Machine Learning
1st Lab Assignment
Instituto Superior Técnico

Linear Regression

1 Introduction

Linear Regression is a simple technique for predicting a real output y given an input $x = (x_1, x_2, \ldots, x_P)$ via the linear model:

$$f(x) = \beta_0 + \sum_{k=1}^{P} \beta_k x_k \tag{1}$$

Typically there is a set of training data $T = (x^i, y^i), i = 1, .., N$ from which to estimate the coefficients $\beta = [\beta_0, \beta_1, ...\beta_P]^T$.

The Least Squares (LS) approach finds these coefficients by minimizing the sum of squares error

$$SSE = \sum_{y=1}^{N} \left(y^i - f(x^i)\right)^2 \tag{2}$$

The linear model is limited because the output is a linear function of the input variables x_k. However, it can easily be extended to more complex models by considering linear combinations of nonlinear functions, $\phi_k(x)$, of the input variables

$$f(x) = \beta_0 + \sum_{k=1}^{P} \beta_k \phi_k(x) \tag{3}$$

In this case the model is still linear in the parameters although it is nonlinear in x. Examples of nonlinear functions include polynomial functions and Radial basis functions. This assignment aims at illustrating Linear Regression. In the first part, we'll experiment linear and polynomial models. In the second part, we'll illustrate regularized Least Squares Regression.

2 Practical assignment

`Note` In this assignment, you'll often find between brackets, as in {`command`}, suggestions of Python commands that may be useful to perform the requested tasks. You should search Python documentation, when necessary, to obtain a description of how to use these commands. At the end of the session you should submit, in fenix, your code along with your answers.

2.1 Least Squares Fitting

1. (T) Write the matrix expressions for the LS estimate of the coefficients of a polynomial fit of degree P and of the corresponding sum of squares error, from training data $T = (x^i, y^i), i = 1, .., N$.
2. Write code to fit a polynomial of degree P to 1D data variables x and y. Write your own code, do not use any Python ready made function for LS estimation or for polynomial fitting [1]. {`matmul, transpose, inv from numpy`}
3. Load the data in files 'data1_x.npy' and 'data1_y.npy' and use your code to fit a straight line to variables y and x. {`load from numpy`}
   (a) Plot the fit on the same graph as the training data. Comment. {`plot, scatter from matplotlib`}
   (b) Indicate the coefficients and the Sum of squared errors (SSE) you obtained.
4. Load the data in files 'data2_x.npy' and 'data2_y.npy', which contain noisy observations of a cosine function $y = cos(2x) + \epsilon$, with $x \in [-1, 1]$, in which ϵ is Gaussian noise with a standard deviation of 0.15. Use your code to fit a second-degree polynomial to these data.
   (a) Plot the training data and the fit. Comment.
   (b) Indicate the coefficients and the SSE you obtained. Comment.
5. Repeat item 4 using as input the data from files 'data2a_x.npy' and 'data2a_y.npy'. This file contains the same data used in the previous exercise except for the presence of a couple of outlier points.
   (a) Plot the training data and the fit. Comment.
   (b) Indicate the coefficients and the SSE you obtained. Compute, in addition, the SSE only on the inliers. Comment on the sensitivity of LS to outliers.

[1] If you are unable to write your own code, you may use `LinearRegression` and `PolynomialFeatures` from `sklearn` (set include_bias to False). However, this option will have a penalty of 2 values in the lab grade.

2.2 Regularization

The goal of this second part is to illustrate linear regression with regularization. We'll experiment with Ridge Regression and Lasso.

1. Load the data in files 'data3_x.npy' and 'data3_y.npy' which contain 3-dimensional features in variable x and a single output y. One of the features in x is irrelevant.
2. (T) Explain Ridge and Lasso regularization methods and explain how Lasso can be used for feature selection and Ridge can not. Use expressions when appropriate.
3. Instantiate Ridge and Lasso models. {`Ridge, Lasso from sklearn.linear_model`}
4. Fit Ridge and Lasso models to your data for values of α in the range 10^{-3} to 10 with step size 0.01. Set the maximum number of iterations to 10000.
5. Plot the Lasso and Ridge coefficients against α, using a logarithmic scale for the α axis. For comparison plot horizontal lines corresponding to the LS coefficients in the same figure ($\alpha = 0$). {`plot from matplotlib`}
6. Comment on what you observe in the plot. Identify the irrelevant feature.
7. Consider now only the Lasso method. Choose an adequate value for α. Plot y and the fit obtained for that value of α and compare with the LS fit. Compute the SSE in both cases. Comment.