EE559 **EE559 Final Project Assignment** **Out Date: 04/10/2014**
Jenkins **Due Date: 05/13/2014**

Introduction

In this project, your goal is to design a classification system, aiming to classify forest cover type with a low error rate. As a part of the project, you are going to implement a number of different classifiers, and try different feature selection or reduction, on the dataset that we supply to you. This dataset includes the labeled training sample set, as well as two testing sample sets that are labeled and unlabeled.

A series of steps are outlined below to help you find a system which gives the (suboptimally) lowest error rate. The classifiers that you will choose to investigate have to follow certain guidelines, which are also given below.

Each part of this project must be done individually. There will be one final project report per student. Detailed guidelines for writing and handing in the final project report will be given in a separate handout.

Dataset：Forest Cover Type

This dataset can be used to predict forest cover type from cartographic variables only (no remotely sensed data). The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains 54 dimensions (features) including binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types). There are 7 forest cover type classes in this database: 1 -- Spruce/Fir; 2 -- Lodgepole Pine; 3 -- Ponderosa Pine; 4 -- Cottonwood/Willow; 5 -- Aspen; 6 -- Douglas-fir; 7 -- Krummholz.

Please see “covertype_info.txt” for more details such as description of each feature if you are interested in it.

In this project, we use 4 subsets from the original dataset which are “training_set”, “testing_set_int_labeled”, “testing_set_final_labeled”, and “testing_set_unlabeled”. In the “training_set”, there are two text documents “train_x.txt” and “train_y.txt”. “train_x.txt” contains 11,340 instances/prototypes (1620 training samples for each cover type class), and each row represents a prototype with 54-dimentional features. “train_y.txt” includes the corresponding labels for the training data. “testing_set_int_labeled” has the similar data representation including 3,780 testing samples (540 testing samples for each cover type class) in “test_x_int_L.txt” with their labels in “test_y_int_L.txt”. “testing_set_final_labeled” also has similar data representation and includes 2,800 testing samples (400 testing samples for each cover type class) in “test_x_final_L.txt” with their labels in “test_y_final_L.txt”.

"testing_set_unlabeled" only has 1,987 testing samples in "test_x_U.txt" but without their true labels known.

Methodology

For most of your work, it is best to use the "training_set" data, and split it as appropriate into validation and training subsets, either by using cross-validation (better if computation time allows), or by implementing a one-pass system with a portion set aside ahead of time as the validation subset. The "testing_set_int_labeled" is an intermediate testing set that can be used after model selection (e.g., parameter optimization) to assess performance of the resulting classifier or feature set. You will likely use this set a number of times. At the end of your project work, once you have finished developing and deciding on the final system, then use the (previously unseen) "testing_set_final_labeled" to estimate performance of your final system on unknowns.

Requirements

In order to design a good pattern recognition system, start with a stepwise analysis of the data, as follows.

1. Baseline performance

To set a reference for the classifier performance, it's necessary to set some type of benchmark. Please use exactly the provided 54-dimensional feature space and run the ***Quadratic Bayes Classifier*** on "training_set". First perform 5-fold cross validation to estimate the performance on the "training_set", and then repeat it 5 times and take the mean as the "baseline". Also, report the standard deviation of your measurement to give a $\pm$range on your mean. Then, test on the "testing_set-_int_labeled", give the accuracy and the confusion matrix.

Please comment on the results obtained.

2. Appropriate Features

(a) Feature Extraction

Note that we have already provided the extracted features in our uploaded dataset, where we have a 54-dimension feature space. The features include Elevation in meters, Slope in degrees, Aspect in degrees azimuth, Wilderness area designation in binary numbers and so on. Again, more details of the features can be found in "covertype_info.txt" as possible reference.

Although we provided the extracted features, it is encouraged to consider some pre-processing or remapping of the 54-dimensional feature as **an option,** which may provide better performance. You need to describe how you manipulate the original feature space to form the new features. You can use classification performance as a guide to evaluate your new features.

(b) Normalization/Scaling

In the discussion session, we mentioned data normalization and techniques such as Extreme Value Normalization and Standard Deviation Measure. In this part, the goal is to evaluate the impact of data normalization on the performance of classifier. You can use baseline classifier for this purpose. Please try at least one data normalization technique (not restricted to these two techniques mentioned above, you can use any data normalization technique) to the original data.

Describe procedure how you normalize the data. Please provide the estimated classification performance using 5-fold cross validation (5 times) on training data. Comment on your observation.

(c) Dimensionality reduction

In this part, the goal is to evaluate the impact of reducing the feature dimension, on the performance of the classifier. You can choose the baseline classifier for this purpose. Pick one or more feature selection/dimension reduction method(s), reduce the feature dimension from original dimension to all the way down to 1 dimension, using an appropriate step size. With the optimal feature dimension from the training data, test on the "testing_set_int_labeled" dataset.

Please turn in the following plots:

- Classification performance (accuracy percentage) vs. dimensionality of feature on training data.
- Scatter plot in feature space for the 2-dimensional case, showing decision boundaries, training data, and testing data.

Compare your results with the baselines results, and try to explain your results as a function of dimension.

3. Comparing classifiers

Part (a):

In this part investigate **at least** three additional classifiers of your choice. You have to include one from **Distribution-free Classification** realm (except SVM), one from **Statistical Classification** realm except the baseline classifier and **Support Vector Machine (SVM)** classifier. For SVM classifier, please at least try both **Linear Kernel** and **RBF Kernel**.

Develop a method for choosing the number of feature dimensions to match the characteristics of each classifier. You can use performance of the classifier as a guide or criterion. For some classifiers you can also estimate the number of degrees of freedom, and number of constraints, as a guide to a reasonable number of parameters and dimensions. You may also want to try normalization techniques on

each classifier.

Hint: for SVM, please consider normalized data for the cross-validation mode ('-v' mode) if you are using libsvm, which may speed up the computation time.

If the classifier has any parameters to be optimized, **maximize its classification** performance by varying (at least one of, but preferably all of) the parameters. Be sure to use a separate validation set, either by using cross validation on the training data or by separating out a validation set from the training dataset before any classifier training or initialization.

Throughout this assignment, in using classifier performance to make choices (e.g., optimizing parameters or choosing among classifiers), adhere to the guidelines given in "Methodology" section above. For example, the final testing set should only be used once, after all decisions have been made and your final pattern recognition system has been specified. This way it will give the best estimate of performance on unknowns.

Part (b):

Examine the performance of different classifiers in two-dimensional feature space (after feature selection/dimensionality reduction to 2 dimensions). Take advantage of the ease of plotting and visualizing the behavior of algorithms in two dimensions. For each of the classifiers you chose, please turn in a scatter plot along with the decision boundary. Include your explanations and interpretation.

Again, for each classifier compare the result with the baseline performance for two dimensions. Comment on the results.

Part (c):

Give a specific description of your best performing system, including any data normalization, the feature set, the classifier, and values of any parameters. Also, for this system, give the following:

(i) Its performance using 5-fold cross validation, averaged over 5 runs (mean and standard deviation) on the training set.

(ii) A confusion matrix and accuracy rate on "tesing_set_int_labeled".

(iii) A confusion matrix and accuracy rate on "tesing_set_final_labeled".

Comments:

Feel free to try different approaches, compare different classifiers, etc. For example, you might try transforming the feature space data in other ways and see if it makes a significant difference. Or, you might try variants of a classifier, or compare different kinds of classifiers.

Throughout this project, where you see a difference in performance (or see no difference where you might have expected a difference), please conjecture as to why; try to understand the results you are observing. You should express your thoughts and explanations in your project report along with your observations.

4. Classification of unlabeled data

In this part, you are going to label the unknown samples ("testing_set_unlabeled" dataset) with your best performing classifier. Please turn in your classification results via DEN Blackboard (and in your hard copy report).

We will test the classification accuracy on this unlabeled dataset. Top 30% for this unlabeled dataset will be given **6pts as extra bonus**; in the middle (30% to 70%) will be given **3pts as extra bonus**;

At the end of the project you must turn in:

(i) Your Matlab code/script (in a "Name_USCID_EE559Project_Code.mat" file)

(ii) Your final classification labels on the unknown "testing_set_unlabeled" dataset (in a "testing_unknown_Label.mat" file)

(iii) A final report (hard copy and pdf file)

Grading Distribution and Criteria

Baseline Performance: 10 pts

Appropriate Feature set: 15 pts

Comparing classifiers: 35 pts

Best performing classifier: 10 pts

Report: 30pts

Extra Credit: 6 pts

Aspects of your project and write-up that will be considered in the scoring include: technical soundness of approach and execution, understanding and interpretation of results, quantity and quality of effort, and clarity and conciseness of report.

Report Write-Up and File Submission

Detailed instructions for the final report write-up will be disseminated later, along with instructions for submission of the *.mat files and pdf file of your report.

Tips

1. For this project you might find PRTools (http://www.prtools.org/) useful, which is based on Matlab. Of course, you may also use the Classification Toolbox we have been using for homework assignments. (Please note, to use the Classification Toolbox you will have to write a small wrapper function to convert two-class functions to multi-class functions).
2. For SVM, we recommend SVM package LibSVM as we used in the homework, which is available from (feel free to use any tools in LibSVM)

   http://www.csie.ntu.edu.tw/~cjlin/libsvm/

   Read the documentation of the package carefully before using it. You can also use any package for SVM that you are familiar with. Please note that in our experience, SVM in PRTools may have some bugs in it.
3. You are required to implement the steps of this assignment using classifiers covered in class, including at least one classifier for each of the 3 topic areas listed above. You are also welcome to try additional classifiers or methods if you like, including ones that have not been covered in class, with the goal of comparing to techniques covered in class.
4. You are welcome to implement this project in Matlab, or in any programming language you are comfortable with.