1. Introduction

In this MP, the experimented feature vectors are the ones obtained from principal component analysis (PCA). The results are compared to those obtained from raw image pixels and from random orthogonal vectors (random projection). In the result, PCA feature vectors appear far more effective than either of raw pixels or random projection. The intuitive reason behind the scene is that PCA produces uncorrelated feature vectors that align in the direction of those main “patterns” of the image, while the raw pixel vectors have high correlation that make the underlying image “patterns” hard to decompose, and the random projection vectors, though have no correlation, make incorrect guesses of the underlying main “patterns”. The classifying algorithm used in the MP is k-nearest-neighbor (kNN), which is not exactly a learning algorithm because it memorizes all of the (projected) images. It learns the position of the training images in the projected space and classify the test data as the closest k neighbors in the space.

1. Methods
   1. Principal Component Analysis (PCA)

The file calc\_PCA.m contains all of the calculation needed for PCA. Specifically, line 4 to 13 describes the process to obtain the PCA basis vectors for the set of images. The mathematical deduction is below.

Given a set of M data, v\_1, v\_2, …, v\_M each with N dimensions, first we normalize them and find the covariance matrix. Hence,

[\Sigma]\_ij = 1/(M-1) \* \Sigma(v\_i\*v\_j^T)

where M-1 is due to the estimation of the population covariance from a sample. Then, we find the eigen-decomposition of \Sigma (in order of descending eigenvalues) to find the orthogonal PCA basis vectors U (6300\*6300, but only the first 6300\*80 is valuable for the reason below).

In this MP, since there are at most 80 samples each with at most 6300 dimensions, only the first 80 eigenvectors will form a projection space, while the rest 6220 eigenvectors are from the null space. To enhance efficiency, we utilize the following mathematical trick in our calculation.

We first find the Gram matrix [G]\_ij = \Sigma(v\_i^T\*v\_j) (80\*80). The eigen-decomposition of G will give an orthogonal matrix V with corresponding eigenvalue k\_i. From singular value decomposition (SVD), we claim that the normalized data matrix X\_tilde (whose i-th column is v\_i-\mu) can be decomposed into U (6300\*80) \* S (80\*80) \* V^T (80\*80), where S has the corresponding sqrt(k\_i) on the diagonal. Hence, we calculate U = X\_tilde\*V\*S^(-1), where [S^(-1)]\_ii = 1/sqrt(k\_i).

* 1. k-Nearest-Neighbor (kNN)

The files knn.m and findlabel.m contains all of the calculation for the kNN classifier. It classifies the test data as the same label that appears for its nearest k neighbors. Mathematically, given M training sample each with N dimensions, each with label l\_1, l\_2, ... , l\_b, denote each sample as v\_i of length N, and define dist(v\_i,v\_j) = ||v\_i-v\_j||^2 (the Euclidean distance squared). For a test data v\_test, we want to find k indexes, say a\_1, a\_2, ... , a\_k, that give the k smallest dist(v\_test,v\_i) value, where 1<=i<=M. Then we label v\_test as the label l\_m that appear the most times in the label set {l(v\_a\_1), l(v\_a\_2), …, l(v\_a\_k)}, where l(v\_i) represents the label of v\_i.

1. Results
2. Discussion

In the result, different features provide different accuracies. The random projection features have the lowest average accuracies (for both 1-NN and 5-NN) because the random guess in the underlying patterns to be projected on can easily go wrong. The reason that the PCA (95% energy) has lower accuracy than raw pixels for both classifiers is that the test data is too similar to several parts of the training data that overfitting occurs. Within each classifier, the average accuracy decreases as raw pixel size increases, because additional dimensionality in the data make the data points sparse and thus hard to classify (i.e. curse of dimensionality).

Different classifiers also result in different accuracies. For any feature, 1-NN performs better than 5-NN, due to the fact that our data set is so small (only 19-20 samples for a label) that the second through fifth nearest neighbor introduces unnecessary classification errors.

(Extra credit) The increase in PCA energy has a positive effect on average accuracies using 5-NN, while it has totally no effect when using 1-NN. This is because the given data set is so similar with such a small size, that the change in PCA energy (i.e. relevant to loss of dimensions) does not change the top closest training data (which is already really close due to overfitting**(?)**), but gives a higher chance for other training data of the same face to reach the second through fifth closest positions.