Alexander Hamidi
Netid: adhamid2
ECE 417 Spring 2016
Mark Hasegawa Johnson

MP2 Report

(Introduction)

This MP is performing both signal understanding and generation. The signal understanding comes from looking for its principal components, which contain the largest variations in the image, as well as in comparing those variations with the variations in all the other images. The signal generation comes from generating a subspace of principal components onto which we project the images. By projection, we in essence generate new "images" which contain the important vibrational information. We also generate new images by rescaling the original to new dimensions and performing the same nearest neighbor analysis on those.

The main algorithms at play are PCA, principal component analysis, and k-nearest neighbor, for comparing projected images. As we have already seen, k-nearest neighbor can be applied to speech recognition, though PCA doesn't quite make sense as speech is one dimensional data. PCA is a dimensionality reduction technique, motivated largely by computational considerations of high-dimensional data, such as image and video. Though the high-dimensional data doesn't have to be such low-level information like images; it can be a 100+ dimensional feature set of, say, flower attributes, for 10000 flowers of 2 or 3 categories. As long as we have a lot of features to work with, and a set of labels, PCA is useful.

(Theoretical Basis)

A problem arises in trying to compute the PCA of a very large dataset, and that is in constructing the covariance matrix of features and computing its eigenvalues and eigenvectors. These images have between hundreds and thousands of features, or pixels, which requires solving the eigenvalue problem for (thousands)x(thousands) dimensional matrices, a computationally ridiculous task. Instead we'd like to solve a much smaller problem, which has the same eigenvectors and eigenvalues. Fortunately, such a process exists.

Once we solve the smaller 80x80 eigenvalue problem of the matrix $\Phi^T\Phi$, where Φ is the mean-centered image matrix, we arrive at the same eigenvalues that solve the much bigger problem $\Phi\Phi^T$, but with different eigenvectors. We'd like the eigenvectors of $\Phi\Phi^T$ so that we can form the eigenfaces, which is a projector matrix. In order to do this, we need to examine the problem more carefully.

$\Phi^T\Phi v_i = \lambda_i v_i$: (4), page 75

Where v_i and λ_i are the eigenvectors and eigenvalues of $\Phi^T\Phi$, respectively. Now multiply by Φ.

$(\Phi\Phi^T)\Phi v_i = \Phi\lambda_i v_i = \lambda(\Phi_i v_i)$: (5), page 75

Thus we can see that Φv_i are eigenvectors of $\Phi\Phi^T$, with appropriate scaling, and with the same eigenvalues λ_i.

The code below forms a projector matrix from the eigenvector matrix V, by taking its columns (individual eigenvectors of $\Phi^T\Phi$, transforming them with Xc, which is the mean-centered data matrix, and scaling by the appropriate eigenvalue.

```
V_sub = [];
for i=1:N
    vk = Xc*V(:,ind(i))/sqrt(D(ind(i),ind(i)));
    V_sub = cat(2,V_sub,vk);
end
```

(Results and Discussion)

From the tables for the accuracies of 1 nearest neighbor and 5 nearest neighbor below, it is pretty clear that 1 nearest neighbor is not reliable enough as it is subject to the whim of noise. The projections as well as the raw features are too close to each other (based on the Euclidean metric) to be able to distinguish them appropriately.

Results, Eigenface Analysis											
1-nn accuracies											
	Raw, 90x70			Raw, 45x35			Raw, 22x17			Raw, 9x7	
	A	0		A	0		A	0		A	0.05
	B	0		B	0		B	0		B	0
	C	0.05		C	0.05		C	0.1		C	0.1
	D	0		D	0		D	0		D	0
	Overall	0.0125		Overall	0.0125		Overall	0.025		Overall	0.0375
	95% E, 90x70			98% E, 90x70			90% E, 90x70			Random, 90x70	
	A	0.05		A	0.1		A	0.05		A	0.15
	B	0		B	0		B	0		B	0
	C	0.05		C	0.05		C	0.05		C	0.05
	D	0		D	0		D	0		D	0
	Overall	0.025		Overall	0.0375		Overall	0.025		Overall	0.05

We see a significant improvement one 5 nearest neighbor is implemented, where the accuracies are between 85 and 95 % on average. The raw features perform particularly well. I think this is because the data is very curated to begin with, as all the faces are captured in similar circumstances, in virtually the same poses, lighting, and such.

Results, Eigenface Analysis											
5-nn accuracies											
	Raw, 90x70			Raw, 45x35			Raw, 22x17			Raw, 9x7	
	A	1		A	1		A	1		A	1
	B	0.85		B	0.85		B	0.9		B	0.9
	C	0.8		C	0.8		C	0.85		C	0.7
	D	0.95		D	0.85		D	0.85		D	0.9
	Overall	0.9		Overall	0.975		Overall	0.9		Overall	0.875
	95% E, 90x70			98% E, 90x70			90% E, 90x70			Random, 90x70	
	A	1		A	1		A	1		A	0.95
	B	0.8		B	0.8		B	0.7		B	0.7
	C	0.75		C	0.75		C	0.65		C	0.5
	D	0.85		D	0.9		D	0.85		D	0.7
	Overall	0.85		Overall	0.8625		Overall	0.8		Overall	0.7125

If we look at the 5nn accuracies, for 95%, 98%, and 90% energy, we can see that the overall accuracies increase with increasing percentage. This is expected as more variation or information is contained as we use more ‘eigenfaces’ to represent our images.

Also for 5nn, it is also worth noting that the accuracies of random projection are really high (~71% on average), considering I put no though into constructing the random projection matrix. This proves that under the right circumstances, and with the right application, randomized algorithms (algorithms with random inputs) work really well.