Statistical Machine Learning
Assignment 2

Please send me

- the **original script** detailing your computations.
  - The script must be **documented**, i.e. the code corresponding to each answer must be delimited and your loops/variables briefly explained.
  - The script must be **executable**: by just running your script, all results should appear **automatically**.
  - Do not use external functions, everything must be coded **by yourself** using elementary linear algebra functions and standard libraries.
- A document (.doc, .pdf) which will contain your answer and your analysis. Do not put your source code in that document. Illustrations, graphs, *etc.*are welcome.

This homework is due **November 11 (Tue.) noon**

Send your homework to marcocuturicameto+report@gmail.com. Please put the word `report` in the title of your email.

Note that this document has several hyperlinks which will not appear on a printed copy.

Least-square and Locally-weighted least-square regression

- Download the `wine quality dataset` available on `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`, read its description and import it using your favorite programming language.
- Divide the dataset **randomly** into 2 folds of 1000 and 3898 points respectively. The first fold will be called the `train` fold, the second will be called the `test` fold.
- Scale all variables in both train and test sets, by substracting to each variable its empirical mean and dividing it by the empirical standard deviation. That is, for each variable j ($1 \leq j \leq 12$) and observation $1 \leq i \leq 4898$, set

$$x_{ij} \leftarrow \frac{x_{ij} - \mu_j}{\sigma_j},$$

where

$$\mu_j = \frac{1}{1000} \sum_{i \in \texttt{train}} x_{ij}, \sigma_i = \sqrt{\frac{1}{999} \sum_{i \in \texttt{train}} (x_{ij} - \mu_j)^2}.$$

(this operation is called ”standardizing” the data, using only the training fold.)

- Estimate a vector β and a constant b such that
$$y \approx \beta^T \mathbf{x} + b.$$
where y is the quality of the wine (variable number 12), and `x` is the vector of all remaining 11 variables, using the standardized data available in the `train` fold and least-square regression.

- Compute the average error on the `train` fold (namely, the average absolute difference between the predicted quality value of the wine and its actual quality). Compute this error on the `test` fold as well. Compare. Can you interpret which variables have the biggest influence on the quality of the wine?.

Given a `train` database of points $\{(\mathbf{x}_i, y)\}_{i=1,\cdots,n}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y \in \mathbb{R}$, least-square regression finds the minimizer of

$$(\beta_\star, b_\star) = \underset{\beta,b}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| y_i - \begin{bmatrix} b & \beta^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} \right\|^2$$

to predict, given a new point $\mathbf{x}_{\text{new}}$, its corresponding predicted variable as $\begin{bmatrix} b_\star & \beta_\star^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}$. A different technique, called locally-weighted linear locally regression, tries to exploit the similarity of the point we are interested in, $\mathbf{x}_{\text{new}}$, with respect to other points in the database,

$$w_i \overset{\text{def}}{=} \text{similarity}(\mathbf{x}_{\text{new}}, \mathbf{x}_i), \quad i = 1, \cdots, n$$

by defining instead

$$(\beta_\sharp, b_\sharp) = \underset{\beta,b}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \left\| y_i - \begin{bmatrix} b & \beta^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} \right\|^2,$$

and using $(\beta_\sharp, b_\sharp)$ to predict the corresponding y variable of $\mathbf{x}_{\text{new}}$ as $\begin{bmatrix} b_\sharp & \beta_\sharp^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}$

- Compute the average error of locally weighted regression on the `test` fold, assuming
$$\text{similarity}(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T \Sigma^{-1} (\mathbf{x}-\mathbf{x}')/2},$$
where Σ is the empirical variance matrix of your `train` fold, namely
$$\Sigma = \frac{1}{n_{\text{train}} - 1} \sum_{i=1}^{n_{\text{train}}} \left(\mathbf{x}_i - \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} \mathbf{x}_j \right) \left(\mathbf{x}_i - \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} \mathbf{x}_j \right)^T.$$
In order to do so, you will have to compute a different $(\beta_\sharp, b_\sharp)$ for **each** element of the test fold. Explain how you can compute $(\beta_\sharp, b_\sharp)$.

- What are the advantages/disadvantages of locally-weighted regression compared to standard regression? In which cases do you think locally-weighted regression might work better than regression?