Data Science Decal Fall 2017 Homework 4

Additional References: https://people.eecs.berkeley.edu/~jrs/papers/machlearn.pdf

1 Principal Component Analysis (PCA)

(1) Visualize eigenvectors (different components) of PCA and see how the clusters change. Do this with the provided, labeled mnist dataset. Each label should have its own color (10 colors total for 10 digits). Graph $X_0, ..., X_9$ where X_i is the input for images with label i (digit i). in 2 dimensions by picking 2 principal components at a time. Then graph X in 3 dimensions. Note: Graph the first 50 images.

You should see that picking different principal components not only explains different variations, but also separates the distributions, or clusters the points over different axis. PCA is useful in clustering when there are too many features, or too many dimensions to cluster over, so we'd rather reduce the dimension of the data or pick which dimensions we want to represent the clusters.

2 Singular Value Decomposition (SVD)

(1) Visualize low rank approximation by graphing the percentage of variation in the data. Pick a random mnist image, then get the SVD of that one image. Graph the original image. Then graph the approximated image with rank 1, 2, ..., 28 (full rank).

For example, the approximated image for rank 1 is:

Given U (28x28), S (28x28) diagonal, $V^\top$ (28x28),

$U[:, :1]S[:1, :1]V[:1]^\top$

The directions that capture the most variation (descending) are the principle components with the largest eigenvalues (descending). These directions most define the data.

(2) Whitening: Compute the low rank approximation, then graph the image without the singular values, thus weighting each orthonormal direction the same. There should be 28 approximated images for rank 1, 2, ..., 28, with rank 28 being that of the original image.

The purpose of this is to emphasize weaker signals, which is especially helpful if your model trains superbly well on datapoints that happen too frequently (which at this point is redundant learning) and performs poorly on other datapoints. This forces the model to learn each direction / type of data more equally (or with equal consideration).