IITM-CS4011 : Principles of Machine Learning — Release date: 18 Aug, 2017, 6:00 pm

Programming Assignment #1 — Deadline: Part (a): 30 Aug, 2017, 11:55 pm
Part (b): 12 Sept, 2017, 11:55 pm

- The goal of this assignment is to experiment with linear methods for classification and regression, logistic regression and backpropagation.
- This is an individual assignment. Collaborations and discussions with others are strictly prohibited.
- You may use Python (recommended), R or Matlab for your implementation. If you are using any other languages, please contact the TAs before you proceed.
- You have to turn in the well documented code along with a detailed report of the results of the experiment electronically in Moodle. Typeset your report in LaTeX.
- Be precise for your explanations in the report. Unnecessary verbosity will be penalized.
- You have to check the Moodle discussion forum regularly for updates regarding the assignment.
- Read the submission instructions carefully before submitting.
- **Please start early.**

Part (a)

Synthetic Dataset Creation

1. You will use a synthetic data set for the classification task. Generate two classes with 20 features each. Each class is given by a multivariate Gaussian distribution, with both classes sharing the same covariance matrix. Ensure that the covariance matrix is not spherical, i.e., that it is not a diagonal matrix, with all the diagonal entries being the same. Generate 2000 examples for each class. Choose the centroids for the classes close enough so that there is some overlap in the classes. Clearly specify the details of the parameters used for the data generation. Randomly pick 30% of each class (i.e., 600 data points per class) as a test set, and train the classifiers on the remaining 70% data When you report performance results, it should be on the left out 30%. Call this dataset as DS1.

Linear Classification

2. For DS1, learn a linear classifier by using regression on indicator variable. Report the best fit accuracy, precision, recall and F-measure achieved by the classifier, along with the coefficients learnt.

k-NN classifier

3. For DS1, use k-NN to learn a classifier. Repeat the experiment for different values of k and report the performance for each value. Technically this is not a linear classifier, but we want you to appreciate how powerful linear classifiers can be.

   - Do you do better than regression on indicator variables or worse?
   - Are there particular values of k which perform better?
   - Report the best fit accuracy, precision, recall and f-measure achieved by this classifier.

Data Imputation

4. For the regression tasks, you will use the Communities and Crime (CandC) Data Set from the UCI repository (http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime). This is a real-life data set and as such would not have the nice properties that we expect. Your first job is to make this dataset usable, by filling in all the missing values.

   - Use the sample mean of the missing attribute. Is this is a good choice?
   - What else might you use? If you have a better method, describe it, and you may use it for filling in the missing data. Turn in the completed data set.

Linear Regression

5. Fit the CandC data using linear regression. Report the residual error of the best fit achieved on test data, averaged over 5 different 80-20 splits, along with the coefficients learnt.

Instructions on how to use 80-20 splits

1. Make 5 different 80-20 splits in the data and name them as $CandC{-}train\,\langle num\rangle\,.csv$ and $CandC - test\,\langle num\rangle\,.csv$.
2. For all 5 datasets that you have generated, learn a regression model using 80% and test it using 20%.
3. Report the average RSS over these 5 different runs.

Regularized Linear Regression

6. Use Ridge regression on the CandC data. Repeat the experiment for various values of λ

- Report the residual error for each value, on test data, averaged over 5 different 80-20 splits, along with the coefficients learnt.
- Which value of λ gives the best fit?
- Is it possible to use the information you obtained during this experiment for feature selection? If so, what is the best fit you achieve with a reduced set of features?

Part (b)

Logistic Regression

7. Download dataset DS2 from here. For this experiment, use only forest and mountain classes in DS2. Perform 2-class Logistic Regression on it. Report per-class precision, recall and f-measure on the same test data you used to test the neural net work. Now perform L_1-regularized Logistic Regression on the same dataset and report similar performance results. Use *l1_logreg* code provided by Boyds Group (`http://www.stanford.edu/~boyd/l1_logreg/`).

Backpropagation

8. Implement original backpropagation algorithm. Use DS2 for training your neural network. Report per-class precision, recall and F-measure on the test data used in Question-1. Now consider the following alternate error function for training neural networks.

$$R(\theta) = \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K}(y_{ik} - f_k(x_i))^2 + \gamma(\sum_k\sum_m \beta_{km}^2 + \sum_m\sum_l \alpha_{ml}^2)$$

where N is the number of training instances, K is the number of output features, $f_k(x)$ is the predicted output vector, y is the original output vector, α and β are the weights and γ is a regularization parameter. Derive the gradient descent update rule for this definition of R. Now train your neural network with this new error function. Report per-class precision, recall and F-measure on the same test data. What will happen when you vary the value of γ? Vary the value of γ from 10^{-2} to 10^{2} in multiples of 10 and repeat the experiment and report the results. Can you figure out the effect of γ in the results? Look at the weights learnt using the new error function. What do you infer from them?

Submission Instructions

Submit a single tar/zip file containing the following files in the specified directory structure. Use the following naming convention: 'rollno_PA1a.tar.gz' / 'rollno_PA1b.tar.gz'.

Note: '*run.py*' script in each code folder should run everything asked and display the results. Turn in coefficients wherever asked in '*coeffs.csv*' file.

```
rollno_PA1a
    Dataset
        DS1-train.csv
        DS1-test.csv
        CandC-train1.csv
        CandC-test1.csv
        ⋮
        CandC-train5.csv
        CandC-test5.csv
    Code
        q1
            run.py
            other code and result files
        ⋮
        q6
            run.py
            other code and result files
    Rollno-report.pdf
    README

rollno_PA1b
    Dataset
        DS2-train.csv
        DS2-test.csv
    Code
        q7
            run.py
            other code and result files
        q8
            run.py
            other code and result files
    Rollno-report.pdf
    README
```