PCA: data compression and online learning

This problem sheet explores applications of batch and online PCA. The first exercise shows how (batch) PCA can be used for compressing data. The second exercise demonstrates how the first principal direction can be found with a simple iterative algorithm (Oja's Rule) and the third exercise applies it to online-learning.

3.1 Data compression (3 points)

Choose a portion of an arbitrary image from the image database (`imgpca.zip` – used also on the previous exercise sheet) and reconstruct it using only the first n PCs for $n \in \{1, 2, 4, 8, 16, 100\}$. To this end,

(a) take a subportion of e.g. 160×320 pixels of the chosen image and partition it into 10×20 tiles.

(b) Calculate the PCs for this small dataset

(c) Reconstruct the tiles using only the first n components

(d) After reconstructing the tiles, stitch them back together and plot the reconstructed image. Compute the squared error between the reconstruction and the original image.

(e) Using the PCs from this image, reconstruct an image region of the same size from a different picture.

3.2 Oja's Rule: Derivation (3 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

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

where η 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:

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

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

$$w_i(t+1) = w_i(t) + \eta y(t) \left[x_i(t) - y(t) w_i(t) \right].$$

3.3 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) in the same plot as the original data.
3. Implement Oja's rule and apply it with a learning-rate parameter $\eta \in \{0.002, 0.04, 0.45\}$ to the dataset. 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