
IV.6 Principal Coordinates Analysis (PCA)

IV.6.1 Introduction

Imagine we are interested in comparing a selection of n different wines from different vineyards. A connoisseur is given samples from pairs of wines and compares them, giving a (somewhat subjective) comparison rating for how similar or dissimilar they are: 0 if the wines cannot be told apart up to 10 if they are totally different. This comparison rating is referred to as the *dissimilarity* between the two wines. At the end of the day the connoisseur has compared all the different wines and assigned ratings to describe the similarity and dissimilarity of each pair. This data is combined into a *dissimilarity matrix* T where T_{ij} is the rating given to the comparison between wines i and j .

Even for a moderate selection, trying to make sense out of this data can be challenging. We want to group the wines into similar-tasting families but, unless there is a very conspicuous characteristic that is different (such as half of the selection being red and the other half white), it may be difficult to identify significant patterns or relationships just by staring at the data.

The idea of Principal Coordinates Analysis (PCA) is firstly to represent the different objects under consideration (in this case the wines) graphically, as points $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ in \mathbb{R}^p for some suitable choice of dimension p . The distance $\|\mathbf{v}_i - \mathbf{v}_j\|$ between points in the plot is chosen to reflect as closely as possible the entry T_{ij} in the dissimilarity matrix T . Our eye is much more capable of spotting relationships and patterns in such a plot than it is spotting them in the raw data. However another problem arises in doing this: we can only visualize two or possibly three dimensions ($p = 2$ or 3) whereas the data would most naturally be graphed in a much higher dimensional, possibly even $n - 1$ -dimensional, space. The second key idea of PCA is to reduce the number of dimensions being considered to those in which the variation is greatest.

IV.6.2 Definitions and useful properties

A real square matrix A is called *positive definite* if

$$\langle \mathbf{x}, A\mathbf{x} \rangle > 0$$

for any $\mathbf{x} \neq \mathbf{0}$.

A real square matrix A is called *positive semi-definite* if

$$\langle \mathbf{x}, A\mathbf{x} \rangle \geq 0$$

for any $\mathbf{x} \neq \mathbf{0}$.

A real matrix of the form $A^T A$ for any real matrix A is always positive semi-definite because

$$\langle \mathbf{x}, A^T A \mathbf{x} \rangle = \langle A \mathbf{x}, A \mathbf{x} \rangle = \|A \mathbf{x}\|^2 \geq 0.$$

A positive semi-definite matrix has all non-negative eigenvalues. The proof is as follows. Let \mathbf{v} be an eigenvector of A with eigenvalue λ so

$$A \mathbf{v} = \lambda \mathbf{v}.$$

Because A is positive semi-definite we have

$$\langle \mathbf{v}, A \mathbf{v} \rangle \geq 0.$$

Using the fact that \mathbf{v} is an eigenvector we also have

$$\langle \mathbf{v}, A \mathbf{v} \rangle = \langle \mathbf{v}, \lambda \mathbf{v} \rangle = \lambda \langle \mathbf{v}, \mathbf{v} \rangle.$$

Combining these two results and remembering that $\langle \mathbf{v}, \mathbf{v} \rangle > 0$ for an eigenvector, we have our result:

$$\lambda \geq 0.$$

IV.6.3 Reconstructing points in \mathbb{R}^p

We begin with a simple example, taking our objects to be a set of n points $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n$ in \mathbb{R}^p . We take the dissimilarity between objects i and j to be the usual distance between those points

$$T_{ij} = \|\mathbf{w}_i - \mathbf{w}_j\|.$$

Is it possible to reconstruct the relative location of the points from the data in T alone?

It is important to notice that any reconstruction that we find is not unique. We are free to rotate, reflect and translate the points and they still satisfy the only requirement that we make of them, namely that the distances between the points are as specified.

Reconstructing two points in \mathbb{R}^p

Consider two points, \mathbf{w}_1 and \mathbf{w}_2 , that are a known distance, say 2, apart. A single line can always be placed between any two points and therefore we expect that the points can be represented in one dimension. If we only know the distance between the points, then a possible representation of them is $\mathbf{v}_1 = [0]$ and $\mathbf{v}_2 = [2]$.

Reconstructing three points in \mathbb{R}^p

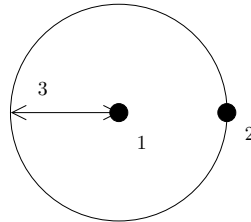
Now consider three points, \mathbf{w}_1 , \mathbf{w}_2 and \mathbf{w}_3 with dissimilarity matrix:

$$T = \begin{bmatrix} 0 & 3 & 5 \\ 3 & 0 & 4 \\ 5 & 4 & 0 \end{bmatrix}.$$

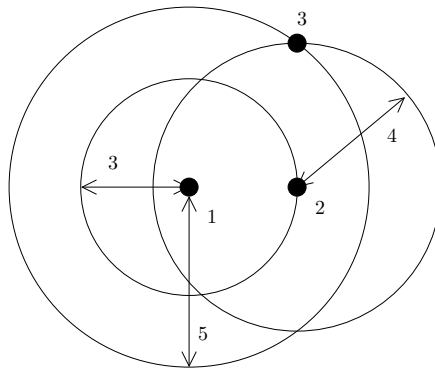
A plane can always be placed through any three points and therefore we expect that the points can be represented in two dimensions. We can find such a representation using trilateration. First we choose a point to represent \mathbf{w}_1 :



Next we draw a circle of radius 3 centred at our first point and choose a second point on the circle to represent \mathbf{w}_2 :



Finally we draw a circle of radius 5 centred at our first point and a circle of radius 4 centred at our second point and choose our third point to be at one of the intersections of these two circles to represent \mathbf{w}_3 :



Thus we have been able to create a representation of the three points: $\mathbf{v}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$
and $\mathbf{v}_3 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Arbitrary number of points in \mathbb{R}^p

The approach above works for a low dimensional space and with few points, but how can we handle many dimensions and/or many points?

Let $\mathbf{v}_1, \dots, \mathbf{v}_n$ be the representation of the points that we are trying to find. We assume that these points are in \mathbb{R}^n (from the discussion above we can infer that the points can be chosen to be in an $n - 1$ dimensional space, but choosing n makes the analysis below a little cleaner).

Let V be the matrix whose rows are the vectors for the points we are trying to find:

$$V = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix}$$

We assume that the points are chosen such that the sum of each column of V is 0, in other words the centroid of the set of points is placed at the origin of \mathbb{R}^n . We are allowed to do this because of the translational freedom that we have in finding a solution.

All we know about the points is the distance that they are required to be apart. We have

$$\begin{aligned} (T_{ij})^2 &= \|\mathbf{v}_i - \mathbf{v}_j\|^2 = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j \rangle \\ &= \langle \mathbf{v}_i, \mathbf{v}_i \rangle + \langle \mathbf{v}_j, \mathbf{v}_j \rangle - 2\langle \mathbf{v}_i, \mathbf{v}_j \rangle. \end{aligned} \tag{IV.1}$$

It is easier to work with this equation in matrix form so we make the following definitions. Let B be the matrix whose entries are the inner products between the points

$$B_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \mathbf{v}_i^T \mathbf{v}_j,$$

or equivalently

$$B = VV^T.$$

Let Δ be the matrix containing the squares of the dissimilarities with entries

$$\Delta_{ij} = (T_{ij})^2.$$

Finally, let Q be the matrix with entries

$$Q_{ij} = \|\mathbf{v}_i\|^2$$

or equivalently

$$Q = \mathbf{q}\mathbf{e}^T,$$

where $\mathbf{q} = \begin{bmatrix} \|v_1\|^2 \\ \|v_2\|^2 \\ \vdots \\ \|v_n\|^2 \end{bmatrix}$ and $\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$.

We can now rewrite (IV.1) as the matrix equation

$$\Delta = Q + Q^T - 2VV^T. \quad (\text{IV.2})$$

We want to find the matrix V , and know the matrix Δ . So the next step is to eliminate Q from this equation. Notice that $\mathbf{q}\mathbf{e}^T = n\mathbf{q}$. This suggests that post-multiplying equation (IV.2) by

$$H = I - \frac{1}{n}\mathbf{e}\mathbf{e}^T$$

will eliminate Q :

$$QH = \mathbf{q}\mathbf{e}^T (I - (1/n)\mathbf{e}\mathbf{e}^T) = \mathbf{q}\mathbf{e}^T - \mathbf{q}\mathbf{e}^T = 0,$$

where 0 here represents the zero matrix. We also have that H is symmetric ($H = H^T$) so $HQ^T = H^T Q^T = (QH)^T = 0$. Therefore pre-multiplying by H eliminates Q^T . Applying these operations to (IV.2) we obtain the matrix equation

$$H\Delta H = -2HVV^T H = -2HVV^T H^T = -2HV(HV)^T.$$

We can simplify this by finding HV :

$$HV = \left(I - \frac{1}{n}\mathbf{e}\mathbf{e}^T \right) V = V - \frac{1}{n}\mathbf{e}\mathbf{e}^T V$$

but each entry of the row vector $\mathbf{e}^T V$ is the sum of each column of V which we assumed was zero. Therefore $\mathbf{e}^T V = \mathbf{0}^T$ and

$$HV = V.$$

We have now obtained the equation

$$VV^T = -\frac{1}{2}H\Delta H,$$

which is an equation that we are able to solve to find V . The matrix on the left is symmetric and positive semi-definite (using the properties we saw in §IV.6.2). The matrix on the right is symmetric (the dissimilarity matrix should be symmetric). In order for a solution to exist, it must also be positive semi-definite. In the example we are considering, we *know* that a solution of the problem exists because the dissimilarity matrix was constructed from a set of points in \mathbb{R}^n . Therefore in this case the matrix on the right *must* be positive semi-definite.

We can find a solution as follows. Because $-(1/2)H\Delta H$ is a real symmetric matrix it can be diagonalized as

$$-\frac{1}{2}H\Delta H = SDS^T,$$

where S is the matrix of normalized eigenvectors and D is the diagonal matrix of eigenvalues. Because it is positive semi-definite all eigenvalues are non-negative, and we can take the square root $D^{1/2}$ of D , where the entries on the leading diagonal are the square roots of the eigenvalues. We can now write

$$VV^T = (SD^{1/2})(SD^{1/2})^T. \quad (\text{IV.3})$$

and a solution for V is

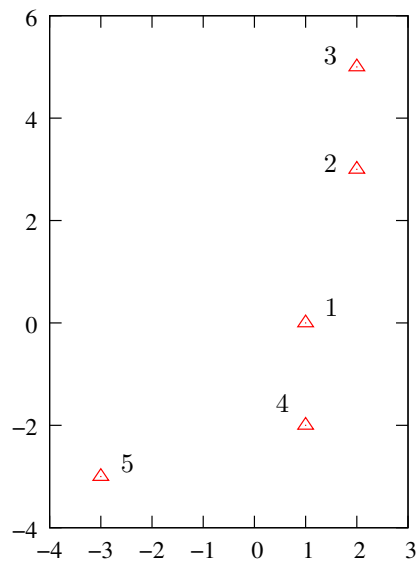
$$V = SD^{1/2}$$

Cautionary note: all we have actually shown is (IV.3). Our claim that $SD^{1/2}$ is a possible solution for V needs more justification. It is not immediately clear that just because this relation between V and $SD^{1/2}$ holds the rows of $SD^{1/2}$ must satisfy the same distance requirements as the rows of V . This result is in fact a particular property of matrices of the form AA^T . Assuming $AA^T = BB^T$, the fundamental idea is that the entries ij of AA^T and BB^T are the dot products of the rows i and j of A and B respectively. This means that corresponding rows of A and B must have the same length, and also the angle between rows i and j of A must equal the angle between the same rows of B for all i and j . Therefore the points represented by the rows of A and the points represented by the rows of B have the same magnitudes and relative angles and so we can find rotations and reflections mapping the points of A to the points of B .

Example

Consider the five points in \mathbb{R}^2 given by

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}, \quad \mathbf{w}_4 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad \mathbf{w}_5 = \begin{bmatrix} -3 \\ -3 \end{bmatrix},$$



The dissimilarity matrix for these points is

$$T = \begin{bmatrix} 0 & \sqrt{10} & \sqrt{26} & 2 & 5 \\ \sqrt{10} & 0 & 2 & \sqrt{26} & \sqrt{61} \\ \sqrt{26} & 2 & 0 & 5\sqrt{2} & \sqrt{89} \\ 2 & \sqrt{26} & 5\sqrt{2} & 0 & \sqrt{17} \\ 5 & \sqrt{61} & \sqrt{89} & \sqrt{17} & 0 \end{bmatrix}$$

Knowing only T , can we find a representation of the points?

Using MATLAB/Octave we have

```
> T
T =

    0.00000    3.16228    5.09902    2.00000    5.00000
    3.16228    0.00000    2.00000    5.09902    7.81025
    5.09902    2.00000    0.00000    7.07107    9.43398
    2.00000    5.09902    7.07107    0.00000    4.12311
    5.00000    7.81025    9.43398    4.12311    0.00000

> H = eye(5)-1/5*ones(5,1)*ones(1,5);
> Delta = T.^2; % this finds the square of each element in T
> [S D] = eig(-0.5*H*Delta*H)
S =
-0.0450409    0.9633529    0.2644287    0.2207222    0.7175132
 0.3690803    0.0063295    0.0398071   -0.6871823    0.3848777
 0.6031796    0.1253245   -0.3538343   -0.2243887   -0.3502305
-0.2791402   -0.1936833    0.6580702   -0.5270235   -0.4611091
-0.6480788    0.1367176   -0.6084717   -0.3885333    0.0419632

D =
56.60551    0.00000    0.00000    0.00000    0.00000
 0.00000   -0.00000    0.00000    0.00000    0.00000
 0.00000    0.00000    5.79449    0.00000    0.00000
 0.00000    0.00000    0.00000    0.00000    0.00000
 0.00000    0.00000    0.00000    0.00000   -0.00000
```

Remember that MATLAB/Octave returns the eigenvectors normalized, so we do not need to normalize the columns of S .

```
> V = S*D^0.5
V =
Columns 1 through 3:
```

```

-0.33887 + 0.00000i   0.00000 + 0.00000i   0.63653 + 0.00000i
 2.77684 + 0.00000i   0.00000 + 0.00000i   0.09582 + 0.00000i
 4.53812 + 0.00000i   0.00000 + 0.00000i  -0.85174 + 0.00000i
-2.10016 + 0.00000i  -0.00000 - 0.00000i   1.58409 + 0.00000i
-4.87593 + 0.00000i   0.00000 + 0.00000i  -1.46470 + 0.00000i

```

Columns 4 and 5:

```

 0.00000 + 0.00000i   0.00000 + 0.00000i
-0.00000 + 0.00000i   0.00000 + 0.00000i
-0.00000 + 0.00000i  -0.00000 - 0.00000i
-0.00000 + 0.00000i  -0.00000 - 0.00000i
-0.00000 + 0.00000i   0.00000 + 0.00000i

```

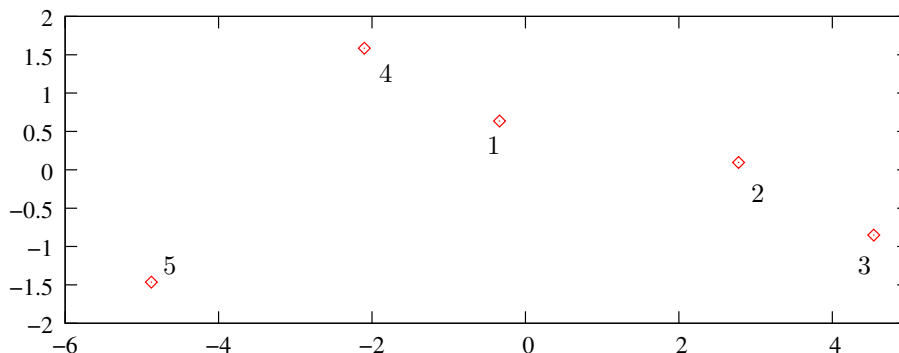
Each row of V is now one of the points we want to find in \mathbb{R}^5 . Notice that only the first and third entries of each row are non-zero (this is a result of the second, fourth and fifth eigenvalues in D being zero). Therefore we can plot the result in \mathbb{R}^2 as follows:

```

> plot(V(:,1),V(:,3),'rs')
> axis([-6 5 -2 2])

```

and we obtain:



This plot is rotated, translated and reflected from the plot that we started off with, but the relative positions of the points are the same as those in the original.

IV.6.4 The dimension of the plot

In the example above, if we had only been given the dissimilarity matrix T our best guess initially would have been that the 5 points came from a four-dimensional space. However we saw that we only needed two dimensions to plot the data: the components in the second, fourth and fifth coordinate directions were all zero because the corresponding eigenvalue

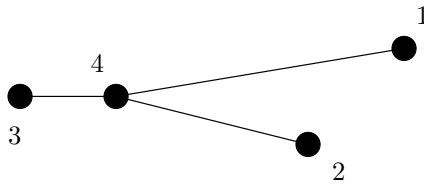
was zero. Often most of the variation takes place in only a few directions (the directions of the largest eigenvalues) and we can look at the projection of the data points onto those coordinate directions (in the example above the first and third coordinates). Because the matrix S in the construction contains orthonormal vectors, the coordinate axes chosen in this way will be orthonormal.

There is a bit of an art (and controversy) to choosing the correct number of coordinates, but if the first two or three eigenvalues are substantially larger than the remainder it is reasonably safe just to use those for a comparison.

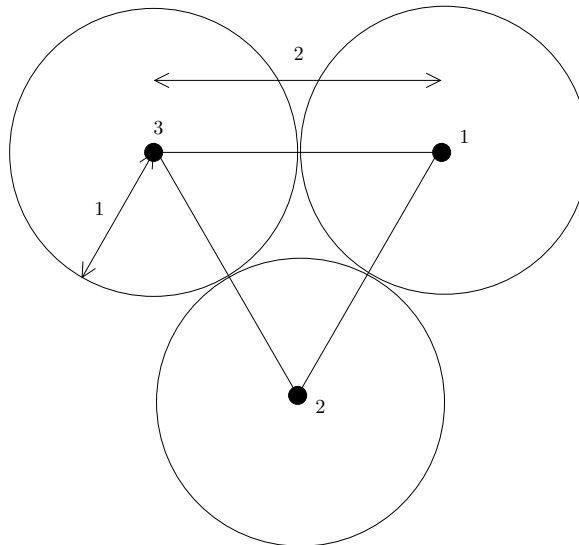
IV.6.5 Real examples

In most real-life applications, the dissimilarity matrix T does not produce a matrix $-H\Delta H/2$ that is positive semi-definite and so taking the square root of the matrix of eigenvalues D gives imaginary numbers. Consequently the decomposition $V = SD^{1/2}$ does not make sense.

An example is four cities connected by roads as follows:



We take the dissimilarity between two cities to be the driving time between them. If the journey time between cities along each road is exactly 1 hour (some roads are much better than others), then cities 1, 2 and 3 must all be an equal distance 2 apart in our representation. Therefore they are the vertices of an equilateral triangle. But now city 4 is only an hour apart from all three other cities and so must be a distance 1 apart from all the vertices of the equilateral triangle. But this is impossible (even with arbitrarily many dimensions):



One solution for this problem is to use only the coordinate directions with real eigenvalues. If these eigenvalues have significantly larger magnitudes than the negative eigenvalues, then this still produces a useful representation of the objects under consideration. This is the approach we will use in the examples below.

A difficult aspect of multi-dimensional scaling is determining how best to measure the dissimilarity between objects. This will depend on such things as what particular properties of the objects we are interested in and whether this data is quantitative (for example if we are comparing people we may be interested in their heights or their hair or eye colour). For our examples, we will assume that an appropriate choice of measurement has already been made.

Example 1

Take the set of objects to be the following towns and cities: Dawson Creek, Fort Nelson, Kamloops, Nanaimo, Penticton, Prince George, Prince Rupert, Trail, Vancouver and Victoria. We index them as

- 1 Dawson Creek
- 2 Fort Nelson
- 3 Kamloops
- 4 Nanaimo
- 5 Penticton
- 6 Prince George
- 7 Prince Rupert
- 8 Trail
- 9 Vancouver
- 10 Victoria

If our interest in the cities is their relative geographical location, then an appropriate measure of dissimilarity is the distance ‘as the crow flies’ between them. Then the entry T_{11} is the distance from Dawson Creek to itself (so 0 km), T_{12} and T_{21} are the distance between Dawson Creek and Fort Nelson, 374 km, and so on. The full dissimilarity matrix in km is:

$$T = \begin{bmatrix} 0 & 374 & 558 & 772 & 698 & 261 & 670 & 752 & 756 & 841 \\ 374 & 0 & 913 & 1077 & 1059 & 551 & 695 & 1124 & 1074 & 1159 \\ 558 & 913 & 0 & 306 & 151 & 384 & 782 & 261 & 260 & 330 \\ 772 & 1077 & 306 & 0 & 318 & 530 & 718 & 453 & 58 & 98 \\ 698 & 1059 & 151 & 318 & 0 & 535 & 912 & 140 & 260 & 295 \\ 261 & 551 & 384 & 530 & 535 & 0 & 504 & 629 & 524 & 610 \\ 670 & 695 & 782 & 718 & 912 & 504 & 0 & 1041 & 753 & 816 \\ 752 & 1124 & 261 & 453 & 140 & 629 & 1041 & 0 & 395 & 417 \\ 756 & 1074 & 260 & 58 & 260 & 524 & 753 & 395 & 0 & 86 \\ 841 & 1159 & 330 & 98 & 295 & 610 & 816 & 417 & 86 & 0 \end{bmatrix}$$

We can now find the PCA representation of the points using MATLAB/Octave (the m-file is on the web in `distance.m`):

```
> T = [0.   374.   558.   772.   698.   261.   670.   752.   756.   841.;
       374.    0.   913.  1077.  1059.   551.   695.  1124.  1074.  1159.;
       558.   913.    0.   306.   151.   384.   782.   261.   260.   330.;
       772.  1077.   306.    0.   318.   530.   718.   453.    58.    98.;
       698.  1059.   151.   318.    0.   535.   912.   140.   260.   295.;
       261.   551.   384.   530.   535.    0.   504.   629.   524.   610.;
       670.   695.   782.   718.   912.   504.    0.  1041.   753.   816.;
       752.  1124.   261.   453.   140.   629.  1041.    0.   395.   417.;
       756.  1074.   260.    58.   260.   524.   753.   395.    0.    86.;
       841.  1159.   330.    98.   295.   610.   816.   417.    86.    0.];

> Delta = T.^2;
> H = eye(size(T)) - 1./length(T)*ones(length(T),1)*ones(1,length(T));
> [S D] = eig(-0.5*H*Delta*H);
```

We would like to sort the eigenvalues such that the largest corresponds to the first coordinate, the next largest to the second, and so on. We can use the MATLAB/Octave command `sort` for this:

```
> [lambda,I] = sort(diag(D),'descend')
lambda =
  1.4615e+06
  4.4276e+05
  7.6808e+02
  2.4605e+02
  1.5347e+02
  3.9772e+00
  1.4233e-11
 -2.9002e+02
 -4.5881e+02
 -1.1204e+03
```

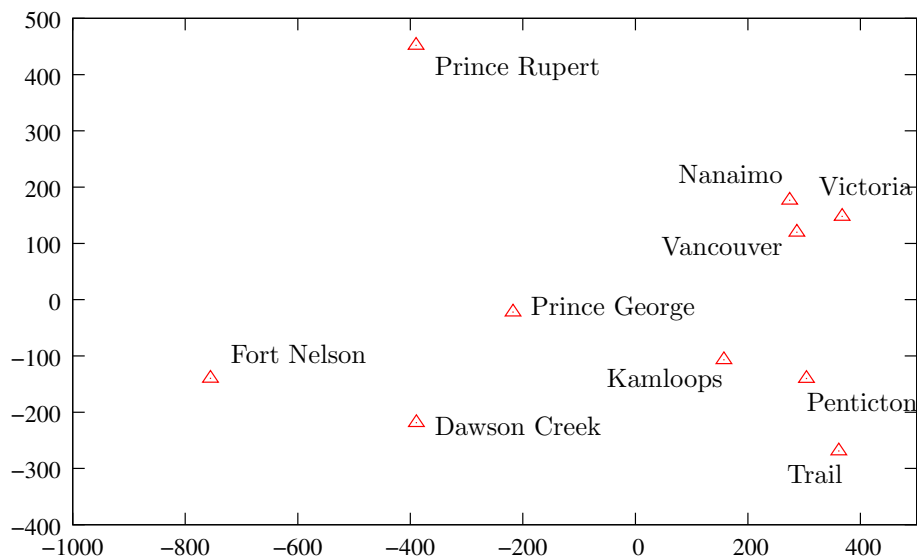
```
I =
  1
  2
  4
  7
  8
 10
  9
  6
  5
  3
```

Here the vector `lambda` is the eigenvalues sorted from largest to smallest and `I` contains the original indices of the elements in the sorted vector. We see that the two largest eigenvalues have substantially greater magnitudes than the following ones. This is what we would expect: it should be approximately possible to represent these towns and cities as points on a plane, so only two dimensions are needed. We also notice that it is not possible to represent the points perfectly in any dimensional space because there are some negative eigenvalues. These result partly from the fact that the distances are measured along the (curved) surface of the Earth and partly from rounding the distances in T .

If we just take the first two coordinates and plot the points

```
> X = S(:,I(1))*sqrt(D(I(1),I(1)));
> Y = S(:,I(2))*sqrt(D(I(2),I(2)));
> plot(X,Y,'bo');
> axis equal
```

we obtain



Notice that although the orientation of the cities is both rotated and reflected, their relative positions are exactly what we would expect.

Example 2

Instead, we may be interested in how easily and quickly we can get between the different towns and cities. In this case, a more appropriate measure of dissimilarity is the *driving*

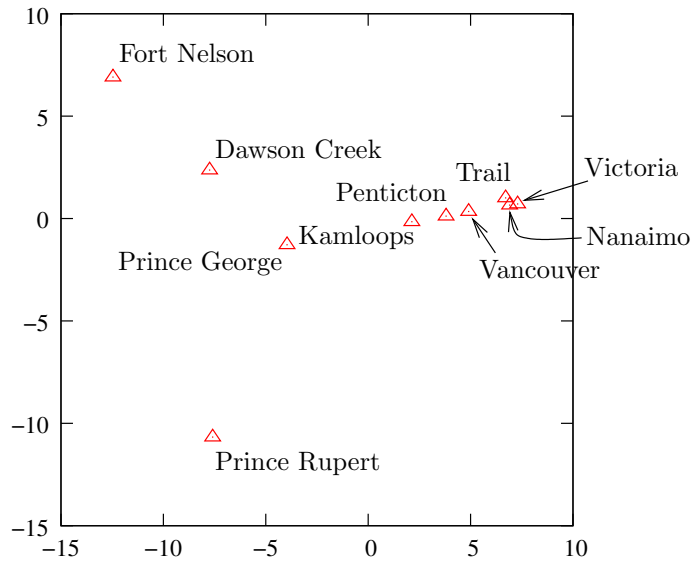
distance between the different towns and cities, or better still the driving *time* between them. The dissimilarity matrix for driving times (in hours and minutes) is:

$$T = \begin{bmatrix} 0:00 & 5:27 & 10:06 & 15:02 & 12:14 & 4:27 & 12:38 & 15:28 & 12:47 & 15:20 \\ 5:27 & 0:00 & 15:42 & 20:39 & 17:50 & 10:03 & 18:14 & 20:55 & 18:23 & 20:56 \\ 10:06 & 15:42 & 0:00 & 5:49 & 2:53 & 5:39 & 13:49 & 5:43 & 3:33 & 6:04 \\ 15:02 & 20:39 & 5:49 & 0:00 & 8:45 & 10:35 & 18:46 & 9:24 & 2:27 & 1:31 \\ 12:14 & 17:50 & 2:53 & 8:45 & 0:00 & 7:47 & 15:57 & 3:29 & 4:25 & 6:56 \\ 4:27 & 10:03 & 5:39 & 10:35 & 7:47 & 0:00 & 8:11 & 11:12 & 8:20 & 10:52 \\ 12:38 & 18:14 & 13:49 & 18:46 & 15:57 & 8:11 & 0:00 & 19:23 & 16:31 & 19:03 \\ 15:28 & 20:55 & 5:43 & 9:24 & 3:29 & 11:12 & 19:23 & 0:00 & 7:09 & 9:41 \\ 12:47 & 18:23 & 3:33 & 2:27 & 4:25 & 8:20 & 16:31 & 7:09 & 0:00 & 3:03 \\ 15:20 & 20:56 & 6:04 & 1:31 & 6:56 & 10:52 & 19:03 & 9:41 & 3:03 & 0:00 \end{bmatrix}$$

If we try the same code as used for the geographical distance with the dissimilarity matrix containing the driving times between the cities we find

```
lambda =
  4.7754e+02
  1.7095e+02
  7.5823e+01
  1.0814e+01
  1.2364e+00
 -1.4217e-15
 -4.6973e-01
 -3.4915e+00
 -1.0043e+01
 -3.3626e+01
```

and using just the first two coordinates we obtain this plot:

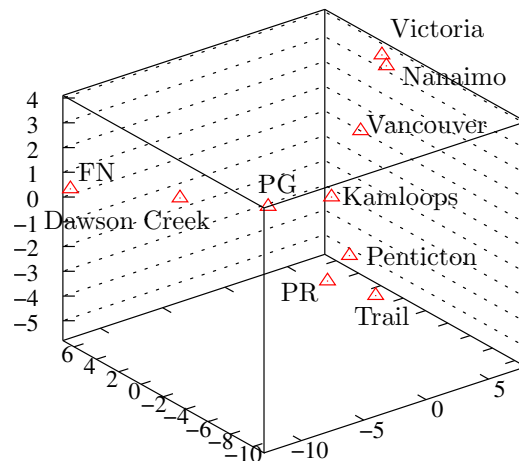


We notice that the structure is similar to that found for the distances, but there are a number of important differences. Roughly speaking major roads appear as straight lines on this plot: Vancouver to Kamloops is approximately Highways 1 and 5, then north from Kamloops to Prince George is Highway 97. Dawson Creek and Fort Nelson are found continuing along Highway 97 from Prince George while Prince Rupert is found by turning on to Highway 16. However there is a problem with what we have found: Trail is placed almost on top of Nanaimo and Victoria, when it is at the other side of the Province!

The problem is that we need to consider more principal coordinates to distinguish between Trail and Vancouver Island: the third eigenvalue is not much smaller than the second. If we add another principal coordinate

```
> X = S(:,I(1))*sqrt(D(I(1),I(1)));
> Y = S(:,I(2))*sqrt(D(I(2),I(2)));
> Z = S(:,I(3))*sqrt(D(I(3),I(3)));
> plot3(X,Y,Z,'bo')
> axis equal
```

then we see the following:



(The plot is much easier to understand if you plot it yourself and rotate it. The m-file is on the website in `hours.m`)

We can now identify Highways 3 and 97 to Trail branching off at Kamloops and the route by ferry to Vancouver Island (where a large part of the time is the ferry crossing which is similar for both Nanaimo and Victoria).