SVD and PCA

QEA

Spring 2016

Using Eigendecomposition to Understand Why the Principle Eigenvector of the Covariance Matrix Captures the Largest Spread of the Data

In the last class you worked on examples involving temperature data and eigenvectors of the covariance matrices formed from that data. Recall that if we have two different data variables x and y (e.g. corresponding to temperatures in Boston and Sao Paolo), with x_i and y_i being different values in the data set we can define a a matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} x_1 - \mu_x & y_1 - \mu_y \\ x_2 - \mu_x & y_2 - \mu_y \\ x_3 - \mu_x & y_3 - \mu_y \\ \vdots & \vdots \\ x_N - \mu_x & y_N - \mu_y \end{pmatrix}$$

where μ_x is the mean of the first column, and N is the number of samples (rows). Recall that the covariance matrix of x and y is $\mathbf{R} = \mathbf{A}^T\mathbf{A}$

You saw that the eigenvector corresponding to the largest eigenvalue of $\mathbf{R}$, which is also called the *principal eigenvector* is in the direction with the largest variation in the data. The eigenvector corresponding to the second largest eigenvalue points in the direction orthogonal to the principle eigenvector in which there is the second largest variation in the data and so on. Of course when you only have two different variables in the data set, the matrix $\mathbf{R}$ has only 2 orthogonal eigenvectors.

To illustrate, consider Figure 1 which shows the centered (mean subtracted) temperatures of Boston vs Sao Paolo. We have also plotted the two eigenvectors, scaled by the square-root of their corresponding eigenvalues, to illustrate the relative variation of the data along the directions of the two eigenvectors. Notice that the principle eigenvector is in the direction of greatest variation in the data. Figure 2 is a similar plot with the temperatures of Boston and Washington DC instead.

In the remainder of this section, we shall prove why it is that the principle eigenvector of the covariance matrix lies in the direction of greatest variation of the data. This proof is rather long, however, all the steps involved have already been introduced in the class. As

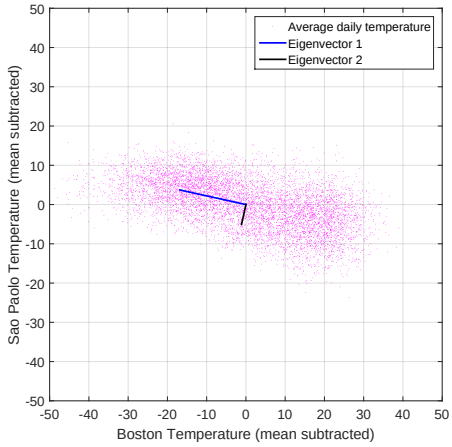

Figure 1: Centered average daily temperatures of Boston vs Sao Paolo, with the eigenvectors of the covariance matrix.

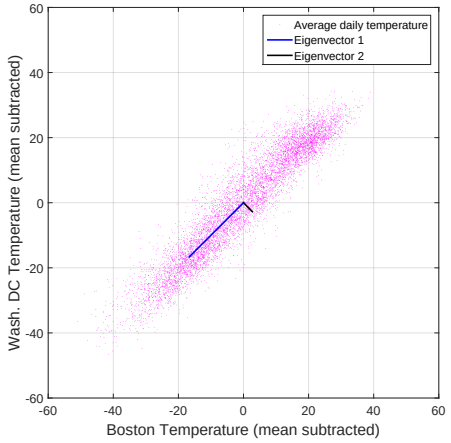

Figure 2: Centered average daily temperatures of Boston vs Washington DC, with the eigenvectors of the covariance matrix.

such it is a good way to bring together many of the concepts we have seen so far. While we have included some questions in the steps of the derivation, they are formulated in the form of questions such as "convince yourself that ....". So if you feel stuck in any one of the questions, please move along in the derivation until you get to the last exercise in this section.

Proof that the principle eigenvector of the covariance matrix is in the direction of greatest variation in the data

Let $\mathbf{b}_i$ be defined as the vector made up of the ith data samples with the mean subtracted out. If for example, we are dealing with two data variables, x and y (e.g. representing temperature in Boston and Sao Paolo),

$$\mathbf{b}_i = \begin{pmatrix} x_i - \mu_x \\ y_i - \mu_y \end{pmatrix}$$

Now let's define a unit vector $\mathbf{u}$ which points in some (at this point unknown) direction, with unit length. Recall from the take home exercise that the dot product of $\mathbf{u}$ and $\mathbf{b}_i$, tells you the length of the component of $\mathbf{b}_i$ that lies in the direction of $\mathbf{u}$. Let's define the length of this projection as w_i. Thus we have

$$w_i = \mathbf{u} \cdot \mathbf{b}_i = \mathbf{u}^T \mathbf{b}_i \tag{1}$$

Hence, you can think of w_i as a measure of "how much" of $\mathbf{b}_i$ lies in the direction of $\mathbf{u}$.

Recall that the variance of the set of data points $w_1, w_2, \cdots w_N$ is

$$\sigma_w^2 = \frac{1}{N-1} \sum_{j=1}^{N} (w_i - \mu_w)^2 \tag{2}$$

where μ_w is the mean or average of $w_1, w_2, \cdots w_N$. Since w_i measures the component of $\mathbf{b}_i$ that lies in the direction of $\mathbf{u}$, σ_w^2 is a measure of the variation in the data set in the direction of $\mathbf{u}$. The goal of the rest of this section is to prove that the $\mathbf{u}$ which maximizes σ_w^2 is precisely the principle eigenvector of the covariance matrix of the data.

The mean of each entry in $\mathbf{b}_i$ is zero because $\mathbf{b}_i$ equals data samples with the means subtracted out, in other words, the entries of $\mathbf{b}_i$ have mean equal to zero. Hence,

$$\begin{aligned} \mu_w &= \frac{1}{N} \sum_{i=1}^{N} w_i = \frac{1}{N} \sum_{i=1}^{N} \mathbf{u}^T \mathbf{b}_i \\ &= \mathbf{u}^T \sum_{i=1}^{N} \frac{1}{N} \mathbf{b}_i = \mathbf{0} \end{aligned}$$

Therefore we can rewrite σ_w^2 as

$$\sigma_w^2 = \frac{1}{N-1}\sum_{j=1}^{N} w_i^2 = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{u}^T\mathbf{b}_i)^2 \tag{3}$$

$$= \sum_{j=1}^{N}\left(\frac{1}{\sqrt{N-1}}\mathbf{u}^T\mathbf{b}_i\right)\left(\frac{1}{\sqrt{N-1}}\mathbf{b}_i^T\mathbf{u}\right) \tag{4}$$

1. Convince yourself that

$$\sigma_w^2 = \mathbf{u}^T\mathbf{A}^T\mathbf{A}\mathbf{u} = \mathbf{u}^T\mathbf{R}\mathbf{u} \tag{5}$$

Since σ_w^2 is the variation in the data set in the direction $\mathbf{u}$, at this point, what is left is to show that the unit vector $\mathbf{u}$ which maximizes the above expression is precisely the principle eigenvector of the covariance matrix $\mathbf{R}$. Our strategy to do this is to find an upper bound to σ_w^2 and then show that the principle eigenvector results in a value of σ_w^2 which meets that upper bound.

An upper bound on the variation

To find the unit vector $\mathbf{u}$ which maximizes (5) involves constrained optimization, since we are trying to maximize σ_w^2 subject to the constraint that $\mathbf{u}$ has unit length. Constrained optimization is a large, rich, and interesting topic. Fortunately (or unfortunately), we can do this particular constrained optimization using the tools we have already developed, without requiring a foray into advanced optimization techniques. To simplify matters, let's introduce a new vector $\mathbf{w}$ such that

$$\mathbf{u} = \frac{\mathbf{w}}{||\mathbf{w}||} = \frac{\mathbf{w}}{\sqrt{\mathbf{w}^T\mathbf{w}}} \tag{6}$$

By this definition, $\mathbf{u}$ is a normalized version of $\mathbf{w}$, hence, $\mathbf{u}$ will have unit length, no matter what $\mathbf{w}$ is. Substituting this definition into (5) yields

$$\sigma_w^2 = \frac{\mathbf{w}^T\mathbf{R}\mathbf{w}}{\mathbf{w}^T\mathbf{w}} \tag{7}$$

Now, we need to maximize the above equation over any $\mathbf{w}$, which will result in a $\mathbf{u}$ which is automatically of unit length. Next, we perform an eigendecomposition on $\mathbf{R}$ as follows

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}.$$

From what we know about the eigendecomposition, the columns of $\mathbf{Q}$ are the eigenvectors of $\mathbf{R}$ and $\mathbf{\Lambda}$ is a diagonal matrix containing the

corresponding eigenvalues in its diagonal entries. Let these eigenvectors be denoted by $\mathbf{v}_1, \mathbf{v}_2$, and the corresponding eigenvalues be λ_1, λ_2. Here, we are assuming that we have two variables x and y in our data set, but these steps carry over directly to the case when we have more than two variables. We further know that the eigenvectors are orthongonal, because $\mathbf{R}$ is a covariance matrix. Additionally, we can normalize the eigenvectors to have unit length (in fact MATLAB's `eig` command produces normalized eigenvectors.

2. Convince yourself that you can write (7) as follows

$$\sigma_w^2 = \frac{\sum_{i=1}^{2} \mathbf{w}^T \mathbf{v}_i \lambda_i \mathbf{v}_i^T \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \tag{8}$$

$$= \frac{\sum_{i=1}^{2} \lambda_i \mathbf{w}^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \tag{9}$$

3. Next, using the fact that the eigenvectors $\mathbf{v}_i$ form a basis in $\mathbf{R}^2$ (for the case of 2 variables), and $\mathbf{w}^T \mathbf{v}_i$ is the projection of $\mathbf{w}$ onto $\mathbf{v}_i$, convince yourself that

$$\mathbf{w} = \sum_{i=1}^{2} \alpha_i \mathbf{v}_i. \tag{10}$$

   where $\alpha_i = \mathbf{w}^T \mathbf{v}_i$.

Using this property, we have that

$$\begin{aligned}
\mathbf{w}^T \mathbf{w} &= \left(\sum_{i=1}^{2} \alpha_i \mathbf{v}_i \right)^T \left(\sum_{j=1}^{2} \alpha_i \mathbf{v}_j \right) \\
&= (\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2)^T (\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2) \\
&= (\alpha_1 \mathbf{v}_1^T + \alpha_2 \mathbf{v}_2^T)(\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2) \\
&= \alpha_1^2 \mathbf{v}_1^T \mathbf{v}_1 + \alpha_1 \alpha_2 \mathbf{v}_1^T \mathbf{v}_2 + \alpha_2 \alpha_1 \mathbf{v}_2^T \mathbf{v}_1 + \alpha_2^2 \mathbf{v}_2^T \mathbf{v}_2
\end{aligned}$$

Note that while the above expression is specific to the case of two variables, this step carries forward in a straightforward but perhaps more tedious way to cases with more than two variables.

4. Using the properties of the eigenvectors $\mathbf{v}_i$, convince yourselves that the above expression is equivalent to the following

$$\mathbf{w}^T \mathbf{w} = \alpha_1^2 + \alpha_2^2 \tag{11}$$

Substituting $\alpha_i = \mathbf{w}^T \mathbf{v}_i$ into the numerator and (11) into the deniminator of (9) yields

$$\sigma_w^2 = \frac{\sum_{i=1}^{2} \lambda_i \alpha_i^2}{\sum_{j=1}^{2} \alpha_j^2} \tag{12}$$

Now let λ_M denote the largest eigenvalue. Hence, we have

$$\begin{aligned}\sigma_w^2 &= \frac{\sum_{i=1}^2 \lambda_i \alpha_i^2}{\sum_{j=1}^2 \alpha_j^2} \\ &\leq \frac{\sum_{i=1}^2 \lambda_M \alpha_i^2}{\sum_{j=1}^2 \alpha_j^2} = \lambda_M \end{aligned} \tag{13}$$

Exercise

5. Let $\mathbf{v}_M$ denote the principle eigenvector of the covariance matrix $\mathbf{R}$. Show that if $\mathbf{w} = \mathbf{v}_M$, then $\mathbf{u} = \mathbf{w}$, and

$$\sigma_w^2 = \lambda_M \tag{14}$$

   Using (13) and the previous expression, convince yourself that $\mathbf{v}_M$ points in the direction of largest variation in the data set.

If we have more than two variables, the eigenvector corresponding to the second largest eigenvalue lies in the direction of greatest variation, among the directions that are orthogonal to the principle eigenvalue. This property can be proved using similar steps as above, except the components that are in the direction of the principle eigenvalue have to be subtracted from the data. In that case, the direction with the second most variation in the original data will become the direction with the most variation in this new set of data points.

Where is this used?

This approach is used in a number of different fields. In the face-recognition project, you will construct covariance matrices of the data and find the directions in which there are the largest variations in the data. These directions will also be the ones in which you can better distinguish between one face and from another. Another application for this approach is lossy data compression. You can represent higher dimensional data using fewer numbers if you only encode the information in the directions of a few of the "strongest" eigenvectors of the covariance matrix. Since these directions are the ones with significant variation, most of the information can still be retained.

Singular Value Decomposition (SVD)

We just met the Eigenvalue Decomposition (EVD), which we used on square matrices. There is no EVD for rectangular matrices, but there

does exist a generalization known as the Singular Value Decomposition, which is one of the most useful matrix decompositions in applied linear algebra. We will begin with a statement of the SVD, consider an example, and then discuss some of the typical applications. See the following webpage for a good geometric discussion of the SVD: http://www.ams.org/samplings/feature-column/fcarc-svd

We would also like you to read and work through the practice problems in the image compression paper linked from the website.

Let $\mathbf{A}$ be a real $n \times m$ matrix. $\mathbf{A}$ has a *singular value decomposition* (SVD) of the form

$$\mathbf{A}_{n\times m} = \mathbf{U}_{n\times r}\mathbf{\Sigma}_{r\times r}\mathbf{V}^T_{m\times r}$$

where $\mathbf{U}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{A}\mathbf{A}^T$, $\mathbf{\Sigma}$ is a diagonal matrix with r positive entries, and $\mathbf{V}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{A}^T\mathbf{A}$. The diagonal entries of $\mathbf{\Sigma}$ are called the singular values and are denoted by σ_i. They are organized so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$: the r nonzero eigenvalues of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are σ_i^2. Please note that this is the *reduced* or *economy* SVD - there is a more general form but this is the most useful in a practical setting.

Here is an example. Consider the rectangular matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Let's first form the matrix $\mathbf{A}^T\mathbf{A}$,

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 84 & 100 \\ 100 & 120 \end{bmatrix}$$

The eigenvalues of $\mathbf{A}^T\mathbf{A}$ are 203.6071 and 0.3929 respectively. The singular values are the square roots of these, namely $\sigma_1 = 14.2691$ and $\sigma_2 = 0.6268$ respectively. The associated eigenvectors are

$$\mathbf{v}_1 = \begin{bmatrix} 0.6414 \\ 0.7672 \end{bmatrix}$$

and

$$\mathbf{v}_2 = \begin{bmatrix} -0.7672 \\ 0.6414 \end{bmatrix}$$

so that the matrix $\mathbf{\Sigma}$ is

$$\mathbf{\Sigma} = \begin{bmatrix} 14.2691 & 0 \\ 0 & 0.6268 \end{bmatrix}$$

and the matrix $\mathbf{V}$ is

$$\mathbf{V} = \begin{bmatrix} 0.6414 & -0.7672 \\ 0.7672 & 0.6414 \end{bmatrix}$$

To determine the $\mathbf{U}$ matrix, we now consider the matrix $\mathbf{A}\mathbf{A}^T$ (we will find another technique for determining $\mathbf{U}$ soon),

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} 5 & 11 & 17 & 23 \\ 11 & 25 & 39 & 53 \\ 17 & 39 & 61 & 83 \\ 23 & 53 & 83 & 113 \end{bmatrix}$$

The eigenvalues of $\mathbf{A}\mathbf{A}^T$ are 203.6071, 0.3929, 0, and 0. The zeros are to be expected, for reasons that we will discuss later. The eigenvectors corresponding to the non-zero eigenvalues are

$$\mathbf{u}_1 = \begin{bmatrix} 0.1525 \\ 0.3499 \\ 0.5474 \\ 0.7448 \end{bmatrix}$$

and

$$\mathbf{u}_2 = \begin{bmatrix} -0.8226 \\ -0.4214 \\ -0.0201 \\ 0.3812 \end{bmatrix}$$

so that the $\mathbf{U}$ matrix is

$$\mathbf{U} = \begin{bmatrix} 0.1525 & -0.8226 \\ 0.3499 & -0.4214 \\ 0.5474 & -0.0201 \\ 0.7448 & 0.3812 \end{bmatrix}$$

The original matrix $\mathbf{A}$ therefore has the SVD

$$\mathbf{A} = \begin{bmatrix} 0.1525 & -0.8226 \\ 0.3499 & -0.4214 \\ 0.5474 & -0.0201 \\ 0.7448 & 0.3812 \end{bmatrix} \begin{bmatrix} 14.2691 & 0 \\ 0 & 0.6268 \end{bmatrix} \begin{bmatrix} 0.6414 & -0.7672 \\ 0.7672 & 0.6414 \end{bmatrix}^T$$

1. Find the SVD of the following matrices by considering the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$. (You can use *eig* in MATLAB.)

   (a)

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

2. Now use the *svd* command in MATLAB and check your work makes sense.

Why does this work?

Let's consider the product $\mathbf{A}^T\mathbf{A}$: all non-zero eigenvalues of this matrix are positive. If λ_i is a non-zero eigenvalue of $\mathbf{A}^T\mathbf{A}$ with eigenvector $\mathbf{v}_i$, then we can write

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{v}_i$$

where $\sigma_i = \sqrt{\lambda_i}$. If we left multiply by $\mathbf{v}_i^T$ we get

$$\mathbf{v}_i^T\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \mathbf{v}_i^T\sigma_i^2\mathbf{v}_i$$

and if we use the fact that $\mathbf{v}_i$ is normalized and the properties of transposes

$$(\mathbf{A}\mathbf{v}_i)^T(\mathbf{A}\mathbf{v}_i) = \sigma_i^2$$

which shows that the length of the vector $\mathbf{A}\mathbf{v}_i$ is just σ_i.

Let's return to our earlier statement

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{v}_i$$

Now if we left multiply by $\mathbf{A}$ we get

$$\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{A}\mathbf{v}_i$$

and so $\mathbf{A}\mathbf{v}_i$ is an eigenvector of $\mathbf{A}\mathbf{A}^T$ with eigenvalue σ_i^2. So $\mathbf{u}_i = \mathbf{A}\mathbf{v}_i/\sigma_i$ is a unit eigenvector of $\mathbf{A}\mathbf{A}^T$ and we have

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i$$

This allows us to write

$$\mathbf{A}\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_r \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_r \end{bmatrix}\begin{bmatrix} \sigma_1 & & \\ & \cdot & \\ & & \sigma_r \end{bmatrix}$$

Since the columns of $\mathbf{V}$ are orthonormal we can right multiply by $\mathbf{V}^T$ to get

$$\mathbf{A} = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_r \end{bmatrix}\begin{bmatrix} \sigma_1 & & \\ & \cdot & \\ & & \sigma_r \end{bmatrix}\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_r \end{bmatrix}^T$$

or simply

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

which is just the SVD of $\mathbf{A}$.

Application of EVD and SVD

Fibonnaci

Here's an application where $\mathbf{A}^n = \mathbf{Q}\mathbf{\Lambda}^n\mathbf{Q}^{-1}$ is very useful. You are probably familiar with the Fibonacci sequence

$$1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad ... \tag{15}$$

where, beginning with the third number of the sequence, each number is a sum of the previous two numbers. This can be described with the recursion relationship

$$u_n = u_{n-1} + u_{n-2} \tag{16}$$

Is there a way to find a number in the sequence without adding up all the previous numbers? There is. Let's start by defining

$$\mathbf{U_k} = \begin{bmatrix} u_{k+1} \\ u_k \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{17}$$

where $0 \leq k \leq n-1$ so that

$$\mathbf{U_0} = \begin{bmatrix} u_1 \\ u_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{U_1} = \begin{bmatrix} u_2 \\ u_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad ...$$

$$\mathbf{U_{n-2}} = \begin{bmatrix} u_{n-1} \\ u_{n-2} \end{bmatrix}, \quad \text{and} \quad \mathbf{U_{n-1}} = \begin{bmatrix} u_n \\ u_{n-1} \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{U}_1 &= \mathbf{A}\mathbf{U}_0 \\ \mathbf{U}_2 &= \mathbf{A}\mathbf{U}_1 = \mathbf{A}(\mathbf{A}\mathbf{U}_0) = \mathbf{A}^2\mathbf{U}_0 \\ \mathbf{U}_3 &= \mathbf{A}\mathbf{U}_2 = \mathbf{A}(\mathbf{A}^2\mathbf{U}_0) = \mathbf{A}^3\mathbf{U}_0 \\ &\vdots \\ \mathbf{U}_{n-1} &= \mathbf{A}^{n-1}\mathbf{U}_0 \end{aligned}$$

Therefore, to find any number in the sequence, u_n, you just need to evaluate A^{n-1} and evaluate the first element of $\mathbf{U}_{n-1}$. That may take some significant computational effort. An easier way is to find an explicit formula for u_n in terms of n by making use of $\mathbf{A}^n = \mathbf{Q}\mathbf{\Lambda}^n\mathbf{Q}^{-1}$.

Find the eigendecomposition of $\mathbf{A}$ and use

$$\mathbf{U}_{n-1} = \mathbf{A}^{n-1}\mathbf{U}_0 = \mathbf{Q}\mathbf{\Lambda}^{n-1}\mathbf{Q}^{-1}\mathbf{U}_0 \tag{18}$$

to get an equation for u_n directly in terms of n. What is the 40th term of the Fibonacci sequence?

Data Compression via Principle Component Analysis

In this example you will perform a simple data compression exercise. You will use temperature data from 3 cities over 10 years, as training data and use it to compress a years worth of temperature data from 3 cities into a 2×365 matrix. In other words, you will represent 3×365 numbers (daily temperature data from 3 cities over 1 year), using 2×365 values. This compression is lossy, in that you will loose some information. However, by representing the data along the two most significant eigenvectors of the covariance matrix, you can reduce this data loss. Please note that while we have laid out the steps you need to take here quite explicitly, it is important for you to fully understand what each step does. You will be using very similar steps in your project.

Exercises

3. Load the file `avg_temperatures_pt2.mat`. You will have 8 data vectors in your workspace. `b_tr, w_tr, s_tr, p_tr` which represent 10 years of training data for the average daily temperatures in Boston, Washington DC, Sao Paolo, and Providence respectively. The vectors `b_new, w_new, s_new, p_new` represent an additional year of data for the three cities – this is the data that you will compress using statistical knowledge of the previous 10 years of data. Create a covariance matrix $\mathbf{R}$ using the 10 years worth of temperature data from Boston, Washington DC and Sao Paolo (in that order).

4. Perform an eigendecomposition of the matrix $\mathbf{R}$, and make a new matrix $\mathbf{V}_p$ which has the 2 eigenvectors corresponding to the 2 largest eigenvalues of $\mathbf{R}$. Let these eigenvectors be $\mathbf{v}_1$ and $\mathbf{v}_2$.

5. Create centered (i.e. subtract the mean), versions of the new temperature data vectors, and create a 3×365 matrix $\mathbf{T}$ which has the centered temperatures of Boston, Washington DC and Sao Paolo as its rows (in that order). This matrix is a representation of the data you are now going to compress. Let the i-th column of $\mathbf{T}$ be $\mathbf{t}_i$.

6. Take the dot product of each column of the matrix $\mathbf{T}$ (which is a vector of the temperature of Boston, Washington DC and Sao Paolo for a given day) with the two eigenvectors in matrix $\mathbf{V}_p$, and save the values. Let this quantities be called α_{1i} and α_{2i}. In other words,

$$\alpha_{1i} = \mathbf{v}_1^T \mathbf{t}_i$$
$$\alpha_{2i} = \mathbf{v}_2^T \mathbf{t}_i$$

You can do this either using a for loop, or using matrix multiplications.

You should now have 365 different values for α_{1i} and α_{2i}, which are a compressed representation of 3×365 different temperature values. Moreover, these values are the components of the temperature data that lie in the directions of the two eigenvectors of the covariance matrix corresponding to the largest eigenvalues. From what we saw in the previous two classes, these vectors represent the two orthogonal directions in the data that have the most amount of variation, and hence the most "important" directions. Of course, there is a third direction (since the temperature vectors live in a 3-dimensional space), which we are discarding. But since this is the direction in which there is the least amount of variation in the data set, we do not lose too much information.

7. You can now check how well your compression worked, by using the values of α_{1i} and α_{2i} to reconstruct 365 different 3×1 vectors each representing the temperatures for the three cities over the 365 days. Let $\hat{\mathbf{t}}_i$ represent the reconstructed temperature vector on the i-th day. Using what you know about projections onto orthonormal vectors, reconstruct $\mathbf{t}_i$ using α_{1i}, α_{2i}, $\mathbf{v}_1$ and $\mathbf{v}_2$. Repeat this for all 365 days.

8. On the same axes, plot the original and reconstructed temperature for Boston. Repeat this for Washington DC and Sao Paolo. Observe how close the reconstructions are, for the different data sets.

9. How accurately do you think you can represent the data if you used 3 eigenvectors instead of 2?

10. If you feel inspired, repeat the above with temperature data for four different cities, and 2 or 3 different eigenvectors.

While this example can be thought of as a "toy" example where we are representing 3 dimensional data using 2 dimensions, there are many applications for which there may be many more dimensions in the data for which accurate representations can be made using only a few dimensions. Additionally, you should note that such dimensionality reduction techniques are not just useful in compression, but they are also useful in speeding up computation. We can often get away with analyzing data over a small number of important dimensions, and this is an important technique when we deal with large amounts of data. Overall, these class of techniques is called Principle Component Analysis (PCA), since we are performing analysis along a few principle component directions of the data.