
The NMF Matlab Toolbox: A Short Tutorial

Yifeng Li*

School of Computer Science
University of Windsor
Windsor, Ontario Canada N9B 3P4
li11112c@uwindsor.ca
yifeng.li.cn@gmail.com

Alioune Ngom

School of Computer Science
University of Windsor
Windsor, Ontario Canada N9B 3P4
angom@cs.uwindsor.ca

Abstract

This is a short tutorial to show how to use the NMF Matlab Toolbox.

1 Installation and Version Information

Simply download the toolbox from the toolbox official website <http://cs.uwindsor.ca/~li11112c/nmf>, and unzip it. After that, add the path of this toolbox into your Matlab using the following commands:

```
addpath(YourNMFToolBoxPath);  
savepath
```

Or use the Set Path dialog.

The current version of this toolbox **v.1.1**. December 30, 2011.

2 Requirement

MATLAB MATLAB 7.9 or later versions;

libSVM The libSVM of MATLAB version, can be find at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. This library is only necessary when support vector machine (SVM) classifier is utilized.

3 Citation

If you are using this toolbox, please use the following citation in your publication to acknowledge our work:

Yifeng Li and Alioune Ngom, "The Non-Negative Matrix Factorization MATLAB Toolbox for High Dimensional Biological Data," School of Computer Science, University of Windsor, Technical Report No. 11-060, May 2011. Available at: <http://cs.uwindsor.ca/~li11112c/nmf>.

4 Algorithm

The toolbox provides NMF optimization algorithms:

nmfnls The non-negative least squares (NNLS) based standard NMF algorithm;

nmfrule The multiple update rules (MUR) based standard algorithm using the Euclidean distance or the KL divergence;

*Yifeng Li's web page is <http://cs.uwindsor.ca/~li11112c>

sparsenmfnnls The NNLS based and Sparse-NMF algorithm;
seminmfnnls The NNLS based Semi-NMF algorithm;
seminmfrule The MUR based Semi-NMF algorithm;
convexnmfrule The MUR based Convex-NMF algorithm;
orthnmfrule The MUR based Orthogonal-NMF algorithm;
kernelseminmfnnls The kernel semi-NMF algorithm based on NNLS optimization;
kernelseminmfrule The kernel semi-NMF algorithm based on update-rule optimization;
kernelnmfdecom The kernel NMF algorithm based on decomposing kernel matrices;
kernelconvexnmf The kernel convex-NMF algorithm;
wnmfrule The MUR based Weighted-NMF algorithm for matrix containing missing values;
nmf The omnibus of the above algorithms.

Please type `help nmfnnls` (or the other function names) in the MATLAB command line for function descriptions.

5 Clustering

Based on the algorithms listed in Section 4, the toolbox provides NMF based clustering method:

NMFCluster Take the coefficient matrix produced by a NMF algorithm, and output the clustering result;
chooseBestk Search the best number of clusters based on dispersion Coefficients.

Please type `help NMFCluster` (or the other function names) in the MATLAB command line for function descriptions. An example named `exampleCluster.m` is provided to learn how to use these functions. The data ALLAML being used is in the `data` subfolder of the toolbox.

6 Biclustering

Based on the algorithms listed in Section 4, the toolbox provides NMF based biclustering method:

biCluster The biclustering method using one of the NMF algorithms;
NMFBicHeatMap Draw the heatmap of the resulting matrices, the generated figures is saved in folder `savedHeatMap` under the current folder by a specific format.

Please type `help biCluster` (or the other function names) in the MATLAB command line for function descriptions. An example named `exampleBiCluster.m` is provided to learn how to use these functions.

7 Feature Extraction

Based on the algorithms listed in Section 4, the toolbox provides NMF based feature extraction method:

nmfnnltest Map the test/unknown data into the feature space generated by `nmfnnls`;
sparsenmfnnls Map the test/unknown data into the feature space generated by `sparsenmfnnls`;
seminmfnnltest Map the test/unknown data into the feature space generated by `seminmfnnls`;
seminmfruletest Map the test/unknown data into the feature space generated by `seminmfrule`;

convexnmfruletest Map the test/unknown data into the feature space generated by `convexnmfrule`;

kernelnmfruletest Map the test/unknown data into the feature space generated by `kernelnmfrule`;

orthnmfruletest Map the test/unknown data into the feature space generated by `orthnmfrule`;

featureExtractionTrain General interface. Using training data, generate the bases of the NMF feature space;

featureExtractionTest General interface. Map the test/unknown data into the feature space generated using `featureExtractionTrain`;

Please type `help featureExtractionTrain` (or the other function names) in the MATLAB command line for function descriptions. An example named `exampleFeatureExtraction.m` is provided to learn how to use these functions.

8 Feature Selection

Based on the algorithms listed in Section 4, the toolbox provides NMF based feature selection method:

featureFilterNMF On training data, select features by various NMF.

Please type `help featureFilterNMF` in the MATLAB command line for function descriptions. An example named `exampleFeatureSelection.m` is provided to learn how to use this function.

9 Classification

Based on the algorithms listed in Section 4, the toolbox provides NMF based feature extraction method:

nnlsClassifier The NNLS classifier;

bootstrapnnlsClassifier The Bootstrap based NNLS (BNNLS) classifier;

nmfClassifier The NMF classifier;

repetitivenmfClassifier The repetitive NMF (RNMF) classifier;

classification The classifier interface including several classifiers;

multiClassifiers Run multiple classifiers on the same training data;

changeClassLabels01 Change the class labels to be in $\{0, 1, 2, \dots, c - 1\}$, where c is the number of unique class labels;

perform Evaluate the classifier performance;

Please type `help nnlsClassifier` (or the other function names) in the MATLAB command line for function descriptions. An example named `exampleClassification.m` is provided to learn how to use these functions. We refer interested users to our Sparse Representation Toolbox in MATLAB for more powerful NNLS classifiers.

10 Overcome Missing Values

In order to disregard the missing values in a matrix, the toolbox also provides Weighted-NMF:

wnmfrule The MUR based Weighted-NMF algorithm;

Please type `help wnmfrule` in the Matlab command line for function descriptions. An example named `exampleClusterWeighted.m` is provided to learn how to use this function.

11 Miscellaneous

The other miscellaneous functions:

matrixNorm Calculate the Frobenius matrix norm;

mergeOption Merge two option structs;

normmean0std1 Normalization to have mean 0 and STD 1;

num2cellstr Convert a numerical matrix into a matrix of string cells;

sparsity Calculate the sparsity of a matrix;

computeKernelMatrix Compute the kernel matrix $k(A,B)$ given a kernel function;

Please type, for example, `help matrixNorm` in the MATLAB command line for function descriptions. These functions are called by the functions in the previous sections.