Exercise Sheet 7 - Matrix decomposition
Prof. Dr. Paul Swoboda

NOTE: For this exercise sheet, you must submit a PDF and two jupyter notebooks.

Exercise 1: Non-negative Matrix Factorization (4 Points)

For this exercise, we aim to implement non-negative matrix factorization. Our dataset consists of 19x19 face images that we want to decompose. To do this, we'll rewrite our feature tensor, denoted as $A \in \mathbb{R}^{N\times M}$, which contains N vectorized images.

$$A_{ij} \approx (WH)_{ij}$$

$W \in \mathbb{R}_{+}^{N\times R}$ and $H \in \mathbb{R}_{+}^{R\times M}$ are non-negative decomposition matrices and R is the rank chosen by the user. Our loss function we want to minimize is given by

$$L(W, H) = ||A - W \cdot H||^2$$

where the $||X||^2 = \sum_{ij} X_{ij}^2$. The optimization algorithm is going to be

$$\begin{aligned} W^{t+1} &\leftarrow [W^t - \alpha \nabla_W L(W^t, H^t)]_+ \\ H^{t+1} &\leftarrow [H^t - \alpha \nabla_H L(W^t, H^t)]_+ \end{aligned}$$

where $\alpha > 0$ is the step size for the gradient descent and $[X]_+ = \max(X, 0)$ is the pointwise projection onto the set of nonnegative numbers. Derive the optimization steps by calculating the gradient for W and H (2 points for deriving by hand & 2 points for correct implementation).

Download the jupyter notebook on Ilias and implement the optimization process. Update the matrices iteratively while ensuring they remain non-negative.

Exercise 2: Moore-Penrose Pseudo-Inverse (1 Point)

Download the housing prices dataset provided on Ilias along with the corresponding jupyter notebook. Solve the regression problem using the Moore–Penrose pseudo-inverse.

a) Use numpy.linalg.pinv() to directly compute the pseudo-inverse.

b) Use numpy.linalg.svd() to compute the singular value decomposition. Derive the pseudo-inverse from the SVD result and compare it to part a).