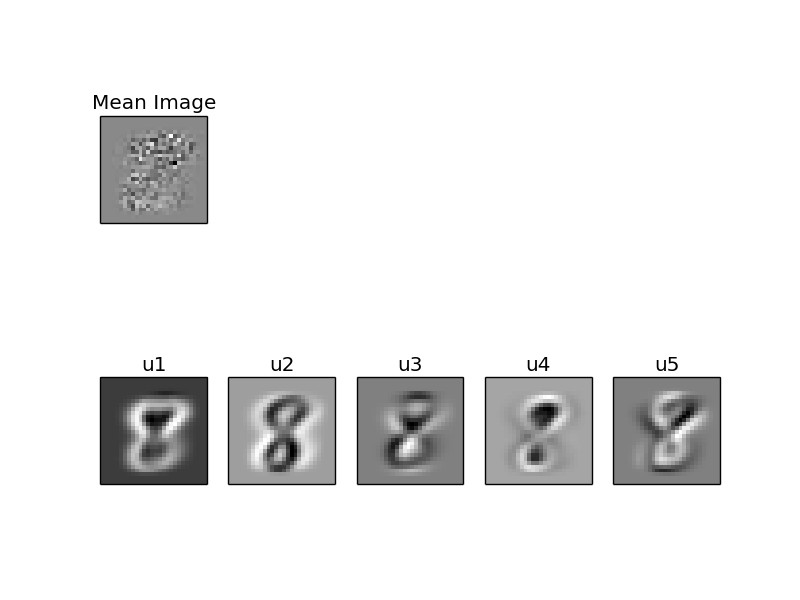
Assignment #6 – Programming Part

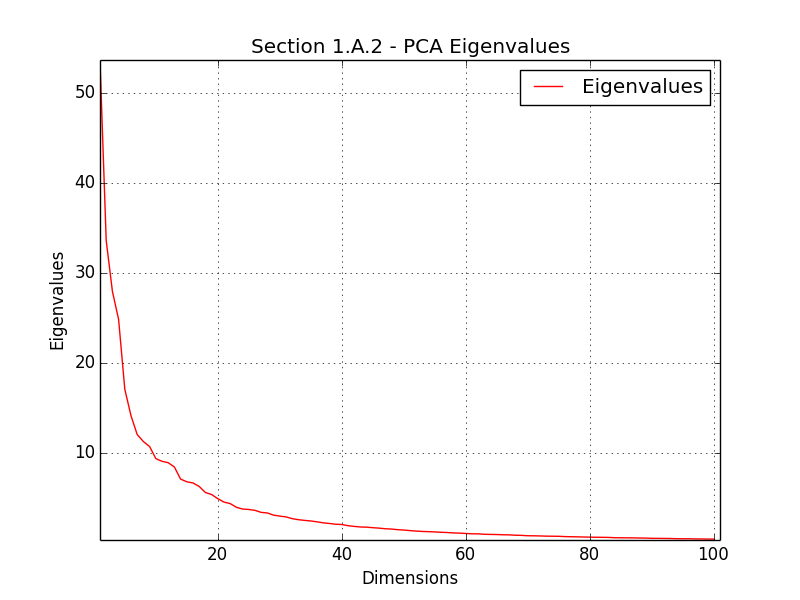
**Code Location:** home/cc/students/cs/orperel/intro\_to\_ml/ex6

**Question (a)**

After performing PCA on ‘8’ digits, we get the following 5 eigen-images (compared with the training data mean-image):



The eigenvalues:



The original feature space of the images was pixels.

The PCA feature space captures uncorrelated features that retain as much variance in the data is possible. The first principal components capture most of the variance in the data (on par with the corresponding eigenvalues), as seen from the second graph.

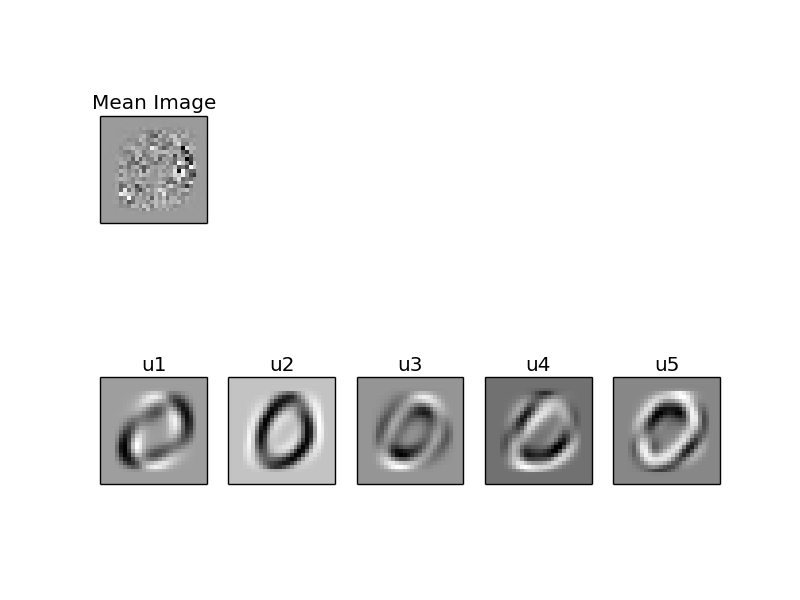
Notably, the eigenvector images we obtain resemble ‘8’ digits from the training set, and can be interpreted as patterns that describe where ‘8’ digits tend to differ from each other. Since the eigen-images form an actual basis (data images are approximated by a weighted sum of the eigen-images), we expect them to roughly describe the digit’s attributes as we weight more or less of each attribute (we note that some of these attributes are entangled within each eigenimage).

For example: u1 describes the general structure of “wide” 8, with wider area on the top of the digit. u2 adds width to the frame of the digit. u3 might describe the attribute of a digit leaning to the left or right, as well as narrow lower part.

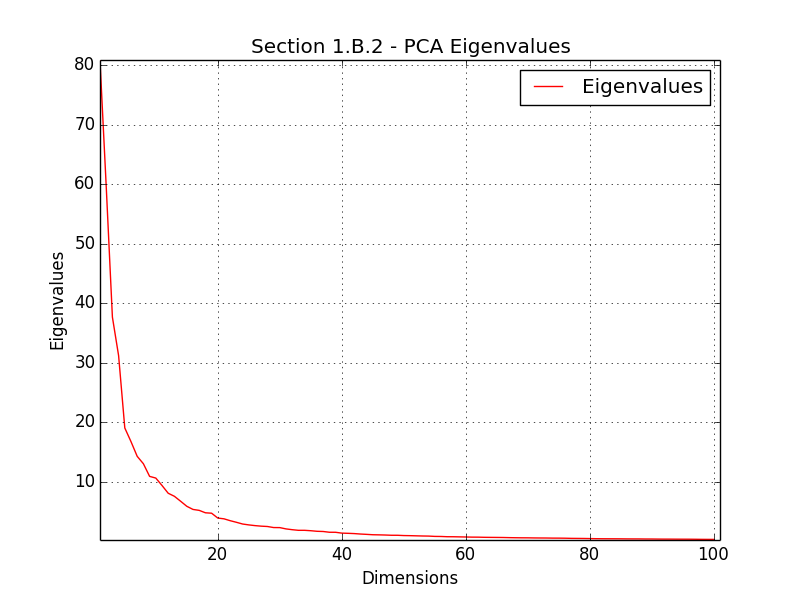
Compared with the noisy mean image, which has no clear semantic meaning, we observe that PCA basis gives better interpretation to the data.

**Question (b)**

After performing PCA on ‘0’ digits, we get the following 5 eigen-images (compared with the training data mean-image):



The eigenvalues:



Compared with the ‘8’ digits, the ‘0’ dataset also models approximately 93% of the variance in the first 20 principal components. The first PC has a larger eigenvalue (more variance) than the first PC of the ‘8’ digits data.

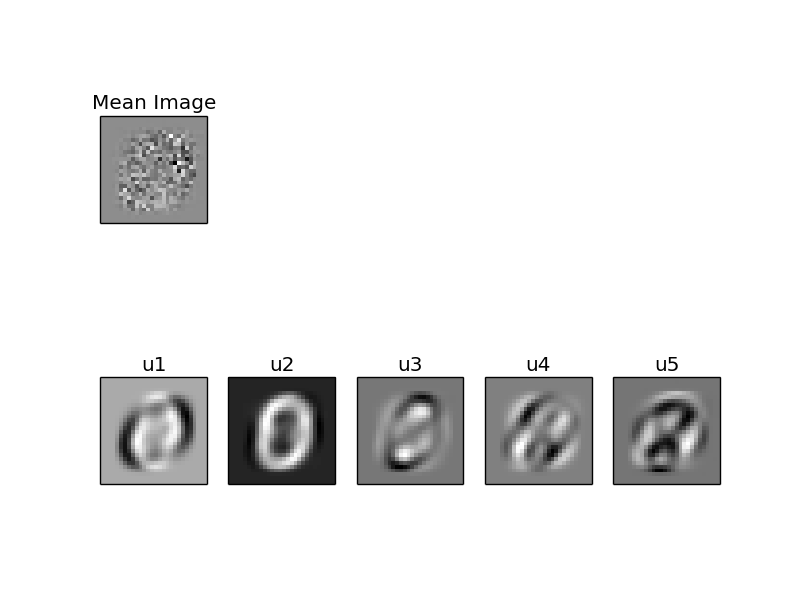
The eigenvector images resemble zero digits, and can be interpreted as describing a weighted sum of attributes of zero digits, in each image.

For example:

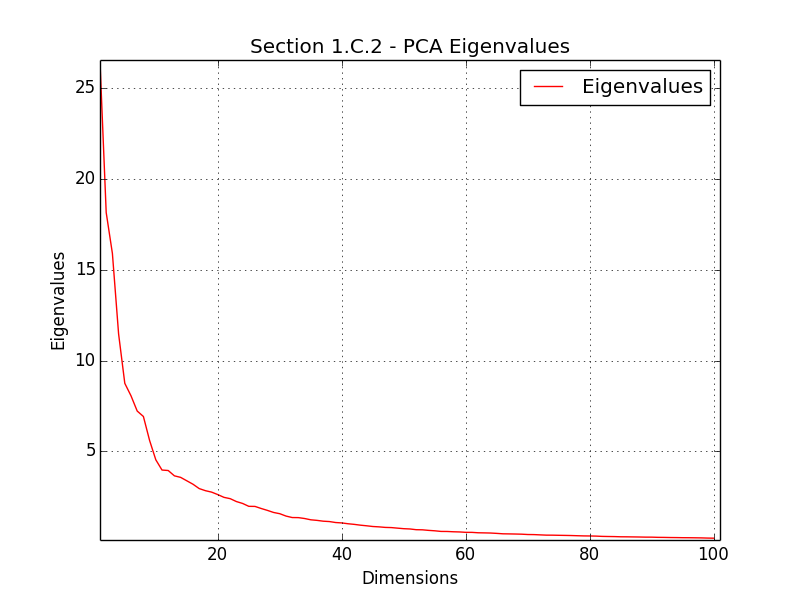
* u1 is a narrow skeleton of a zero digit.
* u2 applies weight to the frame of the zero digit.
* u3 can be roughly evaluated as the attribute of digits leaning to the right.

**Question (c)**

After performing PCA on ‘8’ and ‘0’ digits, we get the following 5 eigen-images (compared with the training data mean-image):



The eigenvalues:



Here the eigenvector images look like a mixture of ‘8’ and ‘0’ digits (the first 2 PCs lean more to ‘0’), suggesting that the PCA basis models entangled attributes of ‘0’ and ‘8’ digits at the same time.

u1, u2 provide the general structure of ‘0’ and ‘8’ digits.

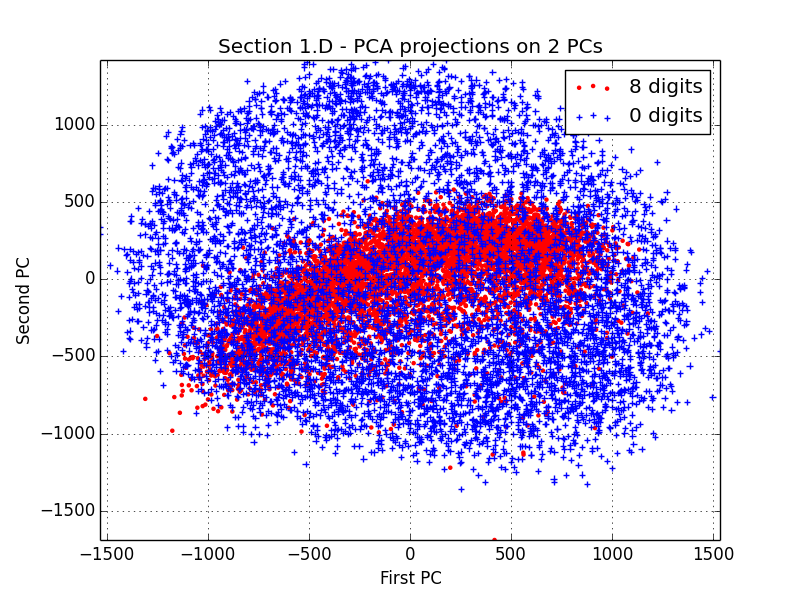
u3 can be interpreted as weakly capturing the shape of ‘0’ and puts a strong emphasis on the holes of the ‘8’ digit.

u4 (very) roughly controls digits leaning to the right.

u5 can be interpreted as assisting in the separation of ‘0’ and ‘8’ digits.

The magnitude of eigenvalues of this test case is smaller than that of the datasets of ‘0’ and ‘8’ separated. This result suggests that the principal components of the joint training set capture less variance than the previous cases. Since the joint training data contains more varied samples than the separated cases, it’s reasonable to expect it to exist in a subspace of higher dimension than the subspace of ‘8’ or ‘0’ digits separated. Therefore PCA has a hard time extracting principal components that model high variance.

**Question (d)**



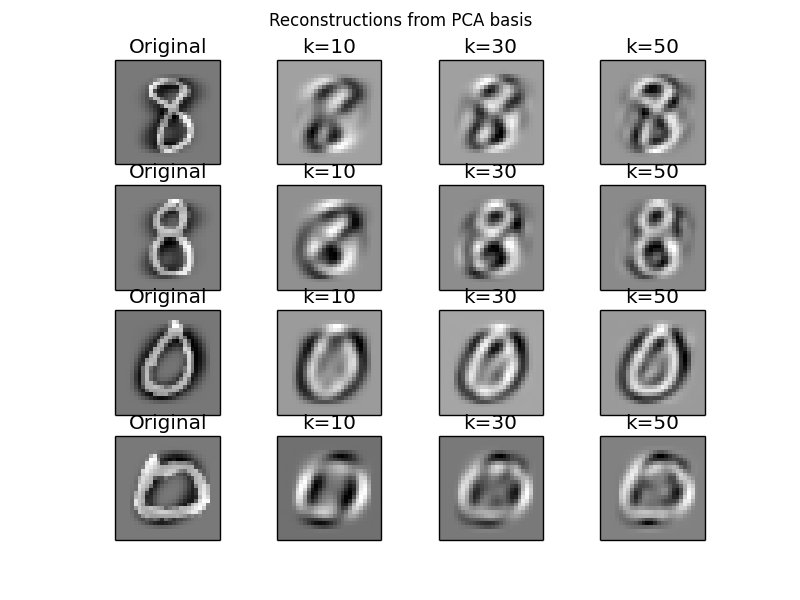
The plot describes the projection of ‘8’ and ‘0’ digits on their first and second PCs.

As noted from the plot, both digit types have high variance along the 1st PC; ‘0’ has slightly higher variance. ‘0’ digits have higher variance than ‘8’ digits along the 2nd PC.

The differences in variance between the digit types are evidence that hints towards the eigenvector images we got in question d: Both eigenvector images look more like zeros but contain slightly weaker marks of an eight shape, implying that the attributes embedded in those axes is related a bit more to ‘0’ digit attributes, which explains why projected ‘0’ maintain more variance in the 2d space.

The plot suggests that reconstructing ‘0’ digits from this PCA basis (k=2) will yield visually better, more varied results than ‘8’ digits.

**Question (e)**

****

As the results show, reconstructing the image using more principal components gives better results. Using the first 10 PCs yields the imprecise, blurred shapes of the digits, that don’t necessarily represent their visual attributes well. Using 30 PCs, we’re able to retain some rough details of the original images. With 50 PCs we’re able to gain a blurry reconstruction of the originals.

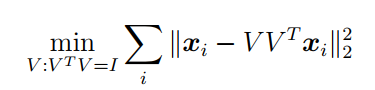
This implies that the combined digits manifold indeed exists in a subspace of more than 50 dimensions.

To sum up:

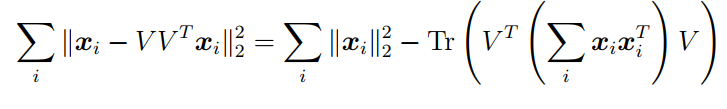
* Reconstruction with k=50 is able to capture the prominent visual features of the digits.
* In order to preserve fine details we should opt for a subspace of higher dimension.

**Question (f)**

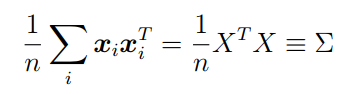
As explained in class - the PCA objective is:



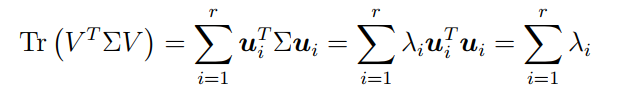
It is equivalent to:



Where:



And:



Therefore we can express the objective by:

Meaning – the more eigenvalues, the more variance we capture, the lower the sum of residuals becomes (lower error for the PCA objective).

Indeed the graph we obtained supports the theoretical claims, as the more PCs we use, the lower the PCA objective becomes. Also – the first PCs contain more variance than the latter ones, which is also evident by the slope of the graph, which is steep in the beginning and gradually becomes smaller as we add more PCs:

