Introduction to Data Analysis 2

In this laboratory you will get a chance to examine a larger dataset (University of East Anglia's Climate Research Unit weather station dataset), quality control the output from that dataset, select a portion of that dataset based on a period of time (1990 to 2005) for further analysis and then perform a more advanced data analysis technique on this dataset (Principal Component Analysis). This will allow you to complete some of the more common tasks usually completed when working with larger datasets where more automation is required.

Principal Component Analysis

Principal Component Analysis (PCA) is a useful statistical technique for finding patterns in data of high dimension and has found many applications, such as facial recognition and image compression. The application of PCA can be described very simply as diagonalizing the covariance matrix, this process can be used to determine underlying correlations in a set of measured variables and to reduce a complex dataset to a lower dimension to reveal the potentially hidden, simplified structures that underlie that data. The goal of PCA could be described as the identification of the most meaningful basis to re-express a dataset. The hope being that this new basis will filter out the noise in measurements and reveal the underlying fundamental structure.

The steps needed to complete the PCA analysis are:

1, Organize the relevant data as an M $\times$ N data matrix **S**, which consists of M measurement types (voltage, temperature, humidity, luminosity) with N being the number of samples for each measurement.

2, Remove the mean value from each measurement type, such that we form a new vector **S'** which represents the anomalies (differences between the original values and the mean).

$$\boldsymbol{S'} = \boldsymbol{S} - \overline{\boldsymbol{S}}$$

3, Derive the covariance matrix ($\mathbf{C_S}$) of the matrix **S'** using:

$$\boldsymbol{C_S} = \frac{1}{N-1}\boldsymbol{S'^T S'}$$

4, Calculate an eigenvector decomposition of the covariance matrix.

This process which effectively diagonalizes the covariance matrix means that the different dimensions (basis functions) making up the matrix become orthogonal which mean that the PCA represents an "optimal" representation of the data. For example, suppose that we have chosen the first basis function to represent the spatial structure of our data, and that we now wish to make the best possible choice of a second basis function. Clearly the worst possible choice would be to make the second basis function the same as the first, because in that case

the second function would contribute no additional information beyond what was already available in the first. This suggests that the second basis function should be "as different as possible" from the first; more precisely, the second function should be spatially uncorrelated with the first, and this is equivalent to the requirement of spatial orthogonality. Extending this reasoning, it is clear that a set of basis functions should be chosen so that each is spatially orthogonal to each of the others.

More details on the PCA scheme are detailed in Shiens et al. (2009)

An atmospheric science application

Empirical orthogonal function (EOF) analysis is effectively a commonly used application of PCA used in atmospheric science. This technique is applied to a group of time series data and was originally developed by Lorenz (1956). Given any space-time meteorological field, EOF analysis finds a set of orthogonal spatial patterns along with a set of associated uncorrelated time series (or principal components). The original purpose of EOF analysis was to reduce the large number of variables of the original data to a few variables, while still explaining the majority of the variance. More recently, EOF analysis has been used to extract individual modes of variability known as teleconnections, such as El Nino Southern Oscillation. Effectively finding unobserved statistical patterns. By truncating the set of EOFs used it can also be used as a way to remove high frequency noise in datasets by reconstructing the data from a subset of the EOF patterns identified. More details are available in Hannachi et al. (2007).

The EOF technique decomposes a space-time field into a set of spatial patterns of variability, their time variation and provides a measure of the 'importance' of each pattern. The methodology used to complete the truncation task is exactly the same as the PCA analysis previously detailed. Note that gridded climate data normally come as a three-dimensional datasets, two-dimensional in space and one-dimensional in time. It is possible to complete the mathematics in 3D or higher, but much simpler to complete the mathematical operations on a 2-D matrix and thus we transform the date into a 2-D matrix, X. For example, the dataset might have been gridded temperatures at a range of latitudes and longitudes. By flattening this dimension to a set of particular stations we would create the 2D matrix (one spatial and one temporal dimension). For a 2D matrix X_{mn} which might represents M measurements at different locations in a spatial pattern each measured N times we may write:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1M} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2M} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3M} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{NM} \end{bmatrix}$$

Given that the aim of the analysis is to examine variations, we examine the anomalies of the observations, *X'*. So we must remove the time average at each specific location from the dataset. We then form the covariance matrix from the anomalies, which can be written as:

$$C = \frac{1}{N-1} X'^T X'$$

Note that superscript T identifies the transpose of the anomaly matrix. The aim of the analysis is to find the linear combination of all the variables, i.e. grid points, that explains maximum variance. That is to find a direction $a = (a_1, a_2 \ldots, a_p)^T$ such that the product of the anomaly matrix, *X'*, and the direction, *a*, has maximum variability. Now the variance of the (centered) time series *X'a* is given by:

$$var(X'a) = a^T C a$$

To maximize the variance associated with the product *X'a* we solve the eigenvalue problem which can then be written as:

$$Ca = a\Lambda$$

This is the traditional eigenvalue problem which involves finding the eigenvalues Λ and the eigenvectors *a* that are non-trivial solutions of the above equation. Note that by definition the covariance matrix C is symmetrical and therefore diagonalisable. The kth EOF is simply the kth eigenvector a_k of C after the eigenvalues, and the corresponding eigenvectors, have been sorted in decreasing order. The covariance matrix is also semi-definite, hence all its eigenvalues are positive. The eigenvalue Λ_k corresponding to the kth EOF gives a measure of the explained variance by a_k. The explained variance in percentage terms can be written as:

$$= \frac{\Lambda_k}{\sum_{k=1}^{M} \Lambda_k} \times 100$$

The resultant data can be used to identify the most important patterns, identify the variance related to them, and also potentially select unimportant patterns. The latter possibility can allow us to identify a sensible truncation point based on ensuring a certain portion of the variance associated with the initial dataset is kept.

We may write $X_m(t_i)$ in an alternative form:

$$X_m(t_i) = \sum_{k=1}^{M} Y_{km} Q_k(t_i)$$

where Y_{km} are unknown time-independent basis functions, these are the EOFS, and $Q_k(t_i)$ are the time dependent coefficients called principal components, PCs. The number of Y's is equal to M, because the spatial information is contained in the Y's and the number of elements in $Q_k(t_i)$ is N. If the sum in the equation above is taken over all of the Y's, then we recover the input field, with no loss of information.

However, as one usage of EOF analysis is to reduce the large number of variables of the original data to a few variables, we can truncate the series:

$$p_m^K(t_i) = \sum_{k=1}^{K} Y_{km} Q_k(t_i) + r_m^K(t_i)$$

where K<M, and r_m^K is the error associated with the truncation.

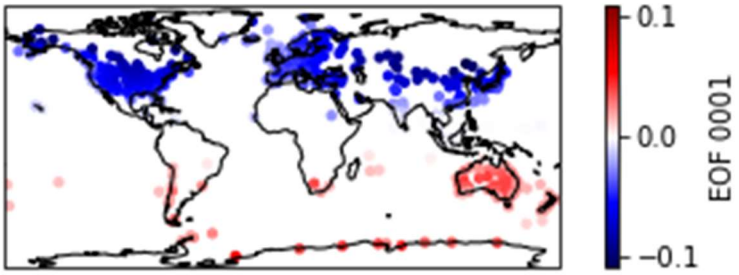
Task 1: Apply PCA to an artificial dataset specified by the code in PCA1.py. In this example, you should make a function or functions that complete the Principal Component Analysis. Note it is important to make this code using functions because this will allow you to reuse this code easily in later Tasks.

Task 2: The code read_station_data_example_code.py (available on the LEARN site) reads in a sample station dataset in the standard CRU format. Use the read functions in that code and the glob command (you will need to use the glob package) to write code that reads in ALL the CRU weather station data included in the Station_data.zip file. Note that there are at over 1000 individual files representing monthly temperature data from individual stations.

Task 3: Examine the various time series associated with the different weather station records to find all those records which display valid measurements between January 1990 and December 2005. Note this condition will only be true for a portion of the weather station datasets. For the records/stations that meet this criteria, you should concatenate that portion of those stations temperature records into a 2D matrix, remembering to save corresponding metadata (station number, latitude, longitude) in separate arrays. You should then write the data to a file for later usage. You might want to use the *numpy.argwhere* command to find relevant indices and also ensure that there are no missing data values in each temperature array subset (missing values are specified by a -99.0 value in the CRU datasets).

NOTE WITH LARGE DATASETS OR DATASETS LIKE THIS ONE WHICH CONTAIN A LARGE NUMBER OF FILES YOU WILL WANT TO TEST YOUR CODE ON SUBSET OF THE DATA. THEN ONCE YOUR CODE WORKS RUN IT ONCE OVER THE ENTIRE DATASET TO GET THE FINAL SINGLE DATA FILE.

Task 4: Read in the quality controlled weather station data file and perform PCA on the dataset. Produce a map displaying the various values of the 1st EOF across the Earth and also the corresponding time series. This might look like the figure below. You might want to use the matplotlib function *plt.scatter* and the command *ax1=plt.subplot(2,1,1,projection=ccrs.PlateCarree())* where *ccrs* is the imported using the command *import cartopy.crs as ccrs.*



Task 5: Reconstruct a filtered time series for the first weather station that passes the quality control in the dataset by retaining only the first three EOFs. Compare this time series with the input time series in the same figure.

References:

Hannachi, A., Jolliffe, I. T., & Stephenson, D. B. (2007). Empirical orthogonal functions and related techniques in atmospheric science: A review. International Journal of Climatology, 27(9), 1119-1152. https://doi.org/https://doi.org/10.1002/joc.1499.

Lorenz, 1956, Empirical orthogonal functions and statistical weather prediction. Technical report, Statistical Forecast Project Report 1, Dept. of Meteorology., MIT.

Shiens, J., 2009, A Tutorial on Principal Component Analysis, http://arxiv.org/pdf/1404.1100.pdf