INF264 - Homework 2

Pierre Gillot   Natacha Galmiche

August 26th 2020

1 Univariate linear regression with gradient descent

In this exercise, we consider the simplest case of linear regression, namely univariate linear regression. Given a dataset (x, y) where $x \in \mathbb{R}^N$ is a column of N one dimensional samples and $y \in \mathbb{R}^N$ are the associated target values, this is equivalent to solving:

$$\min_{(w_0, w_1) \in \mathbb{R}^2} \frac{1}{N} \sum_{n=0}^{N-1} \left(y_n - (w_0 + w_1 x_n) \right)^2 \iff \min_{w \in \mathbb{R}^2} \frac{1}{N} \| y - \overline{x} w \|_2^2 \quad (1)$$

where $\overline{x} = \left[\overrightarrow{1} | x \right]$ is the vertical concatenation of x with a column vector of N ones.

It is well known that linear problems of the form

$$\min_W \| Y - XW \|_2^2 \quad (2)$$

admit a closed form solution given by

$$(X^t X)^{-1} X^t y. \quad (3)$$

Unfortunately, in many real-world scenarios this inversion problem is too costly and/or unstable to compute directly. In such situations, the gradient descent method is a well-suited solution. In the following questions, you will derive and implement the gradient descent method for the univariate linear regression problem:

1. Derive the gradient of the mean-squared error (MSE) loss for the univariate linear regression problem, that is

$$\nabla_w L = \left(\frac{\partial L}{\partial w_0} (w_0, w_1), \frac{\partial L}{\partial w_1} (w_0, w_1) \right) \quad (4)$$

   where

$$L(w_0, w_1) = \frac{1}{N} \sum_{n=0}^{N-1} \left(y_n - (w_0 + w_1 x_n) \right)^2. \quad (5)$$

2. Implement an algorithm which, given a learning rate $\eta > 0$, iteratively updates the weights $w = (w_0, w_1)$ by following the opposite direction of the gradient (also known as the "steepest descent") of the loss at the current estimate, that is:

$$w_{t+1} = w_t - \eta \ \nabla_{w_t} L \tag{6}$$

Hint: You will have to implement a function which for any input $w = (w_0, w_1)$ returns the gradient $\nabla_w L$, based on the equations obtained in the previous question.

3. Use your algorithm to learn the dataset contained in the file 'unilinear.csv'. You can set the number of gradient descent iterations to 40 and the learning rate to 0.1.

4. Directly compute the exact closed form solution and compare it to the approximate solution obtained previously.

5. Perform the gradient descent estimation of the weights for different values of η (for example: 0.0001, 0.001, 0.01, 0.1...). What do you notice ? Why is η referred to as the "learning rate" ?

2 Logistic Regression

Logistic Regression (also called Logit Regression) is a generalized linear model which is commonly used to estimate the probability that an instance belongs to a particular class (e.g., what is the probability that this email is spam?). If the estimated probability is greater than 0.5, then the model predicts that the instance belongs to that class (called the positive class, labeled "1"), or else it predicts that it does not (i.e., it belongs to the negative class, labeled "0"). This makes it a binary classifier.

Let's use the iris dataset to illustrate Logistic Regression. This is a famous dataset that contains the sepal and petal length and width of 150 iris flowers of three different species: Iris-Setosa, Iris-Versicolor, and Iris-Virginica.

2.1 Logistic regression on two classes

In order to try logistic regression on two classes you need to divide the Iris dataset into two classes (for example: "Iris-Virginica" and "not Iris-Virginica"). Then, fit a linear regression model to the petal width of the Iris-Virginica (Read data DESC for more information). Finally, fit a logistic regression to the same feature.

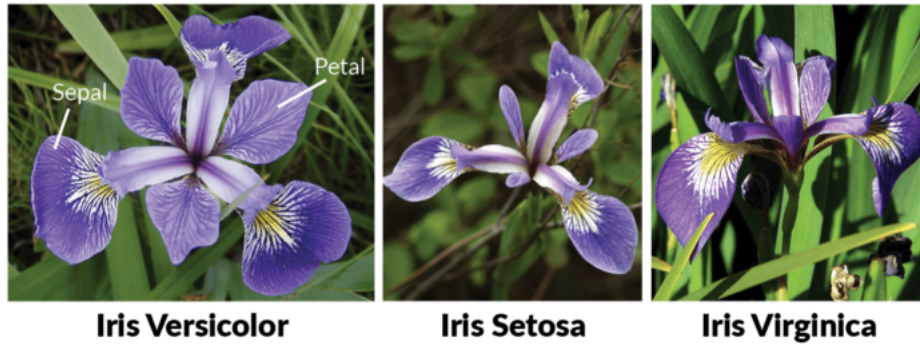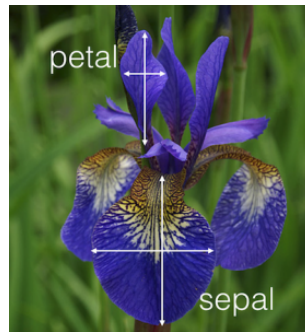1. Compare the linear and the logistic regression models on plots.

Figure 1: Iris dataset

2.2 Multi class logistic regression

Your task now is to build a multi-class classifier that can distinguish between Iris-Virginica, Iris-Setosa, and Iris-Versicolor. Furthermore, instead of using one feature, now you have to use two features — petal length and petal width — to train your model.

2. Plot the boundaries for predicting the labels of the dataset. How is it different than the one that was in the previous week's assignment for KNN?

Read more about the Logistic Regression parameters here: `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html` online document.