Principal Component Analysis

Intuition

The goal of Principal Component Analysis (PCA) is to reduce the number of features in a dataset whilst preserving the variance in the remaining features. It is vital that variance in the dataset is maximised so that we get the best possible performance from our Machine Learning Model. The PCA algorithm involves computing the principal components and performing a change of basis. We will show later that these principal components are the eigenvectors of the covariance matrix of the dataset.

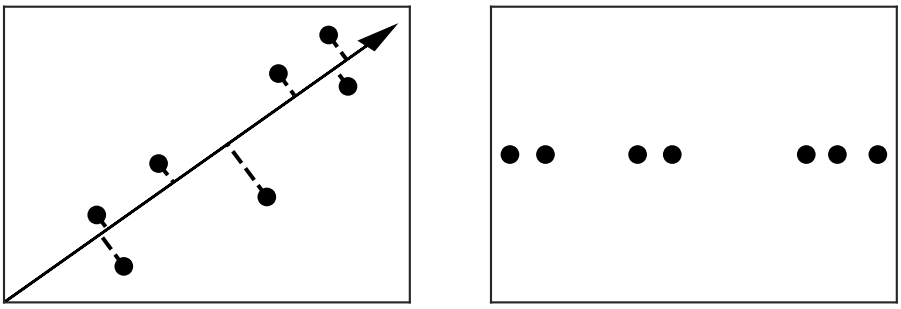

Figure 1: Data points in a plane can be projected onto a one-dimensional line.

Consider the two-dimensional data on the left of Figure 1. We can project these points onto a vector to create a new feature in one-dimensional space. There are infinitely many vectors we could draw on the plane, so which is the best one to choose to maximise the variance, or spread, of the points?

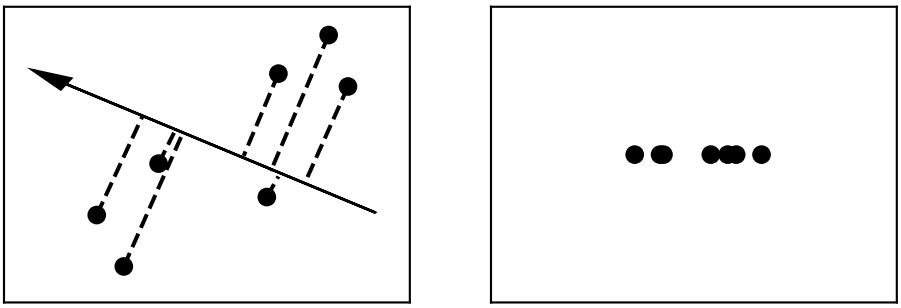

Figure 2: Projecting the data points along this vector results in a feature with minimal variance.

Projecting along the vector shown in Figure 2 is a lot less favourable than projecting along the vector in Figure 1. To maximise the variance in the new feature, we need to project

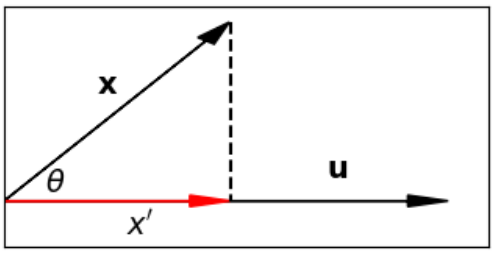

Figure 3: The dot product $\mathbf{u}^T\mathbf{x} = ||\mathbf{x}||\,||\mathbf{u}||\cos\theta$. The scalar projection of $\mathbf{x}$ along $\mathbf{u}$ is $x' = ||\mathbf{x}||\cos\theta$.

the data along a vector that is directed along the line of greatest variance in the data, as in Figure 1.

It can be shown that if we project the two-dimensional position vector of the point $\mathbf{x}$ onto the unit vector $\mathbf{u}$, which spans a one-dimensional line, the scalar projection along $\mathbf{u}$, x', is the dot product of $\mathbf{x}$ and $\mathbf{u}$. That is, we have

$$x' = \mathbf{u}^T\mathbf{x} = ||\mathbf{x}||\,||\mathbf{u}||\cos\theta = ||\mathbf{x}||\cos\theta \tag{1}$$

where θ is the angle between $\mathbf{x}$ and $\mathbf{u}$.

Thinking in terms of the scalar projection we can see that the sum of projections of all data points will be greatest when the angles between the $\mathbf{x}_i$ and $\mathbf{u}$ are minimised, as $\cos\theta$ is largest for small θ. This is further indication that we desire the vector which we will be projecting onto to lie along the line of maximal variance in the data, as this is the direction in which the angles between the $\mathbf{x}_i$ and $\mathbf{u}$ will be minimised.

Now, if $\mathbf{X}$ is the matrix of all the (multi-dimensional) samples, $\mathbf{x}_i$, and $\mathbf{U}$ is the matrix of all the vectors we are projecting onto,$\mathbf{u}_i$, the projected data is computed with the transformation $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$. The matrix multiplication method relies on taking the dot product between rows of the first matrix and columns of the second, so assuming $\mathbf{U}^T$ has its rows composed of the $\mathbf{u}_i$, and $\mathbf{X}$ has the $\mathbf{x}_i$ in the columns, each element of $\mathbf{Z}$ will be the result of a dot product between one of the $\mathbf{u}_i$ and one of the $\mathbf{x}_i$ (see equation (11)). This means that the projection of the data onto the lower-dimensional subspace is still governed by the dot product, and the intuition that we developed above remains valid.

So the obvious question now is how do we find the directions of greatest variance in the data so that we can project along them? We will show later that the eigenvectors of the covariance matrix of the data lie along the directions of greatest variance.

The Covariance Matrix

The matrix $\mathbf{X}$, with m features and n samples has the covariance matrix $\mathbf{C}_X$ defined by

$$\mathbf{C}_X = \begin{pmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) & \dots & \text{cov}(x_1, x_m) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) & \dots & \text{cov}(x_2, x_m) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(x_m, x_1) & \text{cov}(x_m, x_2) & \dots & \text{cov}(x_m, x_m) \end{pmatrix} \tag{2}$$

where

$$\text{cov}(x_i, x_j) = \frac{1}{n} \sum_{k=1}^{n} (x_{ik} - \bar{x_i})(x_{jk} - \bar{x_j}) \tag{3}$$

and $\bar{x_i}$ and $\bar{x_j}$ denote the means of the features x_i and x_j, respectively.

The covariance matrix is square and symmetric and as its name suggests measures the covariance between each pair of elements of a given vector (or matrix). The variance of a feature is defined as the sum of the squared distances of each sample from the mean value. Mathematically, this is

$$V(x_i) = \frac{1}{n} \sum_{k=1}^{n} (x_{ik} - \bar{x_i})^2 = \text{cov}(x_i, x_i) \tag{4}$$

Therefore, the diagonal elements of the covariance matrix are the feature variances. The off-diagonal elements measure the correlation, or the redundancy between different features. Redundant features add no useful information as they are highly-correlated with other features and essentially carry the same information.

The PCA objective can be formulated with regards to the covariant matrix $\mathbf{C}_Z$ of the projected data $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$. When we project the data we simultaneously want to reduce the number of features and maximise variance in the remaining features. The features we would like to remove are those that are redundant. So what would we like the covariance matrix of the projected data to look like? We want to maximise the variance in the data (the diagonal elements) and minimise the redundancy (the off-diagonal elements). This objective corresponds to a diagonal covariance matrix $\mathbf{C}_Z$.

Choosing U for the Transformation $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$

As discussed in the previous section, we can formulate the objective of PCA as diagonalising the covariance matrix $\mathbf{C}_Z$ of the projected data $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$. The first step in the PCA algorithm is to mean normalise the features. This means we can write equation (3) as

$$\text{cov}(x_i, x_j) = \frac{1}{n} \sum_{k=1}^{n} x_{ik} x_{jk} \tag{5}$$

and it follows that

$$\mathbf{C}_X = \frac{1}{n}\mathbf{X}\mathbf{X}^T \tag{6}$$

Performing the transformation $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$ we obtain

$$\mathbf{C}_Z = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T = \frac{1}{n}\mathbf{U}^T\mathbf{X}(\mathbf{U}^T\mathbf{X})^T = \frac{1}{n}\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}^T\mathbf{C}_X\mathbf{U} \tag{7}$$

where we have used the identity $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.

The theory of eigendecomposition states that any real, symmetric matrix $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{Q}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{A}$. If we choose $\mathbf{U}$ to be the orthogonal matrix whose columns are the eigenvectors of $\mathbf{C}_X$, we can use the fact that $\mathbf{C}_X$ is a real, symmetric matrix and equation (8) to write

$$\mathbf{C}_X = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{9}$$

An orthogonal matrix has the property that its transpose is equal to its inverse, that is $\mathbf{U}^T = \mathbf{U}^{-1}$. Therefore, substituting into (7) we obtain

$$\mathbf{C}_Z = \mathbf{U}^T\mathbf{C}_X\mathbf{U} = \mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{U} = \mathbf{U}^{-1}\mathbf{U}\mathbf{D}\mathbf{U}^{-1}\mathbf{U} = \mathbf{D} \tag{10}$$

where we have used the identity $\mathbf{U}^{-1}\mathbf{U} = \mathbf{I}$. We have shown that to achieve the goal of PCA, a diagonal covariance matrix $\mathbf{C}_Z$, we make the orthogonal projection $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$, where $\mathbf{U}$ has the eigenvectors of $\mathbf{C}_X$ as its columns.

Proof that the Eigenvectors of $\mathbf{C}_X$ Lie Along the Directions of Greatest Variance in X

Suppose we have the matrices $\mathbf{U}$ and $\mathbf{X}$ with some vectors $\mathbf{u}_i$ and $\mathbf{x}_i$ in the columns, respectively:

$$\mathbf{U} = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_k \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}$$

The transformation $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$ will produce the matrix

$$\mathbf{Z} = \begin{pmatrix} \longleftarrow & \mathbf{u}_1 & \longrightarrow \\ \longleftarrow & \mathbf{u}_2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \mathbf{u}_k & \longrightarrow \end{pmatrix} \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^T\mathbf{x}_1 & \mathbf{u}_1^T\mathbf{x}_2 & \dots & \mathbf{u}_1^T\mathbf{x}_n \\ \mathbf{u}_2^T\mathbf{x}_1 & \mathbf{u}_2^T\mathbf{x}_2 & \dots & \mathbf{u}_2^T\mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_k^T\mathbf{x}_1 & \mathbf{u}_k^T\mathbf{x}_2 & \dots & \mathbf{u}_k^T\mathbf{x}_n \end{pmatrix} \tag{11}$$

The matrix $\mathbf{Z}$ has its samples in the columns (like $\mathbf{X}$). That is

$$\mathbf{Z} = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \tag{12}$$

Therefore, the sample $\mathbf{z}_i$ is given by

$$\mathbf{z}_i = \begin{pmatrix} \mathbf{u}_1^T \mathbf{x}_i \\ \mathbf{u}_2^T \mathbf{x}_i \\ \vdots \\ \mathbf{u}_k^T \mathbf{x}_i \end{pmatrix} = \begin{pmatrix} z_{1i} \\ z_{2i} \\ \vdots \\ z_{ki} \end{pmatrix} \tag{13}$$

where, for example

$$z_{1i} = \mathbf{u}_1^T \mathbf{x}_i \tag{14}$$

Let's consider just the first feature in $\mathbf{Z}$ which we denote as z_1. This feature runs along the top row of $\mathbf{Z}$ and is the first element in every $\mathbf{z}_i$. From equation (4) the variance of z_1 is given by

$$V(z_1) = \frac{1}{n} \sum_{i=1}^{n} (z_{1i} - \bar{z}_1)^2 = \frac{1}{n} \sum_{i=1}^{n} z_{1i}^2 \tag{15}$$

where we have assumed that the data has been mean normalised, i.e., $\bar{z}_1 = 0$. We substitute equation (14) into (15) to get

$$V(z_1) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{u}_1^T \mathbf{x}_i)^2 \tag{16}$$

The dot product is symmetric with respect to its arguments, i.e., $\mathbf{a}^T \mathbf{b} = \mathbf{b}^T \mathbf{a}$. Therefore

$$V(z_1) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{u}_1^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{u}_1 = \mathbf{u}_1^T \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \mathbf{u}_1 = \mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 \tag{17}$$

As previously discussed, our aim in PCA is to maximise the variance in the projected data. So for the feature z_1, our goal is to find the vector $\mathbf{u}_1$ that maximises the variance. However, we notice that we can make the variance in z_1 larger simply by increasing $\mathbf{u}_1$. This does not help us find the optimal $\mathbf{u}_1$ to project onto, so we restrict $\mathbf{u}_1$ to $||\mathbf{u}_1||^2 = 1$ in order to focus on identifying its direction.

The problem at hand has now become one of constrained optimisation:

$$\max_{\mathbf{u}_1} \mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 \quad \text{subject to} \quad ||\mathbf{u}_1||^2 - 1 = 0 \tag{18}$$

The local optima of a function subject to a constraint can be found using the method of Lagrange multipliers. For a function $f(x)$ subject to the constraint $g(x) = 0$, we construct the Lagrangian function

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x) \tag{19}$$

and find the stationary points of $\mathcal{L}$. From our constrained optimisation problem specified in (18) we can form the Lagrangian

$$\mathcal{L}(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 - \lambda_1(||\mathbf{u}_1||^2 - 1) \tag{20}$$

Utilising the chain rule, the condition to find the stationary points of L is

$$d\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \mathbf{u}_1} d\mathbf{u}_1 + \frac{\partial \mathcal{L}}{\partial \lambda_1} d\lambda_1 = 0 \tag{21}$$

As $d\mathbf{u}_1$ and $d\lambda_1$ are independent, we must have

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_1} = \frac{\partial \mathcal{L}}{\partial \lambda_1} = 0 \tag{22}$$

Calculating the partial derivatives, we obtain

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_1} = 2\mathbf{C}_X \mathbf{u}_1^T - 2\lambda_1 \mathbf{u}_1^T \tag{23}$$

and

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = 1 - \mathbf{u}_1^T \mathbf{u}_1 \tag{24}$$

Setting equation (23) equal to zero yields

$$\mathbf{C}_X \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \tag{25}$$

Premultiplying both sides by $\mathbf{u}_1^T$ gives

$$\mathbf{u}_1^T \mathbf{C}_X \mathbf{u}_1 = \mathbf{u}_1^T \lambda_1 \mathbf{u}_1 \tag{26}$$

Therefore, we have

$$V(z_1) = \lambda_1 \mathbf{u}_1^T \mathbf{u}_1 = \lambda_1 \tag{27}$$

Finally, we can write

$$\mathbf{C}_X \mathbf{u}_1 = V(z_1) \mathbf{u}_1 \tag{28}$$

We recognise equation (28) as an eigenvalue equation, where the eigenvalue is the variance in the feature z_1. Hence, the eigenvector $\mathbf{u}_1$ required to maximise the variance of z_1 points in the direction of the variance, and is an eigenvector of the covariance matrix $\mathbf{C}_X$.

When projecting the data onto the lower-dimensional space we maximise the variance by choosing the $\mathbf{u}_i$ associated with the largest eigenvalues of $\mathbf{C}_x$. The eigenvector associated with the largest eigenvalue is known as the principal component.

The PCA Algorithm

1. Perform mean normalisation, or if required scale the features.
2. Compute the covariance matrix $\mathbf{C}_X$ of the data $\mathbf{X}$. If $\mathbf{X}$ has the dimensions $(m \times n)$, where m is the number of features and n is the number of samples, then $\mathbf{C}_X$ will have the dimensions $(m \times m)$.
3. Compute the eigenvalues and eigenvectors of $\mathbf{C}_X$.
4. Create a matrix, $\mathbf{U}$, with the eigenvectors of $\mathbf{C}_X$ arranged in columns, and sorted in order of descending associated eigenvalues:

$$\mathbf{U} = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_m \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}, \quad (m \times m)$$

5. Reduce $\mathbf{U}$ to the desired dimensionality:

$$\mathbf{U} = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_k \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}, \quad (m \times k), \;\; k < m$$

6. Multiply the data $\mathbf{X}$ by $\mathbf{U}^T$:

$$\mathbf{Z} = \mathbf{U}^T\mathbf{X}, \;\; (k \times m) \cdot (m \times n) = (k \times n)$$

   i.e., $\mathbf{Z}$ has n samples with k features.

References

Deisenroth, M. S., Faldo, A. A., Ong, C. S.(2020). *Mathematics for Machine Learning*. Cambridge University Press.

Ng, A. *Machine Learning*. Offered by Stanford University on Coursera.

Shlens, J.(2014). *A Tutorial on Principle Component Analysis*.

Appendix

Differentiating the magnitude squared of a vector

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{x}}||\mathbf{x}||^2 &= \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{x}) \\
&= \frac{\partial}{\partial \mathbf{x}}(x_1^2 + x_2^2 + ... + x_n^2) \\
&= \begin{pmatrix} \frac{\partial}{\partial x_1}(x_1)^2 \\ \frac{\partial}{\partial x_2}(x_2)^2 \\ \vdots \\ \\ \frac{\partial}{\partial x_n}(x_n)^2 \end{pmatrix} \\
&= \begin{pmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{pmatrix} \\
&= 2\mathbf{x}
\end{aligned}
$$