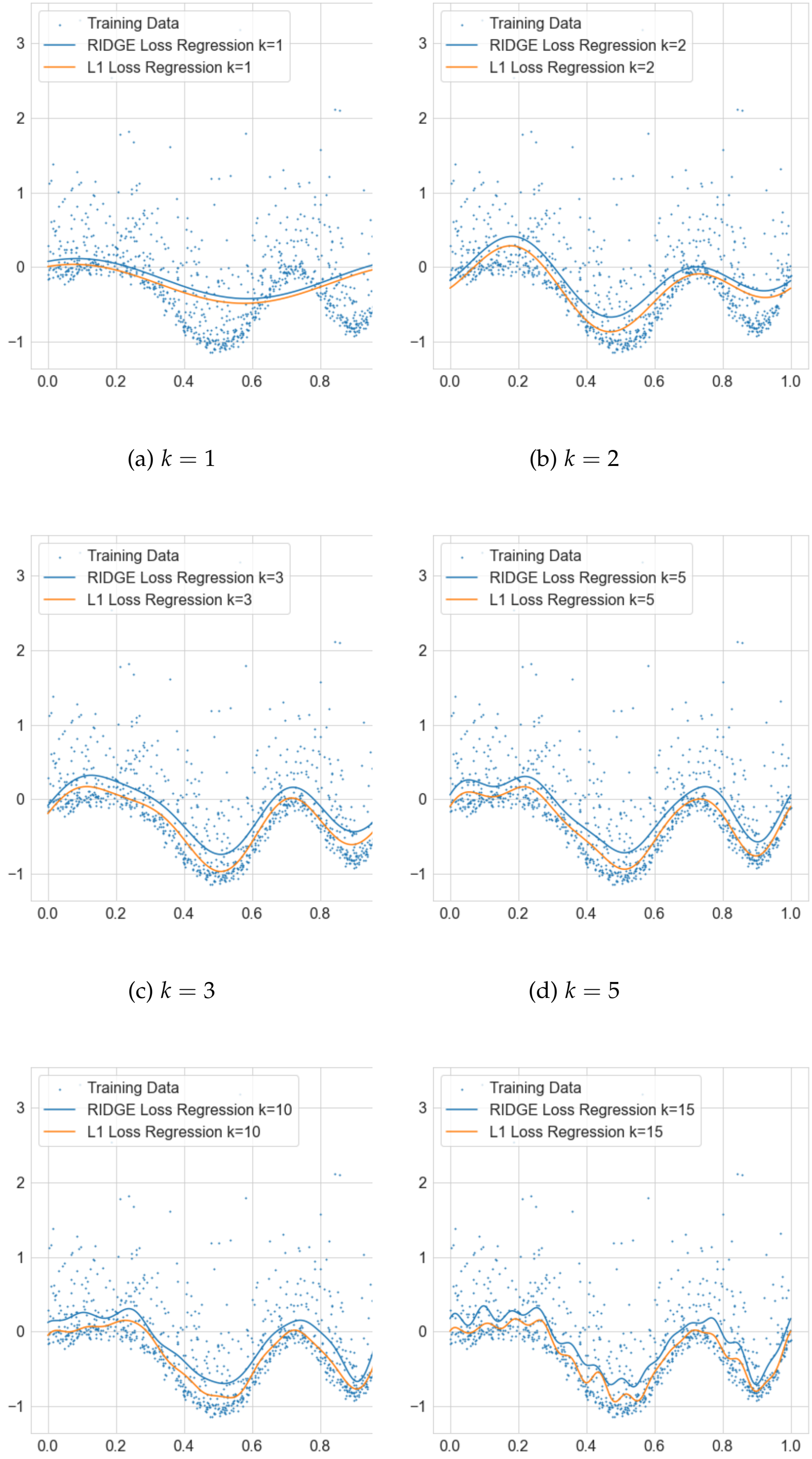
Answer:
See *ridge.py* function *Basis*.

(c) In the first example we have only one feature $(d\ =)$ and thus we want to learn the function $f : \mathbb{R} \rightarrow \mathbb{R}$. The data is given in **onedim.data.py** containing $Xtrain, Xtest, Ytrain, Ytest \in \mathbb{R}^{1000\times 1}$. First Plot the training data $(Xtrain, Ytrain)_{i=1}^{1000}$.

(a) **(1 Points)** Which loss functino (L_1 or L_2) is more appropriate for this kind of data? Justify this by checking the data plot. Use in the next part only the regression method which your chosen loss (that is either regression or $L_1 - loss$ with $L_2 - regularizer$).

Answer:

We find that the ridge regression introduces a consistent bias towards the outliers in the data for all our plots for different values of k. Thus in order to avoid this we use the *L1LossRegression* which is less biased towards the outliers.



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 5$

(e) $k = 10$

(f) $k = 15$

Figure 1: Comparison of L1 and Ridge Regression for different values of $k = 1, 2, 3, 5, 10, 15$. We note that that ridge regression introduces a bias towards the outliers in the data

(b) **(1 Points)** Use the basis function with $k = 1, 2, 3, 5, 10, 15, 20$ from part b. to to fit the regularized version of the loss chosen in the previous part. Use regularization paramater $\lambda = 30$. Plot the resulting function f_k (using as x e.g 1000 evenly spaced points in $[0, 1]$) for all values of k together with the training data with:

$$f_k(x) = \langle \phi(x), w_k \rangle = \sum_{i=1}^{2k+1} w_i^k \phi_i(x) \tag{1}$$

Compute the loss, that is

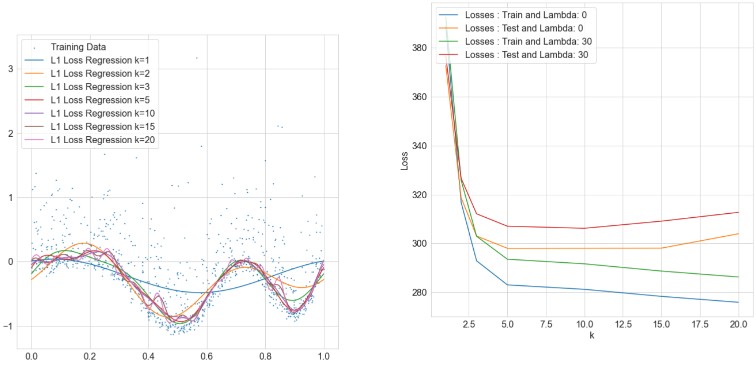
$$\frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i)) \tag{2}$$

on the training and test data and plot training and test loss as a function of k. Repeat the same for $\lambda = 0$ (unregularized version). How does increasing k affect the estimated function f_k ?

What is the behavior of training and test error for increasing k (explanation on paper).

Answer:

We note that increasing the value of k increases the function expressivity it becomes rougher and begins to overfit to training data. This can be seen in the loss curves, where th test loss fails to go down after $k = 5$ while the training loss continues to decrease. This effect is more pronounced without regularization. Where we note the curve for non-regularized Train loss has lowest value as k increases.



(a) L1 Loss Regression for all k

(b) Training and Test Loss for L1 Loss Regression.

Figure 2: Note that Test loss stabilizes around k=5 and increases subsequently. Also note that overfitting is worse for Lambda=0 vs Lambda=30

(d) On observes overfitting when we use large number k of basis functions. We want to avoid this phenomenon by introduing a normalization of the basis functions according to their complexity. One possible way to do this is to define a measure of complexity $\Omega(f) \in \mathbb{R}^+$ as

$$\Omega(f) = \int_0^1 |f'(x)|^2 dx \tag{3}$$

where f' is the first derivative of f at x and introduce new Fourier basis functions $\{\Psi_i(x)\}_{i \in \mathbb{N}^0}$ as

$$\Psi_0(x) = \Phi_0(x) \tag{4}$$

and

$$\Psi_i(x) = \frac{1}{\sqrt{\Omega(\phi_i)}} \Phi_i(x) \tag{5}$$

$i \in \mathbb{N}^+$ where $\mathbb{N}^0 := \{0, 1, 2, \ldots\}$ and

(a) **(1 point)** Show the new Fourier basis functions $\Psi = \Psi_{i i \in \mathbb{N}^+}$ all have the same complexity $\Omega(\Psi_i)$.

Answer:

Consider

$$\Omega(\Phi_i) = \int_0^1 |(\Phi_i)'(x)|^2 dx \tag{6}$$

We know consider the case where $i = 2l - 1$ and $i = 2l$ for $l \in [1..k]$.

$$
\begin{aligned}
\Omega(\Phi_{i=2l-1}) &= \int_0^1 |(\Phi_i)'(x)|^2 dx && (7)\\
&= \int_0^1 |(\cos(2\pi lx))'|^2 dx && (8)\\
&= \int_0^1 |-2\pi l \sin(2\pi lx)|^2 dx && (9)\\
&= \int_0^1 (4\pi^2 l^2)\sin^2(2\pi lx) dx && (10)\\
&= (4\pi^2 l^2)\int_0^1 \sin^2(2\pi lx) dx && (11)\\
&= (4\pi^2 l^2)\int_0^1 \frac{1-\cos(4\pi lx)}{2} dx && (12)\\
&= (2\pi^2 l^2)\int_0^1 1-\cos(4\pi lx) dx && (13)\\
&= (2\pi^2 l^2)\left[x - \frac{1}{4\pi l}\sin(4\pi lx)\right]_0^1 && (14)\\
&= (2\pi^2 l^2)\left[1-0-0+0\right] && (15)\\
&= 2\pi^2 l^2 && (16)\\
\sqrt{\Omega(\Phi_{i=2l-1})} &= \sqrt{2\pi^2 l^2} && (17)\\
&= \sqrt{2}\pi l && (18)
\end{aligned}
$$

Now consider :

$$
\begin{aligned}
\Omega(\Phi_{i=2l}) &= \int_0^1 |(\Phi_i)'(x)|^2 dx && (19)\\
&= \int_0^1 |(\sin(2\pi l x))'|^2 dx && (20)\\
&= \int_0^1 |2\pi l \cos(2\pi l x)|^2 dx && (21)\\
&= \int_0^1 (4\pi^2 l^2) \cos^2(2\pi l x) dx && (22)\\
&= (4\pi^2 l^2) \int_0^1 \cos^2(2\pi l x) dx && (23)\\
&= (4\pi^2 l^2) \int_0^1 \frac{1+\cos(4\pi l x)}{2} dx && (24)\\
&= (2\pi^2 l^2) \int_0^1 1+\cos(4\pi l x) dx && (25)\\
&= (2\pi^2 l^2) \left[x + \frac{1}{4\pi l} \sin(4\pi l x) \right]_0^1 && (26)\\
&= (2\pi^2 l^2) \left[1 + 0 - 0 + 0 \right] && (27)\\
&= 2\pi^2 l^2 && (28)\\
\sqrt{\Omega(\Phi_{i=2l})} &= \sqrt{2\pi^2 l^2} && (29)\\
&= \sqrt{2}\pi l && (30)
\end{aligned}
$$

Thus the normalization constant is the same for both basis functions.

Now we consider the complexity of the new basis functions

Ψ_i:

$$\begin{aligned}
\Omega(\Psi_{i=2l-1}) &= \int_0^1 |(\Psi_i)'(x)|^2 dx && (31)\\
&= \int_0^1 |\frac{1}{\sqrt{\Omega(\Phi_{i=2l-1})}}(\Phi_{i=2l-1})'(x)|^2 dx && (32)\\
&= \int_0^1 |\frac{1}{\sqrt{2}\pi l}(-2\pi l \sin(2\pi l x))|^2 dx && (33)\\
&= \int_0^1 |\sqrt{2}\sin(2\pi l x)|^2 dx && (34)\\
&= \int_0^1 2\sin^2(2\pi l x) dx && (35)\\
&= 2\int_0^1 \frac{1-\cos(4\pi l x)}{2} dx && (36)\\
&= \int_0^1 1-\cos(4\pi l x) dx && (37)\\
&= \left[x - \frac{1}{4\pi l}\sin(4\pi l x)\right]_0^1 && (38)\\
&= 1-0-0+0 && (39)\\
&= 1 && (40)
\end{aligned}$$

Similarly for $\Psi_{i=2l}$ we derive the complexity as :

$$\begin{aligned}
\Omega(\Psi_{i=2l}) &= \int_0^1 |(\Psi_i)'(x)|^2 dx && (41)\\
&= \int_0^1 |\frac{1}{\sqrt{\Omega(\Phi_{i=2l})}}(\Phi_{i=2l})'(x)|^2 dx && (42)\\
&= \int_0^1 |\frac{1}{\sqrt{2}\pi l}(2\pi l \cos(2\pi l x))|^2 dx && (43)\\
&= \int_0^1 |\sqrt{2}\cos(2\pi l x)|^2 dx && (44)\\
&= \int_0^1 2\cos^2(2\pi l x) dx && (45)\\
&= 2\int_0^1 \frac{1+\cos(4\pi l x)}{2} dx && (46)\\
&= \int_0^1 1+\cos(4\pi l x) dx && (47)\\
&= \left[x + \frac{1}{4\pi l}\sin(4\pi l x)\right]_0^1 && (48)\\
&= 1-0-0+0 && (49)\\
&= 1 && (50)
\end{aligned}$$

(b) **(1 point)** Derive the explict form of the new basis functions $\{\Psi_i\}_{i\in\mathbb{N}^0}$ and implement a modified version function **DesignMatrix = FourierBasisNormalized(X,k)**:

i. input: the input data matrix $X \in \mathbb{R}^{n \times 1}$ and maximal frequency k of the Fourier basis.

ii. output: design matrix $\Phi \in \mathbb{R}^{n \times (2k+1)}$ using the normalized Fourier basis $\{Psi_i\}_{i=0...2k}$

Answer: See *ridge.py* function *FourierBasisNormalized*.

(c) **(1 point)** Repeat the experiment from part c. with both old (not normalized) basis Φ_i and the new basis function Ψ_i, using both least squares and ridge regression with regularization parameter $\lambda = 30$, when using $|phi_i$ and $\lambda = 0.5$ when using Ψ_i. How does the new basis function affect the estimation of the function $f_k = \langle w^k, \Psi(x) \rangle$? What is the difference in terms of training and test error for the various k (explanation on paper)?

Answer:

(e) **(2 points)** We now consider a modified problem where instead of penalizing the weights one directly penalises the gradient of the estimated function $f_w(x) = \langle w, \Psi(x) \rangle$:

$$w^k = argmin_{w \in \mathbb{R}^2 k} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f_w(X_i))^2 + \lambda \Omega(f_w) \tag{51}$$

where $\Omega(f)$ is defined in part d. Show that when using the normalized Fourier basis Ψ_i without the constant function Ψ_0 the above optimization problem is equivalent to ridge regression that is $\Omega(f_w) = ||w||^2$.

Zip all plots (.png), scripts (.py), test (.pdf). In addition to the functions mentioned above, there should be scripts to reproduce all the results you submit (plots, losses).

References