M1 Master's Degree in Cognitive Science 2024–2025 | November 14th
TA: Hugo Ninou (hugo.ninou@ens.fr)

MATHEMATICAL TOOLS FOR DATA SCIENCE, MACHINE LEARNING AND STATISTICAL MODELLING

TD2: DIMENSIONALITY REDUCTION

Click here to access the notebook for this tutorial.

You should copy the notebook from Google colab (go to file->save a copy in drive) in order to be able to save your changes.

1 Toy example : A spring in a black box

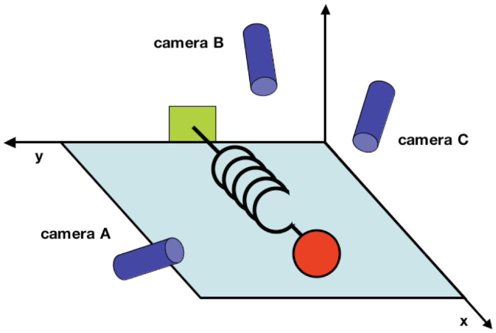

Figure 1: Illustration of the spring in a black box (from *A tutorial on Principal Component Analysis*, 2014, Jonathon Shlens)

Let's consider a toy example of a mass attached to a spring. The spring is frictionless and the mass is released at a small distance from its equilibrium point. This is the physicists ideal spring and the mass will oscillate indefinitely along the x-axis. Let's now imagine that this system is put into a black box and that as a naive experimenter ignorant about physics, you are trying to figure out what's happening in that box. You may choose to put three cameras at different positions in the box and record the movement of the mass. The results are shown in fig.2. Of course, ideally we should have used only one camera and measure the mass' position along the x-axis. But we didn't know that *a priori*. Additionally, we have to deal with noisy data : this could be due to imperfect cameras, an imperfect spring or frictions.
The cameras' data is stored in the matrix X.

$$X = \begin{bmatrix} x_1^0 & ... & x_1^T \\ y_1^0 & ... & y_1^T \\ x_2^0 & ... & x_2^T \\ y_2^0 & ... & y_2^T \\ x_3^0 & ... & x_3^T \\ y_3^0 & ... & y_3^T \end{bmatrix} \tag{1}$$

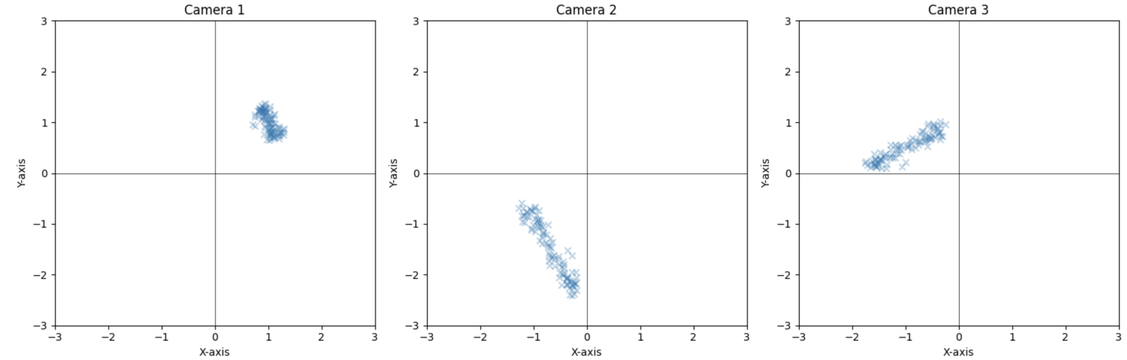

Figure 2: Camera recordings of the mass' movement

1. What is the intrinsic dimensionality of the mass' position ? What is the dimensionality of the data stored in X ?

   We would like to reduce the dimensionality of the data to reflect the intrinsic dimensionality of the system's movement. In other words, we would like to change bases to best re-express our data. To that end, we will perform a principal component analysis (PCA).

2. What kind of basis results from implementing a PCA ?

3. What is the covariance matrix C of our data ? What size is it ?

4. Find the eigenvalues and eigenvectors of C using an eigenvector decomposition thanks to the function "np.linalg.eigh".

5. Plot the eigenvalues and the projection of the data in X along the first two principal components. What is the problem here ?

6. Solve the previously found issue, and plot the correct eigenvalues and projection of X on the first two PCs.

7. Compare the variance of the data projected on the first PC with the first eigenvalue of the correct correlation matrix.

8. Same question for the second PC and second eigenvalue.

9. Compute the singular value decomposition (SVD) of the data. What is the expression linking the eigenvalues of C and the singular values σ_i ?

10. Show a scatter plot of the first two rows of $U^T X$ and ΣV^T, explain what you see.

2 Performing PCA on neural data

A Monkey was trained to perform a center-out reaching task (fig.3). The monkey is holding a controller, during the task it is first asked to maintain the controller in a central position. The monkey is then cued one of the eight targets arranged circularly around the central position. During this "CUE" period, the monkey has to keep maintaining the controller in the central position. It is only when the monkey is given the "GO" signal that it should move the controller to the cued target. Neural recordings were collected during this task (courtesy of Yifat Prut's lab) resulting in a total of 777 recorded neurons. This data contains for each neuron a matrix of size (#trials × #time steps) that has for value 1 if the neuron was firing and 0 otherwise. The data for the first neuron is shown in the notebook in the form of a raster plot.

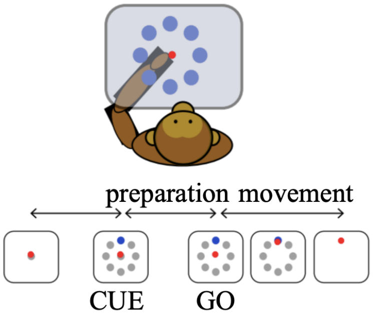

Figure 3: Center-out reaching task

1. Based on your knowledge of PCA, why would you say that using binary data is not ideal for implementing a PCA ? covariance

   Instead of binary data, we are rather going to use smoothened data where every spike is replaced by a Gaussian with parameter $\sigma = 100$ms.

2. Show the eight raster plots of the first neuron by grouping the trials per cued target (each raster plot should correspond to a given cued target). You may use the function "raster_plot()".

3. There are 271 trials for the first neuron, but each neuron was recorded a different amount of times. Store the number of trials for each neuron in the list "nb_trials"

4. Store the number of trials per neuron for each of the 8 targets in the matrix "nb_trials_per_c".

   We are now going to try implementing a naive PCA on the data we have. We would like to define a matrix X containing all of our data and look at the eigenvectors of XX^T. However we do not have the same number of trials for each neuron, the data cannot fit in a matrix, how should we proceed then ? One solution is to only select neurons that have more than 100 trials and select only 100 trials for each neuron. This way we would end up with a matrix X of size $neurons \times (100 \times nb\_time\_frames)$.

5. Build the matrix X by only selecting neurons with at least 100 trials.
   Hint: The resulting matrix should be a numpy array of size (652,400100)

6. Perform a PCA on this matrix X and store the first three PCs in "PC1" "PC2" and "PC3". Project the data on the first three PCs, what can you tell from this plot ? What source of variance does this PCA seem to capture ? not cues variance, variance trials, through time

   The targets are not cued in the same order going from one neuron to another. Due to the cued targets ordering being shuffled from a neuron to another, the correlation matrix XX^T could not catch the variance across neurons related to a specific target being cued. Our goal in the following will be to reorganize the data so that trials with the same cued target are grouped together.

7. Build a filter for neurons that have at least 12 trials for each of the cued targets. The filter should be an array of size (nb_neurons) and its element in position i should read True if neuron i has at least 12 trials for each of the 8 cued targets and False otherwise. How many neurons *nb_valid_neurons* satisfy this condition ?

8. Now build a matrix X_{targ} of size $nb\_valid\_neurons \times (8 \times 12 \times 4001)$ that for each valid neuron contains the first 12 trials for each target. The data corresponding to a given target should be located at the same place for each neuron.

2D vs 3d (more simple)

9. Compute the PCA for X_{targ}, plot the cumulative explained variance and plot the data along the first 3 PCs. How is this plot different from the first PCA ? What source of variance is now being captured by the covariance matrix which was not captured before ?

   Cues variance, trial variance through time

   We do not fully manage to recover axes that capture the variance linked to the different cued targets. This is due to other sources of variance being too strong relative to those we're interested in. Here are the three sources of variance in our dataset : variance through time, inter-trial variance, and variance through cued targets.

   We would like our PCA to only pick up the variance through cued targets. In order to do so, we will look at a time window going from -50 to 0 ms relative to the GO signal, i.e. from timebin 1950 to 2000. Additionally, we will average the data over the trials that share the same cued target to reduce the inter-trial variance.

10. Why does averaging over the trials would reduce the inter-trial variance ? What theorem seen in the class does this refer to ?

    lln : cancelling noise (reducing noise you reduce the
    vazriance trials (large part of variance comes from noise)

11. Build a matrix X_s in the same way as X_{targ} but with the requirements mentioned above, perform its PCA and project the data on the first 3 PCs. What can you tell from this plot ? Does the first PCs seem to explain the variance linked to the cued targets ?

    small timesteps intervalle

    PC allow to distinguish btw cues, but not the variance intra cues (trials bc averaged)

12. Project the data from the same time window without averaging over trials X_{st} on the principal components computed for X_s. What can you tell from this plot ?

    plein de petits traits: parce qu'on n'a pas averages over trials!

13. Plot X_{st} with a time window going from [-1sec;1sec] relative to go, i.e. 1000 to 3000 time bin, without averaging over the trials.

14. How would you further improve this data representation ? Knowing that different neurons tend to have different intrinsic variances in their firing rate, what modification to this procedure would you do ?