

Exercise 7 - 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 $\lambda \in \mathbb{R}^+$
 - 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 matlab code for L_1 -loss (with and without L_2 -regularizer) is provided in the zip-file. It requires the installation of CVX - you can find a link on the course webpage.

b. (**1 Point**) Write a Matlab function `Basis(X, k)`:

- 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:

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

c. In the first example we have only one feature, thus we want to learn a function $f: \mathbb{R} \rightarrow \mathbb{R}$. First plot the training data (`plot(Xtrain, Ytrain, 'l')`).

- (**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 $\lambda = 10$. Plot the resulting functions (use `x = 0 : 0.01 : 1`) for all values of k together with the training data,

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

Save the plots using the command `saveas(gcf, 'PlotFunctions', 'png')`.

Compute the loss, that is

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

on the training and test data (variable names `trainloss` and `testloss`) and plot training and test loss as a function of k . Save the plots using the command `saveas(gcf, 'PlotLoss', 'png')`. Save your training and test loss in a file `LossFirstEx`, use

```
save LossFirstEx trainloss testloss
```

Repeat the same for $\lambda = 0$ (unregularized version) - append a 0 to all filenames e.g. `LossFirstEx0`.

How does increasing k affect the estimated functions f_k ? What is the difference in terms of k of the regularized and unregularized regression method. The last two questions have to be answered on paper.

d. The second example is a real dataset. The task is to predict the total number of violent crimes per 100K population (output variable $Y \in \mathbb{R}$) from a set of features (input variables $X \in \mathbb{R}^{99}$) capturing all sorts of properties of the cities and their population.

- **(2 Points)** Use a linear design with an offset, that is $f(x) = \langle w, x \rangle + b$, (add a feature which is 1 for every data point, $\mathbf{X} = [\mathbf{X}, \mathbf{1}(\text{size}(\mathbf{X}, 1), 1)]$) and fit the data using least squares regression. Compute the training loss, that is $\frac{1}{n} \sum_{i=1}^n \|Y - X * w\|_2^2$, on the training set. Save this `trainloss` with `saveLossSecondExtrainloss`.
- **(2 Points)** You are now free to use any set of basis functions and any regression method. Write a function `Prediction2(X)` which given a set of testpoints - a matrix $X \in \mathbb{R}^{n \times 99}$ outputs the predictions of your chosen learning method as a column vector $f \in \mathbb{R}^{n \times 1}$. For the best results on our hidden test set (which have to be better than the results of linear least squares) we have the following prizes
 1. **(10 Bonus Points)** for the winner
 2. **(5 Bonus Points)** for the second best prediction
 3. **(3 Bonus Points)** for the third best prediction

Submission instructions

- We accept both handwritten and electronic submissions. So you can choose what is more convenient for you. In any case, you should specify full names and immatriculation IDs of all team members. Obviously, programming tasks you can submit only electronically.
- Handwritten submissions should be submitted in the lecture hall of Monday's lecture (before the lecture starts).
- Electronic submissions should be zipped, containing the m-files (`Basis` etc.), your plots (png files) and the matlab data files (.mat) and emailed to the corresponding tutor:
 - a. Apratim Bhattacharyya (Wednesday 8-10): `abhattach@mpi-inf.mpg.de`
 - b. Maksym Andriushchenko (Thursday 8-10): `s8mmandr@stud.uni-saarland.de`
 - c. Max Losch (Friday 16-18): `mlosch@mpi-inf.mpg.de`

If not all 3 students belong to the same tutorial group, then you should email your submission to **only** one tutor (e.g. to the tutor of the first author of your homework), so please do not put other tutors in copy of the email.

The email subject must have the following form: “[ML18/19 Exercise] Sheet X”, where X is the number of the current exercise sheet. Then please specify in the email full names and immatriculation IDs of all team members. Then please attach all your files as a single zip archive, which consists of your immatriculation IDs, e.g. “2561234_2561235_2561236.zip”.

- Reminder: you should submit in groups of 3. Otherwise, we will later on merge the groups smaller than 3 students.

Hints:

- In order to have several plots in one figure you have to use
`figure, hold on`
... all your plotting commands ...
`hold off`
- `gcf` is a handle to the current figure (save the figure just after it was created).
- In order to distinguish the curves for different values of k draw them in different colors using:
`Colors = jet(NVals),`
`for k = 1 : NVals, plot(x, Output(:, k), 'linestyle', '-', 'color', Colors(k, :)), end`
where `Output(:, k)` is a matrix containing the estimated function values at x .
- `norm(x)` computes the Euclidean norm of a vector x .
- Linear system, $Ax = b$, can be solved in Matlab using the backslash operator $x = A \setminus b$.
- More details on the features for the second task can be found in the data-file.