TP3 Data Mining:Linear Regression - L2 Regularizer - Gradient Descent
OBLIGATORY

Tuesday 27th April, 2021
deadline: Monday 17th May, 2021, 23:59

1 Objective

Optimization method like gradient descent (GD) can be used to minimize the cost function of linear regression. But for linear regression, there exists an analytical solution. That means we can obtain the variables for linear regression in one step calculation by using the right formula. In this TP, you are going to solve the Linear Regression with and without $L2$ regularizer using both the analytical solution and gradient Descent.

2 Description

Linear independent attributes. First you should generate a random tall matrix $\mathbf{X}$ and a response vector $\mathbf{y}$ and find the solution of the linear system. Do as follows:

- **Create a function** that generates (**use seed** for reproducibility) a $\mathbf{X}$: $n \times d$, $x_i \sim \mathcal{N}(0,1)$ wide matrix randomly[1]
- Define the target values $\mathbf{y}$ as a linear function of the columns[2] of $\mathbf{X}$, i.e. $\mathbf{Xw = y}$
- Solve analytically the optimization problem $min_{\mathbf{w}}||\mathbf{y - Xw}||^2$ to find the weights $\mathbf{w}$ and compare the solution you got with the one you used to define the target values, is it exactly the same? Comment
- Solve $min_{\mathbf{w}}||\mathbf{y - Xw}||^2$ using GD and compare the solution you got with the one you used to define the target values and the one you got with the analytical solution, is it exactly the same? Comment

[1] Make your **function generic**, but use as default values $n = 100$, $d = 10$
[2] As above make your **function generic** but use a default $d = 10$ and $\mathbf{w} \sim \mathcal{N}(0, 10)$

- For GD run your algorithm for 100 iterations and use learning rate $\{0.1, 0.01, 0.001, 0.0001\}$. For the different learning rates compute the Mean square error over the test data and plot them (x-axis lr, y-axis mse). Comment how the different learning rate influences the result.

Repeat the procedure above but this time add a small random noise $N(0, 1)$ to each target value **y**. Discuss the results. Do the solutions you find match exactly the target values? Explain what happens.

Linear dependencies in the attributes. Now you should study what happens when there are linear dependencies. To do so:

- first take the **X** from above and add few features that are linear combinations[3] of the initial ones and produce a new matrix **X** .
- Define the target value **y** as a linear function of the columns of **X** use the same rule as above to produce the **w**.
- Solve analytically the optimization problem $min_{\mathbf{w}}||\mathbf{y'} - \mathbf{X'w}||^2$. Discuss your results. Here we know that a solution exists, but is there only one? Are there more **ws** that satisfy $Xw = y$? How does the presence of feature dependencies affect the existence of different solutions?
- Solve the optimization problem $min_{\mathbf{w}}||\mathbf{y'} - \mathbf{X'w}||^2$ using GD. Compare the solution you got using GD and the closed form solution with the one you used to define the target values. Comment in details

L2 regularizer. Add a $l2$ regularizer to your optimization problem.

- Derive the analytical solution when we add the $l2$ regularizer (Ridge Regression). You should include to your report all the steps of your calculations.
- Using the same **X'** and **y'** that you generated above compute the weights using the closed form solution. Compare the results that you get with and without regularizer and comment in details explaining de difference that there are in the two different cases (take into account your data and try to understand if this results is expected based on the theory). Try different $\lambda = \{0.0001, 0.01, 0.1, 0, 1, 10\}$ and compute and plot (and comment) the mse that you get using test data for the different values of λ.
- Using the same **X'** and **y'** that you generated above compute the weights using GD using $\lambda = \{0.0001, 0.01, 0.1, 0, 1, 10\}$. Compare, explain, comment in details your results. Compute and plot the mse that you get using test data for the different values of λ. For each λ compare the solution you got using GD and the closed form solution with the one you used to define the target values.

[3] To make our lives simple just add $d = 2$ new features which are simple copies of the first $d/2$ features

Fell free to try different learning rates in order to improve the performance of your model. (Here you don't need to comment about the learning rate but you have to use the same learning rate (and tell us which one do you use) for the different λ that you have to compare).

- Repeat the same exercise using online stochastic gradient descent (online SGD) and SGD with mini batch = 1/10 of your training data to update the weights. Comment

3 General Instructions

This TP is obligatory. You are going to fill a few missing functions in the regression.py python script and to add extra function that needed to implement the exercises that we ask . So first of all read and understand the given python script. To run your code you have to run the main_regression.ipynb notebook.

You have to submit a **formal** report (.pdf or jupyter notebook) and your code.

Reminders

- When we minimize the error/ cost/ loss function using Gradient Descent (GD) the weights are updated after seeing all the training instances
- When we minimize the cost function using Stachastic Gradient Descent (SGD) the weights are updated after seeing a mini batch the training instances. When the size of the mini batch is equal to 1 is called single sample update/ online SGD and the weights are updated each time we see a new instance.