Statistical Machine Learning
Assignment 1

This homework is due **May 8th (Sun.) 11:59 PM**

As you can see below, this homework involves both math questions and programming questions.
Send your completed homework **in pdf format** to marcocuturicameto+report@gmail.com. Please put the word `report` in the title of your email. The pdf can be a scanned copy of a handwritten assignment (specially if you need to write a lot equations) or a word/latex document **exported as pdf** (please don't send me a .doc document). Please **send me your code in a zipped folder**. I should be able to run that code on its own.

Positive Definite Matrices

A square $n \times n$ matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if

$$\forall x \in \mathbb{R}^d, x \neq 0 \Rightarrow x^T A x > 0.$$

Alternatively, A is said to be positive *semi*-definite if

$$\forall x \in \mathbb{R}^d, x^T A x \geq 0.$$

1. Suppose A is a symmetric matrix. What can you say about its eigenvalues?
2. Suppose A is positive definite and symmetric. Prove that all the eigenvalues of A are positive.
3. Prove that the sum of two symmetric positive definite matrices $A, B \in \mathbb{R}^{d\times d}$ is positive definite.
4. Prove that if A is symmetric positive definite, then $\det A > 0$ and thus A is invertible. On the contrary, show that if $\det A > 0$, then A is not necessarily positive definite (you just need to provide a counterexample).
5. Prove that if A is positive *semi*definite and $\lambda > 0$, then $(A + \lambda I)$ is positive definite.
6. Prove that if $X \in \mathbb{R}^{d\times n}$ then XX^T and $X^T X$ are both positive semidefinite.
7. Prove that if $X \in \mathbb{R}^{d\times n}$ has rank d, then XX^T is positive definite (invertible).
8. Let $X \in \mathbb{R}^{d\times n}$ be a matrix, and $Y \in \mathbb{R}^n$. Prove that $\min_{\alpha\in\mathbb{R}^d} \|X^T\alpha - Y\|_2^2 + \lambda\|\alpha\|_2^2$ is attained for $\alpha = (XX^T + \lambda I)^{-1} XY$.
9. Compare this formula with the formula provided in Lecture 1. What is the advantage of introducing a positive λ parameter in the optimization above?

Least-square and Locally-weighted least-square regression

- Download the `wine quality dataset` (white wines) 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 $\mathbf{x}$ 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

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

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,

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

by defining instead

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

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?