Lab Course Machine Learning
Exercise 5

Prof. Dr. Dr. Lars Schmidt-Thieme,
Mofassir ul Islam Arif
Information Systems and Machine Learning Lab
University of Hildesheim
Submission: 29.11.2019 LearnWeb 3115

November 23, 2019

1 Exercise Sheet 5

Classification datasets:
Bank Marketing: (use bank.csv) https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
Regression datasets:
Wine Quality : http://archive.ics.uci.edu/ml/datasets/Wine+Quality

You are required to pre-process given datasets.

- 1. Convert any non-numeric values to numeric values. For example you can replace a country name with an integer value or more appropriately use hot-one encoding. [Hint: use hashmap (dict) or pandas.get_dummies]. Please explain your solution.
- 2. If required drop out the rows with missing values or NA. In next lectures we will handle sparse data, which will allow us to use records with missing values.
- 3. Split the data into a train(80%) and test(20%).
- 4. Normalize the data

2 Linear Classification with Gradient Descent

Exercise 1: Regularization (8 Points)

For each dataset given above

You have to implement *Ridge Regression* using mini-Batch Gradient Descent (mini-BGD) algorithm. Now your SGD algorithm will have three hyperparameters i.e. 1) learning rate (stepsize) α, 2) regularization constant *lambda* and 3) number of mini Batches *batchsize*.

- Implement Ridge Regression using mini-BGD algorithm
- You are free to use any algorithm for selecting learning rate i.e. (AdaGrad, Bold-Driver or fixed stepsize)
- Pick three values of α_0 and *lambda*, these values should be picked from relatively small to large. You should keep a fixed *batchsize = 50.*
- Train you model for each combination of the picked values of α and λ, and for each training epoch (an epoch is equal to going over all mini-batches once) record RMSE on training and test data.
- For each combination of α_0 and λ, plot $RMSE_{train}$ and $RMSE_{test}$ per iteration. [Hint: you can plot $RMSE_{train}$ on positive axis and $RMSE_{test}$ on negative axis of same plot].

3 Hyper-parameter tuning

Exercise 2: Hyper-parameter tuning and Cross validation (12 Points In this section you will implement *grid search* with *k-fold cross-validation* for model selection i.e. choosing best hyperparameters. You will use your implementation from Exercise 1: *Ridge Regression* using miniBatch Gradient Descent (mini-BGD) algorithm.

- Pick a range of α_0 and λ defined on grid. You can choose fixed $batchsize = 50$.
- Implement k-fold cross-validation protocol for grid search. For each combination of α_0 and λ you will perform k-fold cross-validation. let $k = 5$ in this case.
- Keep track of mean performance (i.e. RMSE value) across $k - folds$ for each set of hyperparameters. Plot on the grid α_0 vs λ the RMSE score for all combinations. [Hint: you can use a 3D plot with axes=α_0, λ, RMSE]
- Finally, for the optimal value of $alpha_0$ and λ, train your model on complete training data and evaluate on test data.
- Plot RMSEtrain and RMSEtest per iteration. [Hint: you can plot RMSEtrain on positive axis and RMSEtest on negative axis of same plot]. Compare your result with results in previous plots.

[Hint: If you were unable to complete Exercise 1, you can still complete Exercise 2 by using linear regression implementation from Exercise Sheet 3 and adding regularization term. There will be some penalty for this.]

3.1 ANNEX

- Following lecture is relevant this exercise https://www.ismll.uni-hildesheim.de/lehre/ml-16w/script/ml-04-A3-regularization.pdf
- You can use numpy or scipy in build methods for doing linear algebra operations
- You can use pandas to read and processing data
- You can use matplotlib for plotting.
- You should not use any machine learning library for solving the problem i.e. scikit-learn etc. If you use them you will not get any points for the task.