Machine Learning
Assignment 3
DIKU, Fall 2016

Sune Hellfritzsch
fjc657@alumni.ku.dk

December 11, 2016

1 Summarization by the mean

We're tasked with proving that the mean vector b minimizes the mean-squared error, i.e., the function

$$\text{argmin}_{b\in\mathbb{R}^d}\ \frac{1}{N}\sum_{i=1}^{N}\|x_i - b\|^2$$

is minimized by

$$b = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad .$$

First, we expand the sum of the objective function to get an overview of what is happening:

$$\frac{1}{N}\left(\|x_1 - b\|^2 + \cdots + \|x_N - b\|^2\right)$$

where x_i and b are d-dimensional vectors. Then, we remind ourselves that the norm of a vector is defined as $\|u\| = \sqrt{u^T u}$. Hence, the function to be minimized can be written as

$$\frac{1}{N}\Big((x_1 - b)^T(x_1 - b) + \cdots + (x_N - b)^T(x_N - b)\Big) \quad .$$

Next, each of the dot products can be expanded, such that for each datapoint, we get a sum of terms each depending on the d'th element of the mean vector:

$$\frac{1}{N}\left(\underbrace{\left((x_{11} - b_1)^2 + \cdots + (x_{1d} - b_d)^2\right)}_{(x_1-b)^T(x_1-b)} + \cdots + \underbrace{\left((x_{N1} - b_1)^2 + \cdots + (x_{Nd} - b_d)^2\right)}_{(x_N-b)^T(x_N-b)}\right) \quad .$$

Now, we expand each squared term. For the sake of overview an simplicity, we only expand the term $(x_1 - b)^T(x_1 - b)$.

$$\frac{1}{N}\left(\left(x_{11}^2 + b_1^2 - 2x_{11}b_1 + \cdots + x_{1d}^2 + b_d^2 - 2x_{1d}b_d\right) + \ldots\right) \quad .$$

This objective function is a function of the d-dimensional vector b, and we find the minimum of this as

$$h'(b) = \underline{0}$$

where the derivative with respect to b is set equal to the zero vector, which is also d-dimensional. Thus, this equation is solved as

$$\begin{pmatrix} \frac{\partial h}{\partial b_1} \\ \vdots \\ \frac{\partial h}{\partial b_d} \end{pmatrix} = \underline{0}$$

with each partial derivative solved independently, as they do not depend on the same d'th element of the mean vector. The first equation can be solved as

$$\left(\frac{2}{N}b_1 - \frac{2}{N}x_{11}\right) + \cdots + \left(\frac{2}{N}b_1 - \frac{2}{N}x_{N1}\right) = 0 \quad \Longrightarrow \quad \frac{1}{N}\sum_{i=1}^{N} b_1 = \frac{1}{N}\sum_{i=1}^{N} x_{i1}$$

corresponding to the first element of the mean vector. The rest can be solved in the same manner, and will result in an optimizing vector equal to b.

2 PCA for high dimensional data and small samples

We're tasked with proving that if a vector v is an eigenvector of the matrix $X_0 X_0^T$, with eigenvalue λ, then $X_0^T v$ is an eigenvector of $X_0^T X_0$.

We remind ourselves that a vector, v is an eigenvector of a matrix A, iff.

$$Av = \lambda v \quad .$$

Thus, if

$$X_0 X_0^T v = \lambda v$$

we can obtain the following by multiplying by X_0^T on both sides of the equal sign

$$X_0^T X_0 (X_0^T v) = \lambda (X_0^T v)$$

which concludes our prove.

3 The Traffic Sign Recognition Data

In this section I've used Python version 3.5.2, along with the following Python libraries: `cvs` for reading in the data, `numpy`, for data manipulation, and `matplotlib`, for plotting.

3.1 Data understanding and preprocessing

For this you could go the long way around, as I did initially, and group all the occurences of categories manually. But `matplotlib` has a function named `hist()` that can do this for us, and with the argument `normed=True` it also displays this result as frequencies. The result is displayed in Figure 1.

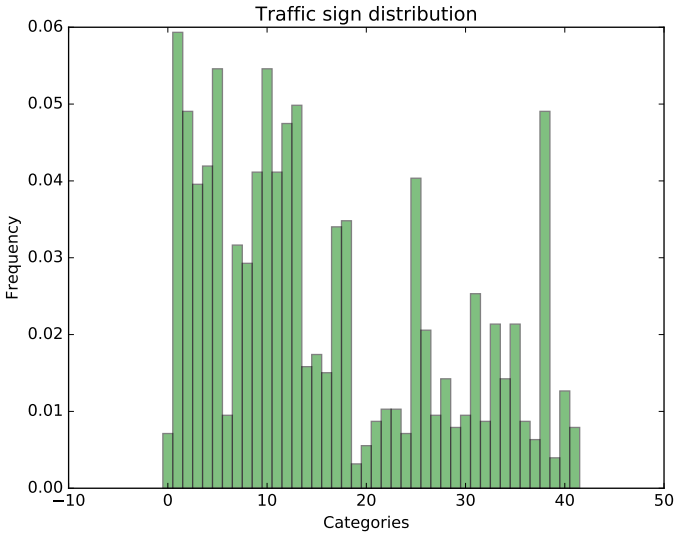


Figure 1: Distribution of traffic sign categories.

3.2 Principal component analysis

Again, I wrote the `numpy.cov()` myself before realizing that it existed. The built-in is faster so I use that instead, but my version is provided in the source code.

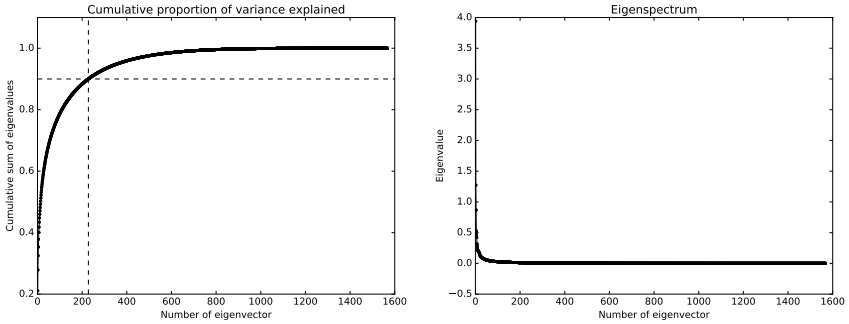
Next, I used `numpy.linalg.eigh()` to compute the eigenvalues and eigenvectors. From that I computed the number of eigenvectors needed to explain 90% of the variance. This can be done rather easy in the following manner:

```
# Cumulative proportion of variance
explainedVar = np.cumsum(eValues / np.sum(eValues))

# Number of eigenvectors needed to explain  90% of the variance.
numEigs = len(explainedVar[explainedVar < 0.9]) + 1
```

In Figure 2a I've displayed the explained variance as a function of the eigenvectors, which are sorted in descending order according to their eigenvalues. In the same manner, the eigenvectors are plotted against their eigenvalues in Figure 2b.

Furthermore, in Figure 3 I'v projected the zero mean data onto the first two principal components. The coloring makes it easy to see our groups, but without,



(a) Explained variance displayed as a cumulative sum of the eigenvectors.

(b) Eigenvectors and their eigenvalues in descending order.

Figure 2: Visual representation of eigenvalues.

one would have some problems distinguishing between the round and diamond shapes sign groups. Also shown in this figure, is the cluster centroids from a run of a 4-means clustering algorithm. More about this in the next section.

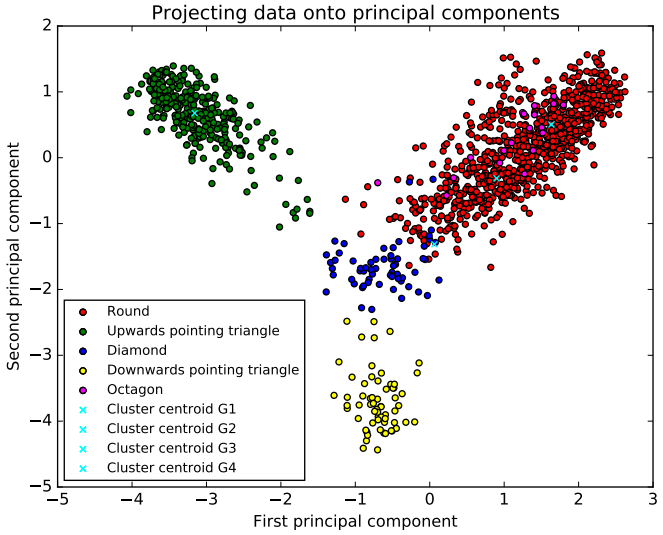


Figure 3: Zero mean data projected onto first two principal components, along with cluster centroids from 4-means algorithm.

3.3 Clustering

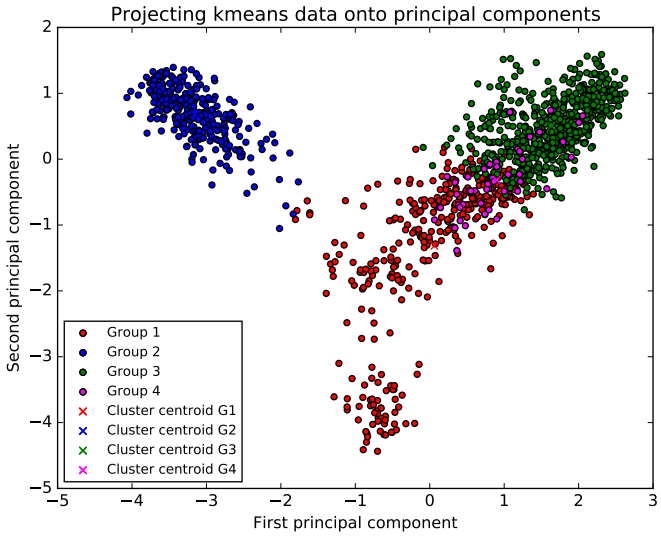


Figure 4: 4-means grouping projected onto first two principal components. The centroids are the same as in Figure 3.