Polynomial regression using different linear regression techniques

Project description

The aim of this project is to perform a polynomial regression on a simple toy dataset and on a more complex real dataset. In particular, the goal is to assess the performance of different regression models: Ordinary Least Squares (OLS), Lasso and Ridge.

The regression task can be generally described as a way to estimate the relationship between a dependent variable y and an independent variable x. This relationship can be formulated as:

$$y = f(x) + \epsilon \tag{1}$$

where $f(x)$ represents the underlying true dependency between x and y, and ϵ is a measure of the noise introduced in the process of measuring y.

In selecting the polynomial regression, we are restricting the function $f(x)$ to assume a polynomial representation:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p \tag{2}$$

The training of the regression model will correspond to the selection of the β_j parameters, such that the model's prediction 'fits' the observed values.

The selection of the optimal parameters is equivalent to the solution of the linear system of the type $\mathbf{Ax} = \mathbf{b}$:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{3}$$

The linear system admits a single solution only if the matrix $\mathbf{A}$ is invertible. This requires that $n = p$ and that $\mathbf{A}$ is a full rank matrix. In most cases, this requirement is not fullfilled and the system can admit no solutions (overdetermined system) or infinite solutions (undertermined system). In this case, the best combination of parameters β_i is found following an optimization process, where the objective is to minimize a loss function. The difference between OLS, Lasso and Ridge lies in the formulation of the loss function.

Tasks

In this project, you will complete the following tasks:

Task 1: Simulate the trajectory of a projectile with initial velocity magnitude $v_0 = 10$ m/s and initial position $x_0 = 0$ m, $y_0 = 2$ m for the time it takes to reach the ground. The launch angle θ is equal to $50°$. The x, y position as well as the velocity magnitude $v = \sqrt{v_x^2 + v_y^2}$ are tracked by a sensor which has a 1 m/s uncertainty in the velocity measurement and a 0.5 m uncertainty in the position measurement. This is equivalent of saying that the measured value is sampled from a normal distribution with mean equal to the true value and standard deviation equal to the experimental uncertainty. The position and velocity is tracked every 0.01 s. The sketch of the synthetic experiment is reported in Fig. 1.

The steps to solve the first exercise are:

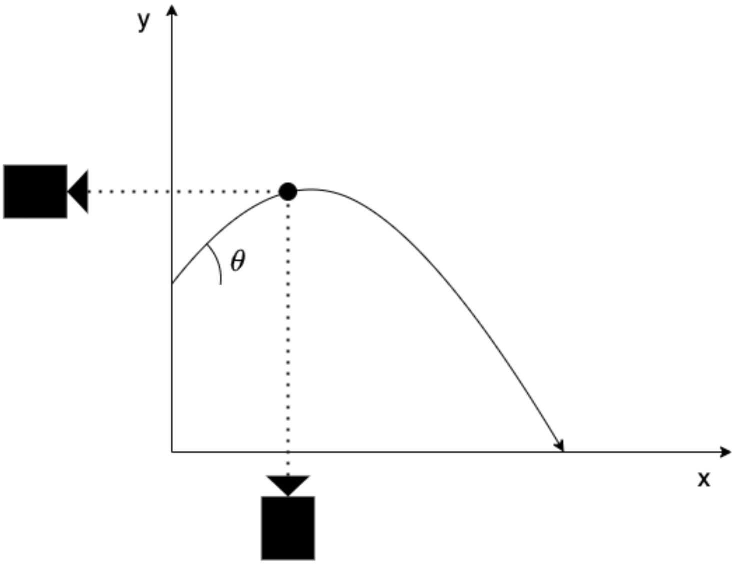

Figure 1: Sketch of the virtual experiment.

- Build a polynomial regression using the OLS method to predict the y position of the projectile using the noisy x measurements as input. The dataset containing x and y has to be divided into a training part and a test part. This can be done using the `train_test_split` function in the `sklearn` package. Usually, the test size is set to 20 % of the entire dataset.

  To perform the regression, you have to first build the matrix $\mathbf{A}$ in Eq. 3. This can be done by using the `PolynomialFeatures` function, where you will set the degree of the polynomial to 2. After you have built the matrix $\mathbf{A}$, you will have to scale both $\mathbf{A}$ and the vector of noisy measurements y using the function `StandardScaler`.

  The actual regression is done using the `LinearRegression` function, in which the parameters to input are the scaled matrix $\mathbf{A_0}$ and the vector $\mathbf{b_0}$, which contains the noisy measurements of the scaled y position.

  Once the model is trained, you can judge the quality of the model by using the `score` attribute of the model, in which you have to input the testing data. Do not forget to scale the testing data as well.
- Build the same OLS regression model using a 20 degree polynomial. Then compare the obtained prediction with the Lasso regression model. You can use the `Lasso` model from the `sklearn` package. Try different values of the `alpha` parameter.
- The optimal value of the α parameter can be inferred from the data using a process called cross-validation. The function `LassoCV` uses an iterative cross-validating algorithm to select the best value of α. Compare the results obtained with the one obtained by manually selecting α.
- Repeat the same process by applying the `Ridge` and `RidgeCV` regressions and compare the results to the OLS results.
- Finally, compare the Lasso and Ridge regression results in extrapolation. To do that, you can predict y by using as input a vector x which is 30% bigger than the one used for training.

Task 2: Build a regression model to predict the daily energy consumption of 32 European countries. The dataset can be found here. Download the `time_series_60min_singleindex.csv` file, which contains the aggregate energy consumption as well as the consumption for each country.

You can use the polynomial regression method that you prefer, however the choice has to be justified.

Note that polynomial regression is not the best choice to build a regression based on a time series. The goal of this task is to apply the polynomial regression to a real dataset. Optionally, you can try to use the Gaussian Process Regression (GPR), which should produce a much better model, given the correct choice of kernel.

Written report

Your written report should contain the following:

Section 1: A brief introduction of the mathematical formulation of the polynomial regression. In particular, explain how the different loss functions influence the behaviour of the model. This section should include also a discussion regarding the risk of overfitting, and which strategies can be adopted to mitigate this risk.

Section 2: Results corresponding to **Tasks 1**, compared to the analytical solution. Each result has to be justified by referring to the mathematical formulation of the model.

Section 3: Results corresponding to **Tasks 2**, including relevant figures.

Section 4: Final discussion and conclusions, highlighting the different situations in which it is preferable to use one model over the others. Don't forget to cite the relevant literature to strengthen your conclusions.

Resources

The Scikit-Learn User Guide offers a good introduction on the mathematical formulation of the linear regression techniques. Moreover, it can be used to access the functions' documentation.

You can find a more in depth discussion of linear regression and shrinkage methods in the third chapter of the book "The Elements of Statistical Learning" [1].

If you encounter problems in the download of the data or installation of the libraries you can contact Alberto Procacci on TEAMS or at alberto.procacci@ulb.be.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2009.