Signal and Information Processing

Alejandro Ribeiro

April 26, 2016

Contents

Chapter 1

Principal Component Analysis

1.1 The DFT and iDFT as Hermitian Matrices

We have seen thus far in the course that the DFT is a hugely useful tool to us in a variety of applications. Recall the definition of the DFT:

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \tag{1.1}$$

Recall that this can also be written as an inner product of a signal x with a complex exponential e_{kn}:

$$X(k) = \langle \mathbf{x}, \mathbf{e}_{kN} \rangle \tag{1.2}$$

This conceptualization of the DFT has been extremely useful so far in frequency analysis and image processing. However, from a broader perspective, this is still ″beginner″ mode. Armed with our knowledge of signal processing, we are ready to switch to ″advanced″ mode.

What, exactly, is ″advanced″ mode? Well, in a word, linear algebra. (Two words, actually). We know that linear algebra is a well-established branch of mathematics with a myriad of profound and insightful theorems and results that make complicated applied mathematics much simpler. Now, if only there were a way to write this DFT formula above as a matrix, so that we could apply the powerful mathematics of linear algebra...

As you may have guessed by now, there is. How is this done? Well, let's write out our signal $\mathbf{x}$ and our complex exponential $\mathbf{e}_{kn}$ as vectors.

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{pmatrix} \qquad \mathbf{e}_{kN} = \frac{1}{\sqrt{N}} \begin{pmatrix} e^{j2\pi k0/N} \\ e^{j2\pi k1/N} \\ \vdots \\ e^{j2\pi k(N-1)/N} \end{pmatrix} \tag{1.3}$$

It's easy to see that the kth component of $\mathbf{X}(k)$ (the DFT of $\mathbf{x}$) can be written as the kth element of $\mathbf{x}$ times the kth element of $\mathbf{e}_{kn}$. From here you should start to see how this will come together to form another representation of the DFT.

Let's take a break and introduce a bit of notation. We will introduce a matrix operator, the Hermitian, denoted by an uppercase H. The Hermitian of a matrix is simply its conjugate transpose. That is, $\mathbf{A}^H = (\mathbf{A}^*)^T$, where $*$ and T denote the conjugate and transpose operations, respectively. You should be familiar with these operations from prior mathematics coursework.

With this new notation, we can see that

$$\langle \mathbf{x}, \mathbf{e}_{kN} \rangle = \mathbf{e}_{kN}^H \mathbf{x} = X(k) \tag{1.4}$$

Thus, the kth DFT component can be written as a matrix (in this case, vector) multiplication! Now let's try to do the whole thing in one step. We can ”stack” N complex exponentials into a matrix and vary their components as we go across and down the rows and columns of a single matrix to form a matrix that performs the entire DFT operation in one simple multiplication. Let's see what this looks like. Note that we are defining the DFT matrix as a Hermitian in the beginning, but we will see why we do this later.

$$\mathbf{F}^H = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix} \tag{1.5}$$

We can now write the entire DFT as a matrix multiplication, by multiplying this $\mathbf{F}^H$ matrix by our signal $\mathbf{x}$. If you are having trouble visualizing this, examine the following diagram:

1.1.1 Properties of the DFT Matrix

As we can see from the above matrix, the (k, n)th element of the DFT matrix $\mathbf{F}^H$ is the complex exponential with indices k and n.

$$\left(\mathbf{F}^H\right)_{kn} = e^{-j2\pi(k)(n)/N} \tag{1.6}$$

Note that by using properties of exponents to pull the exponential outside the parentheses, we can write the rows of this matrix as indexed powers. That is,

$$e^{-j2\pi(k)(n)/N} = \left(e^{-j2\pi(k)/N}\right)^{(n)} \tag{1.7}$$

We say that $\mathbf{F}^H$ is Vandermonde. This is an interesting property, but will not be discussed further in this course.

We can also note that this complex exponential can be written by reversing the indices. That is, $e^{-j2\pi(k)(n)/N} = e^{-j2\pi(n)(k)/N}$ due to commutativity of multiplication. Then, the (k, n)th element of the DFT matrix is the same as the (n, k)th element of the DFT matrix. This matrix, therefore is symmetric. That is, $(\mathbf{F}^H)^T = \mathbf{F}^H$.

We can then write:

$$\mathbf{F}^H = (\mathbf{F}^H)^T = \begin{bmatrix} \mathbf{e}_{0N}^* & \mathbf{e}_{1N}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix} \tag{1.8}$$

Similarly, conjugate transposing the matrix twice returns the original matrix, so:

$$(\mathbf{F}^H)^H = \mathbf{F} \tag{1.9}$$

This should spark a notion of invertibility, one of the key properties of linear algebra. We will get there soon. But first let's note that we can write this matrix $\mathbf{F}$ as a matrix of complex exponentials as follows.

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}^T_{0N} \\ \mathbf{e}^T_{1N} \\ \vdots \\ \mathbf{e}^T_{(N-1)N} \end{bmatrix} \tag{1.10}$$

In other words, $\mathbf{F}$ and$\mathbf{F}^H$ are Hermitians of each other. That is, the nth row of $\mathbf{F}$ is the nth complex exponential $\mathbf{e}^T_{nN}$, and the kth column of of F^H is the kth conjugate complex exponential $\mathbf{e}^*_{kN}$ Again, we will see why this property is so important shortly.

1.1.2 The Product of F and its Hermitian

We are getting close to the key result that will allow us to abstract the process of signal processing. We know that $\mathbf{F}$ and $\mathbf{F}^H$ are related to each other via the conjugate transpose, but our intuition would suggest that there exists a deeper relationship beyond this superficial one. Our intuition is indeed correct. Let's see what happens if we multiply $\mathbf{F}^H$ by $\mathbf{F}$.

$$\begin{array}{cc} & \begin{bmatrix} \mathbf{e}^*_{0N} & \cdots & \mathbf{e}^*_{kN} & \cdots & \mathbf{e}^*_{(N-1)N} \end{bmatrix} \\ \begin{bmatrix} \mathbf{e}^T_{0N} \\ \vdots \\ \mathbf{e}^T_{kN} \\ \vdots \\ \mathbf{e}^T_{(N-1)N} \end{bmatrix} & \begin{bmatrix} \mathbf{e}^T_{0N}\mathbf{e}^*_{0N} & \cdots & \mathbf{e}^T_{0N}\mathbf{e}^*_{kN} & \cdots & \mathbf{e}^T_{0N}\mathbf{e}^*_{(N-1)N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{e}^T_{kN}\mathbf{e}^*_{0N} & \cdots & \mathbf{e}^T_{kN}\mathbf{e}^*_{kN} & \cdots & \mathbf{e}^T_{kN}\mathbf{e}^*_{(N-1)N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{e}^T_{(N-1)N}\mathbf{e}^*_{0N} & \cdots & \mathbf{e}^T_{(N-1)N}\mathbf{e}^*_{kN} & \cdots & \mathbf{e}^T_{(N-1)N}\mathbf{e}^*_{(N-1)N} \end{bmatrix} = \mathbf{F}^H\mathbf{F} \end{array} \tag{1.11}$$

We can see that the (n,k)th element of the resulting matrix is $\mathbf{e}^T_{nN}\mathbf{e}^*_{kN}$. Let's recall the orthonormality of complex exponentials. That is,

$$\mathbf{e}^T_{nN}\mathbf{e}^*_{kN} = \delta(n-k) \tag{1.12}$$

By virtue of this amazing fact, all of the non-diagonal elements in the matrix go to zero, and the diagonal elements go to 1! Our multiplication of a matrix by its Hermitian results in the identity matrix!

$$\mathbf{F}^H\mathbf{F} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I} \tag{1.13}$$

This is precisely the definition of a matrix inverse! Therefore, the DFT matrix and its Hermitian are inverses of each other. Essentially all of linear algebra rests on the invertibility of matrices, so the importance of this fact should be starting to become clear. However, this will not be true for all matrices - the DFT matrix is one example of a matrix where this is the case. Under what circumstances, then, will a matrix's Hermitian also be its inverse? Well, we define a Hermitian matrix to be just that. That is, a matrix is Hermitian if its Hermitian is also its inverse. In other words:

Theorem 1 *A matrix* $\mathbf{A}$ *is Hermitian iff* $\mathbf{A}^H\mathbf{A} = \mathbf{I} = \mathbf{A}\mathbf{A}^H$

We see that this is true for the DFT matrix $\mathbf{F}^H$. That is:

Theorem 2 *The DFT matrix* $\mathbf{F}$ *is Hermitian, since* $\mathbf{F}^H\mathbf{F} = \mathbf{I} = \mathbf{F}\mathbf{F}^H$

We have just shown this to be the case, so we will not repeat the proof.

1.1.3 Why Is This Important?

We have so far established that the DFT can be written as a matrix multiplication. So why do we care? To put it simply, this turns something that was hard (sums and formulas) into something that is easy (matrix multiplication). Because we have now moved from the realm of "signal processing" to linear algebra, there are well-established mathematics to provide the foundation of what we do from here on out.

Next, we saw that the DFT matrix $\mathbf{F}^H$ is Hermitian. That is, the conjugate transpose of the DFT matrix is the inverse of the DFT matrix. This is a hugely useful fact. Why? Because taking the inverse of a matrix is extremely computationally expensive, whereas taking the conjugate transpose of a matrix is trivial. If I asked Alice to take the conjugate transpose of a 10 x 10 matrix and I asked Bob to take the inverse of a 10 x 10 matrix, who would finish first? Surely, it would be Alice. Furthermore, if the matrix that Alice and Bob inverted were Hermitian, they would arrive at the same result, yet Alice would have done a fraction of the work! This is why this representation of the DFT is so powerful.

Next, we will generalize even further. We will see that any Hermitian matrix can be used as a transform to process signals, and we will find that certain matrices perform this task better than others. This is the basis of principal component analysis. We will arrive there shortly, but before we do, let's briefly revisit the iDFT and see how linear algebra makes this operation easier as well.

1.1.4 The iDFT in Matrix Form

We can repeat everything we did to write the DFT in matrix form for the iDFT. For the sake of concision, we will not do that here. However, recalling the definition of the iDFT, we can write the formula in vector format as follows.

$$\tilde{x}(n) \;=\; \mathbf{e}_{nN}^T\mathbf{X} \;=\; \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \tag{1.14}$$

We then proceed as before to write the iDFT as a matrix multiplication.

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}(0) \\ \tilde{x}(1) \\ \vdots \\ \tilde{x}(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^T\mathbf{X} \\ \mathbf{e}_{1N}^T\mathbf{X} \\ \vdots \\ \mathbf{e}_{(N-1)N}^T\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^T \\ \mathbf{e}_{1N}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix}\mathbf{X} = \mathbf{F}\mathbf{X} \tag{1.15}$$

Again, for help visualizing this operation, refer to the following figure.
We can now see that the iDFT is, as the DFT, just the matrix product $\tilde{\mathbf{x}} = \mathbf{FX}$.

1.1.5 Inverse Theorem Revisited (Again)

We will continue the tradition of following every introduction of an important concept with another proof of the inverse theorem and Parseval's theorem. This time, however, will be much quicker. All of the linear algebra above will be the key as to why. We can now be much smarter and much more efficient with our proof by using these tools of matrices that we have just learned. In fact, we can do the whole proof in one line. We'll do that now.

Theorem 3 *The iDFT is, indeed, the inverse of the DFT.*

Proof: Write $\tilde{\mathbf{x}} = \mathbf{FX}$ and $\mathbf{X} = \mathbf{F}^H\mathbf{x}$ and exploit the fact that $\mathbf{F}$ is Hermitian.

$$\tilde{\mathbf{x}} \;=\; \mathbf{FX} \;=\; \mathbf{FF}^H\mathbf{x} \;=\; \mathbf{Ix} \;=\; \mathbf{x} \tag{1.16}$$

■

Furthermore, this theorem is true for *any transform pair with transformation matrix T, provided that T is Hermitian*. That is,

$$\mathbf{X} = \mathbf{T}^H\mathbf{x} \qquad \Longleftrightarrow \qquad \tilde{\mathbf{x}} = \mathbf{TX} \tag{1.17}$$

As long as $\mathbf{T}^H\mathbf{T} = \mathbf{I}$.

1.1.6 Parseval's Theorem Revisited (Again)

Again, we will do the "advanced mode" proof of Parseval's theorem, i.e. energy conservation.

Theorem 4 *The DFT preserves energy* $\Rightarrow \|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = \mathbf{X}^H\mathbf{X} = \|\mathbf{X}\|^2$

Proof: Use iDFT to write $\mathbf{x} = \mathbf{FX}$ and exploit the fact that $\mathbf{F}$ is Hermitian

$$\|\mathbf{x}\|^2 \;=\; \mathbf{x}^H\mathbf{x} \;=\; (\mathbf{FX})^H\,\mathbf{FX} \;=\; \mathbf{X}^H\mathbf{F}^H\mathbf{FX} \;=\; \mathbf{X}^H\mathbf{X} \;=\; \|\mathbf{X}\|^2 \tag{1.18}$$

■

Again, note that this theorem is true for *any transform pair with transformation matrix T, provided that T is Hermitian*. That is,

$$\mathbf{X} = \mathbf{T}^H\mathbf{x} \qquad \Longleftrightarrow \qquad \tilde{\mathbf{x}} = \mathbf{TX} \tag{1.19}$$

As long as $\mathbf{T}^H\mathbf{T} = \mathbf{I}$.

See how easy all of these once arduous tasks have become with this new ”advanced” system?

1.1.7 The DCT in Matrix Form

We have seen that the DFT can be defined as a matrix multiplication, and we have seen how useful this result has been. But is this a rule or an exception? That is, are there other transforms that can be encoded in matrices in addition to the DFT? The answer, of course, is yes. We will see one soon in the PCA transform. And we have already seen one in the DCT. We will not delve into too much detail with the DCT since we will not use it in this form, but it is easy to verify that we can construct a Hermitian matrix $\mathbf{C}$ for that performs the iDCT, defined as:

$$\mathbf{C} = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((1)+1/2)}{N}\right] \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((N-1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((N-1)+1/2)}{N}\right] \end{bmatrix} \tag{1.20}$$

The DCT matrix, then would simply be $\mathbf{C}^H$.

As we just saw, the inverse theorem and Parseval’s theorem hold for the DCT since its transform matrix is Hermitian.

1.1.8 Designing Transforms for Adapted Signals

We have now seen that we can write every transform we have learned in this class as a matrix multiplication. In the case of the DFT and the DCT, the transform matrices $\mathbf{F}^H$ and $\mathbf{C}$ do not change, regardless of the signal $\mathbf{x}$. We have seen that the DFT and the DCT provide us with a myriad of useful information about a signal’s oscillations and rates of chance, which provides us with a tremendous amount of insight about the signal itself that may have been disguised originally. However, because the transform matrices are always the same, the performance of these transforms is irrespective of the signal that we input.

However, this does not have to be the case. If we knew something about our signal, we could, conceivably, design a specialized transform matrix $\mathbf{T}$ to do all the same things as the

DFT and DCT, but better. Our only requirement, as we have seen, is that **T** be Hermitian. We will see in the coming sections that we can "know something" about a signal by means of a stochastic model. By understanding the principles of stochastic modeling, we will then be able to discover the power of principal component analysis. This technique is an alternative method of transformation that uses information contained in the eigenvectors of the covariance matrix of a data set to determine the transform matrix **T**.

1.2 Stochastic Signals

Before we get into the details of principle component analysis, we will first lay out some foundations in probability.

1.2.1 Random Variables

A random variable X models something that is random, with two important points. Firstly, the random phenomenon being modeled is one that has several different possible outcomes, and secondly, you have an idea of how likely that these outcomes may appear. Therefore, a random variable represents all possible values that an event can take as well as a measure of how likely those values are. A higher likelihood corresponds with a higher chance to observe a certain value x. By convention, random variables are usually represented in uppercase (i.e. X), whereas the values that it can take are represented in lowercase (i.e. x).

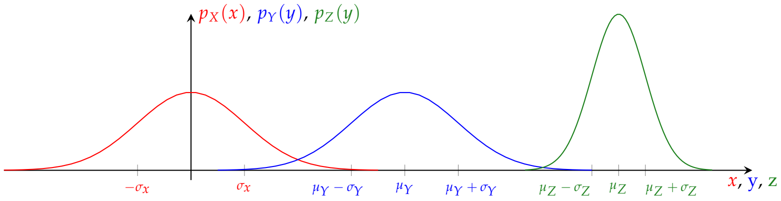

Figure 1.1. Examples of random signals X, Y, and Z with Gaussian, or normal, probability distributions. The red curve represents random variable X, which takes values around 0. The blue curve representing Y has a shifted center around μ_Y, and the green curve representing Z is centered around μ_Z. Also notice that the values of X and Y are equally distributed about their respective centers, whereas the values of Z are more concentrated about its mean.

Probabilities measure the likelihood of observing different outcomes. A larger probability indicates that an outcome is more likely to be observed over many realizations. The probability that the random variable X takes values between x and x' is the term $\mathrm{P}(x < X \leq x')$. This probability can be described with a probability density function $p_X(x)$. A probability distribution function (pdf) tells you about how likely a variable is around a value x (but not what the probability of x is itself).

$$\mathrm{P}(x < \mathrm{X} \leq x') = \int_x^{x'} p_X(u)\,du \tag{1.21}$$

A random variable X is defined as a Gaussian, or normal, variable if its pdf is of the form

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2} \tag{1.22}$$

where the mean μ determines the center and the variance σ^2 determines the width of the distribution. The previous figure illustrates the effects of different μ and σ^2, where $0 = \mu_X < \mu_Y < \mu_Z$, and $\sigma_X^2 = \sigma_Y^2 > \sigma_Z^2$.

1.2.2 Expectation and Variance

The expectation of a random variable is an average of the possible values weighted by their likelihoods.

$$\mathbb{E}\left[X\right] = \int_{-\infty}^{\infty} x p_X(x)\, dx \tag{1.23}$$

In a regular average, you would sum all of the values and divide by the number of values. However, in an expectation, you weight the values x by their relative likelihoods $p_X(x)$.

For a Gaussian random variable X, the expectation is the mean μ.

$$\mathrm{var}\left[X\right] = \int_{-\infty}^{\infty} \left(x - \mathbb{E}\left[X\right]\right)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \sigma^2 \tag{1.24}$$

The variance of a random variable is a measure of the variability around the mean. A large variance tells you that the likely values are spread out around the mean, and a small variance means that the the most likely values are concentrated around the mean.

$$\mathrm{var}\left[X\right] = \mathbb{E}\left[\left(X - \mathbb{E}\left[X\right]\right)^2\right] = \int_{-\infty}^{\infty} \left(x - \mathbb{E}\left[X\right]\right)^2 p_X(x)\, dx \tag{1.25}$$

For a Gaussian random variable X, the variance is the variance σ^2. For a Gaussian random variable X the variance is the variance σ^2

$$\mathrm{var}\left[X\right] = \int_{-\infty}^{\infty} \left(x - \mathbb{E}\left[X\right]\right)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \sigma^2 \tag{1.26}$$

The variance is important because it tells you how random the random variable is. We usually care more about the variance than the expectation. In many cases, we subtract the mean from our signals so that we may focus only on the differences in variability between the signals. For example, in the previous image with the red, blue, and green Gaussian curves, it can be seen that the differences between the blue and green curves are more important than those between the blue and red curves because of the different variances.

1.2.3 Random Signals

A random signal **X** is a collection of random variables with length N.

$$\mathbf{X} = \left[X(0),\ X(1),\ \ldots,\ X(N-1)\right]^T \tag{1.27}$$

Each random variable has its own pdf $p_{X(n)}(x)$, which describes the likelihood of the random variable X(n) taking a value around x. These individual pdfs are also called marginal pdfs.

Joint outcomes are also important in the description of a random variable. The joint pdf $p_{\mathbf{X}}(\mathbf{x})$ says how likely the signal $\mathbf{X}$ is to be found around the collection of values x $\mathbf{x}$.

$$\mathrm{P}(\mathbf{x} \in \mathcal{X}) = \iint_{\mathcal{X}} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \tag{1.28}$$

1.2.4 Face Images

We can extend the idea of random signals to the dataset we are working with in the lab, AT&T's Database of Faces. We can imagine all possible images of human faces as a random signal X. That's a lot of faces to consider, so we'll actually reduce X to the collection of 400 face images in the Database of Faces. In this case, each random variable of the random signal X represents each of the images and the likelihood of each of them being chosen (e.g. 1/400 each).

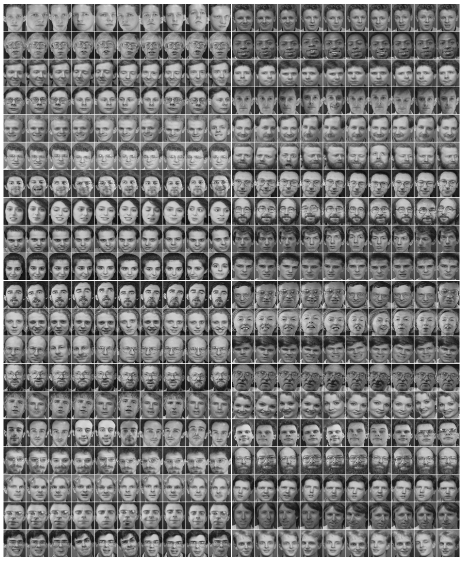

Figure 1.2. Face images of the AT&T Database of Faces.

The face images are the realizations of the random variable. A realization $\mathbf{x}$ in our data set is an individual face pulled from the set of possible outcomes. Realizations are considered just regular signals, not random signals.

As a side note, we can consider an image as a 2-D matrix. In the dataset we are working with, each image is a 112 × 92 image, so each image can be stored in a matrix of size 112 × 92.

$$\mathbf{M}_i = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,92} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,92} \\ \vdots & \vdots & \ddots & \vdots \\ m_{112,1} & m_{112,2} & \cdots & m_{112,92} \end{bmatrix} \tag{1.29}$$

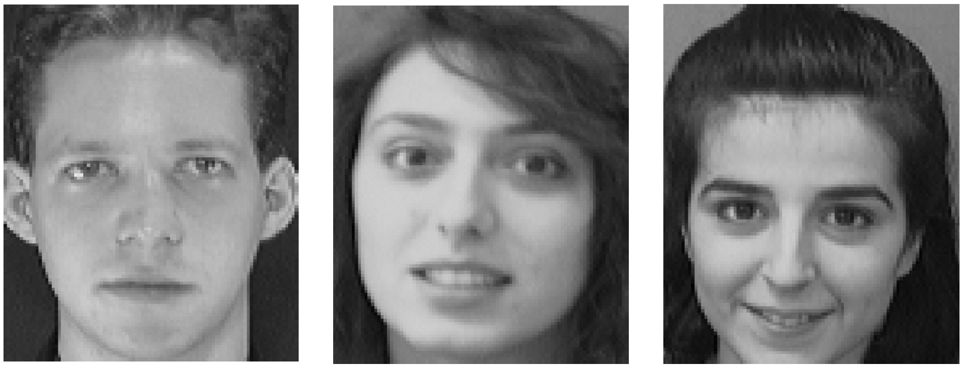

Figure 1.3. Three possible realizations of the random signal X.

To make these images more manageable, we want to remove one dimension by stacking the columns of the image into a vector with length 10,304 (112 multiplied by 92, the total number of pixels in each image). This is called vectorization.

$$\mathbf{x}_i = \left[m_{1,1},\ m_{21},\ \ldots,\ m_{112,1},\ m_{1,2},\ m_{2,2},\ \ldots,\ m_{112,2},\ \vdots,\ m_{1,92},\ m_{2,92},\ \ldots,\ m_{112,92} \right]^T \tag{1.30}$$

1.2.5 Expectation, Variance, and Covariance

As a refresher of definitions, a signal's expectation $\mathbb{E}\left[\mathbf{X}\right]$ is the concatenation of individual expectations.

$$\mathbb{E}\left[\mathbf{X}\right] = \Big[\mathbb{E}\left[X(0)\right],\ \mathbb{E}\left[X(1)\right],\ \ldots\ \mathbb{E}\left[X(N-1)\right]\Big]^T = \iint \mathbf{x} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \tag{1.31}$$

The variance of the nth element Σ_{nn} measures the variability of the nth component.

$$\Sigma_{nn} = \text{var}\left[X(n)\right] = \mathbb{E}\left[\left(X(n) - \mathbb{E}\left[X(n)\right]\right)^2\right] \tag{1.32}$$

We also would like to know how similar each of the signals are, which is what the covariance describes. The covariance Σ_{nm} between two signal components $X(n)$ and $X(m)$ can be written as

$$\Sigma_{nm} = \mathbb{E}\left[\left(X(n) - \mathbb{E}\left[X(n)\right]\right)\left(X(m) - \mathbb{E}\left[X(m)\right]\right)\right] = \Sigma_{mn} \tag{1.33}$$

The covariance Σ_{nm}, measures how much X(n) predicts X(m). If $\Sigma_{nm} = 0$, then the components are unrelated, called orthogonal. If $\Sigma_{nm} > 0$, the components move in the same direction, and if $\Sigma_{nm} < 0$, they move in the opposite direction. This should seem very familiar, because the interpretation of the inner product we learned with Fourier transforms is similar.

1.2.6 Covariance Matrix

The covariance matrix is a way of concatenating or vectorizing these covariance matrices between each pair of components. To illustrate this, let us first assume that $\mathbb{E}\left[\mathbf{X}\right] = \mathbf{0}$ so that the the covariances are $\Sigma_{nm} = \mathbb{E}\left[X(n)X(m)\right]$. This will also show why we usually subtract out the mean of a signal such that the signal is centered around 0. Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the outer product $\mathbf{x}\mathbf{x}^T$. We can write the outer product $\mathbf{x}\mathbf{x}^T$ as:

$$\mathbf{x}\mathbf{x}^T = \begin{bmatrix} x(0)x(0) & \cdots & x(0)x(n) & \cdots & x(0)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(n)x(0) & \cdots & x(n)x(n) & \cdots & x(n)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(N-1)x(0) & \cdots & x(N-1)x(n) & \cdots & x(N-1)x(N-1) \end{bmatrix} \tag{1.34}$$

The expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ implies the expectation of each individual element of the matrix.

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \mathbb{E}[x(0)x(0)] & \cdots & \mathbb{E}[x(0)x(n)] & \cdots & \mathbb{E}[x(0)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(n)x(0)] & \cdots & \mathbb{E}[x(n)x(n)] & \cdots & \mathbb{E}[x(n)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(N-1)x(0)] & \cdots & \mathbb{E}[x(N-1)x(n)] & \cdots & \mathbb{E}[x(N-1)x(N-1)] \end{bmatrix} \tag{1.35}$$

We can then rewrite the (n, m) element of the matrix $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ as the covariance $\Sigma_{n,m}$. The result is a covariance matrix.

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix} \tag{1.36}$$

So, we can define the covariance matrix of a random signal $\mathbf{X}$ as $\mathbf{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$. More generally, when the mean is not null, we define the covariance matrix as

$$\mathbf{\Sigma} := \mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}\left[\mathbf{x}\right]\right)\left(\mathbf{x} - \mathbb{E}\left[\mathbf{x}\right]\right)^T\right] \tag{1.37}$$

There are a couple of details to the structure of covariance matrices. When the mean is null, the (n, m) element of $\mathbf{\Sigma}$ is the covariance $\Sigma_{n,m}$. The diagonal of $\mathbf{\Sigma}$ contains the autovariances $\Sigma_{nn} = \text{var}\left[X(n)\right]$, the variance between a signal and itself. The covariance matrix is also symmetric:

$$((\mathbf{\Sigma}))_{n,m} = \Sigma_{nm} = \Sigma_{mn} = ((\mathbf{\Sigma}))_{mn}$$

.

The covariance matrix tells you what it means to have change on a specific random signal. Because it defines and helps us understand the variability for a specific signal, it can be used to transform the signal. After all, the DFT was built on the notion of change with respect to complex exponentials.

1.3 Principle Component Analysis Transform

From the covariance matrix, we understand the variability for a specific signal. We can derive further information on the order of significance of specific components through the concept of eigenvalues and eigenvectors.

1.3.1 Eigenvectors and Eigenvalues of the Covariance Matrix

Consider a vector $\mathbf{v}$ with N elements, $\mathbf{v} = [v(0), v(1), \ldots, v(N-1)]$. We say that $\mathbf{v}$ is an eigenvector of Σ if for some scalar $\lambda \in \mathbb{R}$,

$$\mathbf{\Sigma v} = \lambda \mathbf{v} \tag{1.38}$$

where λ is the eigenvalue associated to $\mathbf{v}$. In other words, the product $\Sigma\mathbf{v}$ results in a vector in the same direction as $\mathbf{v}$, but with a different-scaled length. More formally, $\Sigma\mathbf{v}$ is collinear with $\mathbf{v}$. This is not the case for non-eigenvectors $\mathbf{w}$, where $\mathbf{w}$ and $\Sigma\mathbf{v}$ point in different directions. This fact simplifies things for us, because usually when you multiply a matrix with a vector, you get a vector with a different direction and length, but with the covariance matrix and an eigenvector, their product only affects the length.

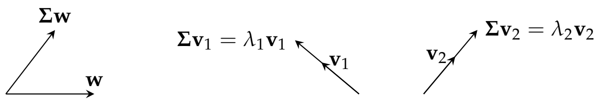
$\mathbf{\Sigma w}$
$\mathbf{w}$
$\mathbf{\Sigma v}_1 = \lambda_1 \mathbf{v}_1$
$\mathbf{v}_1$
$\mathbf{v}_2$
$\mathbf{\Sigma v}_2 = \lambda_2 \mathbf{v}_2$

Figure 1.4. Three vectors multiplied by a covariance matrix Σ, where $\mathbf{w}$ is a non-eigenvector and $\mathbf{v}_i$ are eigenvectors. Notice how the product $\Sigma\mathbf{v}_i$ is simply a scaled version of that vector $\mathbf{v}_i$.

If $\mathbf{v}$ is an eigenvector, $\alpha\mathbf{v}$ is also an eigenvector for any scalar $\alpha \in \mathbb{R}$. This is because we are simply changing the length of the vector while maintaining the direction.

$$\mathbf{\Sigma}(\alpha\mathbf{v}) = \alpha(\mathbf{\Sigma v}) = \alpha\lambda\mathbf{v} = \lambda(\alpha\mathbf{v}) \tag{1.39}$$

So, eigenvectors are defined by a constant. To keep things consistent, we will be using normalized eigenvectors with unit energy, i.e. $||\mathbf{v}||^2 = 1$. To normalize an eigenvector $\mathbf{v}$ with $||\mathbf{v}||^2 \neq 1$, divide $\mathbf{v}$ by the norm $||\mathbf{v}||$. We will also be assuming that there are N eigenvalues and distinct associated eigenvectors. This is not necessarily true, as there are a few details where this is not the case, but we will assume such for our case.

1.3.2 Ordering of Eigenvalues and Eigenvectors

Theorem 5 *The eigenvalues of* $\mathbf{\Sigma}$ *are real and nonnegative* $\Rightarrow \lambda \in \mathbb{R}$ *and* $\lambda \geq 0$.

Proof: To prove this, we begin by writing $\lambda = \mathbf{v}^H\mathbf{\Sigma v}/||\mathbf{v}||^2$. To show that λ is real, we can write that

$$\mathbf{v}^H\mathbf{\Sigma v} = \mathbf{v}^H(\mathbf{\Sigma v}) = \mathbf{v}^H(\lambda\mathbf{v}) = \lambda\mathbf{v}^H\mathbf{v} = \lambda||\mathbf{v}||^2 \tag{1.40}$$

To show that $\mathbf{v}^T\mathbf{\Sigma v}$ is nonnegative, and assuming $\mathbb{E}[\mathbf{x}] = \mathbf{0}$, we can say

$$\mathbf{v}^H\mathbf{\Sigma v} = \mathbf{v}^H\mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]\mathbf{v} = \mathbb{E}\left[\mathbf{v}^H\mathbf{x}\mathbf{x}^H\mathbf{v}\right] = \mathbb{E}\left[\left(\mathbf{v}^H\mathbf{x}\right)\left(\mathbf{x}^H\mathbf{v}\right)\right] = \mathbb{E}\left[\left(\mathbf{v}^H\mathbf{x}\right)^2\right] \geq 0 \tag{1.41}$$

■

From this proof, we can now order eigenvalues from largest to smallest, e.g. $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1}$. The eigenvectors also inherit the same order as their associated eigenvalue, e.g. $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}$. This order is important because it tells us something about how much change happens over different directions. Since eigenvalues scale eigenvectors, a larger scaling from a larger eigenvalue displays larger variability.

1.3.3 Orthonormality of Eigenvectors

Theorem 6 *Eigenvectors of $\mathbf{\Sigma}$ associated with different eigenvalues are orthogonal.*

Proof: Normalized eigenvectors $\mathbf{v}$ and $\mathbf{u}$ are associated with eigenvalues $\lambda \neq \mu$.

$$\mathbf{\Sigma v} = \lambda \mathbf{v}, \qquad \mathbf{\Sigma u} = \mu \mathbf{u} \tag{1.42}$$

Since the matrix $\mathbf{\Sigma}$ is symmetric, we have $\mathbf{\Sigma}^H = \mathbf{\Sigma}$, and it follows that

$$\mathbf{u}^H \Sigma \mathbf{v} = \left(\mathbf{u}^H \Sigma \mathbf{v}\right)^H = \mathbf{v}^H \Sigma^H \mathbf{u} = \mathbf{v}^H \Sigma \mathbf{u} \tag{1.43}$$

If we make $\Sigma \mathbf{v} = \lambda \mathbf{v}$ on the leftmost side and $\Sigma \mathbf{u} = \mu \mathbf{u}$ on the rightmost side, then

$$\mathbf{u}^H \lambda \mathbf{v} = \lambda \mathbf{u}^H \mathbf{v} = \mu \mathbf{v}^H \mathbf{u} = \mathbf{v}^H \mu \mathbf{u} \tag{1.44}$$

From this, we conclude that the eigenvalues are different. This relationship can only be true if $\mathbf{v}^H \mathbf{u} = 0$, or if $\mathbf{v}$ and $\mathbf{u}$ are orthogonal. ■

1.3.4 Eigenvectors of Face Images

Below are visualizations in 1-D and 2-D of the first four eigenvectors of the covariance matrix.

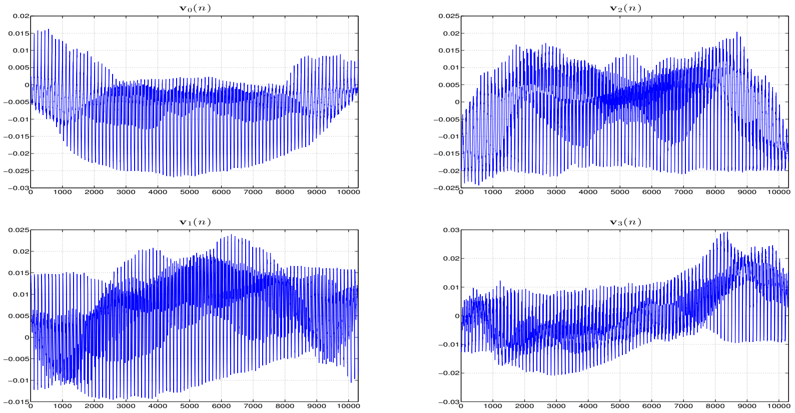

Figure 1.5. One dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$.

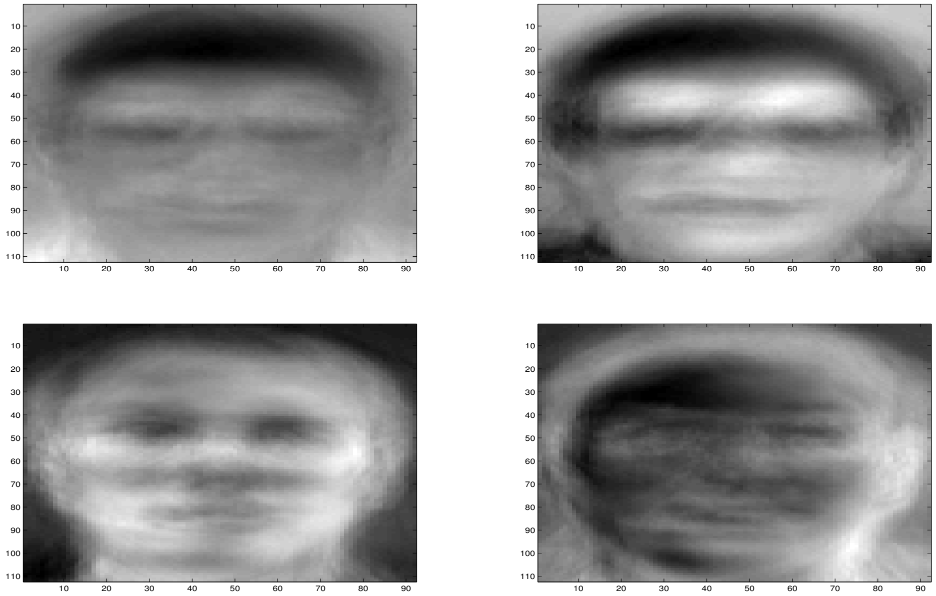

Figure 1.6. Two dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$.

1.3.5 Eigenvector Matrix

Define the matrix $\mathbf{T}$ whose kth column is the kth eigenvector of $\mathbf{\Sigma}$,

$$\mathbf{T} = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}] \tag{1.45}$$

Since the eigenvectors $\mathbf{v}_k$ are orthonormal, the product $\mathbf{T}^H\mathbf{T}$ is the identity matrix.

$$\mathbf{T}^H\mathbf{T} = \begin{array}{c} \\ \begin{bmatrix} \mathbf{v}_0^H \\ \vdots \\ \mathbf{v}_k^H \\ \vdots \\ \mathbf{v}_{N-1}^H \end{bmatrix} \end{array} \begin{array}{c} \begin{bmatrix} \mathbf{v}_0 & \cdots & \mathbf{v}_k & \cdots & \mathbf{v}_{N-1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{v}_0^H\mathbf{v}_0 & \cdots & \mathbf{v}_1^H\mathbf{v}_k & \cdots & \mathbf{v}_0^H\mathbf{v}_{N-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_k^H\mathbf{v}_0 & \cdots & \mathbf{v}_k^H\mathbf{v}_k & \cdots & \mathbf{v}_k^H\mathbf{v}_{N-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_{N-1}^H\mathbf{v}_{N-1} & \cdots & \mathbf{v}_{N-1}^H\mathbf{v}_k & \cdots & \mathbf{v}_{N-1}^H\mathbf{v}_{N-1} \end{bmatrix} \end{array} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

Thus, because we observe that $\mathbf{T}^H\mathbf{T} = \mathbf{I}$, we show that eigenvector matrix $\mathbf{T}$ is Hermitian.

1.3.6 Principle Component Analysis Transform

As we discussed earlier, any Hermitian matrix $\mathbf{T}$ can be used to define an information processing transform like DFT or DCT. As such, we can use the Hermitian eigenvector matrix to define a transform, namely the principle component analysis (PCA) transform.

We define the PCA transform as $bby = \mathbf{T}^H\mathbf{x}$, and the inverse PCA (iPCA) transform as $\tilde{\mathbf{x}} = \mathbf{T}\mathbf{y}$.

We can quickly prove that the iPCA is truly the inverse of the PCA since $\mathbf{T}$ is Hermitian.

$$\tilde{\mathbf{x}} \;=\; \mathbf{T}\mathbf{y} \;=\; \mathbf{T}\left(\mathbf{T}^H\mathbf{x}\right) \;=\; \mathbf{T}\mathbf{T}^H\mathbf{x} \;=\; \mathbf{I}\mathbf{x} \;=\; \mathbf{x} \tag{1.46}$$

From this result, we show that the inverse of PCA transform $\mathbf{y}$ is an equivalent representation of $\mathbf{x}$, meaning we can go back and forth using the defined PCA transform without losing information or changing the signal.

We can also show Parseval's theorem holds because $\mathbf{T}$ is Hermitian.

$$\|\mathbf{x}\|^2 \;=\; \mathbf{x}^H\mathbf{x} \;=\; (\mathbf{T}\mathbf{y})^H\,\mathbf{T}\mathbf{y} \;=\; \mathbf{y}^H\mathbf{T}^H\mathbf{T}\mathbf{y} \;=\; \mathbf{y}^H\mathbf{y} \;=\; \|\mathbf{y}\|^2 \tag{1.47}$$

This means that modifying the elements y_k means altering the energy composition of the signal.

The PCA transform is defined for any signal $\mathbf{x}$, but we expect it to work well only when x is a realization $\mathbf{X}$.

We can compare the expanded mathematical forms of the iPCA with the iDFT as well.

$$x(n) = \sum_{k=0}^{N-1} y(k)v_k(n) \quad \Leftrightarrow \quad x(n) = \sum_{k=0}^{N-1} X(k)e_{kN}(n) \tag{1.48}$$

As we can see, the forms are the same except that they use different bases for the expansion. As we had developed a "sense" with the DFT to view signals through the frequency domain, we have now developed a new "sense" that is not generic, but rather adapted to the random signal $\mathbf{X}$.

If we perform PCA on a face image with 10,304 pixels , we can see that there is substantial energy in the first 15 PCA coefficients $y(k)$ with $k15$, since these have the most significant coefficient magnitudes. Furthermore, almost all of the energy of the face image is contained in the first 50 PCA coefficients. Being able to go from 10,304 pixels to only 50 to represent almost the entire image is a compression factor of more than 200.

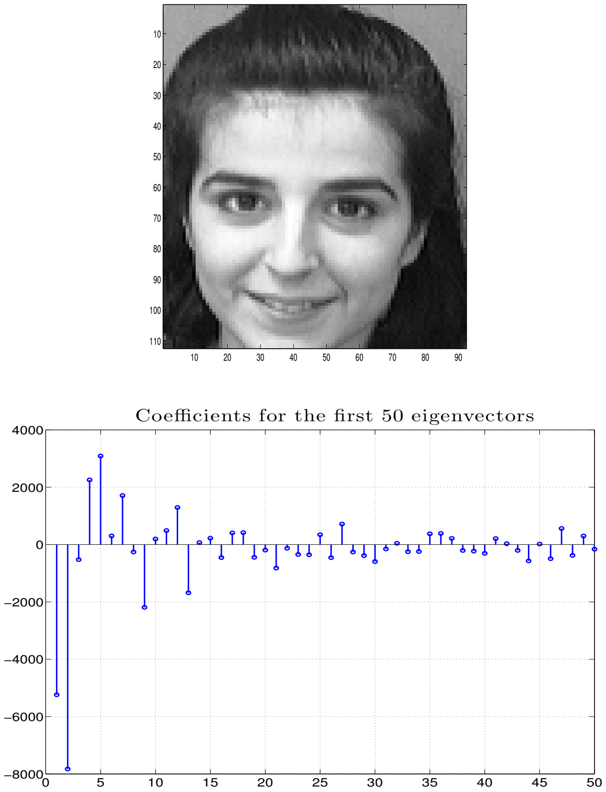

Figure 1.7. Face image and its PCA coefficients for the first 50 eigenvectors.

We can see the impact of using a relatively small number of principle components in reconstruction. As seen in the images below, increasing the number of coefficients increases the accuracy of the reconstruction. Once we reach 50 principle components used in reconstruction, we obtain a reconstructed image that is almost identical to the original image.

If we examine the PCA transform for two different images of the same person, we can see that the coefficients are similar, even if the pose, orientation, or expression of the face are different.

However, if we examine the PCA transform for two different images with similar poses and expressions, we end up with different PCA coefficients. This observation will be useful in performing facial recognition, which we will discuss later.

1.4 Dimensionality Reduction

1.4.1 Compression with the DFT

We have already seen compression in the context of the DFT and DCT in 1D and 2D, where we compressed voice and image signals. We can transform a signal $\mathbf{x}$ into the frequency domain with the DFT $\mathbf{X} = \mathbf{F}^H\mathbf{x}$, and we can recover the original signal from $\mathbf{X}$ through the iDFT $\mathbf{x} = \mathbf{FX}$. We performed compression by retaining K of the N DFT coefficients and using only those through the iDFT to obtain the approximated signal $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}}(n) = \sum_{k=0}^{K-1} X(k)e^{j2\pi kn/N} \tag{1.49}$$

We can also define the compressed DFT as

$$\tilde{\mathbf{X}}(k) = X(k) \quad \text{for} \quad k < K, \qquad \tilde{\mathbf{X}}(k) = 0 \text{ otherwise} \tag{1.50}$$

and we can define the reconstructed signal from the iDFT as $\tilde{\mathbf{x}} = \mathbf{F}\tilde{\mathbf{X}}$.

1.4.2 Compression with the PCA

Performing compression with the PCA is mathematically almost identical as the DFT, except that we use a different transformation matrix. The DFT uses the matrix $\mathbf{F}^H$, which is a matrix of complex exponentials, and the PCA uses $\mathbf{T}^H$, which is a matrix constructed from the eigenvectors of the covariance matrix. Just as we can transform a signal $\mathbf{x}$ into the frequency domain, we can transform $\mathbf{x}$ into the eigenvector domain with the PCA $\mathbf{y} = \mathbf{T}^H\mathbf{x}$. We can recover $\mathbf{x}$ from $\mathbf{y}$ through the iPCA $\mathbf{x} = \mathbf{Ty}$. We can compress $\mathbf{x}$ by retaining K out of all N PCA coefficients, which are the K eigenvectors associated with the K first eigenvalues, to write

$$\tilde{\mathbf{x}}(n) = \sum_{k=0}^{K-1} y(k)\mathbf{v}_k(n) \tag{1.51}$$

We equivalently define the compressed PCA as

$$\tilde{\mathbf{y}}(k) = y(k) \quad \text{for} \quad k < K, \qquad \tilde{\mathbf{y}}(k) = 0 \text{ otherwise} \tag{1.52}$$

and define the reconstructed signal from the iPCA as $\tilde{\mathbf{x}} = \mathbf{T}\tilde{\mathbf{y}}$.

1.4.3 Why Keep First K Coefficients?

We keep the first K coefficients in compression strategically to represent as much of the original signal with as few pieces of data as possible. With the DFT, we would like to retain the coefficients that represent faster oscillations and, in turn, faster variation, because these signals tell us more about how significant or distinct the signal is. In fact, the first K DFT coefficients do not always correlate to the components with the fastest variation, so sometimes we keep the K *largest* coefficients instead. In PCA, when we examine the eigendecomposition of the covariance matrix, we wish to keep the eigenvalues with larger values because they represent more variability and therefore more dominant features. These larger eigenvalues correspond to eigenvectors with lower ordinality. Eigenvectors with large ordinality represent finer signal features, which we can often omit while still retaining the majority of a signal.

1.4.4 Dimensionality Reduction

A more accurate name for PCA compression is called dimensionality reduction. When we are taking the K first PCA coefficients, we are not compressing the signal per se. Rather, we are reducing the number of dimensions.

As an example, consider the covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix} \tag{1.53}$$

The covariance matrix has the eigenvectors $\mathbf{v}_0$ and $\mathbf{v}_1$, which are in the 45° and -45° directions, with eigenvalues $\lambda_0 = 2$ and $\lambda_1 = 1$.

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

We can draw the covariance matrix as an ellipse using the eigenvectors as the directions of the major and minor axes and the eigenvalues as the lengths of the corresponding eigenvectors, as shown below.

We currently have 2 signals to describe the dataset with, one for each eigenvector $\mathbf{v}_0$ and $\mathbf{v}_1$. Say you could only describe the set of data with 1 signal instead of 2. Which one would we pick? We would choose the signal with the longer-length eigenvector, $\mathbf{v}_0$, because the longer length represents a large eigenvalue, which describes a signal with more variability. This describes something that is more important than the other signal, and you gain more information when you view the signal along the direction of greater variability. With this in mind, we can now reduce the dimension and study the one-dimensional signal $\tilde{\mathbf{x}} = y(0)\mathbf{v}_0$ instead of the original two-dimensional signal $\mathbf{x}$.

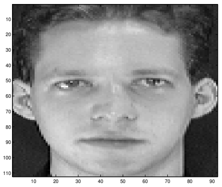

Figure 1.8. Original Image

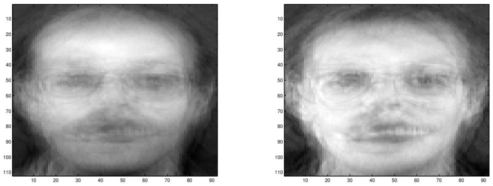

Figure 1.9. Reconstructed images using 1 and 5 principle components

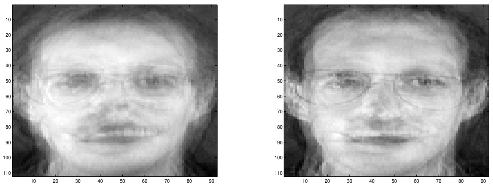

Figure 1.10. Reconstructed images using 10 and 20 principle components

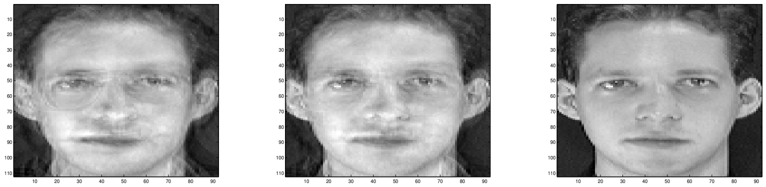

Figure 1.11. Reconstructed images using 30, 40, and 50 principle components

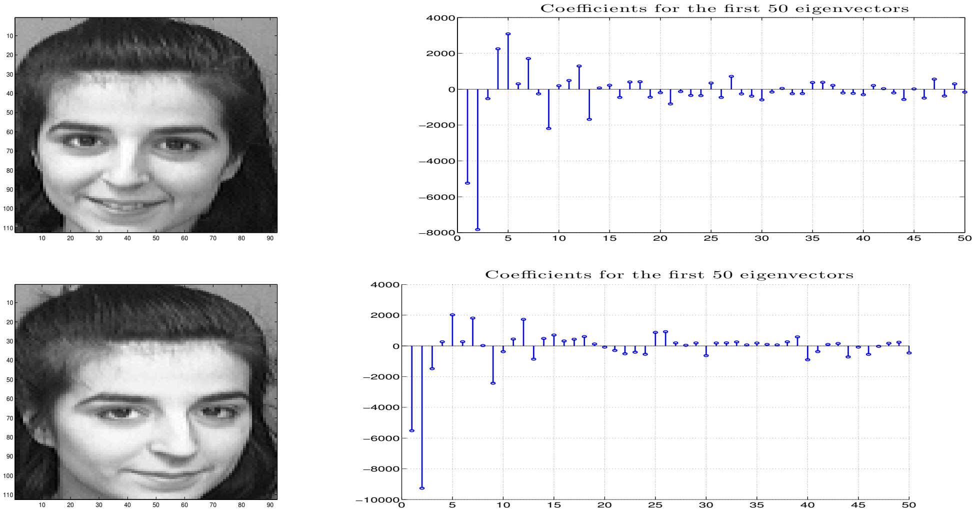

Figure 1.12. Two face images of the same woman but in different poses and expressions, and their associated PCA coefficients for the first 50 eigenvalues. Notice that the first two coefficients are almost identical.

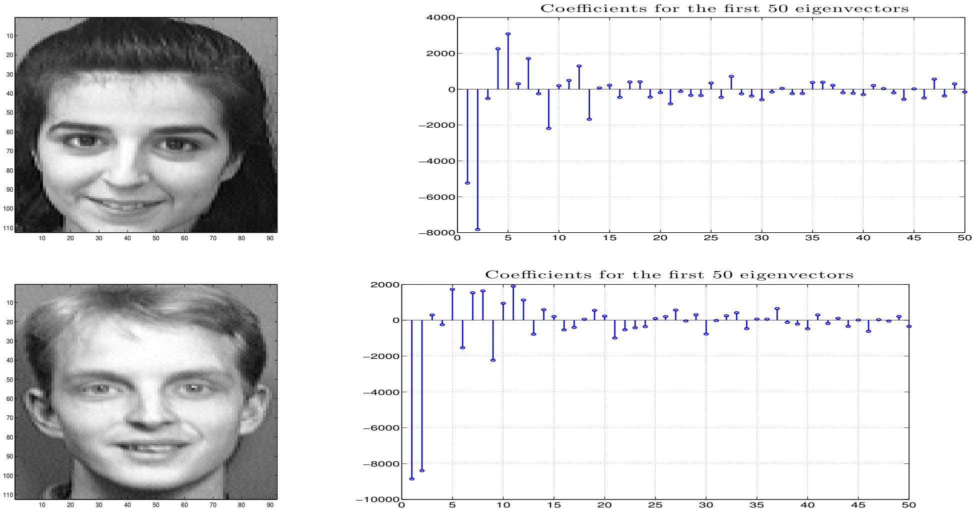

Figure 1.13. Two face images of two different people with similar poses and expressions, and their associated PCA coefficients for the first 50 eigenvalues. In contrast to the comparison of two images of the same person, the PCA coefficients of two images of different people are much different.

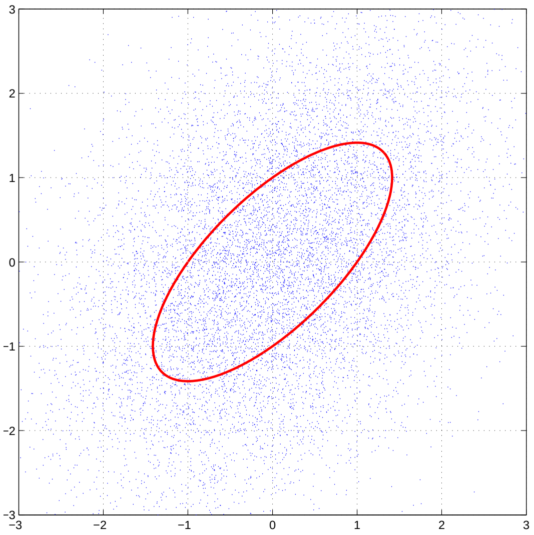

Figure 1.14. An example dataset with covariance matrix $\mathbf{\Sigma}$