- The goal of this assignment is to experiment with some of the concepts taught in the class till now.

- This is an individual assignment. Collaborations and discussions with others are strictly prohibited.

- You may use Matlab, Python for your implementation. If you are using any other languages, please contact us 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.

- Your report should contain detailed answer for all of the questions asked below.

- Look at the end of the assignment for submission instructions.

- Submission deadline: 1st October, 2015.

Linear Classifier

1. You will use a synthetic data set for the classification task. Generate two classes with 10 features each. Each class is given by a multivariate Gaussian distribution, with both classes sharing the same covariance matrix. Ensure that the covarianve matrix is not spherical, i.e., that it is not a diagonal matrix, with all the diagonal entries being the same. Generate 1000 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 40% of each class (i.e., 400 data points per class) as a test set, and train the classiers on the remaining 60% data. When you report performance results, it should be on the left out 40%. Call this dataset as DS1.

2. For DS1, learn a linear classifer by using regression on indicator variable. Report the best fit achieved by the classifer, along with the coeffcients learnt.

3. For DS1, use k-NN to learn a classifer. 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 classifers can be. Do you do better than regression on indicator variables or worse? Are there particular values of k which perform better?

Linear Regression

1. 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 complete data set.

2. Fit the above data using linear regression. Report the residual error of the best fit achieved, along with the coeffcients learnt.

3. 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

- Make 5 different 80-20 splits in the data and name them as *CandC-train < num > .csv* and *CandC-test< num >.csv*.
- For all 5 datasets that you have generated, learn a regression model using 80% and test it using 20%.
- Report the average RSS over these 5 different runs.

Feature Extraction

1. 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.

2. 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?

Support Vector Machines

1. You have been provided with training instances for an image classification problem (DS4). You have to train an SVM to classify the test images into either of the following four categories: coast, forest, inside-city, mountain. Follow the instructions below for extracting the features from the images.

Instructions for Feature Extraction

You are given a set of scene images. In this step, the requirement is to extract some features from the images that can be used as input to our SVM. There are many feature extraction techniques. For this assignment, we will follow a color histogram based approach. This is not the best technique for feature extraction, but most likely, the easiest.

- Read image.
- Extract red, green and blue channels from the variable you read into in 1. The sequence is r-g-b.
- For every channel divide it into 32 bins and find frequency.
- Concatenate these 32 dimensional feature vectors for every channel to find a 96D vector for the whole image. (sequence r-g-b)
- Normalize the features before using them.

Use the training data to build classification models using the following kernels.

- Linear kernel
- Polynomial kernel
- Gaussian kernel
- Sigmoid kernel

Come up with the kernel parameters for the various models. You can use a fraction of data supplied to do a n-fold cross validation to find the best model parameters.

Note

- You are expected to use the libsvm library available at `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`. LibSVM has interfaces for matlab and python. It can also be used a standalone tool on the command line.
- Once you learn the svm, you are expected to turn in the model learnt as a matfile or dumped using pickle(if done in python)

Bayesian Parameter Estimation

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

   (a) Maximum Likelihood Estimation assuming likelihood $L \sim \text{Multinomial}(n_1, n_2, \ldots, n_k, N)$, where k is the size of the vocabulary, n_w is the number of times word w appears in the document d and $N = \sum_i n_i$.

   (b) Maximum Likelihood Estimation assuming likelihood $L \sim \text{Bernoulli}(i, p)$, where p is the parameter of the Bernoulli distribution and $i \in \{0, 1\}$. In our case, we would have k Bernoulli distributions.

   (c) Bayesian Parameter Estimation assuming that prior $p \sim \text{Dir}(\vec{\alpha})$, where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$, the elements of the vector are the parameters of the Dirichlet distribution.

   (d) Bayesian Parameter Estimation, assuming that prior $p \sim \text{Beta}(\alpha, \beta)$, where α and β are the parameters of the Beta distribution.

Naive Bayesian Classifier

Naive Bayesian Classifier takes a collection of words as input and predicts the category of the given text. Naive Bayes algorithm for text classification involves two stages : training and classification. In the training stage, various probabilities are estimated based on the training examples. In the classification stage, the estimated probabilities are used to evaluate the likelihood of each class for the given document. The document is then assigned a label with the highest likelihood score. Let $C = \{C_1, C_2, \ldots, C_m\}$ be the set of labels and $V = \{w_1, w_2, \ldots, w_n\}$ 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 $i = 1, 2, \ldots, m$. Note that $\sum_{i=1}^{k} p(C_i) = 1$.
- Within class word probabilities: $p(w_j|C_i)$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. Note that $\sum_{j=1}^{n} p(w_j|C_i) = 1$ for $i = 1, 2, 3, \ldots, k$.

Note- For some words, the conditional probability can end up becoming 0, to prevent this, we perform a "add one smoothing" (`https://en.wikipedia.org/wiki/Additive_smoothing`)

Description of the dataset:

The data set contains a collection of spam and legitimate emails. Each token (word, number, punctuations, etc.) is replaced by a unique number througout 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 ith element is the frequency of the ith 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 tesing and so on.

- 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.

- Refer to chapter 13 of Manning, Raghavan and Schutze for further reference on implementation of Naive Bayes for text classification.

- Your code should take $trainMatrix_{p \times k}$, $trainlabel_{p \times 1}$, $testMatrix_{r \times k}$ and $test-label_{r \times k}$ 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 file containing the following files in the specified directory structure. Use the following naming convention: 'cs5011_pa1_rollno.tar.gz'
data
    DS1_train.csv
    DS1_test.csv
    .
    .
code
    Source Code
report.pdf