# Contents

[Contents](#h.uajqwvplen87)

[Introduction](#h.fm59t19tqb3k)

[Emotion Data](#h.3w7pr9a80k35)

[Preprocessing](#h.ugim1wihtkr8)

[Correlation Based Feature Selection](#h.mmqvaf4cdlgk)

[Principal Component Analysis](#h.zakx5m20w6l4)

[Constructing Trees](#h.gjnghx1el9kk)

[Implementing early stopping](#h.agwquom3alrf)

[10-fold cross-validation](#h.earcv1b6gai2)

[Results](#h.q8ezxyqjoye1)

[Additional Questions](#h.a7uacqhacxfl)

[Challenges](#h.ydcy2lo41b22)

[Conclusion](#h.nb37ls6n3ipm)

[Appendix A:  Tools, Sources and Scripts](#h.1dg79eq05q2y)

[Materials](#h.9ansa151px9b)

[Files attached:](#h.1yytxl1eh7ge)

# 

# Introduction

The purpose of this lab was to compare different methods of dimensionality reduction.  
Correlation Based Feature Selection (CFS) was implemented from scratch. A built in MATLAB Principal Component Analysis (PCA) was used. Both methods were used to create a new dataset from the emotions data supplied for the lab, with fewer dimensions than the original.

**Objectives**

The objective was to compare the performance of both methods of dimensionality reduction.

# Emotion Data

Data provided in emotions\_data.mat file contains

1. A matrix x of dimensions 612×136 , (There are 612 pieces of emotion data and 136 is the dimensionality of the feature vector computed by concatenating the  x and y coordinates of 68 facial points).
2. A vector y of dimensions 612×1, containing the emotion labels of the corresponding examples. These labels are numbered from 1 to 6 and correspond to an emotion follows:

1=anger, 2=disgust, 3=fear, 4=happiness, 5=sadness and 6=surprise

# Preprocessing

## Correlation Based Feature Selection

A function was implemented as described by section 9a of the Lab Manual. This essentially picks the features from the training data that has the strongest correlation to the training labels.

CFS makes use of correlation information between features and targets as well as between features. The idea is that features which are more correlated to the target are more likely to affect classification results and we remove redundancy by eliminating features which are least correlated with the target and have bigger correlation to other features. This theoretically results in a smaller list of features to build the decision tree on.

In short, we are looking for features which have more correlation with the targets and less correlation with other features1.

In CFS, no projections are made, only selection of attributes.

## Principal Component Analysis

PCA was applied directly to all emotion data using the built in pca function of matlab. The minimum number of new features that covers 95% of the ‘energy’ of the original data was calculated as 43.

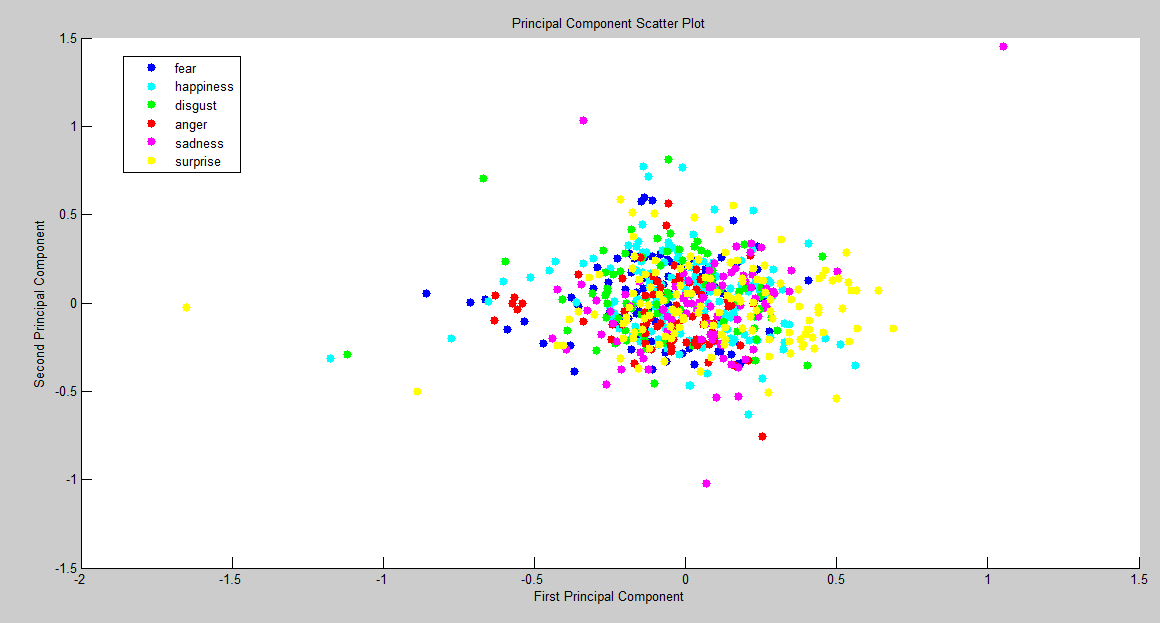


Figure: Scatter plot of first two principal components

# Constructing Trees

## Implementing early stopping

Several early stopping methods were tested such as stopping tree branching when information gained approached zero and when same features were selected as best successively. After observing that some branches contained many nodes with the same classification with one outlier, we decided to stop the growth of such branches. Below are the decision trees constructed with and without using early stopping:

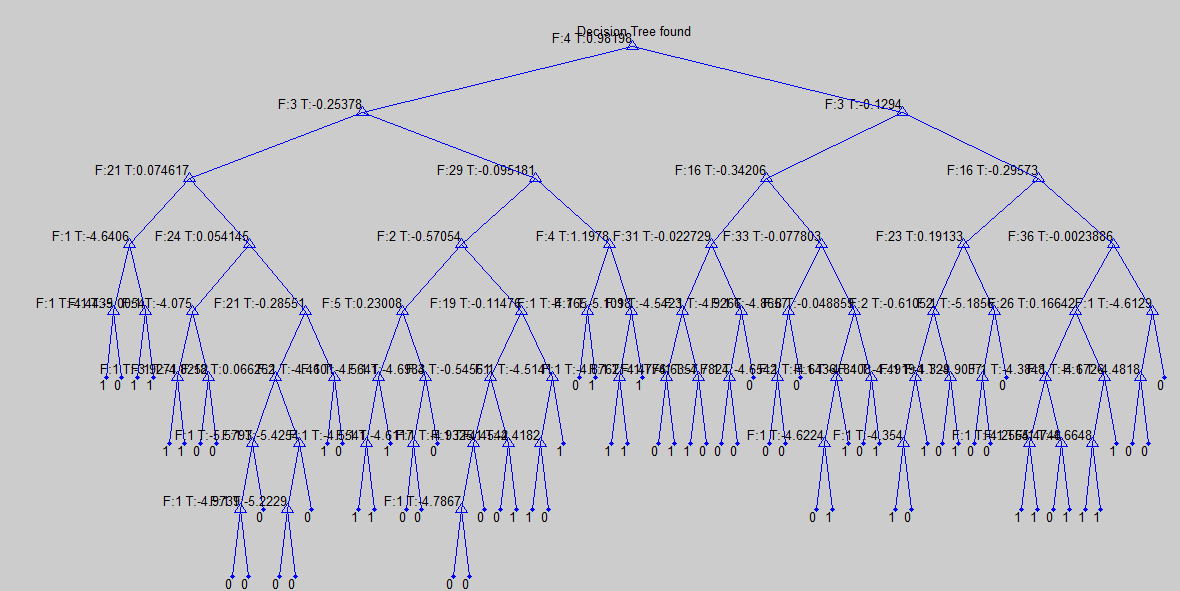


Figure: Emotion 4 tree constructed using PCA applied data and early stopping

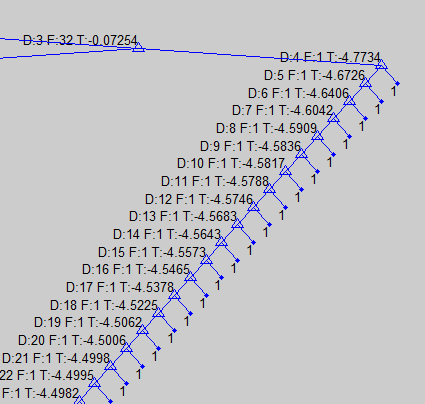
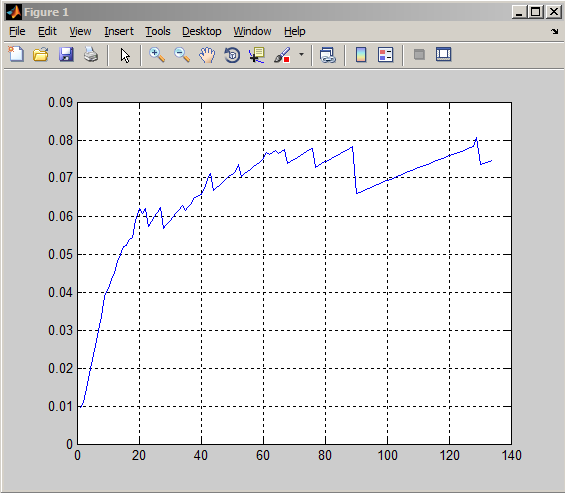


Figure: A branch of Emotion 4 tree of PCA applied data constructed without using early stopping

# 10-fold cross-validation

**Correlation Based Feature Selection**

As we had issues with getting CFS to give us a smooth curve which we can select features from, we decided to take the first 100 features with highest correlation and run it through cross-validation anyway since we were running out of time. The results were as expected not as accurate as using the original features. 

However we managed to get our CFS algorithm to somewhat work at a very late stage which meant we had to skip tree generation. For emotion 1, we interpreted the plot of CFS to indicate that the first 68 features were the most significant and hence could have been used for selection using C4.5 instead of the original 137 features for each emotion. As a result of power-cuts (excuses, excuses), we didn’t have time to run the CFS script over all six emotions.

**Principal Component Analysis**

PCA was applied to data directly. The minimum number of new features, k, that covers 95% of the ‘energy’ of the original data was calculated as 43. New matrix with reduced dimensionality was reconstructed in original space by using the formula below:

x\_reduced = x\*COEFF(:,1:k);

where x is our original data and COEFF is the principal components matrix.

This matrix with reduced dimensionality was then used in training and testing of the decision tree using cross validation technique.

# Results

**Correlation Based Feature Selection**

**Test results of decision trees constructed using all samples:**

Confusion matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **61** | 3 | 3 | 4 | 1 | 2 |
| **2** | 62 | **2** | 5 | 5 | 2 | 8 |
| **3** | 53 | 5 | **9** | 11 | 6 | 6 |
| **4** | 98 | 7 | 12 | **6** | 9 | 11 |
| **5** | 60 | 6 | 4 | 3 | **5** | 6 |
| **6** | 100 | 7 | 4 | 8 | 8 | **10** |

Data is from “CFSTreeResults.mat”

Evaluation Statistics:

Accuracy is 38.4%.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **Specificity** | 0.3067 | 0.9470 | 0.9464 | 0.9340 | 0.9508 | 0.9305 |
| **Recall** | 0.8243 | 0.0238 | 0.1000 | 0.0420 | 0.0595 | 0.0730 |
| **Precision** | 0.1405 | 0.0667 | 0.2432 | 0.1622 | 0.1613 | 0.2326 |
| **F score** | 0.2402 | 0.03509 | 0.1417 | 0.0667 | 0.0870 | 0.1111 |

**Principal Component Analysis**

**Test results of decision trees constructed using all samples:**

Confusion matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **74** | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | **84** | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | **90** | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | **143** | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | **84** | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | **137** |

Data is from “pca\_fulltrees.mat”

**Test results of decision trees constructed using all samples and early stopping described earlier in our report:**

Confusion matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **69** | 2 | 0 | 2 | 1 | 0 |
| **2** | 19 | **59** | 2 | 2 | 0 | 2 |
| **3** | 19 | 5 | **55** | 1 | 5 | 5 |
| **4** | 25 | 14 | 16 | **68** | 9 | 11 |
| **5** | 25 | 9 | 4 | 0 | **45** | 1 |
| **6** | 35 | 23 | 12 | 10 | 9 | **48** |

Data is from “pca\_fulltrees\_earlystopping.mat”

**Cross validation results of trees constructed using early stopping method described earlier in our report:**

Confusion matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **43** | 7 | 4 | 3 | 9 | 8 |
| **2** | 43 | **20** | 5 | 6 | 3 | 7 |
| **3** | 43 | 3 | **20** | 10 | 4 | 12 |
| **6** | 66 | 11 | 13 | **38** | 6 | 9 |
| **5** | 42 | 12 | 3 | 5 | **14** | 8 |
| **6** | 71 | 12 | 14 | 5 | 7 | **28** |

Evaluation statistics:

Accuracy is 42%.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **Specificity** | 0.5111 | 0.9147 | 0.92529 | 0.9381 | 0.9450 | 0.9073 |
| **Recall** | 0.5810 | 0.2380 | 0.2222 | 0.2657 | 0.1666 | 0.2043 |
| **Precision** | 0.1405 | 0.3076 | 0.3389 | 0.5671 | 0.3255 | 0.3888 |
| **F score** | 0.2263 | 0.2684 | 0.2684 | 0.3619 | 0.2204 | 0.2679 |

Data is from “pca\_crossvalidation\_earlystopping.mat”

**Cross validation results of trees constructed without early stopping:**

Confusion matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **1** | **18** | 13 | 7 | 14 | 16 | 6 |
| **2** | 10 | **28** | 9 | 11 | 13 | 13 |
| **3** | 8 | 16 | **31** | 15 | 8 | 12 |
| **4** | 8 | 11 | 17 | **84** | 10 | 13 |
| **5** | 16 | 14 | 12 | 8 | **26** | 8 |
| **6** | 7 | 15 | 12 | 12 | 8 | **83** |

Evaluation statistics:

Accuracy is 48%.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **Specificity** | 0.9089 | 0.8692 | 0.8908 | 0.8720 | 0.8958 | 0.8905 |
| **Recall** | 0.2432 | 0.3333 | 0.3444 | 0.5874 | 0.3095 | 0.6058 |
| **Precision** | 0.2686 | 0.2886 | 0.3522 | 0.5833 | 0.3209 | 0.6148 |
| **F score** | 0.2553 | 0.3093 | 0.3483 | 0.5853 | 0.3151 | 0.6102 |

Data is from “pca\_crossvalidation.mat”

# **Additional Questions**

**Question 1: Explain why you needed to repeat CFS six times, but could apply PCA only once.**

CFS requires measurement of correlation of features to the target value. This correlation can only be measured on a single target at a time.   
  
Repeating CFS six times allows the pre-processing system to select data features specific to a certain emotion. For example, coordinates around the upper portion of the face will not change much when smiling, but will do so greatly when an emotion of surprise is present. Selecting such specific data features is only possible when running CFS once for each emotion.

PCA analyzes the correlation matrix of the set of features in our data. Principal components are determined without regarding their relation to the targets. As such it is enough to apply PCA to data once.

**Question 2: Why can’t you directly infer what features are most informative after applying PCA?**

PCA selects variables according to magnitude ( Largest to smallest) of variables’ coefficient. So PCA seeks to replace p (more or less correlated) variables by k<p uncorrelated linear combinations (projections) of the original variables.

So to choose an optimal k features ( out of 136, the original features), those k principal components are ranked by importance using their explained variance, and each variable contributes with varying degree to each component.[2]

The problem with using PCA is that measurements from all of the original variables are used in the projection to the lower dimensional space. So it's not that PCA "selects" the most important features but rather it finds linear combinations of existing variables and the user decides how many of those new combinations to keep. [3]

**Question 3: How can you use PCA to analyse latent variables affecting the variance in your data?**

PCA is used to find the latent structure of data to obtain a reduced dimensional representation of it. PCA involves calculating the covariance matrix of centralized data and then calculating its eigenvectors. First k principal components are selected that cover enough variance in data. Principal components are the underlying latent variables that affect the variance in our data.

**Discussion**

One advantage of implementing dimensionality reduction techniques is that they reduce the computation time greatly.

We observed that the test results for trees built with PCA applied data were less accurate than the results in our last lab. This was because when we reconstructed our data back to its original space with reduced features, we lost some of the information in the original data.

# Challenges

Implementing early stopping conditions was challenging. We tried to limit tree growth when information gain approached zero or when the successively selected features were identical. Both methods caused a drop in accuracy.

Figuring out the CFS algorithm was challenging. We initially did not get an ascending value for maximum CFS, even when we did, we couldn’t find a peak. We suspect it has something to do with the way we calculate the sum of correlations between features and only somewhat managed to figure it out close to submission time. Both the built-in MATLAB correlation function corr() and our custom written pearsonPMCC() returned seemingly random NaN for some input arrays. This caused issues when processing outputs, as MATLAB will generally return NaN from wny function where the input array contains NaN. We attempted to rectify this by setting all NaNs to zero. However, this produced non-continuous results at best, which were hard to analyse.

After implementing pruning on our previous decision tree algorithm, results were not as crisp as before and the tree even failed at classifying its own training data. Although it is understood that this can be the result of generalization(which is what we would like to achieve), it is difficult to judge by just looking at the confusion matrix if we had also gotten rid of crucial features during dimensionality reduction.

# Conclusion

The objective of dimensionality reduction is to work around the curse of dimensionality. This is usually done to better visualize the data and to assist in a shorter training time. However, there is a risk of losing information if not done carefully which leads to a loss in test accuracy.

in CFS, we remove features which do not correlate to the target and retain features which do and not do any transformation to the values.

in PCA, we do a projection to find better correlation with the target which means changing the features entirely.

**References**

1. Correlation-based Feature Selection for Machine Learning - Mark A. Hall <http://www.cs.waikato.ac.nz/~mhall/thesis.pdf>
2. Using PCA for Features Selectio <http://stats.stackexchange.com/questions/27300/using-principal-component-analysis-pca-for-feature-selection>
3. Using PCA to select Most Important Features http://stats.stackexchange.com/questions/132976/does-pca-mean-selecting-most-important-features-and-ignoring-the-others

# Appendix A:  Tools, Sources and Scripts

## Materials

1. Matlab R2014a
2. Data from provided emotions\_data.mat file

## Files attached:

**Scripts folder:**

1-convertNum.m

2-convert1D.m

3-ConstructDecisionTree.m - constructs trees without early stopping

4-ConstructDecisionTreev2.m - constructs trees with early stopping technique

4-ChooseAttribute.m

5-Entropy.m

6-TestTree.m

7-TestSample.m

8-ConfusionMatStats.m

9-PCA.m - displays scatter plot of the first two principal components

10-PCAKfold.m - Cross validation of pca applied data

11-CFSFun4.m - Calculating a CFS array and outputting selected relevant features.

**Data folder:**

1-pca\_fulltrees.mat

2-pca\_fulltrees\_earlystopping.mat

3-pca\_crossvalidation\_earlystopping.mat

4-pca\_crossvalidation.mat

5-CFS emotion1 results.mat

# 