

Exercise 6 - Regression in Practice

We consider the regression problem where the input $x \in \mathbb{R}^d$ and the output $y \in \mathbb{R}$. We use either L_1 or L_2 -loss.

- a. **(2 points)** In the following you have to implement least squares and ridge regression (both L_2 -loss).

- **w = LeastSquares(Designmatrix,Y):**
 - input: design matrix $\Phi \in \mathbb{R}^{n \times d}$ and the outputs $Y \in \mathbb{R}^n$ (column vector)
 - output: weight vector w of least squares regression as column vector
- **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 the regression with L_1 -loss is already provided in `L1LossRegression(Designmatrix,Y,Lambda)`.

- b. **(1 Point)** Let us assume $d = 1$. 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:

$$\phi_0(x) = 1, \quad \phi_{2l-1}(x) = \cos(2\pi l x), \quad \phi_{2l}(x) = \sin(2\pi l x), \quad l = 1, \dots, k.$$

- c. In the first example we have only one feature ($d = 1$), thus we want to learn a function $f : \mathbb{R} \rightarrow \mathbb{R}$. The data is given in `onedim.data.npy`, containing `Xtrain`, `Xtest`, `Ytrain`, `Ytest` $\in \mathbb{R}^{1000,1}$. First plot the training data, that is the points $(X_{train,i}, Y_{train,i})_{i=1}^{1000}$.

- **(2 Points)** Which loss function (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 with your chosen loss (that is either ridge regression or L_1 -loss with L_2 -regularizer).
- **(4 Points)** Use the basis functions with $k = 1, 2, 3, 5, 10, 15, 20$ from part b. to fit the regularized version of the loss chosen in the previous part. Use regularization parameter $\lambda = 30$. Plot the resulting functions 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

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

Compute the loss, that is

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

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

- d. One observes overfitting when we use a large number k of basis functions. We want to avoid this phenomenon by introducing 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

$$\Omega(f) = \int_0^1 |f'(x)|^2 dx, \quad (1)$$

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

$$\psi_0(x) = \phi_0(x) \quad \text{and} \quad \psi_i(x) = \frac{1}{\sqrt{\Omega(\phi_i)}} \phi_i(x), \quad i \in \mathbb{N}^+,$$

where $\mathbb{N}^0 := \{0, 1, 2, \dots\}$, and $\mathbb{N}^+ := \mathbb{N}^0 \setminus \{0\}$.

- **(1 point)** Show that the new Fourier basis functions $\Psi = \{\psi_i\}_{i \in \mathbb{N}^0}$ all have the same complexity $\Omega(\psi_i)$.
 - **(2 points)** Derive the explicit form of the new basis functions $\{\psi_i\}_{i \in \mathbb{N}^0}$ and implement a modified function `DesignMatrix = FourierBasisNormalized(X,k)`:
 - input: input data matrix $\mathbf{X} \in \mathbb{R}^{n \times 1}$ and maximal frequency k of the Fourier basis
 - output: design matrix $\Phi \in \mathbb{R}^{n \times (2k+1)}$ using the normalized Fourier basis $\{\psi_i\}_{i=0..2k}$
 - **(2 points)** Repeat the experiment from part c with both the old (not normalized) basis ϕ_i and the new basis functions ψ_i , using both least squares and ridge regression with regularization parameter $\lambda = 30$ when using ϕ_i and $\lambda = 0.5$ when using ψ_i . How does the new basis functions affect the estimation of the function $f_k = \langle w^k, \Psi(x) \rangle$? What is the difference in terms of the training and test error for the various k (explanation on paper)?
- e. **(2 points)** We now consider a modified problem where instead of penalising the weights one directly penalises the gradient of the estimated function $f_w(x) = \langle w, \Psi(x) \rangle$:

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

where $\Omega(f)$ is defined in (1). Show that, when using the normalized Fourier basis $\{\psi_i\}_{i=1..2k}$ 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), text (.pdf). In addition to the functions mentioned above, there should be scripts to reproduce all the results you submit (plots, losses).

Hints:

- Linear systems, $\mathbf{Ax} = \mathbf{b}$, can be solved with `numpy.linalg.solve(A, b)`.
- In part e you have to use the fact that the functions $(\psi_i)_{i=1..n}$ are mutually orthogonal, i.e.

$$\forall i, j : \int_0^1 \psi_i(x) \psi_j(x) dx = \delta_{ij} c_i, \quad \text{with } \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else} \end{cases} \quad \text{and for some } c_i \in \mathbb{R}.$$

- You can load the data with `numpy.load(<file_name>).item()`, which gives a dictionary with the four matrices.