Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

1st Lab Assignment

Shift: ______ Group Number: ______

Number_______ Name____

Number_______ Name____

Linear Regression

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

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

Typically there is a set of training data T={($\mathbf{x}^i$, y^i), i=1,.., N} from which to estimate the coefficients $\boldsymbol{\beta}=[\beta_0, \beta_1, \ldots,\beta_P]^T$. The Least Squares (LS) approach finds these coefficients by minimizing the sum of squares error

$$SSE = \sum_{i=1}^{N} (y_i - f(\mathbf{x}^{\boldsymbol{i}}))^2$$

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, $\phi_k(\mathbf{x})$, of the input variables

$$f(\mathbf{x}) = \beta_0 + \sum_{k=1}^{K} \beta_k \phi_k(\mathbf{x})$$

In this case the model is still linear in the parameters although it is nonlinear in **x**. Examples of nonlinear function 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. The second part of this assignment requires MatLab's Statistics Toolbox.

1. Least Squares Fitting

1. 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 Matlab code to fit a polynomial of degree P to 1D data variables *x* and *y*. Write your own code, do not use any Matlab ready made function for LS estimation or for polynomial fitting. You should submit your code along with your report.

3. Load the data in file 'data1.mat' and use your code to fit a straight line to variables *y* and *x*.

    a. Plot the fit on the same graph as the training data. Comment.

    b. Indicate the coefficients and the error you obtained.

4. Load the data in file 'data2.mat', which contains noisy observations of a cosine function $y = cos(2x) + \varepsilon$, 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 error you obtained. Comment.

5. Repeat item 4 using as input the data from file 'data2a.mat'. This file contains the same data used in the previous exercise except for the presence of an outlier point.

    a. Plot the training data and the fit. Comment.

    b. Indicate the coefficients and the error you obtained. Comment.

2. Regularization

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

1. (T) Write the expression for the cost function used in Ridge Regression and Lasso and explain how Lasso can be used for feature selection.

2. Load the data in file 'data3.mat' which contains 3-dimensional features in variable **x** and a single output y. One of the features in **x** is irrelevant. Use function `lasso` with default parameters (type help for more information on this function) and obtain regression parameters for different values of the regularization parameter λ (the values for lambda are returned in `FitInfo.Lambda`). Use function `lassoPlot` to plot the coefficients against λ. For comparison plot the LS coefficients in the same figure ($\lambda = 0$).

```
[B,FitInfo] = lasso(x,y);
lassoPlot(B,FitInfo,'PlotType','Lambda','XScale','log');
```

3. Comment on what you observe in the plot. Identify the irrelevant feature.

4. Choose an adequate value for λ. Plot y and the fit obtained for that value of λ. Compare with the LS fit. Compute the error in both cases. Comment.

5. Repeat the previous items but using ridge regression (function `ridge)` instead of Lasso. Use the same λ values as in Lasso.