ECS797 – Machine Learning for Visual Data Analysis

Lab 3: Age Estimation by Regression

Yi-Chin, Hsieh 210830653

2.1 Read in the training and test data by loading the ".mat" file

```
% path
database_path = './data_age.mat';
result_path = './results/';

% initial states
absTestErr = 0;
cs_number = 0;


% cumulative error level
err_level = 5;

%% Training
load(database_path);
```

1. set "database_path" and "result_path"

2. initial "absTestErr" and "cs_number"

3. set cumulative error level

   err_level: 5

4. load the mat file from "database_path"

2.2 Call the Matlab built-in function "regress()", which takes the training features and labels as input and learn a linear regression model.

nTrain: number of training samples, 500

nTest: number of testing sample, 502

xtrain: feature, 500*201

ytrain: label, 500*1

```
nTrain = length(trData.label); % number of training samples
nTest  = length(teData.label); % number of testing samples
xtrain = trData.feat; % feature
ytrain = trData.label; % labels

w_lr = regress(ytrain,xtrain);
```

Training:

1. Take the face AAM features as input
2. Learn a regression function to map the input to the known age

2.3 Read in the test data, and apply the learned linear regression model to estimate the age for each test data point

xtest: feature, 502*201

ytest: label, 502*1

```
%% Testing
xtest = teData.feat; % feature
ytest = teData.label; % labels

yhat_test = xtest * w_lr;
```

Testing:

1. Extract AAM features
2. Predict the age using the learned regression function

2.4 Compute the MAE and CS value (with a cumulative error level of 5) by comparing the estimated ages with the ground truth ages.

```
absTestErr = abs(yhat_test-ytest);
mae = sum(absTestErr)/size(ytest,1);
fprintf('MAE = %f\n',mae);
cs = sum(absTestErr <=err_level >0)/size(ytest,1);
fprintf('CS = %f\n',cs);

MAE = 7.704359
CS = 0.412351
```

- MAE (mean average error)

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error
y_i = prediction
x_i = true value
n = total number of data points

In this case, x is the ground truth ages. y is the estimated ages. And n is the number of testing samples.

- CS (cumulative score)

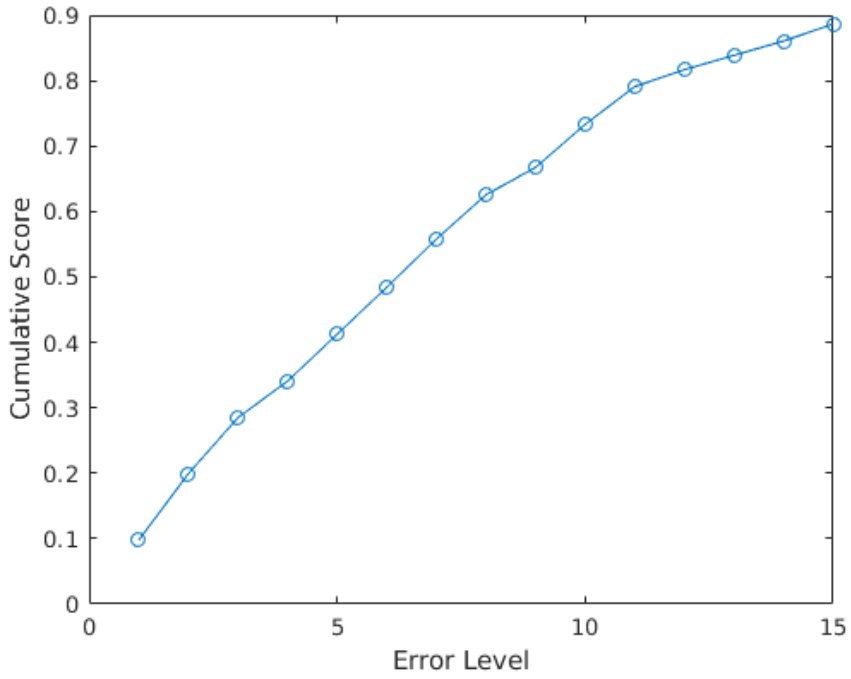
$$\mathrm{CS}(j) = N_{e \leq j} / N \times 100\%$$

In this case, N is the absolute error of testing samples, and j is the cumulative error level.

2.5 Vary the cumulative error level from 1 to 15 and generate a plot of the CS value against the cumulative error level.

```
y = zeros(1,15); %cs
for i=1:15
    y(i) = sum(absTestErr <=i >0)/size(ytest,1);
end
plot(1:15,y,'-o');
xlabel('Error Level');
ylabel('Cumulative Score');
```

When the cumulative error increases, the cumulative score rises. Because [***].



2.6 Compute the MAE and CS values (with cumulative error level of 5) for both partial least squares regression model and the regression tree model by using the Matlab built-in functions.

- Partial least squares regression model
  - [XL, YL, XS, YS, beta, PCTVAR, MSE, stats] = plsregress(X, y, Ncomp);
  - Input
    - x: predictor variable
    - y: response variable

        - Ncomp: number of components
    - Output
        - XL: predictor loadings
        - YL: response loadings
        - XS: predictor scores
        - YS: response scores
        - beta: coefficient estimates for PLS regression
        - PCTVAR: percentage of variance
        - MSE: mean squared error
        - stats: model statistics

```
%partial least square regression
disp('partial least square regression');
[XL, YL, XS, YS, beta, PCTVAR, MSE, stats] = plsregress(xtrain,ytrain, 10);
yhat_test_plsr = [ones(size(xtest,1),1) xtest]*beta;
absTestErr_plsr = abs(yhat_test_plsr-ytest);
mae_plsr = sum(absTestErr_plsr)/size(ytest,1);
fprintf('MAE = %f\n',mae_plsr);
cs_plsr = sum(absTestErr_plsr <=err_level >0)/size(ytest,1);
fprintf('CS = %f\n',cs_plsr);
```

```
partial least square regression
MAE = 6.070294
CS = 0.525896
```

- The regression tree model
    - A decision tree with binary splits for regression.
    - fitrtree: returns a regression tree.
    - predict: returns the predicted response values.

```
%regression tree model
%disp('regression tree model');
tree = fitrtree(xtrain, ytrain);
yhat_test_rtm = predict(tree, xtest);
absTestErr_rtm= abs(yhat_test_rtm-ytest);
mae_rtm = sum(absTestErr_rtm)/size(ytest,1);
fprintf('MAE = %f\n',mae_rtm);
cs_rtm = sum(absTestErr_rtm <=err_level >0)/size(ytest,1);
fprintf('CS = %f\n',cs_rtm);
```

```
regression tree model
MAE = 8.235005
CS = 0.503984
```

2.7 Compute MAE and CS values (with cumulative error level of 5) for Support Vector Regression by using LIBSVM toolbox (n by using the LIBSVM toolbox (http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

- LIBSVM

1. Download LIBSVM from https://www.csie.ntu.edu.tw/~cjlin/libsvm/

2. Unzip it and put it in the same file as the Matlab file
3. Change path to “labsvm-3.25/matlab”
4. Input “make” in the command window to set up the interface of LIBSVM
5. Back to “lab3” file
6. Use “svmtrain” to train the model
7. Use “svmpredict” to make the prediction

```
%LIBSVM
addpath(genpath('libsvm-3.25/matlab'));
disp('Support Vector Regression by using LIBSVM toolbox');
%Training inpur: training label, training feature, libsvm options
model = svmtrain(ytrain, xtrain, '-s 3 -t 0');
%Testing input: testing label, testing feature, model, libsvm options
yhat_test_libsvm = svmpredict(ytest, xtest, model);
absTestErr_libsvm = abs(yhat_test_libsvm-ytest);
mae_LIBSVM = sum(absTestErr_libsvm)/size(ytest,1);
fprintf('MAE = %f\n',mae_LIBSVM);
cs_LIBSVM = sum(absTestErr_libsvm <=err_level >0)/size(ytest,1);
fprintf('CS = %f\n',cs_LIBSVM);
```

```
MAE = 5.623867
CS = 0.565737
```