Multiway Lecture - Amino Exercise (for Python)

April 23, 2020

Computational Data Analysis

Course 02582

Andreas Baum, andba@dtu.dk

Required Python3 packages: *dill*, *numpy*, *scikit-learn*, *tensorly*, *matplotlib*

1 Background

Excitation-emission fluorescence spectroscopy is used in analytical chemistry to detect and quantify chemical constituents, e.g. amino acids. Such constituents typically absorb light in a certain spectral range (excitement) followed by the subsequent phenomena of light emission (fluorescence). Hereby, the emitted light may indicate different wavelengths than the light used for excitation.

This provided data set consists of five simple laboratory-made samples. Each sample contains different amounts of tyrosine, tryptophan and phenylalanine dissolved in phosphate buffered water. The samples were measured by fluorescence (excitation 250-300 nm, emission 250-450 nm, 1 nm intervals) on a PE LS50B spectrofluorometer. The multiway array $\mathcal{X}$ to be decomposed is hence $5 \times 61 \times 201$. (Bro et al. 1997)

For further information about these data refer to: Bro, R, "PARAFAC: Tutorial and applications", *Chemometrics and Intelligent Laboratory Systems*, **1997**, 38, 149-171

2 Questions

2.1 Install all required Python3 packages (maybe you require upgrades)! Load pickled data (`Amino.pkl`) using dill!

This has been done for you already. The tensor $\mathcal{X}$ is stored in variable `x`. Amino acid concentrations of the three amino acids tryptophan, tyrosine and phenylalanine are stored in `amin_conc`. `scale_em` and `scale_ex` contain the axis labels for the emission and the excitation mode. The tensor $\mathcal{X}$ has three modes and its dimensionality is $5 \times 61 \times 201$.

[ ]:
```
import dill
dill.load_session('Amino.pkl')
```

```
# concentrations of the three amino acids in the five samples
amino_conc
colnames_amino = ['tryptophan', 'tyrosine', 'phenylalanine']

# axis labels for the emission and excitation mode
scale_em = emax
scale_ex = exax

# dimensionality of the tensor X is ...
print(x.shape)
```

2.2 Visualize all 5 Excitation/Emission (EX/EM) profiles (= samples) using surface plots! Ideally these data should be describable with three PARAFAC components. Why?

Hint: You can find Python code on how to surface plot in the casein exercise.

2.3 Decompose the three-way tensor using PARAFAC!

For the following assume the notation of the decomposition as $\mathcal{X} = A(B \odot C)^T + \mathcal{E}$, where $\mathcal{X}$, A, B, C and $\mathcal{E}$ represent the three-way tensor, PARAFAC scores (sample mode loadings), loadings for the emission and excitation mode and residuals, respectively, and $\odot$ represents the Khatri-Rao product.

[ ]:
```
from tensorly.decomposition import parafac
# Define number of components
numcomp = 3

# random state is set to reproduce solution handout results
model1 = parafac(x, numcomp, init='random', random_state=2020)
loadings = model1[1]
```

2.4 Plot all PARAFAC mode loadings (A, B and C)! How do these relate to the 5 EX/EM profiles above?

[ ]:
```

```

2.5 Scatter plot the PARAFAC scores versus the appropriate known amino acid concentrations!

Hint: You are expected to obtain three scatter plots here, one for each amino acid.

[ ]:

2.6 Compute the outer vector products $b_i c_i^T$ for all PARAFAC components $1, 2, ..., I$ and surface plot the result!

[ ]:

2.7 Reconstruct the 5th EX/EM profile using the PARAFAC model and surface plot the result!

[ ]:
```
from tensorly.tenalg import khatri_rao
from tensorly import fold
```

2.8 Reconstruct the 5th sample using the PARAFAC model, but remove the tyrosine signals! Surface plot the result!

[ ]:

2.9 (Optional) Compute the core tensor, such that $\mathcal{G} = \mathcal{X} \times_1 A^{-1} \times_2 B^{-1} \times_3 C^{-1}$.

Hint: Use the function `multi_mode_dot` for sequential calculation of n-mode products.

[ ]:
```
from tensorly.tenalg import multi_mode_dot
```

2.10 (Optional) Calculate the core consistency diagnostic (corcondia - CCD), such that $CCD = 100 \cdot (1 - \frac{||\mathcal{I} - \mathcal{G}||_F^2}{||\mathcal{I}||_F^2})$)! What is the purpose of the CCD?

Hint: $\mathcal{I}$ represents a perfect super-diagonal core tensor, i.e. it contains ones along its diagonal and zeros otherwise.

[ ]:

2.11 (Optional) Unfold the data and perform PCA! Can three PCA components resolve the three amino acids? Does PCA yield a unique solution?

PCA was performed for you already!

[ ]:
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from tensorly import unfold
```

```
# Unfold, mean-center and perform PCA
x_unfold = unfold(x, mode=0)
scaler1 = StandardScaler(with_mean=True, with_std=False)
model2 = PCA(numcomp)
pca_scores = model2.fit_transform(scaler1.fit_transform(x_unfold))
```