TERM PROJECT [ROBT407]

NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

In this project students are required to implement the theoretical knowledge learned about a class of learning machines such as the Linear and Logistic Regression, Support-Vector Machines. This project also aims to give you an important hands on experience in implementation of a specific algorithm and using available tools in ScikitLearn.

PLEASE NOTE: The assignment should be implemented in Python - Jupyter Notebook enviroment.

- RELEASE DATE: 08/10/2018
- DUE DATE: 18/10/2018

METHOD OF DELIVERY

Assignment deliverables should be submitted via Moodle to the course instructor before the due date.

LEVEL OF COLLABORATION ALLOWED

Collaboration is allowed on this assignment – each group should consist of maximum of **two students**. Discussions on course materials and implementation of the project are encouraged. Though, each team should write the final solutions uniquely and understand them fully. External resources can be consulted, but not copied from.

ESTIMATED TIME FOR COMPLETION

60 hours

ASSIGNMENT DELIVERABLES

- A well documented code in Jupyter Notebook implementing algorithm routines.
- Also, Jupyter Notebook report describing in sufficient detail the work of a student – the approach for solving the problem, implementation, results, difficulties, limitations, etc.
- **Statement** in your report clearly stating the contribution of each member in a team.
- **In addition, convert your Jupyter notebook to a PDF file and submit.**

GRADING CRITERIA

- 50% - Implementation (well documented Jupyter Notebook with code cells)
- 10% - The accuracy of the best model that has been selected after cross-validation
- 20% - overall work and report quality
- 20% - discussion (for example of success/failure; limitations, etc.)

1 Linear Regression -20%

Implement the least-squared linear regression algorithm in Section 3.2 of LFD to compute the optimal $(d+1)$-dimensional w that solves

$$\underset{w}{\text{minimize}} \sum_{n=1}^{N} (y_n - (w^\intercal x_n))^2 \tag{1}$$

- Generate a training data set of size 100 as directed by Exercise 3.2 of LFD. Generate a test set of size 1000 of the same nature.
- Run the pocket algorithm (Homework 1) on the training set for $T = 1000$ to get w_{pocket}.
- Run the linear regression algorithm to get w_{lin}. Estimate the performance of the two weight vectors with the test set to get $E_{test}(w_{pocket})$ and $E_{test}(w_{lin})$, in terms of the 0/1 loss (classification).
- Repeat the experiment (with fresh data sets) 100 times and plot $E_{test}(w_{pocket})$ versus $E_{test}(w_{lin})$ as a scatter plot.
- Based on your findings in the previous problem, which algorithm would you recommend to your boss for this data set? Why?

2 Gradient Descent for Logistic Regression -20%

Consider the formulation,

$$\begin{aligned} &\underset{w}{\text{minimize}} \quad E(w) \\ &\text{where,} \quad E(w) = \frac{1}{N}\sum_{n=1}^{N} E^{(n)}(w), \\ &E^{(n)}(w) = \ln\Big(1 + \exp(-y_n\big(2w^{\intercal}x_n\big)\Big) \end{aligned} \tag{2}$$

Implement the fixed learning rate stochastic gradient descent algorithm for Eq.2.

a) initialize a $(d+1)$ -dimensional vector $w^{(0)}$, say, $w^{(0)} \leftarrow (0, 0, ..., 0)$

b) for $t = 1, 2, ..., T$

- randomly pick one n from $\{1, 2, ..., N\}$.
- update

$$w^{(t)} \rightarrow w^{(t-1)} - \eta \nabla E^{(n)}(w^{(t-1)}) \tag{3}$$

- Assume that

$$g_1^t(x) = \text{sign}\Big((w^{(t)})^T x\Big) \tag{4}$$

- where $w(t)$ are generated from gradient descent algorithm above.

  Run the algorithm with $\eta = 0.001$ and $T = 2000$ on the IRIS data set after splitting it into D_{train} (80%) and on the D_{test} (20%) You can use get the IRIS data as follows or from the Scikitlearn

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

  .

- Plot $E_{in}(g_1^{(t)})$ and $E_{test}(g_1^{(t)})$ as a function of t, and briefly state your findings.

3 Practical design of a learning algorithm -60%

- Given training data consisting of input-output pairs, **model selection** in machine learning is a process that builds a model to predict the output from the input usually by learning optimal adjustable hyperparameters. Many models exist in literature to perform such tasks, including linear, logistic models, svms etc. Finding and comparing methods to optimally select models, which will perform best on new test data, is the objective of this assignment.

The following steps summarize the important steps in model selection:

1. Consider a dataset D (from any domain e.g. credit/ medical / digit recognition)
2. Apply an algorithm of your choice (e.g. Lin Reg, Logistic Reg. etc) on D
3. Estimate its generalization error (E_{test})
4. **If**: generalization error smaller than what exists in the literature for the same dataset:
   - End of the process: Outcome
5. **Else**:
   - Go back to step 2 with another algorithm or change the learning strategy

As we have also studies in the model validation lecture (see Texbook section 4.3) the V-fold cross validation is one commonly used method to estimate generalization error (or to perform model selection), especially when there is little training data.

Specific tasks:

- Review the lectures on Hyperparameters and Model Validation (https://goo.gl/ro5WGq) and Feature Engineering (https://goo.gl/MqoCxm)

Task 0: Load Optical Recognition of Handwritten Digits Data Set

```
from sklearn import datasets
digits = datasets.load_digits()
# read the description
print(digits.DESCR)
```

Task 1: Using the Linear & Logistic Regression implemented in Section2 and 3 implement a routine that uses tenfold cross validation for model selection on digits dataset.

Task 2: Perform GridSearchCV based tenfold cross validation using the Scikit-learn module and compare the performance of Linear, Logistic regression and Linear SVMs and polynomial kernel SVMs on digits dataset.

* For each model plot the validation curves using Scikit-Learn to justfy your selection of a final model. Explain shortly how you addressed bias-variance tradeoff, and that your model has not overfitted the digits dataset.

For Task 1 implement the following model selection steps:

1. Write a function that divides the on digits dataset (training) set (of size m) into n disjoint sets $S_1, ..., S_n$ of equal size n/m
2. For each S_i:
   - Train a classifier (e.g. Lin Reg. Log Reg) on $S \backslash S_i$
   - Test it on $S_i \leftarrow error(i)$
3. Output the average error

This gives an estimate of the generalization error of the classifier when trained on $n - n/m$ data. Report the comparative analysis of performance of Linear and Logistic regression and when you change the number of folds in validation (5 fold vs 10 fold vs 20 fold vs loocv). Refer to chapter 4 of the textbook for more details.