IITM-CS5011 : Introduction to Machine Learning — Given on: Sep 6, 6pm

Programming Assignment #1 — Due on : Sep 20, 6pm

- The goal of this assignment is to experiment with feature extraction methods, linear methods for classification and regression, logistic-regression and Naive Bayes Classifiers.
- This is an individual assignment. Collaborations and discussions with others are strictly prohibited.
- You may use R, Matlab or Python 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.

Synthetic Data-set 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. Specify clearly 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 at 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 I 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 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 above 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.

Regularized Linear Regression

6. Use Ridge-regression on the above data. Repeat the experiment for different 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?

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.

Feature Extraction

7. You have been provided with a 3-dimensional dataset (DS3) which contains 2 classes. Perform PCA on the dataset and extract 1 feature and use the data in this projected space to train linear regression with indicator random variables. Use the learnt model to classify the test instances. Report per-class precision, recall and f-measure. Also you have to report the 3-D plot of the dataset and the plot of the dataset in the projected space along with the classifier boundary.

8. Now use the same dataset and perform LDA on it and project the dataset to the derived feature space. Report per-class precision, recall and f-measure. Also you have to report the 3-D plot of the dataset and the plot of the dataset in the projected space along with the classifier boundary. What do you infer from these two experiments? Which feature extraction technique performs better for this scenario? Why?

Logistic Regression

9. 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/`). The dataset can be found at:
(`https://drive.google.com/open?id=0B5siSlREOlLOa1ZrWE9DRzBud2M`)

Naive Bayesian classifier

10. Design and implement a Bayesian Spam Filter that classifies email messages as either spam (unwanted) or ham (useful), that is, $y_i \in \{\text{spam, ham}\}$ using a Naive Bayes Classifier(explained later) for the following four scenarios:

    - Maximum Likelihood Estimation assuming likelihood $L \sim \text{Multinomial}(n_1, n_2, \cdots, n_k, N)$, where k is the size of the vocabulary, n_w is the number of times word w appears in the document d and $N = \sum_i n_i$
    - Maximum Likelihood Estimation assuming likelihood $L \sim$ Bernoulli (i, p), where p is the parameter of the Bernoulli Distribution and $i \in \{0, 1\}$. In our case, we have k Bernoulli Distributions.
    - Bayesian Parameter Estimation assuming that prior $p \sim \text{Dir}(\alpha_1, \cdots, \alpha_k)$, where $(\alpha_1, \alpha_2, \cdots, \alpha_k)$ are the parameters of the Dirichlet distribution.
    - Bayesian Parameter Estimation, assuming that prior $p \sim \text{Beta}(\alpha, \beta)$, where α and β are the parameters of the Beta distribution.

Naive Bayes Classifier

Naive Bayesian Classifier takes a text as input. The text is just a collection of words. It predicts the category of the input text. Naive Bayes algorithm for text classification involves two stages - training and classification. In the training stage, various probabilities are estimated based on the counts of training example features. In the classification stage, the estimated probabilities are used to evaluate the likelihood of each class for the input text. The text is then assigned a label with the highest likelihood score. Let $C = \{C_1, \cdots, C_m$ be the set of labels and $V = \{w_1, w_2, \cdots, w_n\}$ be all the words in the vocabulary. You would need to estimate the following propbabilities in the training stage:

- Class priors: $p(C_i)$ for $i = 1, 2, \cdots, m$. Note that $\sum_{i=1}^{m} p(C_i) = 1$.
- Within class word probabilities: $p(w_j|C_i)$ for $i = 1, 2, \cdots, m$ and $j = 1, 2, ..., n$. Note that $\sum_{j=1}^{n} p(w_j|C_i) = 1$ for $i = 1, 2, 3, \cdots, m$

Description of the Dataset (Q10):

The data set contains a collection of spam and legitimate emails. Each token (word, number, punctuations, etc.) is replaced by a unique number throughout the dataset. In order to get the emails to an usable by Naive Bayes, some preprocessing is required. Each document should be represented by a term frequency vector of size k, where the i_{th} element is the frequency of the i_{th} term in the vocabulary (set of all distinct words in any text document from the training set). Do not consider the token "Subject:" as part of the vocabulary. Files whose names have the form spmsg*.txt are spam messages. Files whose names have the form *legit*.txt are legitimate examples.

Tasks

- You are required to implement Naive Bayesian classifier for 4 different scenarios as described previously. In the last two cases you are expected to try out different parameters for the prior distribution. Report the results after performing 5-fold cross validation (80-20 split). You have 10 folders in the dataset. Use 1-8 for training, 9-10 for testing in one fold. In the next fold, use 3-10 for training, 1-2 for testing and so on.
- Refer to chapter 13 of Manning, Raghavan and Schutze for further reference on implementation of Naive Bayes for text classification.
- Comment on the impact of choice of parameters on the performance. Also comment on the performance of the classifier under different scenarios. Plot a PR-curve (refer Page 145-146 Manning, Raghavan and Schutze) for each scenario and for different parameter setting in the last 2 scenarios.

- Your code should take $trainMatrix_{p\times k}, trainLabel_{p\times 1}, testMatrix_{r\times k}$ and $testLabel_{r\times 1}$ in the first two scenarios. p and r are number of documents in training and test set respectively and k is the size of the vocabulary. In the last 2 scenarios it should also take the parameters for the prior distribution as input. The code should output precision, recall, f1-measure for spam class and plot PR-Curve for the best model obtained in terms of performance.

Submission Instructions

Submit a single tarball/zip file containing the following files in the specified directory structure. Use the following naming convention: 'cs5011_a1_rollno.tar.gz'.

cs5011_a1_rollno

Dataset
DS1-train.csv
DS1-test.csv
DS2-train.csv
DS2-test.csv
CandC-train1.csv
CandC-test1.csv
CandC-train2.csv
CandC-test2.csv
CandC-train3.csv
CandC-test3.csv
CandC-train4.csv
CandC-test4.csv
CandC-train5.csv
CandC-test5.csv

Report
rollno-report.pdf

Code
all your code files