CSE474/574: Introduction to Machine Learning(Fall 2013)

Instructor: Sargur N. Srihari
Teaching Assistants: Zhen Xu, Yu Liu

Project 2: Classification

Due Date: Monday, December 2, 2013 (11:59pm)

1 Overview

This project is to implement and evaluate classification algorithms. You will implement two different methods of classification and compare their performance with a publicly available package.

You need to write a report and submit all your code through cse_submit command. After you have finished the project, you may need to make a demonstration of your project to the TAs.

2 Classification of Handwritten Numerals

This project is about multi-class classification. The task is handwritten digit recognition where there are ten digits (0-9). Examples of each of the digits are given below.

Figure 1: Examples of each of the digits

You are provided with two types of input data, feature vectors and raw images:

1. GSC features extracted from each of the image: each image is represented by a 512-bit vector, the first 192 are G (gradient), the next 192 are S (structural) and the last 128 are C (concavity). [1]

2. Handwritten digit images: in case you want to experiment with other feature extractors, you are given handwritten images (image type .png). These are segmented images scanned at a resolution of 100ppi and cropped.

[1] The GSC feature extractor was developed at CEDAR/UB. If you are interested in the details see: `http://www.cedar.buffalo.edu/~srihari/papers/JFS2008-color.pdf`

1. **Data set:**

   Training Data
   Both the feature vectors and images can be found at UB learns. Each digit has 2000 samples available for training. Figure 2 shows variant '0's in the data set.

Figure 2: Examples of digit '0's

   Testing Data
   The feature vectors and images correspond to 1500 digit images (150 for each digit). You are expected to predict the labels for the test set and submit a vector of labels.

2. **Classifiers:**

   You are to implement two different classifiers on your own and also choose a publicly available classifier (one among those listed in Appendix A). The three classifiers are to be evaluated and compared using different evaluation metrics (see Appendix B). The two classifiers to implement are:

   (a) Logistic Regression (LR)
   (b) Neural Network (NN)

   You are encouraged to implement other classifiers, such as Naive Bayes, Bayesian logistic regression, *etc*, and find better existing classification packages with comparable performance. You will get extra bonus points in this project for your extra work.

3. **Training and Testing:**

   Training
   Name your training program `"train_lr.m"`, `"train_nn.m"` `"train_blr.m"` and so on. The input of the training programs is an $X = N \times (D+1)$ matrix, where N is the number of the training samples and $D+1$ is the length of each training sample vector consisting of D features and the corresponding classification label. To select the appropriate model parameters for testing, you may want to further decompose the training set into training and validation set and tune your parameters using the validation set. Or you can use cross validation on the training set to select your model.

   Testing
   Once you are satisfied with your model on your training set, for each of the test feature vector provided, the class needs to be predicted using the model you learned. Name your testing program `"test_lr.m"`, `"test_nn.m"`, `"test_blr.m"` and so on. Your testing programs should take a $Y = N' \times D$ matrix as input and output a $T = N' \times 1$ vector of classification labels. In testing phase N' is the size of the testing data set.

4. **Optional:**

Extract Features from Data: The accuracy of hand written digital recognition is determined by both the feature extraction and classification methods. Before you use the classification method to train the data, you should process the original images into a matrix containing category labels (the first column) and feature vectors. Since you are already provided with the extracted features, you can start with the provided feature vectors, but you are encouraged to develop your own features and compare the results obtained from your feature with the one that from provided features. This input matrix will be feed into your classification model.

Submission:

Project Report: Write a project report where you explain your model, the intuitive choice of methods and parameters. Your report should discuss and compare the performance of each of the methods implemented using appropriate evaluation metrics in Appendix B. Discuss the relative role of the features and classifiers on performance. Additional grading considerations will include creativity in choice of models, accuracy and completeness in interpreting your statistics, and the clarity and flow of your report.

Matlab code: Matlab functions `"train_lr.m"`, `"train_nn.m"` ..., that train the classifier and `"test_lr.m"`, `"test_nn.m"`, ..., that predict the labels in order for us to verify that your code produces the labels you submitted. One script or .m file for other models should be submitted, for example, nn_model.m for neural network, gp_model.m for Gaussian process, svm.** (appropriate suffix for different programming languages) for SVM, dl_model.py for deep learning. You should also include short usage/information about arguments, return values of your functions in your report or as comment in your code in order for us to verify that your code produces the results you submitted.

Results: The final sets of class labels for each of the test feature vector. Submit these as a text files `"classes_lr.txt"`, `"classes_nn.txt"`, ..., containing all the class labels.

All the above mentioned files should be submitted via the CSE submit script:
`submit_cse474 yourfiles` (for undergraduate students)
`submit_cse574 yourfiles` (for graduate student)

A Appendix: Downloadable Classification Packages

A.1 Neural Network

You can use the Matlab's neural network toolbox to do the classification and compare the results. The link is `http://www.mathworks.com/help/nnet/index.html`. There are some explanations and examples under this link. The neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior. It is an adaptive system that changes its structure based on training samples that flows through the network.

A.2 Gaussian Process

You need to download and use the Gaussian process for machine learning code to do the classification and compare the results. The link is `http://www.gaussianprocess.org/gpml/code/matlab/doc/`. The Gaussian process is a generalization of the Gaussian probability distribution to functions. It is a collection of random variables, any finite number of which have a joint Gaussian distribution. In Gaussian process, the function is treated as a very long vector, in which each element is specified by the function value $f(x)$ for the input x. It may take a long time to do the classification.

A.3 Support Vector Machine

You need to download and use the Support Vector Machine (SVM) code to do the classification and compare the results. LIBSVM is an integrated software for support vector classification, regression and distribution estimation. It supports multi-class classification. The link is `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`. There are also some explanation and examples in this link. SVM constructs a hyperplane or set of hyperplanes in a high dimensional space. It is widely used in machine learning areas for classification, regression, or other tasks.

A.4 Deep Learning

You need to download and use the Deep Learning code to do the classification and compare the results. The link is `http://deeplearning.net/tutorial/`. There are also some explanation and examples in this link. Deep Learning is a new area of Machine Learning research. It is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text.

B Appendix: Evaluation Metrics

B.1 Error Rate

The percentage of error labeled test points. For example, if you have 1500 digit images, after applying classification method, your algorithm has wrongly labeled 150 digit images. Then the error rate is $150/1500 = 10\%$.

B.2 Reciprocal Rank

The reciprocal rank is the multiplicative inverse of the rank of the first correct classification label. For example, for a specified digit image, for which the ground truth is 0, if your algorithm ranks it as the first, then the *reciprocal rank* $= 1/1 = 1$. If it is ranked as the 3rd one, then the *reciprocal rank* $= 1/3$.