Project Blind Signal Separation

Selected Topics in Biomedial Signal Processing

November, 2021

1 Practical arrangements

To take advantage of the new graphical user interfaces (GUIs), Tensorlab 4.0 beta will be used in this project. This toolbox can be downloaded from https://www.tensorlab.net/summerschool/tensorlab4.0beta.zip and is installed by adding the directory to your path. Tensorlab 4.0 requires Matlab 2016b or newer. Older versions of Matlab are not recommended, as they do not support the new GUIs. Some useful methods are

- `gui_cpd` launches the GUI for the computation of a CPD.
- `gui_mlvsd` launches the GUI for the computation of an MLSVD.
- `U = cpd(T,R)` computes a rank-R CPD with factor matrices `U{1},U{2},...` of a tensor `T`.
- `cpd_nls` computes a CPD using the NLS approach.
- `[U,S,sv] = mlsvd(T)` computes the multilinear singular value decomposition with factor matrices `U{1},U{2},...` and core tensor `S` of a tensor `T`, and the multilinear singular values `sv`.
- `Tn = noisy(T,snr)` adds Guassian noise with a certain `snr` to a tensor `T`.
- `rankest(T)` estimates the rank of the tensor.
- `mlrankest(T)` estimates the multilinear rank of the tensor.
- `frob(T)` computes the Frobenius norm of a tensor.
- `tmprod(T,U,1:N)` computes the tensor-matrix product of tensor `T` with matrices `U{1},U{2},...` in modes `1,2,...`.
- `U = cpd_rnd(size_tens, R)` creates random factor matrices for rank-R CPD.

Use `help <algorithm>` for more information. Additionally you can have a look at some of the demos at the tensorlab website. Some algorithms accept extra options which can be set using a struct or using key-value pairs. For example, to create random, complex factor matrices, you can use

```
options = struct;
options.Imag = @rand;
U0 = cpd_rnd([10, 10, 10], 5, options);
% or
U0 = cpd_rnd([10, 10, 10], 5, 'imag', @rand);
```

Other useful options

- for the `cpd` algorithm are `options.Initialization = @cpd_rnd`
- for the `cpd_nls` algorithm the `MaxIter` options can be set to increase or decrease the number of iterations, `TolFun` and `TolX` can be used to change the tolerances.

This project is also accompanied by some additional m-files and data files (see additional zip-file):

- `ex1data.mat`: dataset `Tn` for exercise 1
- `ex2data.mat`: mixing matrix `A` for exercise 2
- `ex3data.mat`: dataset `T` and factor matrices `A`, `B` and `C` for exercise 3
- `ex4data.mat`: signal `x` for exercise 4
- `aci.m`: COM2 algorithm for ICA
- `ex_eeg.m`: starting code file for exercise 5
- `tansform_back.m`: create topoplot from factor matrix
- `demosignal3_963.mat`: EEG signal with epileptic seizure for exercise 5
- `22system10_20deze.loc`: location of electrodes
- `make_scales.m`: create scales for wavelet transform
- `multiwavelet.m`: tensorization by wavelet transform
- `normalise_3D.m` extract part of EEG signals and convert to tensor using wavelet transform

Your results must be bundled in a report with maximum 5 pages of text and 10 pages of figures.

2 Exercises

2.1 MLPCA

A 9-dimensional vector is sampled 100 times under 100 different conditions, which results in a tensor $\mathcal{T}$ of size $9 \times 100 \times 100$; see `Tn` in `ex1data.mat`. Use MLPCA to analyze this data. Hint: visualize the data and the recovered components as a point cloud.

2.2 Source separation with ICA and PCA

Consider $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}$ with $\mathbf{s}, \mathbf{x}, \mathbf{n} \in \mathbb{R}^3$ and $\mathbf{A} \in \mathbb{R}^{3\times 3}$. The noise $\mathbf{n}$ is zero-mean white Gaussian with covariance matrix equal to $\sigma_n^2\,\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The three independent sources $\mathbf{S}$ are randomly drawn from a uniform distribution over $[-0.25, 0.25]$. (Hint: `rand`.) Consider 500 samples: $\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}$, with $\mathbf{X}, \mathbf{S}, \mathbf{N} \in \mathbb{R}^{3\times 500}$. The mixing matrix `A` can be found in `ex2data.mat`.

Determine the average separation quality over 100 independent trials for a signal-to-noise ratio (SNR) equal to 0, 5, 10, ..., $50\,\mathrm{dB}$ in the following cases:

- To estimate the sources $\mathbf{S}$ from the observations $\mathbf{X}$ using ICA. You can use the COM2 algorithm (`aci.m`).
- Try to separate the sources by means of PCA.

The source and noise values vary over the trials but the mixing matrix remains fixed. Summarize the results in plots and interpret.

Hints:

- The quality of the source separation may be assessed as follows. Call the estimate of the mixing matrix $\hat{\mathbf{A}}$, and make sure that its columns are optimally ordered and scaled. (Optimal scaling means that the columns of $\hat{\mathbf{A}}$ are least-squares estimates of the columns of $\mathbf{A}$.) Let $\mathbf{F} = \mathbf{A}^{\dagger} \cdot \hat{\mathbf{A}}$. Then the signal-to-interference ratio (SIR) can be defined as

$$10\log_{10}\left(\frac{\|\mathrm{diag}(\mathbf{F})\|^2}{\|\mathbf{F}-\mathrm{diag}(\mathbf{F})\|^2}\right) \quad \text{(dB)}.$$

The SIR can be computed by means of `sir.m`.

- The SNR can be defined as $10\log_{10}\left(\frac{\|\mathbf{AS}\|^2}{\|\mathbf{N}\|^2}\right) \simeq 10\log_{10}\left(\frac{\|\mathbf{AS}\|^2}{1500\cdot\sigma_n^2}\right)$ (dB), in which the norm is the Frobenius norm (`frob`).
- The function `noisy` can be used.

2.3 Synthetic CP

Load `ex3data.mat`. The tensor $\mathcal{T}$ was generated as $\mathcal{T} = \sum_{r=1}^{3} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$. Plot the components of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$. All factors are present in the frontal slice $\mathcal{T}(:,:,4)$.

1. Try to estimate the components from the noiseless slice $\mathcal{T}(:,:,4)$ by means of PCA and explain.
2. Add zero-mean Gaussian noise to $\mathcal{T}$ with SNR $15\,\mathrm{dB}$. Compute the multilinear singular values of the noisy tensor. Compute the CPD of the noisy tensor for different numbers of rank-1 terms. Show the results (convergence curves, rank-1 terms, ...) in a plot and discuss. Mention which algorithm and motivate which parameter settings were used.

2.4 Harmonic retrieval

In `ex4data.mat`, a signal $x(t)$ that is sampled at $100\,\mathrm{Hz}$, is given. We know that this signal is a noisy mixture of sines, cosines and exponentials, but the SNR is unknown. Use ESPRIT, LMLRA-based harmonic retrieval and CPD-based harmonic retrieval to answer the following questions:

- How many poles are in the systems and how can you see this?
- Which are the poles of the system?

Address both questions for the three techniques by performing harmonic retrieval after suitable matricization/tensorization techniques and decompositions. You do not have to discuss the relative performance of the results of the three methods. Some hints:

1. `hankelize`.
2. For CPD-based harmonic retrieval, remember the relation between exponentials and rank-1 terms.
3. Manual creation of a complex initialization for the CPD might be necessary; see `cpd_rnd`.

2.5 EEG

Download and install EEGLAB from http://sccn.ucsd.edu/eeglab. Make sure you have the Wavelet Toolbox installed. Open the file `ex_eeg.m`. Load the EEG data `demosignal3_963.mat`. The sample frequency is $250\,\mathrm{Hz}$. Inspect the 21 channels using `eegplot`. An epileptic seizure occurs near $t = 52$ seconds. Use `normalise_3D.m` to normalize and wavelet-transform the data near the seizure into a third-order tensor. What do the modes of the resulting tensor correspond to? Compute the CPD of the tensor. Look at the fit in function of the number of rank-1 terms. Look also at the factor matrices and rank-1 terms (using `plot` and `transform_back`). Make topoplots of the first mode for a decomposition in two rank-1 terms. Interpret these topoplots and link their meaning to plots of the components of the other two modes.