D5: Databases and Information Systems
Data Mining and Matrices, SS 2015
Programming assignment #2: NMF and CX
Due: **12 July 2015** at 23:29 CEST

max planck institut
informatik

*You must hand in machine-typed report in PDF format and a script file (e.g.* `*.R` *file). The report must explain your approach to the problem, the results you obtained, and your interpretation of the results. Naturally, the report must also answer to any direct question presented in the problems. You can, and in many cases should, add plots and other illustrations to your answers. The script file must show every step you have taken to solve these tasks, that is to say, if we run the script file we must get the same results you reported and see the same figures you presented. You can discuss these problems with other students, and you are encouraged to discuss with the tutor, but everybody must hand in their own answers and own code. Return your answers by email to* `dmm15@mpi-inf.mpg.de`*. Remember to write your name and matriculation number to every answer sheet!*

## Task 1: ALS vs. multiplicative NMF

Download the data and utility files from `http://resources.mpi-inf.mpg.de/departments/d5/teaching/ss15/dmm/assignments/assignment_2.zip`. That package contains file `assignment2.R`. You can fill your answers to that file and return it as a part of your solution.

Your first task is to implement alternating least squares based NMF algorithm and Lee and Seung's multiplicative NMF algorithm. For the former, you can truncate negative values to zero. Your implementations should be reasonably efficient.

Load the `news` data. It consists of a subset of the 20-newsgroups dataset (`http://qwone.com/~jason/20Newsgroups/`). The subset contains 100 news articles from each of the following Usenet groups: `sci.crypt`, `sci.med`, `sci.space`, and `soc.religion.christian`. Terms have been stemmed, stop words removed, and only the 800 most popular terms have been retained. The data is given in form of an $400 \times 800$ document-term matrix; entry $(d, w)$ denotes the term frequency (tf) of word $w$ in document $d$.

Run the two NMF algorithms on the `news` data. Compare the reconstruction errors and convergence rates. Notice that any two runs of the algorithm might result to very different outcomes, depending on the initial $\boldsymbol{W}$ and $\boldsymbol{H}$. Also, the default 300 iterations might not be enough (or it might be too much) for the methods to converge. Play around with the number of re-starts and iterations. Based on your experiments, which one of the two methods you consider better for this data and why?

## Task 2: Analysing the data

In this task we try to analyse the `news` data. Before proceeding further, normalize the data such that it sums to unity. Then use one of the methods you implemented in the first task to find $k = 4$ NMF of the data and study the top-10 terms of the right factor matrix $\boldsymbol{H}$. Can you infer some "topics" based on these terms? Recall that the terms are stemmed,. The topics can be very broad (e.g. "terms associated with sports") and they might not be the ones of the newsgroups. Also, some factors might not correspond to any sensible topic. Argue why (or why not) you think the factors correspond to the topics you claim they do.

Repeat the analysis with $k = 2, 4, 8, 16, 32$. How do the results change with increased $k$? Can you name the single best rank for this data?

Repeat the analysis, but this time using the generalized K–L divergence optimizing version of NMF (provided in `utils.R`). Do the results change? Are they better or worse? Is different $k$ better with K–L divergence than with Euclidean distance?

# Task 3: Clustering and pLSA

In this task, we study the use of pLSA's as a dimensionality reduction tool, and compare it to Karhunen–Lóeve transformation. We use the (normalized) `news` data. The documents of the data came from 4 newsgroups. Your task is to cluster the documents in such a way that the clusters correspond to the newsgroups. To evaluate the quality of the clustering, we use normalized mutual information (NMI).[1] NMI takes values from $[0, 1]$ and obtains value 0 for perfect match. Notice that NMI does not care about cluster labels or the ordering of the clusters. You can use the provided function `nmi.news` to evaluate a clustering (see provided example in `assignment2.R`).

To compute the pLSA, first compute the K–L divergence optimizing NMF (of the normalized data), and then use the provided script to normalize $\boldsymbol{W}$ and $\boldsymbol{H}$ to obtain decomposition of type $\boldsymbol{W}'\boldsymbol{\Sigma}\boldsymbol{H}'$. Use also another script to compute normalization of type $\boldsymbol{\Sigma}\boldsymbol{W}'\boldsymbol{H}'$, where $\boldsymbol{\Sigma}$ is $n$-by-$n$ diagonal.

Cluster the normalized newsgroup data into 4 clusters using each of the methods below and compute the NMI. Try different ranks for the matrix factorizations. Which clustering(s) perform well, which do not? Why?

1. $k$-means,
2. $k$-means on the first $k$ principal components (Karhunen–Lóeve transform, similar to Task 3 of Assignment 1),
3. $k$-means on the $\boldsymbol{W}$ matrix of the NMF (using K–L divergence),
4. $k$-means on the $\boldsymbol{W}'$ matrix of factorization $\boldsymbol{W}'\boldsymbol{\Sigma}\boldsymbol{H}'$ obtained from the NMF, and
5. $k$-means on the $\boldsymbol{W}'$ matrix of factorization $\boldsymbol{\Sigma}\boldsymbol{W}'\boldsymbol{H}'$ obtained from the NMF.

# Task 4: CX decomposition

In this task you implement the two-phase CX and apply it to the `worldclim` data we used in the first assignment.

First, you have to implement the two-phase CX algorithm (slide 14 from 2015-06-09; you can also see Boutsidis et al. 2008 or Boutsidis et al. 2010 for more information). For the RRQR algorithm, you can use the `qr` algorithm from R that computes the (pivoted) QR decomposition (see `assignment2.R`).

We apply the CX decomposition to the European climate data we used in the first assignment. Load the data and normalize it to $z$-scores. The columns of the data are the climate variables, but instead we want to select some locations to $\boldsymbol{C}$. To that end, apply the CX decomposition to the transpose of the climate data. How would you describe the columns of $\boldsymbol{C}$? How about rows of $\boldsymbol{X}$?

The assignment file shows few ideas on how to visualize the decomposition. Based on these, do the results make sense? Can you interpret them? Try different values of $k$ and sample different number of columns at the sampling phase. How do the results change? Does it help to sample more columns (w.r.t. the final number of columns)? Overall, is CX decomposition a good match for this data? Argue!

[1] Strictly speaking, we are using the normalized metric variant $D(X, Y)$, `http://en.wikipedia.org/wiki/Mutual_information#Metric`