Non-Negative Matrix Factorization

Algorithm

Inputs: nonnegative $m \times n$ matrix V, r output dimension
Outputs: non-negative matrices W, H of sizes $m \times r$ and $r \times n$, such that $V = WH$

Steps:

1. Initialize W and H (various methods exist, see references).
2. Update the matrices with the following steps until convergence:

$$H_{[i,j]}^{n+1} = H_{[i,j]}^{n} \frac{((W^n)^T V)_{[i,j]}}{((W^n)^T W^n H^n)_{[i,j]}}$$

$$W_{[i,j]}^{n+1} = W_{[i,j]}^{n} \frac{(V(H^{n+1})^T)_{[i,j]}}{(W^n H^{n+1} (H^{n+1})^T)_{[i,j]}}$$

Implementation

Baseline 1: a straightforward implementation with loops

Baseline 2: replacing the loops with BLAS library calls

Optimized version: The same matrices are used in multiple steps of the update rules, re-ordering, tiling and unrolling the computation can result in much better data re-use than what is possible with the BLAS interface.

References:
https://en.wikipedia.org/wiki/Non-negative_matrix_factorization#Algorithms
https://www.cs.cornell.edu/~bindel/class/sjtu-summer19/lec/2019-05-30.pdf
https://arxiv.org/pdf/2109.03874v1.pdf