Ecole Polytechnique Fédérale de Lausanne

Machine Learning Project

Milestone 1 report

Cs-233(a): Machine Learning

Matheo Godenzi
Constant Riolo
Lucas Beck

11 nov. 2022

Methods Summary:

Linear regression:

The regression tasks are implemented using a closed-form solution. The hyperplane parameters are computed using the analytical solution on the training data and are further multiplicated with the test data to predict its future values.

Ridge Regression:

The ridge regression, or regularized linear regression, adds a regularizer to penalize the complexity of the linear regression model (overfitting). Our version uses the sum of the squares of the weights as the regularizer. This regularizer is influenced by a hyperparameter lambda, which will be tested with different values to yield the best regression result.

Logistic regression:

First, during training, a matrix of size (N, C) with random weights is implemented. Then, those weights are updated using a gradient descent method, which includes a learning rate parameter that is defined beforehand. The gradient is essentially a computation of the error of the current prediction labels (from a softmax function that returns the probabilities of each sample belonging to a class using the weights, minus the true training labels) multiplied by the input training data. The weights update happens until a threshold of max iterations that is defined previously or stops if the accuracy is perfect at any point, and predicted labels of the training data computed with the final weights are returned. Finally, these resulting weights are also used to compute the labels of the test data, and thus classify its samples.

Cross validation:

To assess our methods, different validation scores were implemented depending on the task, such as the macro F1 score for classification and MSE for regression.

A K-fold cross validation has been implemented, which takes folds of the training data and separates them into a validation fold and training folds. The model is then trained on the training set and the accuracy, F1 score or MSE is found on the validation set, depending on the method. This is repeated K times, each time selecting a different fold of the data for the validation set. In the end, K different validation accuracies will be found and averaged. This will represent the performance of our model.

This cross validation model can also be used to find the best hyperparameter of a method, for example the lambda of the ridge regression.

Analysis of the Cross validation function results:

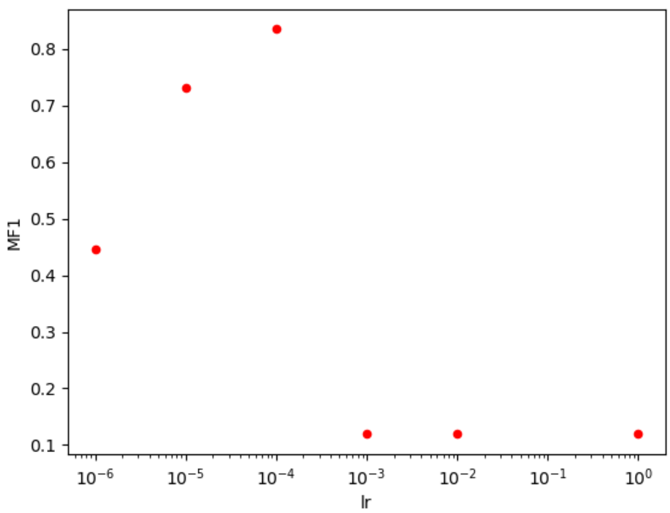
The purpose of this part is to determine what is the best hyperparameter for each method between the set of hyperparameters we gave to our function.

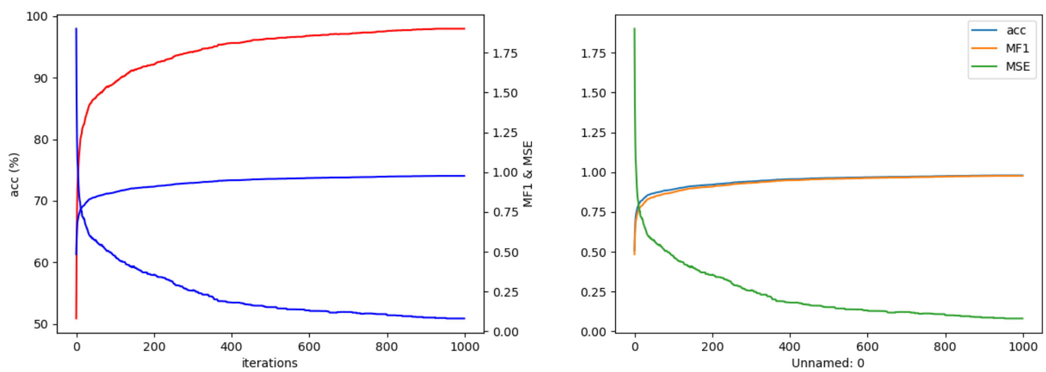
Logistic regression:

We ran the code for cross validation with the following values for “lr” using logistic regression method : [1, 0.01, 0.001, 0.0001, 0.00001, 0.000001]

The best “lr” turned out to be 0.0001 with an accuracy of 82,68 and a macro F1 score of 0.8046.

The graph below represents the train values of MF1 for each value of “lr”.





The graph on the left represents the accuracy in %, the graph on the right represents the accuracy from 0-1.

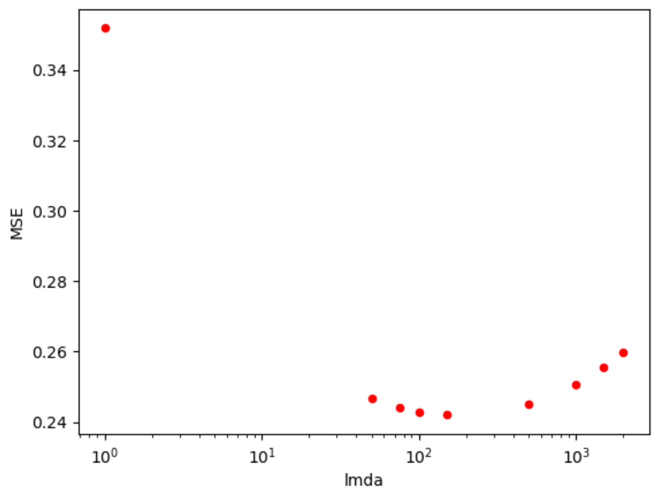
The graphs above represent the values of accuracy (blue), MF1 (orange), MSE (green) for a given value of “lr” over 1000 iterations. This operation is produced over a training dataset.

We clearly see the accuracy growing with each iteration and reaching 100% or 1. The MF1 is also growing and reaching a value of 1. The MSE is drastically decreasing over the 1000 iterations, which is the expected result.

Ridge Regression:

We ran the code for cross validation with the following values for “lmda” using ridge regression method : [1,50, 75, 100, 150, 500, 1000, 1500, 2000]

The best “lmda” turned out to be 150 and has a train MSE value of ~0.24. The test MSE value is ~0.365. Under that value the model underfitst the data; it is not close enough to it. Above that value, it overfits it, which would then be a perfect model as fas as the training data is concerned but would do more poorly on testing data.



Linear Regression:

We ran the code for ridge_regression using lmda=0, which allows us to run the Linear regression model. We found a MSE value of ~2901.98

For the h36m dataset, we have a lot more coordinates than samples, therefore a linear regression is an unstable model because of the infinite possibilities to draw a line through the points. This is the reason why we have such a high MSE compared to the other datasets (music, movies).

To verify our model, we imported the music dataset and ran a linear regression over the cross validation that gave us an MSE of ~0.50. This is the expected value, which means that our model is correct.

Discussion :

The ridge regression and the logistic regression are used for slightly different tasks : in our case (human poses dataset) the logistic regression (classification task) tries to classify the sequence into one of the four action classes, whereas the ridge regression (regression task) tries to predict the poses in the next sequence (lasting one second) after looking at a sequence of two seconds.

Now, the logistic regression, when using cross-validation, takes up to 444 seconds and up to 392 seconds with 1000 iterations. That process is very costly but reliable, with both the Final classification accuracy and the Final macro F1 score reaching 71.2% and 0.69 respectively.

On the other hand, the ridge regression is much faster (5 seconds) but much less reliable as the number of data samples is smaller than the number of features of each sample (i.e. the number of dimensions).

There’s a tradeoff between speed and precision here, and depending on the application needed one would pick one of the two models, as choosing between four key moves is not that far apart from predicting the end of a moving sequence.