

PCA: batch preprocessing and online-PCA

3.1 Preprocessing (1 point)

- (a) Load the dataset `pca2.csv`. Compute the Principal Components PC1 and PC2 and plot the data in the coordinate system PC1 vs. PC2 – What do you observe?
- (b) Remove Observations 17 and 157 and redo the first two steps. What is the difference?

3.2 Whitening (3 points)

- (a) Load the dataset `pca4.csv` and check for outliers in the individual variables.
- (b) Do PCA on a reasonable subset of this data. Use a scree plot to determine how many PCs represent the data well.
- (c) “Whiten” the data, i.e. create a set of 4 *uncorrelated* variables with *mean 0* and *standard deviation equal to 1*. This can be done e.g. using the transformation

$$\mathbf{Z} = \tilde{\mathbf{X}}\mathbf{E}\mathbf{\Lambda}^{-1/2}.$$

The new variables z_i form the columns of \mathbf{Z} , \mathbf{E} is a matrix containing in its columns the normalized eigenvectors of the covariance matrix \mathbf{C} of the centered data $\tilde{\mathbf{X}}$ (variables column-wise) and $\mathbf{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues.

- (d) Make 3 heat plots of the (i) 4x4 covariance matrix \mathbf{C} , (ii) the covariance matrix of the data projected onto PC1-PC4, and (iii) of the whitened variables.

3.3 Oja's Rule: Derivation (2 points)

Consider a linear connectionist neuron whose output $y = y(t)$ at time t is an inner product of the N -dim input vector $\mathbf{x} = \mathbf{x}(t)$ with the N -dim weight vector \mathbf{w} :

$$y = \mathbf{w}^T \mathbf{x}.$$

The Hebbian update rule for learning the weights can be written as

$$w_i(t+1) = w_i(t) + \varepsilon y(t) x_i(t), \quad i = 1, 2, \dots, N$$

where ε is the learning-rate parameter and t the iteration step. As was shown in the lecture, the Hebbian learning rule leads to a divergence of the length of the weight vector. Therefore, the following normalization was introduced by Oja:

$$w_i(t+1) = \frac{w_i(t) + \varepsilon y(t) x_i(t)}{\left(\sum_{j=1}^N [w_j(t) + \varepsilon y(t) x_j(t)]^2 \right)^{\frac{1}{2}}}$$

Task: Derive an approximation to this update rule for a small value of the learning-rate parameter ε by Taylor-expanding the right hand side of this equation with respect to ε around $\varepsilon = 0$. Show that neglecting terms of second or higher order in ε gives *Oja's rule*:

$$w_i(t+1) = w_i(t) + \varepsilon y(t) [x_i(t) - y(t) w_i(t)].$$

3.4 Oja's Rule: Application (4 points)

The file `data-onlinePCA.txt` contains observations from an artificial experiment run over an interval of time (i.e. the first datapoint was observed at $t_0 = 0$ and the last at $t_N = 10s$).

1. Make a scatter plot of the data and indicate the time index by the color of the datapoints (you can e.g. break the full dataset into 10 blocks corresponding to 1 second length each and therefore use 10 different colors).
2. Determine the principal components (using batch PCA) and plot the first PC (e.g. as an arrow or the endpoint of it) for each of the 10 blocks in the same plot as the original data – together with the original data.
3. Implement Oja's rule and apply it with a learning-rate parameter $\varepsilon \in \{0.002, 0.04, 0.45\}$ to the dataset. In each iteration ~~randomly sample one~~ **take the next**¹ data point and apply the learning step. Plot the weights at each timestep (as points whose x vs. y coordinates are given by the weight for x and y) in the same plot as the original data (use the colors from 1. to indicate the time index for each plotted weight). Interpret your results.

Total points: 10

¹In contrast to the algorithm of the lecture where datapoints are sampled randomly here we do not want to lose the temporal relation between the data points and therefore process the data in the original order.