

Information on how to run the code for the simulation and real data in the paper "TPRM: Tensor partition regression models with applications in imaging biomarker detection" by Michelle F. Miranda, Hongtu Zhu and Joseph G. Ibrahim

1 How to download the toolboxes

To run the codes for the simulation, and to run TPRM on your data, you will need to install the following toolboxes:

1. k-Wave toolbox — this toolbox is used to generate 3D spherical signals, and it was developed by Bradley Treeby, Ben Cox, and Jiri Jaros. All documentation and download information can be found in <http://www.k-wave.org/>
2. Tools for NIfTI and ANALYZE image Toolbox — this toolbox is used to load, make, and save imaging in NIfTI and ANALYZE data. In our simulation, it is used to read the template \mathcal{G}_0 and to generate the 3D rendered image of the signals overlaid on the template. All information on this toolbox can be found in <http://www.mathworks.com/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image>
3. Tensor toolbox — this toolbox is used to create tensor structures and decompose tensor data. It was developed by Brett W. Bader, Tamara G. Kolda and others. All documentation and information on how to download it can be found in <http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html>.

2 Simulations

To run the simulations described in Section 3.1 of the paper, we provide one file for each scenario in the paper. The files are named SimulationTPRM_Scenario followed by the scenario number. Running the code is straightforward. You will need the template shown in Figure , which is provide here (Template_DS2). If you wish to run exactly the same 10 folds we ran in the paper, use the file Sim_10fold_sets.mat (the default option). If you want to generate your own data split, uncomment the lines corresponding to it and comment the line `load('Sim_10fold_sets')`. The code will call the functions `sample_delta_j.m`, and `trunc_norm.m`. These are functions for the MCMC sampling algorithm.

3 Real Data

To run TPRM on your data you will need to use `ADNI_FinalModel.m`. This is assuming you already know the size of the partitions you want to consider. Load your own dataset into a 4D tensor, with the last dimension corresponding to subjects, call this tensor `YTotal`. `YTotal` is $R^{p_1 \times p_2 \times p_3 \times N}$. If you have $p_1 \neq p_2 \neq p_3$ or want your partitions to be of different size in each dimension, use `func_dividepartition_notequal.m` when partitioning the tensor.

The main code will also call the following functions: `func_dividepartition.m` or `func_dividepartition_notequal.m`, `sample_deltaj.m`, `trunc_norm.m`, and `functionIC.m`.

If you wish to run on your data the code for the FPCA described in Section 4.1 of the paper, you can look at the file `ADNI_10fold_fpca.m`. It is straightforward to follow the code. You may have to alter the code for different Matlab versions. It works well for Matlab version 2014b.

Any questions, please send an email to michellemirandaest@gmail.com, with subject line `sffamily` TPRM code question.