

Lab 2

Zeyuan Xu

October 16, 2017

1 Task 1

Covariance matrix is real and symmetric in this case, therefore its eigenvectors are nonzero and orthogonal vectors. Therefore they are linearly independent. So eigenvectors of S form an orthogonal basis for \mathbb{R}^d . Therefore, a point x_n in the dataset has a unique linear combination of these points. In other words, linear combination of the projection onto these vectors is unique, therefore:

$$x_n = \sum_{i=1}^d \langle x_n, u_i \rangle u_i = \sum_{i=1}^d \langle x_n - \bar{x}, u_i \rangle u_i + \sum_{i=1}^d \langle \bar{x}, u_i \rangle u_i = \sum_{i=1}^d \langle x_n - \bar{x}, u_i \rangle u_i + \bar{x}$$

we have $\sum_{i=1}^d \langle \bar{x}, u_i \rangle u_i = \bar{x}$, since $\bar{x} \in \mathbb{R}^d$, and its projection onto the same set of eigenvectors (which forms a basis for \mathbb{R}^d) is unique.

2 Task 2

By definition, projection of vector x onto subspace V is given by the linear combination of projection onto its basis. Since $\{v_1, \dots, v_k\}$ spans V and are orthonormal, it is an orthonormal basis for \mathbb{R}^d . Therefore we have:

$$pV(x) = \sum_{i=1}^k \frac{\langle x, v_i \rangle}{\|v_i\|^2} v_i = \sum_{i=1}^k \langle x, v_i \rangle v_i$$

as a projection to the subspace V . Now to show that the projection is orthogonal, This is the same to show that $pV(x)^\top p\bar{V}(y) = 0, \forall x, y \in \mathbb{R}^d$. Where $p\bar{V}(x)$ is the complement subspace of V , and $p\bar{V}$ represents the complement projection. We can vectorize the above expression into $pV(x) = QQ^\top x$, where Q is a matrix whose column vectors are v_1, \dots, v_k . Since $QQ^\top = I_k$, $p\bar{V}(y) = (I - QQ^\top)y$. Therefore we have that:

$$\begin{aligned} pV(x)^\top p\bar{V}(y) &= (QQ^\top x)^\top (I - QQ^\top)y \\ &= x^\top QQ^\top y - x^\top QQ^\top QQ^\top y = 0 \end{aligned}$$

Therefore, $pV(x)$ is an orthogonal projection of x onto subspace V .

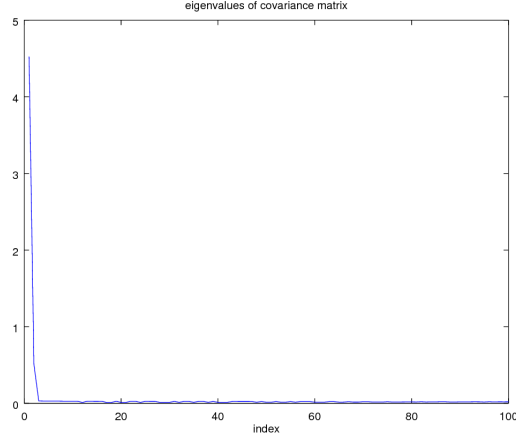


Figure 1: Eigenvalues

3 Task 3

Given:

$$\hat{x}_n = \bar{x} + \sum_{i=1}^m \langle x_n - \bar{x}, u_i \rangle u_i$$

$$x_n = \sum_{i=1}^d \langle x_n - \bar{x}, u_i \rangle u_i + \bar{x}$$

Since $\{u_1, u_2, \dots, u_d\}$ are eigenvectors of real symmetric matrix. We have that $uu^\top = I$:

$$\begin{aligned} J_m &= \frac{1}{N} \sum_{n=1}^N \|\hat{x}_n - x_n\|^2 \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{i=m+1}^D \|\langle x_n - \bar{x}, u_i \rangle u_i\|^2 \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{i=m+1}^D (x_n^\top u_i - \bar{x}^\top u_i)^2 \\ &= \frac{N-1}{N} \sum_{i=m+1}^D u_i^\top S u_i \\ &= \frac{N-1}{N} \sum_{i=m+1}^D \lambda_i \end{aligned}$$

4 Task 4

The plot for eigenvalues of the covariance matrix S for dataset is shown in figure 1: There is indeed a sudden drop of eigenvalues at beginning of the index. From PCA, we know that large eigenvalues of covariance matrix represent high variance, and since most of the eigenvalues are close to zero, the actual dimension of the data can be much less than 100.

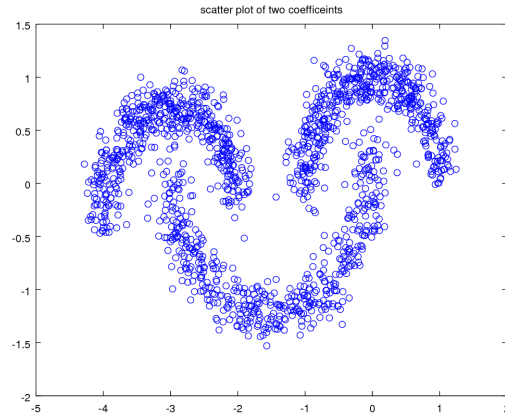


Figure 2: Three moons

5 Task 5

The plot of three moon is shown in figure2.

6 Task 6

The original image is shown in figure 3. The plot of image rotated is shown in figure 4.

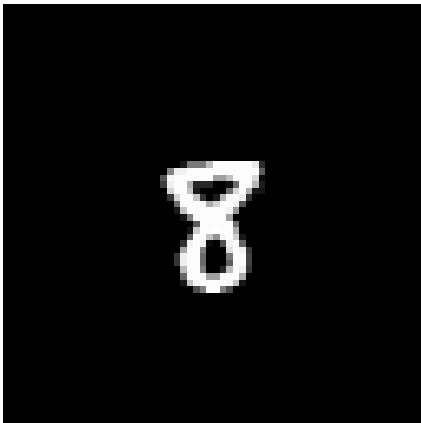


Figure 3: Original image

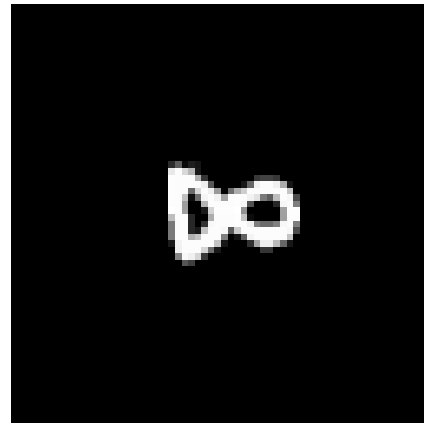


Figure 4: Rotation 90 degrees

7 Task 7

Data is inherently 1-D and identified by only the rotation angle.

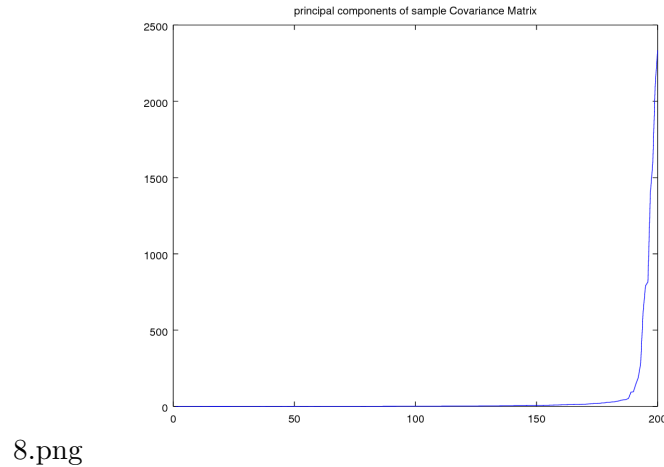


Figure 5: Eigenvalues for Image

8 Task 8

No need to compute all eigenvectors if $N < d$. Since the rank of covariance matrix is N . The plot for eigenvalues of covariance matrix is shown in figure 5.

9 Task 9

images for the projection onto 3 major components are in figure 6, 7, 8. The images don't make sense: they are nowhere similar to the original image, but they are suppose to be just rotations.

10 Task 10

The image for using the 3 principal components to reconstruct original image is shown in figure 9. It is completely different from figure 3 (original). So it shouldn't be resonable to compress the image using only 3 principal components.

11 Task 11

The image of using 198 principal components is shown in figure 10. The image of using 199 principal components is shown in figure 11. It seems that we need at least 199 principal components to reconstruct the image. Therefore, PCA is not useful for this dataset.

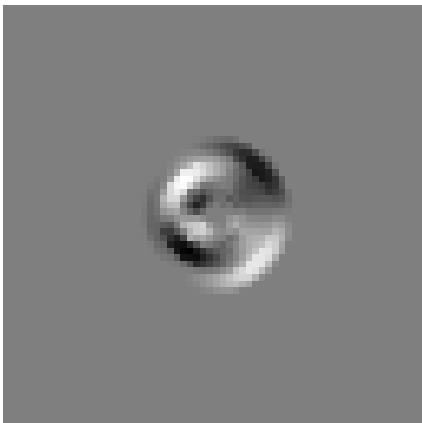


Figure 6: First Principal Component

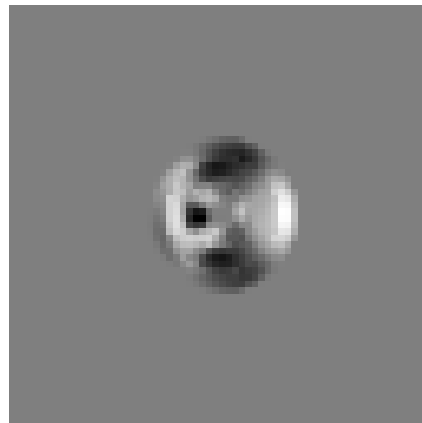


Figure 7: Second Principal Component

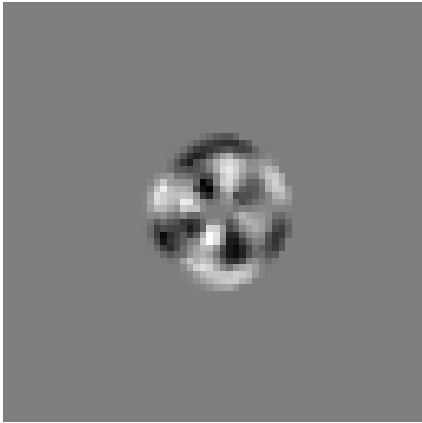


Figure 8: Third Principal Component

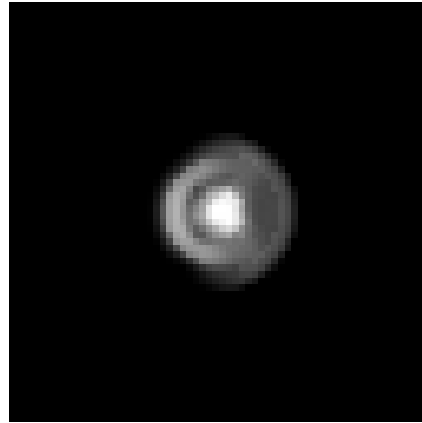


Figure 9: Reconstruction

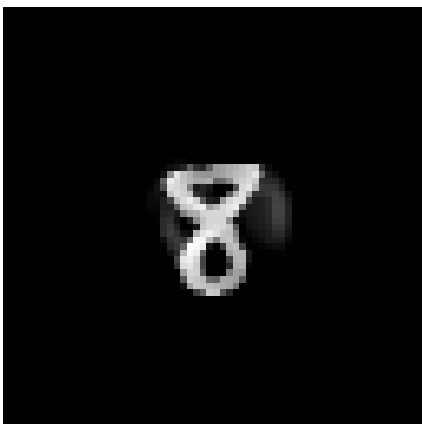


Figure 10: 198 Principal Components

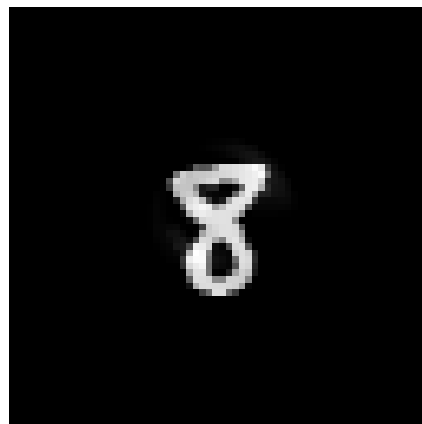


Figure 11: 199 Principal Components