Covariance matrix

Variables in a data set may be related to one another, or they may be independent. The covariance matrix of a data set $\Sigma$ gives us an idea of the first order relationships between different dimensions. The covariance matrix is defined by (1).

$$
\Sigma(i, j) = \frac{\sum_{k=0}^{N} (x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j)}{N-1} \tag{1}
$$

The diagonal entries of the covariance matrix are the variances of each dimension. The off-diagonals show the relationships between different dimensions. If the off-diagonals are close to zero, then the two dimensions are largely independent. If the off-diagonals are large, then the two dimensions are strongly related. A covariance matrix is symmetric, as the covariance of dimensions $i$ and $j$ is the same as the covariance of dimensions $j$ and $i$.

Principal Component Analysis

Principal components analysis uses the covariance matrix $\Sigma$ to identify the primary directions of variation in a set of data. The eigenvectors of the covariance matrix provide a basis space for the data that is tailored specifically to those variation directions. The eigenvalues tell us the relative importance of each of the eigenvectors.

When the each data point is a small(ish) number of features and there are lots of them, we calculate the covariance matrix and compute its eigenvalues and eigenvectors.

Calculating the principal components using the covariance matrix:

- Calculate the covariance matrix for the data set. If we are use the convention of a row for each data point, then we need to tell it that the variables are in the columns, not the rows.

  $$
  \text{mcov} = \text{numpy.cov}( m, \text{rowvar=False} )
  $$

- Calculate the eigenvalues and eigenvectors of the covariance matrix. The eig function in the linalg package provides this capability in numpy. The return value is two arrays. The first contains the eigenvalues. The second contains the eigenvectors as columns of the matrix (meigvec[:,i]).
\[(\text{meigval}, \text{meigvec}) = \text{numpy.linalg.eig}( \text{mcov} )\]

- The eigenvalues tell you the relative importance of the eigenvectors. Looking at the ratio of the eigenvalues is an indication of their relative importance. A commonly used analysis is to look at the cumulative sum of the eigenvalues from largest to smallest as a fraction of their total sum. If the first few eigenvalues represent 95% of the sum of all the eigenvalues, for example, then the corresponding eigenvectors account for almost all of the variation in the data set.

- The eigenvectors tell you the directions of primary variation within the data. They are orthonormal vectors, which means the dot product of any two eigenvectors is zero and they have unit length.

Projecting data onto the principal components

- Select how many principal components to keep. As noted above, it is useful to look at the fraction defined by the cumulative sum of the eigenvalues divided by their total sum. Sometimes you can keep only two or three eigenvectors (for visualization, for example), but other times you may want to choose enough eigenvectors to represent some percentage of the data variation (e.g. 90%).

- Take the dot product of each difference vector with the principal component directions. You can treat the eigenvector matrix as a rotation matrix. Each row of deltadata gets dotted with each column of meigvec.

\[\text{pdata} = \text{deltadata} \ast \text{numpy.matrix}( \text{meigvec} )\]

- The resulting set of feature vectors is a compressed representation of the data.

### 2.1 Toy Example

Let’s perform a principle component analysis on a data set with 2 features:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

![Toy Example Graph](image-url)
We compute the covariance C:
\[
C = \begin{pmatrix}
1.6667 & 1.4167 \\
1.4167 & 1.2158
\end{pmatrix}
\]

And then its eigenvectors (the columns of V) and eigenvalues (the diagonal entries of D):
\[
V = \begin{pmatrix}
0.6492 & 0.7606 \\
-0.7606 & -0.6492
\end{pmatrix}, \quad D = \begin{pmatrix}
0.0068 & 0 \\
0 & 2.8757
\end{pmatrix}
\]

To make sense of the data, let’s reorder the eigenvectors so that the one associated with the largest eigenvalue is first, and then they appear in decreasing order of significance. We call the eigenvectors the “principal components” and the eigenvector associated with the largest eigenvalue is called the first principal component.

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvector</td>
<td>( v_1 )</td>
<td>( v_2 )</td>
</tr>
<tr>
<td></td>
<td>0.7606</td>
<td>0.6492</td>
</tr>
<tr>
<td></td>
<td>-0.6492</td>
<td>-0.7606</td>
</tr>
</tbody>
</table>

We learn that there is one dominant dimension of information in this case. The largest eigenvalue is 99.8% of the sum of all of the eigenvalues. If we plot the eigenvectors on the data, and scale their lengths by their eigenvalues, then we see the two orthogonal directions of information. Below, I plot the two eigenvectors on the scatter plot, scaling the length to indicate which is the dominant vector. Ideally, I would scale the length by the eigenvalue, but the second vector is too small, so I scaled it by 0.1. It still barely shows up, but you can see that it is short and perpendicular (orthogonal) to the first vector.

![Scatter plot with eigenvectors](image)

When we have more than 3 features, it is useful to project the data onto the first few principal components. For this example, we will do that projection in order to understand how it transforms the data.

First, we zero-center the data (\( \Delta X = X - \bar{X}, \Delta Y = Y - \bar{Y} \))

\[
\begin{array}{c|c|c|c}
\Delta X & \Delta Y & \Delta X & \Delta Y \\
1-1.5 & 1.1-1.75 & -0.5 & -0.575 \\
2-1.5 & 2.1-1.675 & 0.5 & 0.425 \\
3-1.5 & 3-1.675 & 1.5 & 1.325 \\
0-1.5 & 0.5-1.675 & -1.5 & -1.175 \\
\end{array}
\]
Then, we project the zero-centered data onto each of the principle components, using the notation $p_1$ for the data projected onto the first principal component and $p_2$ for the data projected onto the second principal component. We then scatter plot the data using the new coordinates.

$$p_1 = (\Delta X \ \Delta Y) v_1$$

$$= \begin{pmatrix} -0.5 & -0.575 \\ 0.5 & 0.425 \\ 1.5 & 1.325 \\ -1.5 & -1.175 \end{pmatrix} \begin{pmatrix} 0.7606 \\ -0.6492 \end{pmatrix}$$

$$= \begin{pmatrix} 0.11 \\ 0.0013 \\ -0.0341 \\ -0.08 \end{pmatrix}$$

$$p_2 = (\Delta X \ \Delta Y) v_2$$

$$= \begin{pmatrix} -0.5 & -0.575 \\ 0.5 & 0.425 \\ 1.5 & 1.325 \\ -1.5 & -1.175 \end{pmatrix} \begin{pmatrix} 0.6492 \\ -0.7606 \end{pmatrix}$$

$$= \begin{pmatrix} 0.7536 \\ -0.6562 \\ -2.0011 \\ 1.9037 \end{pmatrix}$$

When we plot the projection, we see that the data change shape - we no longer have a correlation.