PCA and PCoA

Gui Araujo

1 PCA

Consider a data matrix X. An element x_{ij} of X represents the value of the variable j of sample i. For example, if I measure samples of animal presence data, I will have x_{ij} as the number of observations of animal j on the sample i. The goal of PCA (principal component analysis) is to transform the data matrix X to describe it with new axes (representing new variables) allowing for independent variation between variables. If we do this, we can order these axes or directions by how much variation they show. If we select a subset of directions with the largest data variation, they allow us to better visualize the data in lower dimensions while minimizing the error of discarding information. The most common use of PCA is to take only the two directions containing the highest variation and visualize the data on a 2D plot. Thus, To be able to look at the data in the variables' space with lower dimensions, we always have to discard some information, and PCA is a tool to waste a minimal amount. Those best directions to consider are called the principal components (PCs).

Taking the first PCs is analogous to expanding a function and taking only the leading terms, the dominant orders. In PCAs case, the error comes by discarding the directions with smaller data variation, and the PCA minimizes this error. Data variation is the outcome of sources of change in the data-generating process, the mechanisms producing diversity in the dataset (variation between samples).

The covariance between data variables is a measure of variational structure. If some data variables strongly co-vary, then the variation they jointly produce is along a preferred direction, instead of randomly placed. We can think of the covariances between samples as a representation of statistical **information** contained in the dataset, the information used to cast the differences in samples.

The pairwise covariance matrix for X computes all covariances between any two variables (plus variances on the diagonal). If the data X has N samples and M variables, then the covariance matrix is a square MxM matrix with entries computed as:

$$cov_{kl} = \frac{\sum_i (x_{ik} - \mu_k)(x_{il} - \mu_l)}{N-1}. \tag{1}$$

The parameters μ are the averages of variables, defined as:

$$\mu_k = \frac{\sum_i x_{ik}}{N} \tag{2}$$

If two variables increase together, their covariance gets more positive. If they decrease together, it gets more negative. If they change independently, the factors will tend to cancel each other and drive the covariance near zero. To calculate the covariance from a data set, we are actually building a sample estimator of the theoretical value (and the same holds for the averages). The unbiased sample estimator of the covariance is divided by $(N-1)$ instead of N as a correction for using the unknown average (which correlates a bit with the data). This is called the Bessel's correction.

When working with different variables, each having their own natures and scales, it is always good to standardize them so they are comparable. Therefore, it's best to work with standardised data. This is done by transforming the data matrix X into the matrix Z, with variables rescaled to have zero mean and unit variance:

$$z_{ik} = \frac{x_{ik} - \mu_k}{\sigma_k}, \tag{3}$$

where $\sigma_k = \sqrt{\frac{\sum_i (x_{ik} - \mu_k)^2}{N-1}}$. Then the covariance matrix becomes the same as the (Pearson) correlation matrix:

$$C = cov_{kl} = corr_{kl} = \frac{1}{N-1} \sum_i z_{ik} z_{il} = \frac{Z^T Z}{N-1}. \tag{4}$$

The square matrix $Z^T Z$ is statistically important to a standardized data set Z and is roughly the matrix equivalent of the square of a number. It's a symmetrical matrix and therefore is always diagonalisable (this will make sense in a bit, and it means that the PCA will always be possible). PCA is a change of basis from C to the basis of eigenvectors of C, which in this case are always orthogonal.

Why eigenvectors and eigenvalues? We should know that the data points are objects independent of the frame of reference we use to describe them. We can view them as an invariant cloud of points. Then, we generate a basis of orthogonal axes as a frame to write the data points as vectors. We can change this basis, then altering the way we describe the data, but without reshaping the cloud of data points. It's just a change of perspective. Then, the change of basis to the basis formed by the eigenvectors of a matrix is a transformation that diagonalizes that matrix (i.e. the same unchanged data, as viewed by the transformed perspective, is now described by a diagonal matrix). It means looking at the covariance matrix C from a perspective where the directions of variation are independent, where all covariances between variables are zero, and directions can be considered separately. In this case, we are left only with the eigenvalues as the diagonal elements in the new perspective, which represent the variances in each direction. Then, choosing the directions with highest eigenvalues is the same as choosing the ones with highest variance. Thus, for example, choosing the first two PCs is the same as choosing the two eigenvectors of C with the two highest eigenvalues.

The data in the transformed coordinates, which we visualise with the PCA plot, is simply the projection of the original Z on the chosen eigenvalues (which means Z as viewed from the perspective of the new basis, with variables made independent). The projected data is good to visualise the separation of clusters of data-groups because the separation is driven by variation in the data, which is exactly what the PCs optimize and summarize in up to 3 (visualisable) dimensions. This projection is like the shadow of the data points cast onto the first PC directions (and note that casting shadows means losing information contained in the other dimensions).

2 Change of basis

So the matrix C is square with dimension MxM. It has a set of M eigenvalues λ^m associated with M eigenvectors v^m. They are defined as

$$Cv = \lambda v, \tag{5}$$

which means that

$$(C - \lambda I)v = 0 \rightarrow det(C - \lambda I) = 0, \tag{6}$$

where I is the identity matrix of dimension M. By solving for the determinant of C, we find all λ^m. The eigenvectors v^m are defined up to a multiplicative constant, which we determine by setting their length (or norm) to 1: $\sqrt{\sum_k (v_k^m)^2} = 1$. Then, we order the eigenvectors as successive columns of a matrix by highest to lowest eigenvalue. Define the matrix W as that matrix considering only the subset of leftmost eigenvectors not discarded by the PCA. The projection of the data is given by

$$Z' = ZW. \tag{7}$$

If we choose S PCs, W has a dimension MxS and Z' has a dimension NxS. The new variables calculated for samples in the new data X' are features the PCA extracts as a mix of the original features. The original variables are mixed as we change the perspective of the data and the new ones are made from linear combinations of the original ones.

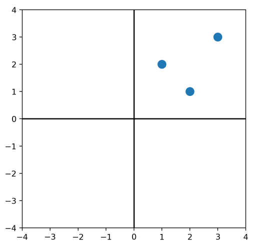
The loadings of each original variable for each PC are the fractions of the variance encoded in the PC that come from that variable. Given a PC m with eigenvalue λ^m and eigenvector v^m, λ^m is the total variance captured by m and each entry of v^m is the weight of each original variable along that direction. Therefore, the loadings for this PC are $\lambda^m v^m$, the vector of M fractions of the variance λ^m corresponding to each feature.

3 PCA example from scratch

Now let's calculate everything using a very simple example. We measure the observation counts of two animals present in 3 different patches, generating a data matrix X with 3 samples and 2 variables:

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 3 \end{bmatrix} \tag{8}$$

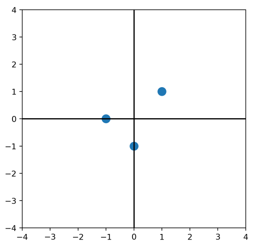
We visualise the data points as:



The mean and standard deviation for each variable (each column) are $\mu_1 = \mu_2 = 2$ and $\sigma_1 = \sigma_2 = 1$. Then, the standardised matrix Z becomes:

$$Z = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 2 & 2 \end{bmatrix} \tag{9}$$

visualised as:



Then, we calculate the matrix of covariances

$$C = \frac{1}{N-1} Z^T Z = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{10}$$

The condition for the eigenvalues of C is

$$\det \left[\begin{bmatrix} 1-\lambda & \frac{1}{2} \\ \frac{1}{2} & 1-\lambda \end{bmatrix} \right] = 0 \tag{11}$$

equivalent to the equation

$$(1-\lambda)^2 = \frac{1}{4} \tag{12}$$

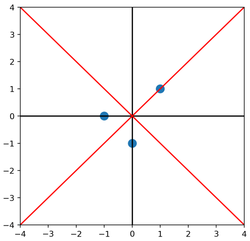
which yields the two eigenvalues $\lambda^+ = 3/2$ and $\lambda^- = 1/2$. The associated eigenvalues are calculated from

$$\frac{1}{2}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{13}$$

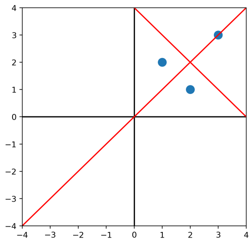
which, normalised, result in

$$v^+ = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad v^- = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{14}$$

These two directions are shown in red below: Note how v^+ is indeed the direction where the variance of data point is the highest and v^- is orthogonal to it. The two red lines are defined by the equations defining the eigenvectors: $x - y = 0$ and $x + y = 0$. The standardized data Z in this case is only shifted from the original data X by a translation of $(-2, -2)$. Transforming back these lines, we can see the same change from the perspective of the original data, with equations $(x-2) - (y-2) = 0$ and $(x-2) + (y-2) = 0$:



The diagonalised version of matrix C is given by the eigenvalues of C. Here,



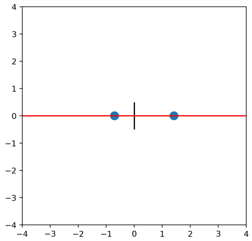
there are no correlations between variables, only the independent variances:

$$C' = \frac{1}{2}\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}. \tag{15}$$

Since this example dataset is so simple, there are only 2 directions in total, but we can consider only v^+ as a PC, discarding v^- as the direction with smaller variance (which means less information about the processes causing variation in the data). We can then project the data Z on v^+:

$$ZW = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}. \tag{16}$$

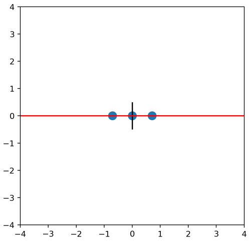
Two of the points coincide (we can see this in the figures above by looking at the data with the perspective of this direction alone). The PCA visualisation with this sole direction as PC is then (with a small black line marking the origin):



We can do the same with v^- as PC:

$$ZW = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}. \tag{17}$$

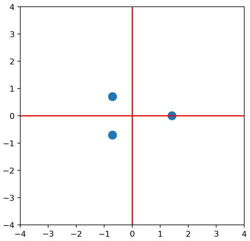
visualised as:



We can also consider both v^+ and v^-, which particularly in this case means no waste of information:

$$ZW = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 2 & 0 \end{bmatrix}. \tag{18}$$

If we calculate the variance of each column, we can verify that they are $\lambda^+ = 3$ and $\lambda^- = 1$. We visualise this data as: Although simple, the example above carries all elements to be generalised for a data matrix X of any dimension NxM.



4 Distances and PCoA

The principle coordinate analysis (PCoA) is very similar to PCA, but it's a different task. PCoA maps the distances between the samples of the dataset Z in a lower-dimension space. In its original variable's space, the data points of Z have a natural relationship represented by their pairwise distances. We can then ask the question: can this same structure be projected into a 2D or 3D space, so we visualise them without distortion? In general, the short answer is no (or yes only for very specific configurations). However, as with PCA, we can do it while minimizing our loss of information and still retaining much of the structure present in the original distances, which means minimizing the distortion of distances. This is analogous to the projection of Earth as a flat 2D map, with the lands being always distorted but recognizable.

Distance is an abstract measure of the relationship between two points z_i and z_j. We can define many types of distances $d(z_i, z_j)$, and they have to follow the following properties:

$$1. \quad d(z_i, z_j) > 0 \tag{19}$$

$$2. \quad d(z_i, z_i) = 0 \tag{20}$$

$$3. \quad d(z_i, z_j) = d(z_j, z_i) \tag{21}$$

$$4. \quad d(z_i, z_j) \leq d(z_i, z_o) + d(z_o, z_j) \tag{22}$$

The notion of a distance between vectors is useful to generate a metric space, which we can think of as casting a cohesive "surface" for the points to inhabit. The distance between two points is then the path on the surface that connects them. Properties 1 and 2 say that we start at distance zero and increase the distance as we go along a path. Property 3 says that going back on the same path travels the same distance. Property 4 ensures that going directly to a destination, without deviating, is always the shortest path in the surface between the two points.

The metric space we perceive in the real classical world carries as ”surface” the usual 3D volume, with distances that we call Euclidean distances:

$$d(z_i, z_j) = \sqrt{\sum_k (z_{ik} - z_{jk})^2}. \tag{23}$$

The real 3D world has $k = 1, 2, 3$ components.

A commonly used distance in community ecology is the Jaccard distance:

$$d_{ij} = \frac{x_i \cup x_j - x_i \cap x_j}{x_i \cup x_j} = 1 - \frac{x_i \cap x_j}{x_i \cup x_j} = 1 - \frac{\sum_k min(x_{ik}, x_{jk})}{\sum_k max(x_{ik}, x_{jk})}. \tag{24}$$

The intersection divided by the union takes the fraction of the union that belongs to both sets, being a similarity measure. The complementary fraction is then a distance between the two sets. The fraction counts the shared portions of the variables’ values.

Another common distance is the Bray-Curtis:

$$d_{ij} = 1 - \frac{2\sum_k (min(x_{ik}, x_{jk}))}{\sum_k (x_{ik} + x_{jk})} = 1 - \frac{\sum_k min(x_{ik}, x_{jk}) + \sum_k min(x_{ik}, x_{jk})}{\sum_k max(x_{ik}, x_{jk}) + \sum_k min(x_{ik}, x_{jk})} \tag{25}$$

An intuition for this expression is: the similarity counts all ”lengths” of vectors that are shared, from the total of lengths among the two vectors, where each variable of each sample gives its own length. Note the comparison to the Jaccard, where Bray-Curtis adds the minimum lengths to both sides of the fraction.

The Jaccard distance is the most intuitive, it comes from the direct overlap of sets, and it is 0.5 if all the lengths of one point are half the lengths of the other, aligned with our intuition (and the Bray-Curtis is 0.666..)). The Bray-

Curtis is higher, and it gives a boost for ”shared presence”, which makes sense for ecology, where we think of shared diversity as more robustly similar than shared abundance exactly (because this can fluctuate a lot more). Both distances go from zero to one and, for only presence/absence data, they become the same.

For the PCoA, we can directly use a distance d or use it to build a measure of similarity between points, which decreases as the distance increases. Suppose we build a function $f_{ij} = f(z_i, z_j)$ representing distance or similarity. Then, we generate a NxN matrix F using f_{ij} as entries. The PCoA is simply to perform the PCA on F instead of Z. But why a PCA on F means mapping distances in lower dimensions? Because, remember, the PCA will recast the variables of the data so that the directions are ordered by how much variational information they hold. In the case of F, That is the information used to generate the distances (or similarities) between data points. By taking the PCs, we minimize the wasted information when mapping the distances in 2D or 3D.

Is there any way to connect PCA of the data Z and the PCoA of Z? Yes, there is a particular f we choose that gives the same results between PCA and PCoA. This means that PCA can be viewed as a particular case of PCoA.

Let’s work out the equivalence. If we consider the classical Euclidean distance, we can see the entries are made from 3 factors:

$$d(z_i, z_j)^2 = d_{ij}^2 = \sum_k z_{ik}^2 + \sum_k z_{jk}^2 - 2\sum_k z_{ik} z_{jk}. \tag{26}$$

The first two terms don’t hold any relationship between the points i and j, as they are particular to only i or j. All the information about the distance is held by the third term. Therefore, we can build a structurally equivalent similarity measure as:

$$s_{ij} = \sum_k z_{ik} z_{jk}. \tag{27}$$

Let's choose this similarity as our F, then note that $F = ZZ^T$ (remember that $C \propto Z^T Z$, similar, but not the same). For the PCA on F, we calculate the covariances of F, then find the eigenvalues/eigenvectors of the correlation matrix, then project F on the PCs' eigenvectors. But wait... shouldn't we standardize F? In this case, no, because F is not a data matrix, and it's already calculated from a standardised data Z.

What are the average values of F's columns?

$$\mu^F = \frac{1}{N}\sum_j f_{ij} = \frac{1}{N}\sum_j\sum_k z_{ik}z_{jk} = \sum_k z_{ik}\sum_j \frac{z_{jk}}{N} = \sum_k z_{ik}\mu_k = 0 \tag{28}$$

Here, μ_k is the average of column k of Z, that's why it's zero. Also, we can't change summation order at will, but in this case it's fine (k goes from zero to M and j goes from zero to N). Then, the covariance matrix of F, which we call C', has the same structure in relation to F as C has in relation to Z:

$$C' = \frac{(ZZ^T)^T(ZZ^T)}{N-1} = \frac{F^2}{N-1}, \tag{29}$$

note that F is symmetric, so $F^T = F$. Then, we need to find the eigenvalues/eigenvectors of F^2. This can actually be done from C:

$$Cv = \lambda v \tag{30}$$

$$Z^T Zv = (N-1)\lambda v \tag{31}$$

$$Z[Z^T Zv] = Z[(N-1)\lambda v] \tag{32}$$

$$F(Zv) = (N-1)\lambda(Zv) \tag{33}$$

We found the eigenvalues/eigenvectors of F in terms of the ones of C. Then, we can multiply by F on both sides and use this very fact:

$$F^2(zv) = (N-1)\lambda F(Zv) = [(N-1)\lambda]^2(Zv) \tag{34}$$

$$\frac{F^2}{N-1}(Zv) = (N-1)\lambda^2(Zv). \tag{35}$$

$$C'(Zv) = (N-1)\lambda^2(Zv) \tag{36}$$

Now we can notice something interesting. The matrix $Z^T Z$ has dimensions MxM, so it has M eigenvalues, while ZZ^T has dimensions NxN, with N eigenvalues. And we just saw that there is a direct correspondence between their nonzero eigenvalues. What happens to all additional eigenvalues the larger of them has? They must all be zero!

Finally, we just project F on Zv, knowing that the order of eigenvalues is the same as the one for C. To do that, we must normalise Zv and multiply F to it. The normalisation is:

$$\sqrt{\sum_i (\sum_k z_{ik} v_k)^2} = \sqrt{\lambda(N-1)}. \tag{37}$$

This is because inside the square root we have the $(N-1)$ times the variance of Zv (that is true because the average of Zv is zero, which holds because Z has zero average and Zv is just part of a change of basis). The variance of Zv is the variance of the transformed data, precisely λ, as we saw. Then, the projection becomes:

$$F' = \frac{FZv}{\sqrt{\lambda(N-1)}} = \frac{ZZ^T Zv}{\sqrt{\lambda(N-1)}} = \sqrt{\lambda(N-1)}Zv. \tag{38}$$

That is the equivalence: the PCA on Z is the same as the PCoA on F with each PC rescaled by $\sqrt{\lambda(N-1)}$. Since it's just rescaling, the structural information is the same.

5 A word on MDS methods

The idea of multidimensional scaling is itself a generalisation of PCoA. The task is the same, however, instead of just mapping through eigenvalues and eigenvectors of the covariances, it becomes a general optimization process. Thus, MDS can become a whole diversity of methods and algorithms (such as metric and non-metric methods, tSNE, uMAP, etc), depending on the theoretical and computational approach to probe the structure of distances/similarities between samples and extract a robust summary of it. The general idea is to minimize a loss function, which penalizes the method by how much the new distances diverge from the original ones. Finding the best result can also become a back-and-forth iterative process. Again, the PCoA can be viewed as a particular case of MDS, for a particular loss function. One example of loss function is the stress, which is the basic metric distance between the result and the original distance:

$$Stress = \sqrt{\sum_{i,j}\left(d_{ij} - \sqrt{\sum_{k}(z'_{ik} - z'_{jk})^2}\right)^2} \tag{39}$$

Here, z'_{ik} are the new data points, which are the result of the MDS, and d_{ij} is the input distance between the original data points in Z.