SI221 Assignment #5:
Linear and logistic regression - 20/10/2020

Instructions

Read all the instructions below carefully before you start working on the assignment.

- Please submit the source code and a pdf report of your work. You can use the Jupyter notebook instead, by submitting the ipynb file. Each file must have as title: *TP_Regression_Student1_Student2*.
- Late assignments will not be corrected.
- You must do this assignment in groups of 2. Please submit no more than one submission per group.
- You must implement a algorithm from scratch.
- Code that does not work will not be considered.
- Send your work to: `nazareth@telecom-paris.fr` and `colombo.pierre@gmail.com`

Practical assignment objective

This assignment is aimed at coding, from scratch, linear and polynomial regression models for prediction and logistic regression for classification.

Reminders about logistic regression and optimization

Optimization

When computing ERM we are given a cost function $\mathcal{L}(\omega)$ we wish to minimize.

Gradient Definition The gradient of the function $\mathcal{L}$ is the vector defined as:

$$\nabla\mathcal{L}(\omega) = \left[\frac{\partial\mathcal{L}(\omega)}{\partial w_1}, \cdots, \frac{\partial\mathcal{L}(\omega)}{\partial w_i}, \cdots, \frac{\partial\mathcal{L}(\omega)}{\partial w_d}\right].$$

Batched Gradient Descent: For ERM the cost function is a sum of losses over the training examples, that is

$$\mathcal{L}(\omega) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_i(\omega)$$

where $\mathcal{L}_i$ is the cost of the $i-th$ training example.

To minimize the function, we itteratively take a step in the (opposite) direction of the gradient

$$\omega(t+1) = \omega(t) - \frac{1}{d} \times \gamma \nabla_d \mathcal{L}(\omega)(t) \tag{1}$$

where γ is the learning rate and where $\nabla_d \mathcal{L}(\omega)(t) = \frac{1}{d} \sum_{i=1}^{d} \nabla \mathcal{L}_i(\omega)$ is the gradient computed on d points selected at random in the training set.

Logistic Regression

In this HW we will treat a classification problem as a regression problem. Recall that in the classification problem $y = sgn(< \mathbf{x},\mathbf{w} >)$ Instead, in logistic regression we consider

$$y = \sigma(< \mathbf{x},\mathbf{w} >)$$

where σ denotes the logistic function and is defined as

$$\sigma(u) = \frac{e^u}{1+e^u}.$$

Here $y \in [0,1]$ and tends to 1 as $< \mathbf{x},\mathbf{w} >$ tends to infinty and tends to 0 as $< \mathbf{x},\mathbf{w} >$ tends to $+\infty$ and tends to 0 as $< \mathbf{x},\mathbf{w} >$ tends to $-\infty$. Hence, $\sigma(< \mathbf{x},\mathbf{w} >)$ can be interpreted as $Pr(y = 1|\mathbf{x},\mathbf{w})$, and this quantity depends on how far $\mathbf{x}$ is from the hyperplane defined by $\mathbf{w}$. In particular, for points close to the hyperplane $\sigma(< \mathbf{x},\mathbf{w} >) \approx 1/2$.

Getting started!

1 Multiple linear regression

In this part we will apply linear regression model for prediction. Let's use the *USA Housing* data set that can be downloaded in `https://www.kaggle.com/vedavyasv/usa-housing`. We want to predict the house prices in US based on some attributes.

Short description: This data set contains 5k instances with 7 attributes: 'Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'. Since house price is a continues variable, this is a regression problem.

Data Preparation: For simplicity, we will not consider the address. Split the data set into a training and test set, containing respectively 75% and 25% of the data set.

1. Taking into account all the data set, compute the sample Pearson correlation coefficients in order to see how the variables (except the address) are correlated with each other. You should obtain Fig. 1. Comment.

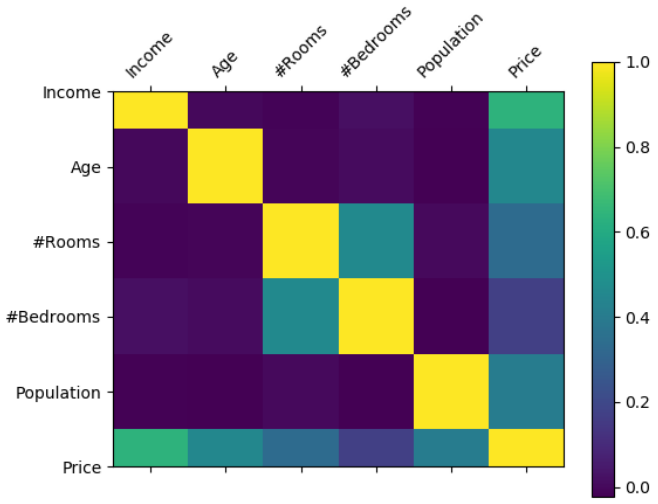

Figure 1: Correlation matrix.

2. By considering a bias column, predict the price of houses with linear regression model with respect to the squared loss, via Eigendecomposition[1]. Compute the estimated bias and coefficients, and the mean absolute error of the test data set. Plot the actual price of the test data set *versus* its prediction. Comment.

3. (*Optional*) Repeat the previous item using polynomial regression. Generate a new feature matrix consisting of all polynomial combinations of the features with degree 2 (or higher degree). Solve the polynomial regression by employing linear regression[2]. Compare the results with linear regression obtained in the previous item.

2 Logistic regression

In this part we will apply logistic regression. Before starting let's warm up with a couple of optional theoretical questions.

(Optional) Theoretical question

- Show that the gradient of the MSE under linear prediction is given by

$$\nabla \mathcal{L}(\omega) = \frac{-1}{m} \times \mathbf{X}^T \mathbf{e}$$

where $\mathbf{X}$ represents the input data, $\mathbf{y}$ the label data and $\mathbf{w}$ are the learnable weights and $\mathbf{e}$ is the error vector defined as

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}$$

[1]For more information check the Book "Understanding Machine Learning: From Theory to Algorithms", by Shai Shalev-Shwartz and Shai Ben-David.

- Show that for Linear regression the cost function (MSE) is convex in $\mathbf{w}$
- Show that for the Logistic regression the gradient can be written as :

$$\nabla \mathcal{L}(\mathbf{w}) = \mathbf{X}^T[\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}]$$

Recall that by convention the matrix $\mathbf{X}$ has N rows, one per input sample. Further, $\mathbf{y}$ is the column vector of length N which represents the N labels corresponding to each sample. Therefore, $\mathbf{X}\mathbf{w}$ is a column vector of length N. The expression $\sigma(\mathbf{X}\mathbf{w})$ means that we apply the function σ to each of the N components of $\mathbf{X}\mathbf{w}$. In this manner we can express the gradient in a compact manner.

We will use use linear regression to do classification. Although this is not a good idea in general, it will work for simple data. We will use a subset of the height-weight data.

Practical questions: Open the provided notebook and read the linear regression code and fill the logistic regression code.