Dimensionality reduction: Applying PCA to the MNIST handwritten digits dataset

December 9, 2015

We have seen in the lecture that performing PCA with q principal components takes a vector $\mathbf{x}$ and maps it to the q-coordinate vector

$$\mathbf{p} = (\mathbf{u}_1^T(\mathbf{x} - \mu), \ldots, \mathbf{u}_q^T(\mathbf{x} - \mu)) \tag{1}$$

Where μ is the mean of the data set and $\mathbf{u}_1, \ldots, \mathbf{u}_q$ are the first q principal axes (or loadings vectors). These q coordinates describe a position inside the q-dimensional affine subspace

$$\mu + \mathrm{Sp}\{\mathbf{u}_1, \ldots, \mathbf{u}_q\}.$$

However we can easily translate the coordinates of Eq. (1) to coordinates in our original space. As we have seen, the orthogonal projection on this affine space is

$$\mathbf{u}_1\mathbf{u}_1^T(\mathbf{x} - \mu) + \ldots + \mathbf{u}_q\mathbf{u}_q^T(\mathbf{x} - \mu) + \mu$$

Which is just the result of multiplying each principal component by its corresponding principal axis and adding the mean vector μ. Note that by varying q, we obtain a series of increasingly better *approximations* to $\mathbf{x}$:

0^{th} order approximation: μ

1^{st} order approximation: $\mu + \mathbf{u}_1\mathbf{u}_1^T(\mathbf{x} - \mu)$

2^{nd} order approximation: $\mu + \mathbf{u}_1\mathbf{u}_1^T(\mathbf{x} - \mu) + \mathbf{u}_2\mathbf{u}_2^T(\mathbf{x} - \mu)$

3^{rd} order approximation: $\mu + \mathbf{u}_1\mathbf{u}_1^T(\mathbf{x} - \mu) + \mathbf{u}_2\mathbf{u}_2^T(\mathbf{x} - \mu) + \mathbf{u}_3\mathbf{u}_3^T(\mathbf{x} - \mu)$

Question 1

Pick a '1' digit of your choice and compare visually (using the **mnist.montage** function) the actual digit with all of its approximations up to some high order (say, 35). Note that for in most cases, 20-30 principal components are enough to get a very nice approximation of the original digit.

Question 2

Do the same for a more "complicated" looking digit (e.g., 3, 4, 5). Since there are more variants to complicated-looking digits we would expect to need more principal components in order to faithfully reconstruct the digit.

Question 3

a.

Propose a lossy compression scheme for the MNIST data set based on these ideas. (don't implement anything - just write a short description)

b.

How much storage overhead does this scheme have? What is the per-digit storage requirement? (a rough guess is sufficient)