Computing Project 2

In this project, we're going to study how patient cohorts can be analysed based on measurements of the activity of many genes. This project is worth 30% of your module mark. You will have until midnight on the night of the 17th December to complete this project. This project requires you to write two scripts. Once you have completed this project, place these scripts into a single .zip archive and upload this archive to the dropbox folder in Week 12 of Blackboard Learn.

When we take multiple measurements from a patient, we can compile the measurements from one patient into a vector, with each component of the vector describing a different measured quantity. If we were to take 15 measurements from each patient then each patient's data could be described in a 15 component vector and if we studied 100 patients then their data would make up 100 data points in a 15 dimensional space.

Such data is considered *high-dimensional* and analyzing high-dimensional data is not easy. In this project, we are going to introduce one approach for analyzing high dimensional data. The approach is called Principal Component Analysis (usually abbreviated to PCA) and involves trying to find the best set of axes (and basis vectors) in the high dimensional space with which to study the dataset.

In practice, it's very difficult to work with large numbers of dimensions. So PCA is used to find a small number of axes in the high dimensional space that describe most of the trends in the data. So for example, we might identify the 2 axes in our 15 dimensional space that account for most of the variation in the data and then choose to work with the data in this 2D subspace. In doing so, we throw away 13 dimensions worth of information, but as long as we choose our 2 dimensions to be those that are most important in the 15D space, then there is a good chance that the data thrown away won't significantly affect any conclusions we draw.

For further details of Principal Component Analysis, see

- http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- https://www.youtube.com/playlist?list=PLBC24FD8C389FE9E4
- http://www.dsea.unipi.it/Members/balestrinow/CP/file/TD_16_6_davies_pca.pdf
- https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/

In this project, we will use PCA twice. In part A, we will work through a small data set in order to see how PCA works. In part B, we will use a larger data set in order to see how PCA is used in practice.

PCA uses a lot of the ideas that we have covered in lectures regarding axes and bases. We will therefore need to draw upon this knowledge as part of this project.

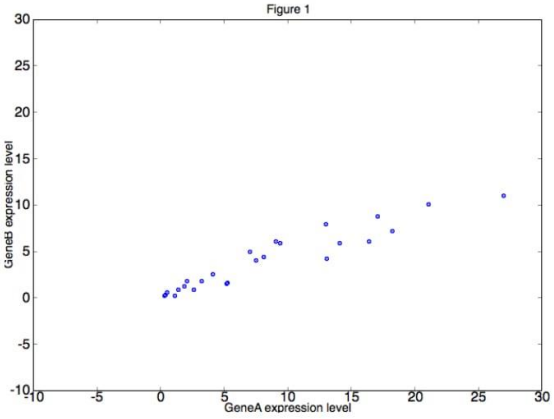
Part A

All of part A should be contained inside one script.

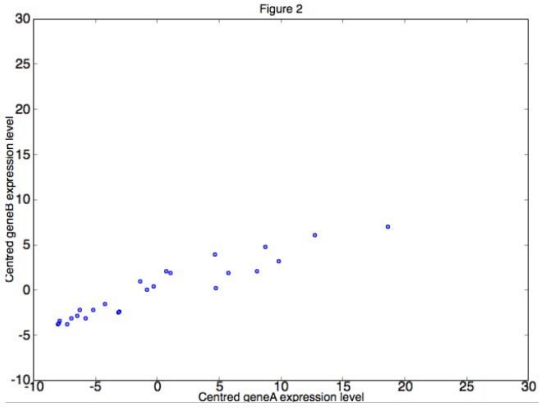
1. We have taken measurements of the activity of 2 genes for 25 patients. In order to store this data, you must create a 2x25 component matrix. Each row of this matrix should correspond to the measurements of activity for a single gene. Each column of this matrix will correspond to a different patient. Populate this matrix with the data for GeneA in the first row and the data for GeneB in the second row

GeneA = [1.1 0.5 27 1.4 1.9 2.6 3.2 8.1 4.1 0.4 9.4 13 14.1 9.1 18.2 2.1 5.3 16.4 7 21.1 0.3 7.5 17.1 5.2 13.1];
GeneB = [0.2 0.6 11 0.9 1.2 0.9 1.8 4.4 2.5 0.3 5.9 7.9 5.9 6.1 7.2 1.8 1.6 6.1 5 10.1 0.2 4.0 8.8 1.5 4.2];

(5 marks)



2. Plot this data [hint: use the scatter command] so that you can compare the activity levels of the two genes across the group of patients. Add the correct titles to the axes and set the range to be -10 to 30 on the x-axis and -10 to 30 on the y-axis. This should yield Figure 1 [hint you'll need the xlabel, ylabel and axis commands]. (5 marks)



3. We need to centre this dataset on the origin. Calculate the mean of each row of gene measurements and create a new 2x25 component matrix. Copy across the data from the first matrix, but subtract the mean gene measurement from each row. Plot this and set the range from -10 to 30 on each axis [hint: again use the scatter command]. This should yield Figure 2. (10 marks)

4. In the next step, we want to identify the axis on which the data shows the greatest variation. To do this, we create an axis, calculate the

coordinate of each data point on this axis and then calculate the variance of these coordinates. We then change the orientation of the axis and repeat this process, calculating a new set of coordinates on the new axis and evaluating their variance before changing the orientation again. If we repeat this process and cover a wide range of axis orientations, we will be able to identify the axis on which the data shows the greatest variance and therefore the single axis on which we have the most information about the data.

To implement this approach we start by considering a new axis that lies perfectly on top of the x-axis. To obtain the coordinate of each data point on this axis, we must create an orthonormal basis vector. To do this, we create a 2 component vector with a x-coordinate of 1 and a y-component of 0. Then, to obtain the coordinates of each data point on this axis, we do a dot product between this basis vector and each of data points in turn. This will give us 25 numbers corresponding to 25 coordinates on this axis, one for each data point. We then calculate the variance in these coordinates and store the value for the variance in a variable. This is the variance that we associate with this orientation of axis. Next, we consider a new axis that lies at an angle 0.01 radians anti clockwise from the x-axis (a radian is a unit of angle like a degree, but more mathematically meaningful – most mathematical functions such as trigonometry functions use radians by default). For this axis, we need to calculate a new orthonormal basis vector. To do so, create a new 2 component vector and again set the x-component to be equal to 1, but now set the y component to be equal to tan(0.02). This gives us a basis vector that lies in the correct direction, but at this stage is not yet orthornormal. To make it orthonormal, we must first normalize it. If $(1, \tan(0.02))^T$ is the basis vector then the orthonormal version will be $(1/\text{sqrt}(1^2+(\tan(0.02))^2))(1, \tan(0.02))^T$. Using this orthonormal basis vector, we then take a dot product with each of the data points in turn, giving us a new set of 25 coordinates. We then calculate the variance in these coordinates to give a new variance value associated with this axis orientation that we store in a variable. Next, we create a third axis, this time at an angle of 0.04 radians to the x-axis and again we create a new orthonormal basis vector. To do so, we create a 2 component vector, setting the first component to 1 and the second component to tan(0.04). We normalize this to create an orthonormal basis vector $(1/\text{sqrt}(1^2+(\tan(0.04))^2))(1, \tan(0.04))^T$ and calculate the coordinates of the data points on this new axis by taking the dot product of each data point in turn with this new orthonormal basis vector. This gives 25 new coordinate values and we can calculate a value for the variance that we store in a new variable associated with this orientation of axis. We can then proceed to consider a new axis at 0.06 radians from the x-axis, calculating an orthonormal basis vector, coordinates and a variance and then repeat for a new orientation of axis.

In radians, 90 degrees corresponds to pi/2, so we repeat this process until we have calculated the variance on all angles of axis from 0 to pi/2

radians in steps of 0.02. To keep track of the variance value calculated for each axis orientation, we should store the value in a component of an array.

This is the approach we will take in the following exercises.

a. Set up a for-loop in which the loop variable runs from 0 to pi/2 in steps of 0.02. This for-loop and the loop variable (which we have usually called *i* or *j* in class) is going to parameterize the steepness of the axis.

(5 marks)

b. Inside this for-loop, create a two component vector in which the first component is 1 and the second component is tan(LoopVariable). Normalise this vector, ie if the vector is (a, b)^T then create a second vector (1/sqrt(a^2+b^2))(a, b)^T. This is the normalized basis vector for the axis.

(5 marks)

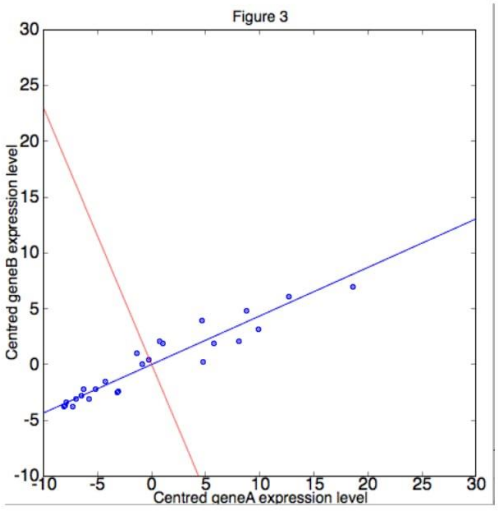
c. Also inside this loop, using the normalized basis vector, calculate the coordinates of all 25 data points on this axis, ie take the dot produce of all 25 data points with the basis vector.

(5 marks)

d. Also inside the loop, calculate the variance in the coordinates you obtain on this axis. The variance for a single set of measurements, $\underline{x}$, is defined to be $var(x) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N-1}$ where $\bar{x}$ is the mean of the set of measurements. (If you don't know what the variance is http://www.wikihow.com/Calculate-Variance)

(5 marks)

e. Also inside the loop, assign the value for the variance you've calculated to a component of an array. You'll want to put a value for the variance in a different element each time you go through the loop so that you can keep track of the variance calculated for each orientation of the axis. (5 marks)



f. Once the loop has finished, you should have an array that contains the variance in the data for each angle of the axis. Find the component of the

array with the greatest variance and calculate the gradient of the axis to which this corresponds. (5 marks)

5. On your version of Figure 2, plot the axis that has the greatest variance. Also plot an axis that is perpendicular to this. This should yield Figure 3. (10 marks)

   An example with only two sets of measurements would usually be too simple to warrant use of PCA. However, it's useful as an exercise. We can see that we have identified an axis that maximally describes the variation in the data (in blue) and a second axis that describes the minimal variation in the data (red). The blue axis would be considered the first principle component of the data set. By identifying the axis on which there is the most variation and (by implication) the axis on which there is the least variation, we could reduce the dimension of the data by keeping the coordinates along the blue axis and ditching the coordinates along the red axis, on the justification that we had selected an axis that maximized the information retained about the systematic variation in the data. Described in this way, the 1D dataset we arrive at would only be an approximation to the original 2D data set, but by choosing axes in this way, we would ensure that it was the best approximation possible.

Part B

All of part B should be contained inside one script.

We will now consider 25 patients where we have measured the expression levels of 5 genes.

1. Construct a 5x25 component matrix where each row corresponds to a different gene and each patient's measurements corresponds to a different column. Populate this matrix with the following data:-

GeneA = [1.1 0.5 27 1.4 1.9 2.6 3.2 8.1 4.1 0.4 9.4 13 14.1 9.1 18.2 2.1 5.3 16.4 7 21.1 0.3 7.5 17.1 5.2 13.1];
GeneB = [11.4 2.7 3.2 4.4 1.4 6.8 2.2 23.6 21.9 10.2 14.7 9.1 3.3 6.1 10.2 4.8 6.6 2.8 8 5.7 0.2 8.0 8.8 11.5 4.2];
GeneC = [0.7 0.1 13.4 0.9 1.4 1.1 1.5 3.1 1.4 0.4 3.7 8.5 9.1 2.9 7.8 0.9 3.6 6.4 3.1 9.4 0.01 3.2 10.1 3.2 5.4];
GeneD = [21.5 14.3 1.1 9 13.6 12.3 12.8 11.1 2.6 2.1 2.9 4.6 7.1 2.4 3.6 11.7 18.3 25.3 15.2 2.6 6.2 6.1 3.3 15.0 7.1];
GeneE = [0.7 0.1 12.7 0.7 1.1 1.1 1.2 6.2 3.0 0.2 4.1 7.3 7.2 5.1 9.0 0.8 2.2 6.9 3.1 10.9 0.2 5.4 9.1 3.0 6.0];

(5 marks)

2. Our first step is to do the higher dimensional equivalent of centering our data on the origin. Calculate the mean measurement for each gene and create a new 5x25 component matrix containing the measurements from the matrix above after subtracting the mean measurement for each gene from each row.

(5 marks)

3. Instead of calculating the axis with the greatest variance manually, there is a mathematical cheat that we can use that makes this process much easier. However, first we much calculate an object called the covariance matrix. The variance for a single set of measurements, $\underline{x}$, is defined to be $var(x) = \sum_{i=1}^{N} \frac{(X_i - \bar{x})^2}{N-1}$ where $\bar{x}$ is the mean of the set of measurements. The covariance between two sets of measurements, $\underline{x}$ and y, is defined to be $cov(x, y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N-1}$ where $\bar{x}$ and $\bar{y}$ are the mean measurement from $\underline{x}$ and y, respectively.
   Create a 5x5 matrix and calculate values for it where the (i, j) component contains the covariance of the set of gene measurements from row i of the 5x25 centred matrix with the set of gene measurements from row j of the 5x5 centred matrix. To achieve this, you will want to set up three nested for-loops; the first over the columns of the covariant matrix, the second over the rows of the covariant matrix and the third over the 25 measurements of gene activity involved in one covariance calculation.

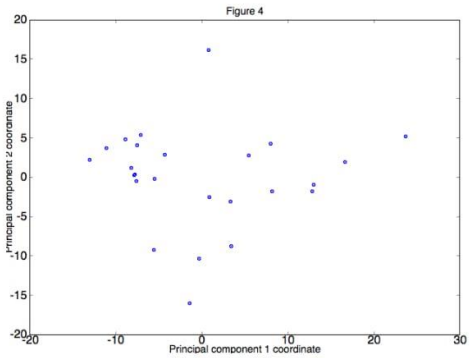
(10 marks)

4. The second part of the mathematical cheat is to calculate the eigenvalues and eigenvectors belonging to the covariant matrix. Eigenvectors and eigenvalues are a powerful concept in mathematics, but would require several dedicated lectures to explain fully. In this exercise, you only need to know how to interpret them. An eigenvalue is just a scalar (ie a number) and an eigenvector is just a vector. The prefix *eigen* comes from the German word for *characteristic*, meaning that they are characteristic numbers and vectors of a matrix. We can take advantage of the built in function in Matlab to calculate them. If *cov* is the name of your covariant matrix then the command you need is

```
[V, D] = eig(cov);
```

The output of this function is two matrices, *V* and *D*. Each column of *V* is a basis vector for an axis. These basis vectors are *orthonormal* and because we have calculated them from the covariant matrix, they describe the axes with the greatest variance in the data. As such, each of these basis vectors is a principal component of the dataset. D is a diagonal matrix containing numbers (eigenvalues) on the lead diagonal and zero everywhere else. Each number (eigenvalue) in D corresponds to a basis vector (eigenvector) in V and the position of each number in D indicates which basis vector it corresponds to, so the number in the first column of D belongs to the vector in the first column of V, the number in the second column of D belongs to the vector in the second column of V and so on. Each number (eigenvalue) tells you the variance along an axis belonging to the corresponding basis vector and the basis vectors with the largest numbers (eigenvalues) are those belonging to axes with the largest variance.
Identify the two basis vectors belonging to axes with the greatest variance. What are their corresponding variances (ie their eigenvalues)? Indicate your answers with a comment in the code. (10 marks)

5. Calculate the coordinates of all of our 5 component data points along the axes belonging to these two basis vectors. You can do this by taking the dot product of each data point with the basis vector of each axis. Plot these coordinates in a new figure and set the range of the axes and label the axes as shown in Figure 4. [Hint: use the scatter function again with the xlabel, ylabel and axis commands.]



(10 marks)

This represents the optimal projection of the data from 5 dimensions into 2 dimensions in such a way that it retains as of many of the trends in the data as possible. In a real world example, you might have hundreds or thousands of genes measured for each patient. However, a PCA of this data would proceed in the same way, enabling you to reduce a thousand dimensional space to the two or three most important dimensions for further study.