MCRLLM

Multivariate Curve Resolution by Log-Likelihood Maximization

Contact info: Ryan Gosselin, Université de Sherbrooke, ryan.gosselin@usherbrooke.ca

Available at: pypi.org
Updated: 2020 04 15

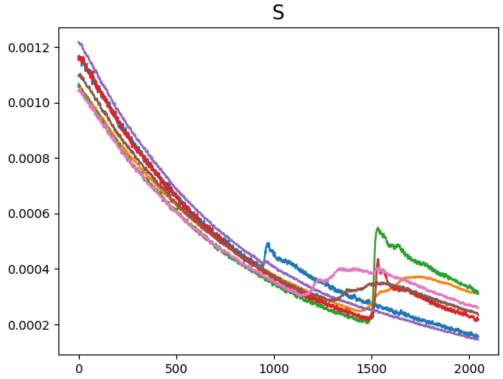
Example 1. 1D spectral image (line-scan)

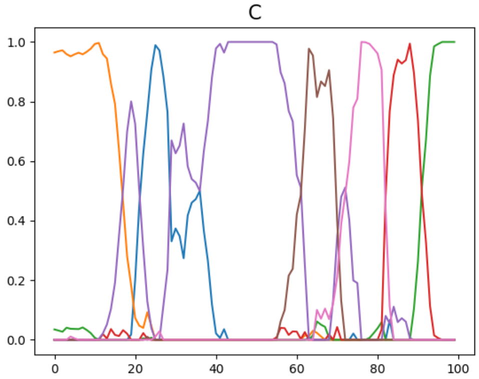
Spectroscopic data is available in the zip file of this pypi module (data_EELS.txt). It represents EELS (Electron Energy Loss Spectroscopy). This data consists of 100 spectra acquired over 2048 energy levels. The dataset described in:

Braidy N. and Gosselin R. (2019) Unmixing noisy co-registered spectrum images of multicomponent nanostructures, Scientific Reports, 9:18797.

```
import numpy as np
import matplotlib.pyplot as plt
from MCRLLM import mcrllm
plt.close('all')

#%% Load Data
X = np.loadtxt('data_EELS.txt', delimiter=',')
X = X.T

#%% MCRLLM
nb_c = 7
decomp = mcrllm(X,nb_c,init = 'Kmeans', nb_iter=20)
C = decomp.C
S = decomp.S

#%% Plot results
plt.figure();plt.plot(S.T);plt.title('S',fontsize=16)
plt.figure();plt.plot(C);plt.title('C',fontsize=16)
```

Output plots





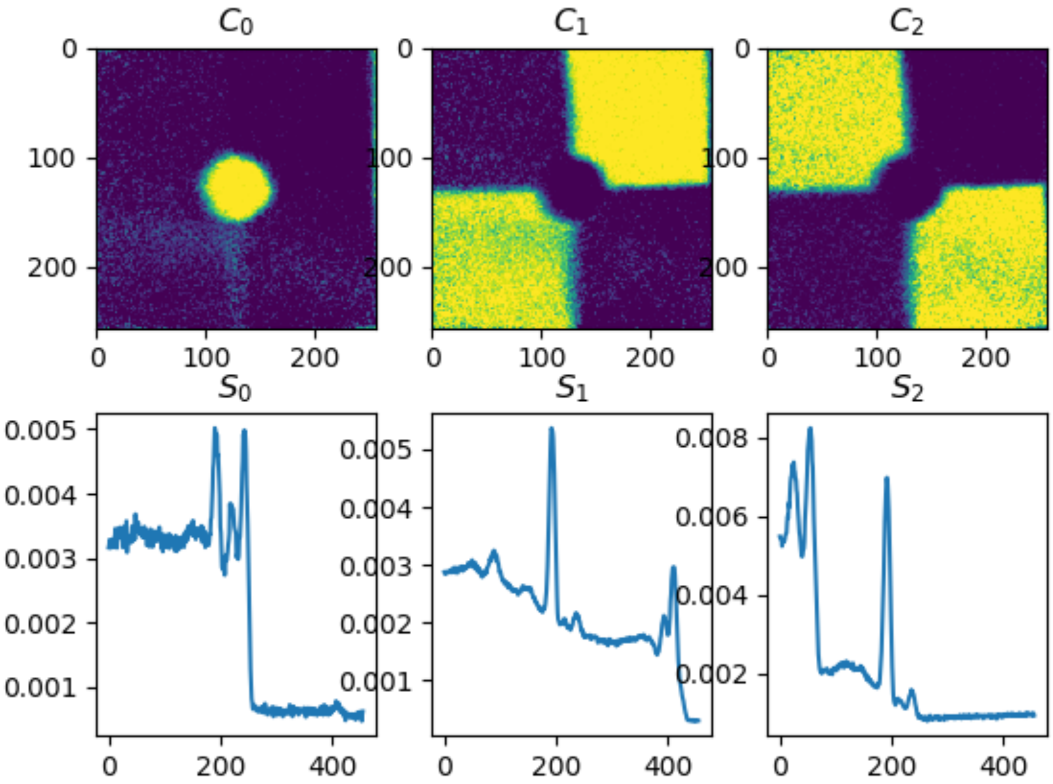
Example 2. 2D spectral image

Spectroscopic data is available in the zip file of this pypi module (data_XPS.npy). It represents XPS (X-ray photoelectron spectroscopy). This data consists of (256×256) spectra acquired over 456 energy levels. The dataset described in:

Lavoie F.B., Braidy N. and Gosselin R. (2016) Including Noise Characteristics in MCR to improve Mapping and Component Extraction from Spectral Images, Chemometrics and Intelligent Laboratory Systems, 153, 40-50.

```
import numpy as np
import matplotlib.pyplot as plt
from MCRLLM import mcrllm
plt.close('all')

#%% Load Data
X = np.load('data_XPS.npy')
dim1 = np.shape(X)[0]
dim2 = np.shape(X)[1]
nb_level = np.shape(X)[2]

X = np.reshape(X, [dim1*dim2 , nb_level])

#%% MCRLLM
nb_c = 3
decomposition = mcrllm(X,nb_c,init = 'Kmeans', nb_iter=3)

C = decomposition.C
S = decomposition.S

#%% Plot results
plt.figure();
plt.subplot(231); plt.imshow(C[:,:,0]); plt.title('$C_0$')
plt.subplot(232); plt.imshow(C[:,:,1]); plt.title('$C_1$')
plt.subplot(233); plt.imshow(C[:,:,2]); plt.title('$C_2$')
plt.subplot(234); plt.plot(S[0,:]); plt.title('$S_0$')
plt.subplot(235); plt.plot(S[1,:]); plt.title('$S_1$')
plt.subplot(236); plt.plot(S[2,:]); plt.title('$S_2$')
```

Output plots


C_0
C_1
C_2
0
100
200
0
100
200
S_0
S_1
S_2
0.005
0.004
0.003
0.002
0.001
0.008
0.006
0.004
0.002
0
200
400