PCA

March 20, 2019

1 Principal Component Analysis

1.0.1 Names: Jarred Parr Edwin Purcell

1.0.2 Due: Wednesday, March 20

We are going to look at a classic data set consisting of physical measurements of 150 irises. There are three species of irises in this set—setosa, versicolor, and virginica—and there are 50 samples of each species. Each sample has four measurements, all of which are in centimeters:

- sepal length
- sepal width
- petal length
- petal width

In the image below, the sepals are labeled as *falls* and the petals as *standards*.

The basic problem is to use the four physical measurements to predict which species a given sample belongs to. This is a standard data set that is used for testing machine learning techniques.

Since we have 150 samples, each of which has 4 measurements, we are looking at 150 data points in $\mathbf{R}^4$. That makes it difficult to visualize. Of course, we could look at scatter plots formed by considering just two measurements at a time, but we'd like to find the best two-dimensional picture of the data. Principal component analysis is the right tool for doing that.

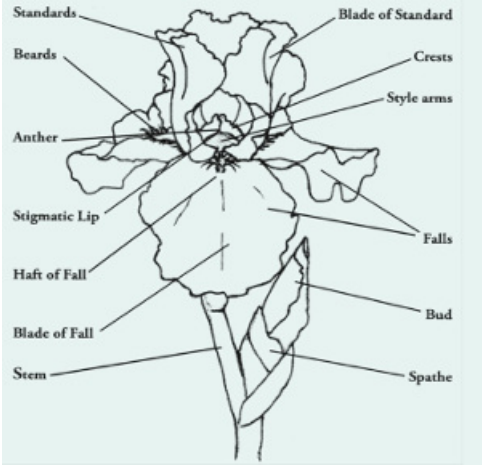
Evaluate the cell below to read in the data set. In addition, you will have two familiar functions `findmean(data)`, which returns the mean of the data, and `demean(data)`, which returns the 4×150 de-meaned data matrix. Remember that you have two other useful functions: `B.matrix_from_columns( list )` and `B.matrix_from_rows( list )`.

```
In [2]: import csv
        import numpy as np

        def findmean(data):
            return vector(np.sum(data, axis=0))/len(data)

        def demean(data):
            mean = findmean(data)
            return matrix(RDF, [datum - mean for datum in data]).T

        def plot2d(M):
            colors = ['red', 'green', 'blue']
```




```
    p = list_plot([])
    for i in range(3):
        p += list_plot(M.columns()[50*i: 50*(i+1)], color = colors[i], aspect_ratio=1,
    return p
```

```
input = csv.reader(open('iris.data'))
data = map(vector, [map(float, datum[:4]) for datum in input])
```

What is the average petal length in centimeters?

```
In [3]: findmean(data)[2]
```

```
Out[3]: 3.7586666666666697
```

Average petal length in centimeters: 3.758

```
In [0]:
```

Construct the covariance matrix C and display it below.

```
In [4]: A = demean(data)
        C = 1/len(data) * A * A.T
```

Find the eigenvalues of the covariance matrix.

```
In [5]: eigenvals = C.eigenvalues()
        eigenvals
```

```
Out[5]: [4.196675163197983,
         0.24062861448333234,
         0.07800041537352656,
         0.023525140278495168]
```

For what percentage of the total variance do the first two eigenvalues account?

```
In [6]: first_two = eigenvals[:2]
        sum(first_two)/sum(C.eigenvalues()) * 100
```

```
Out[6]: 97.76317750248033
```

Find matrices D and Q that orthogonally diagonalize C.

```
In [7]: D, Q = C.eigenmatrix_right()
        D, Q
```

```
Out[7]: (
        [   4.196675163197985                  0.0                   0.0                   0.0]
        [                 0.0  0.24062861448333256                   0.0                   0.0]
        [                 0.0                  0.0  0.07800041537352653                   0.0]
        [                 0.0                  0.0                   0.0 0.023525140278495192],
        )
```

Verify that the columns of Q are orthonormal.

```
In [8]: n(Q.T * Q)
```

```
Out[8]: [     1.00000000000000  7.15009949432602e-16 -2.07146693363023e-16 -1.62695571888723e-1
        [ 7.15009949432602e-16      1.00000000000000  2.81042386322640e-15 -2.88279709793641e-1
        [-2.07146693363023e-16  2.81042386322640e-15      1.00000000000000  1.76204237137238e-1
        [-1.62695571888723e-18 -2.88279709793641e-15  1.76204237137238e-15      1.000000000000
```

Suppose that we would like to create a two-dimensional plot of the de-meaned data set by projecting the data onto the two-dimensional subspace V formed by eigenvectors of C corresponding to the two largest eigenvalues. That is, if $\mathbf{u}_1$ and $\mathbf{u}_2$ are the eigenvectors, we would like to represent a de-meaned data point $\mathbf{x}$ as (c_1, c_2) where the projection of $\mathbf{x}$ onto this subspace is $c_1\mathbf{u}_1 + c_2\mathbf{u}_2$. Find the matrix P such that $P\mathbf{x} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$.

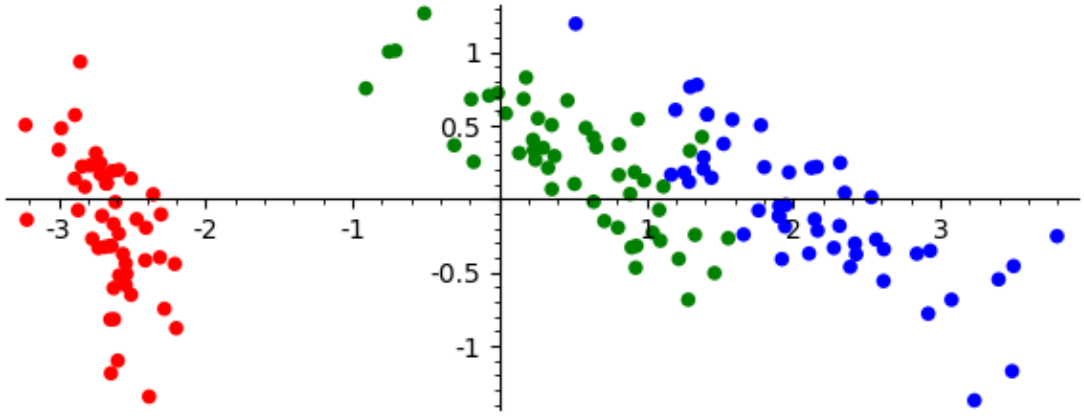
```
In [9]: u1 = Q[:, 0]
        u2 = Q[:, 1]
        P = Q.matrix_from_columns([0, 1])
        P
```

```
Out[9]: [ 0.36158967738144937  -0.6565398832858327]
        [-0.08226888989221476  -0.7297123713264952]
        [  0.8565721052905279   0.1757674034286548]
        [ 0.35884392624821554   0.0747064701350339]
```

The product PA will give a matrix whose columns consist of the de-meaned data points projected onto the plane. Construct this product and use the `plot2d` function to display these projected points. The red points are samples from the setosa species, green are versicolor, and blue are virginica.

```
In [10]: plot2d(P.T * A)
```

Out[10]:



Explain why this plot is wider than it is tall: Because, there are many more rows with widely distributed points. When we do the projection onto the 2d plane, the points become spread along that plane since their dimensionality was so heavily reduced.

Suppose that you discover a new sample but that you only know two measurements: the sepal length is 5.65 cm and the sepal width is 2.75 cm. In this case, you don't know some of the data for this sample. However, let's make the reasonable assumption that the demeaned data point lies in the two-dimensional subspace V. Find the coordinates (c_1, c_2) of this sample. You will probably need to think about this task for a little bit to determine a linear system for the coordinates.

```
In [23]: newpoints = vector([5.65, 2.75, 0, 0])
         dm = newpoints - findmean(data)

         plane = P.matrix_from_rows([0, 1])
         knowndm = vector([dm[0], dm[1]])
         # x = s * vector
         x = P*(plane.inverse() * knowndm)
         x + findmean(data)
```

```
Out[23]: (5.65, 2.75, 3.9859139852715457, 1.2942918609794793)
```

Estimate the other two measurements, the petal length and the petal width: petal length: 3.99 petal width: 1.29

To which species does this sample most likely belong: Versicolor

Suppose you find another sample whose petal length is 1.5 cm and whose petal width is 0.25 cm. Estimate the other two measurements.

```
In [24]: newpoints = vector([0, 0, 1.5, 0.25])
         dm = newpoints - findmean(data)
```

```
plane = P.matrix_from_rows([2, 3])
knowndm = vector([dm[2], dm[3]])
# x = s * vector
x = P*(plane.inverse() * knowndm)
x + findmean(data)
```

```
Out[24]: (6.554842749006653, 4.895105347229306, 1.500000000000019, 0.2500000000000082)
```

Sepal Length: 6.44 Sepal Width: 4.90
To which species does this sample most likely belong: Virginica