Assignment 2

Mustererkennung/Machine Learning WiSe 18/19

Excercise 1. Implementation of Least-Squares Linear Regression

Using the closed-form expression from the lecture, implement Linear Regression in Python (incl. Numpy, Pandas, Matplotlib) on a Jupyter Notebook. Train on the training set of the "ZIP code"-Dataset and test on its test set.

(a) Print out the Confusion Matrix and the accuracy.

(b) What is a good way of encoding the labels?

(c) What is the problem with using Linear Regression for Classification?

Excercise 2. Why use quadratic loss?

Suppose that N observations y_i are produced by a deterministic linear function $\hat{y}(x_i; \theta)$ with parameter θ and an additive measurement error ϵ_i that is independent and identically distributed from a Normal distribution:

$$y_i = \hat{y}(x_i; \theta) + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, 1) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(\frac{-\epsilon_i^2}{2}\right)$$

We want to train a maximum likelihood estimator, which in the linear case is parameterized by a weight-vector $\hat{\mathbf{w}}$, where

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} L(\mathbf{X}, \mathbf{Y}, \mathbf{w}) = \arg\max_{\mathbf{w}} \mathbb{P}[y_1 = \hat{y}(x_1; \mathbf{w}), \ldots, y_N = \hat{y}(x_N; \mathbf{w}) \mid \mathbf{w}]$$

Show that $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \hat{y}(x_i; w))^2$

For Excercise 1, submit as a fully executed Jupyter Notebook, either printed out in the tutorial <u>*or*</u> *as PDF in KVV. For Excercise 2, either include in the Jupyter Notebook (It can do LaTeX)* <u>*or*</u> *submit on handwritten paper in the tutorial.*
In order to pass, you need to do both excercises! Deadline: 01.11.18, 12.15 Uhr.