Non negative Matrix factorization

We have a data matrix X. It contains non-negative entries. Also X has n rows and m columns. We can decompose X into matrices W and H whose shapes are $n \times k$ and $k \times m$ respectively. Thus, $X_{ij} \approx \sum_k W_{ik} H_{kj}$.

Our goal is to determine the decompositions W and H. One way to do so is by minimizing the divergence penalty[1].

$$D(X||WH) = \sum_{i,j} X_{ij} \log \frac{X_{ij}}{(WV)_{ij}} - X_{ij} + (WV)_{ij}$$

We solve the optimization problem by performing multiplicative updates as opposed to additive updates. The update rules are:

$$W_{ik} \leftarrow W_{ik} \frac{\sum_j H_{kj} X_{ij} / (WH)_{ij}}{\sum_j H_{kj}}$$

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i W_{ik} X_{ij} / (WH)_{ij}}{\sum_i W_{ik}}$$

We perform the updates until the divergence is small enough.

Important note: When performing the updates, 0 / 0 divisions may occur in some cases. To avoid complications, we can add a small number to the denominator.

I have ran the code on data from New York Times. Each line in the csv file corresponds to a single document. It gives information about the index of a word and the number of times the word occurs in that document. It is written in the format **index : count**.

[1] https://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf