# CSE 455/555 Spring 2013 Homework: Dimensionality and Component Analysis

Jason J. Corso
Computer Science and Engineering
SUNY at Buffalo
jcorso@buffalo.edu

**Dimensionality**

As we discussed in the class, nearest neighbor algorithms are very intuitive for data in low-dimensions. They assume we should be able to find a good number of examples close to a given point $x$ and use the examples to estimate the class of $x$ (these are its neighbors).

However, in high-dimensions, this assumption breaks down. This makes learning hard and is an instance of the curse of dimensionality. Consider inputs distributed in a $d$-dimensional unit hypercube $[0, 1]^d$. Suppose we define a hypercubical neighborhood around a point $x$ to capture a fraction $r$ of the unit volume. The point and the hypercubical neighborhood are assumed to be fully inside the unit hypercube without loss of generality. Denote the length of each side of the hypercubic neighborhood as $l$.

1. Derive $l$ for $d = 1$ and $d = 2$.
2. Show that $l = r^{1/d}$. What is $l$ with $d = 100$ for $r = 0.1$? In other words, to capture 10 percent of the data in the 100-dimensional space, what is that length of a side of the hypercubic neighborhood?
3. For $d = 10100$, how big on average do you need to make $l$ in order to capture $1\%$ of the data? How big for $10\%$?

**Component Analysis** *Now, let's get our hands dirty!*

As we saw in the case of faces, principal component analysis provides a way of creating an optimal low-dimensional representation of a dataset. Now, let's do such a PCA analysis on handwritten digits.

Download the dataset from the website:
`http://www.cse.buffalo.edu/~jcorso/t/555pdf/homework2-data.tar.gz`. The datafiles are also available on the CSE network at `/projects/jcorso/CSE555/homework2-data`. This is a subset of the LeCun's MNIST dataset containing just the digits 0,1, and 2. The full dataset is available at `http://yann.lecun.com/exdb/mnist/` or in a more convenient Matlab format `http://www.cs.toronto.edu/~roweis/data/mnist_all.mat`. The dataset is split into training and testing pictures. Do all PCA analysis on the training pictures, and reserve the testing pictures until the last part (nearest neighbor classification).

The images are all in the PGM format (Matlab can read these directly). The format is simple:

```
// Line1 P5
// Line2 WIDTH HEIGHT
// Line3 255
// Line4 ROW-WISE-BYTE-CHUNK-OF-PIXELS
```

1. Write a function to perform PCA on a group of images. This will require you to vectorize the images (i.e., do not do IMPCA). Input the number of dimensions $k$ you want to estimate and output the set of eigenvectors and their corresponding eigenvalues (for the largest $k$).
2. Use the PCA function from question 1 to compute the Digit-0-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Describe what you find in both plots.
3. Use the PCA function from question 1 to compute the Digit-2-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Describe what you find in both plots.

4. Use the PCA function from question 1 to compute the Digit-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Compare and constrast what you find in these plots to the ones you created in questions 2 and 3.

5. Implement a nearest-neighbor (NN) classifier to input a training dataset, compute the PCA space, and then take a query image and assign it the class of its nearest neighbor in the PCA space.

6. Use the NN classifier to classify the testing images. Prepare a figure that shows 5 correctly classified images of each class and 5 incorrectly classified images of each class. Prepare a table giving the quantitative results over all of the testing data. Explain your findings.