Title here

Christian M. Lillelund, *201408354, Aarhus University*

***Abstract*—X**

***Index Terms*—IEEE, IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

For humans, it doesn't take much effort to tell the difference between a picture of a dog or a cat. A natural number or a letter. A happy person or a sad person. For computers, these sort of problems are notorious hard to solve and often require many computational resources. Machine learning and computer vision deals with these issues as they encompass a range of algorithms and classification techniques to produce a model or scheme that can tell images apart and recognize similarities. In this report we study the recognition and classification of a data set containing human faces and one featuring handwritten numbers by implementing and testing five commonly used techniques (Nearest class centroid classifier, nearest subclass centroid, nearest neighbor and two perceptron variants) using MATLAB. We split the data in a training and a test set, then train our respective model or classifier and evaluate their ability to classify correctly on the testing set. A version using all the dimensions of the data and a principal component version will be applied. We use some visualization techniques to better communicate the data representation and tables to compare them. We start by going over the basic theory behind the classification schemes, then look at the data, briefly go over implementation details and then turn to results. At the end we review and argue which scheme would make the most sense to use with these two classification problems.

II. Theory

This section will explain fundamental theory behind the dimensional reduction technique (PCA) and five schemes used in this report.

A. Principal Component Analysis

PCA is a procedure that transform a set of observations or samples in dimension D to a lower dimension $D-n$ while still preserving a smaller number of variables explaining the main features $X_1, X_2, ..., X_p$ in the original set. This is particularly useful when dealing with high dimension data, as this can be computationally hard and challenging to visualize. With PCA, we compute principal components d of n original samples with p features, where d is the desired output dimensionality and each dimension is a linear combination of the p features. In practice, we find eigenvectors of the covariance matrix of the original data set, sort them by highest eigenvalue score and use these as weights W in computing a linear transformation

$$y_i = W^T * x_i, i = 1, ..., N$$

. The scattering of the transformed data is the scatter matrix, a function of X:

$$S_T = \sum_{i=1}^{N} [W^T(x_i - \mu)][W^T(x_i - \mu)]^T$$

. The weights W can be found by applying eigenanalysis and taking the eigenvectors with the highest score, more formally optimizing:

$$W* = \arg\max_c Tr(W^T S_T)$$

subject to $W^T W = I$. We end with a data set containing fewer ($d < D$) dimensions.

B. Nearest class centroid classifier

The NCC classifier assigns labels l_i to N observations determined by which class c_k's mean (centroid) the observation x_i is closest to. We make the assumption that each class follow a normal distribution, as they are given equal importance in the classification algorithm. The mean class vector is given by:

$$\mu_k = \frac{1}{N_k} \sum_{i, li=k} x_i, k = 1, ...K.$$

To classify any observation x_i we find the smallest distance to any mean vector and assign the label of that vector to it, more formally:

$$d(x_i, \mu_k) = ||x_i - \mu_k||_2^2$$

C. Nearest subclass centroid classifier

Manuscript received April 19, 2005; revised August 26, 2015.