Hong Kong University of Science and Technology
COMP 4211: Machine Learning
Spring 2019

Programming Assignment 1
Due: 19 March 2019, Tuesday, 11:59pm

1 Objective

The objective of this programming assignment is twofold:

1. To acquire a better understanding of supervised learning methods by using a public-domain software package called `scikit-learn`.
2. To evaluate the performance of several supervised learning methods by conducting empirical study on three data sets.

2 Major Tasks

The assignment consists of the following tasks:

1. To learn to use the linear regression model for regression.
2. To learn to use the logistic regression model for classification.
3. To learn to use the single-hidden-layer neural network model for classification.
4. To conduct empirical study using different supervised learning methods.
5. To write up a report.

Each of these tasks will be elaborated in the following subsections.

2.1 Regression Method

Linear regression is a basic model for regression which is expressed in the form $f(\mathbf{x};\mathbf{w}) = w_0 + w_1x_1 + \cdots + w_dx_d$, where $\mathbf{w}$ denotes the parameters to be learned from data. Note that this basic model has no hyperparameter to set.

2.2 Classification Methods

2.2.1 Logistic Regression

Learning of the logistic regression model should use a gradient-descent algorithm by minimizing the cross-entropy loss.[1] It requires that the step size parameter η be specified. Try out a few values (<1) and choose one that leads to stable convergence. You may also decrease η gradually during the learning process to enhance convergence. A common criterion used for early stopping is when the improvement between iterations does not exceed a small threshold or when the number of iterations has reached a prespecified maximum.

2.2.2 Single-hidden-layer Neural Networks

Neural network classifiers generalize logistic regression by introducing one or more hidden layers. The learning algorithm for them is similar to that for logistic regression as described above.

For the single-hidden-layer neural network model, the number of hidden units H should be determined using cross validation. The generalization performance of the model is estimated for each candidate value of $H \in \{1, 2, \ldots, 10\}$. This is done by randomly sampling 80% of the training instances to train a classifier and then validating it on the remaining 20%. Five such random data splits are performed and the average over these five trials is used to estimate the generalization performance. The value H^* that gives the best performance among the 10 choices of H can then be found. Subsequently, a neural network classifier with H^* hidden units in a single layer is trained from scratch using all the training instances available. In addition, if you wish, you may learn to use the more powerful `model_selection` submodule in `scikit-learn` to facilitate performing grid search for hyperparameter tuning. Since the solution found may depend on the initial weight values chosen randomly, you may repeat each setting multiple times and report the average classification accuracy.

2.3 Empirical Study

You will use three binary classification and regression data sets which are available as a ZIP file (`datasets.zip`). The following table shows the number of features, number of training examples, and number of test examples for each data set.

Data set	#features	#train	#test
fifa	36	13191	4397
finance	26	2754	918
orbits	12	9642	3215

When you load each `.npz` data file, you will find six `NumPy` arrays.

`train_X`	`classification_train_Y`	`regression_train_Y`
`test_X`	`classification_test_Y`	`regression_test_Y`

[1] For simplicity, you are not required to add regularization terms to the loss functions though you may do it if you wish.

Each row of `X` stores the features of one example and the corresponding row of `Y` stores its class label (0 or 1) for classification, and regression label (0 to 1) for regression. As is always the case, the label files for the test sets should not be used for training but only for measuring the accuracy on the test data.

For each of the three data sets, you will evaluate the following methods with respect to the regression and classification accuracy on the training set and the test set separately:[2]

- Linear regression
- Logistic regression
- Neural network with H^* hidden units (H^* determined by cross validation)

You are expected to also report the time required by each method to complete the task, excluding the time needed for loading the data files. For the linear regression model, you are required to compute the squared error $(f(\mathbf{x};\mathbf{w}) - y)^2$ for each data point in the test set and then plot the distribution of the squared error values as a histogram. For the logistic regression model, you are required to visualize the classification results to depict the performance on both the training and test sets. For the neural network model, you should report the performance of each value of $H \in \{1, 2, \ldots, 10\}$ in the cross validation procedure for determining the best value H^*. Furthermore, you should keep in mind to report the best (i.e., lowest) loss of the neural network model on both the training and test sets before the model is overfitted. For reporting the results of the neural network model, you are required to visualize not only the classification results on the training set and test set after training the model, but also the change in performance on the training set and validation set during training the model.

Your programs should be written in such a way that the TA can run them easily to verify the results reported by you.

2.4 Report Writing

In your report, you are expected to present the parameter settings and the experiment results. Besides reporting the accuracy (for both training and test data) in numbers, graphical aids should also be used to analyze the performance of different methods visually. Note that you may not score high if you fail to provide analysis and visualization of your experiment results. Some utilities in `scikit-learn` such as `auc` and `confusion_matrix` are recommended for reporting the experiment results. For the time required by each method to complete the task, you report it in seconds.

3 Some Programming Tips

As is always the case, good programming practices should be applied when coding your program. Below are some common ones but they are by no means complete:

- Using functions to structure program clearly

[2] You may also try to use single-hidden-layer neural networks for the regression tasks but it is not required for this assignment. Please note that the squared loss should be used for regression tasks.

- Using meaningful variable and function names to improve readability
- Using indentation
- Using consistent styles
- Including concise but informative comments

For `scikit-learn` in particular, you are recommended to take full advantage of the built-in classes which can keep your program both short and efficient. Proper use of implementation tricks often leads to speedup by orders of magnitude. Please be careful to choose the built-in models that are suitable for your tasks, e.g., `sklearn.linear_model.LogisticRegression` is not a correct choice for your second task since it does not use gradient descent.

4 Assignment Submission

Assignment submission should only be done electronically using the Course Assignment Submission System (CASS):

`https://cssystem.cse.ust.hk/UGuides/cass/student.html`

There should be two files in your submission with the following naming convention required:

1. **Report** (with filename `report`): preferably in PDF format.
2. **Source code and a README file** (with filename `code`): all necessary code for running your program as well as a brief user guide for the TA to run the programs easily to verify your results, all compressed into a single ZIP or RAR file. The data should not be submitted to keep the file size small.

When multiple versions with the same filename are submitted, only the latest version according to the timestamp will be used for grading. Files not adhering to the naming convention above will be ignored.

5 Grading Scheme

This programming assignment will be counted towards 12% of your final course grade. Note that the plus sign (+) in the table below indicates that reporting without providing the corresponding code will get zero point. The maximum scores for different tasks are as follows:

Grading scheme	Code (60)	Report (+40)
Empirical study on linear regression		
- Build the linear regression model	2	
- Compute the R^2 score of the linear regression model on both the training and test sets	3	+2
- Depict a histogram of the squared errors of the data points in the test set of the linear regression model	10	+3
Empirical study on logistic regression		
- Build the logistic regression model by adopting the gradient descent optimization algorithm, and present the model settings	5	+2
- Compute the accuracy of the logistic regression model on both the training and test sets	5	+3
- Record and visualize the experiment results of the logistic regression model on both the training and test sets	10	+3
Empirical study on neural network model		
- Build the neural network model by adopting the gradient descent optimization algorithm, and present the model settings	5	+2
- Report the parameter tuning results of the neural network model using cross validation	5	+4
- Compute the best (i.e., lowest) loss of the neural network model on both the training and test sets before the model is overfitted	5	+3
- Record and visualize the experiment results of the neural network model, including performance change over time	10	+3
Writing report		
- Present the computing environment for this assignment		+2
- Present the time required by each method to complete the task		+3
- Compare and analyze the performance of all the regression and classification methods involved		+10

Late submission will be accepted but with penalty. The late penalty is deduction of one point (out of a maximum of 100 points) for every minute late after 11:59pm. Being late for a fraction of a minute is considered a full minute. For example, two points will be deducted if the submission time is 00:00:34.

6 Academic Integrity

Please read carefully the relevant web pages linked from the course website.

While you may discuss with your classmates on general ideas about the assignment, your submission should be based on your own independent effort. In case you seek help from any person or reference source, you should state it clearly in your submission. Failure to do so is considered plagiarism which will lead to appropriate disciplinary actions.