MTBI Code/Data Guide

### Available Data:

Feature set:

226 subjects having 13 MRI 3D metrics for each subject. Subject set is divided into three different named parts: 117 subjects (second cycle), 65 old subjects and 44 meso subjects (no NP scores). Among those with NP scores: some do not have all NP test scores, only some. 154 subjects have all NP test scores. (Since we only only try to predict several NP test: letter\_num, digit span (backward and forward), we can use all data where these test results are present)

Ground truth set:

A binary label for all subjects indicating if MTBI or not.

Neurophysiological (NP) test scores only for the 117 second cycle and 65 old subjects taken at three different time steps.

For test types and how to merge different time step results use the file “Combining cognitive variables across cycles” provided by Yvonne.

NP Scores for old\_subjects are on “mTBI data first cycle Lui Grossman Miles 2018 review and update for combining”

NP Scores for 117 subjects are on “LUI\_mTBI\_NPwithZscores June 2018 cleaned up” file

Processing Features:

* Patches: slice\_as\_mask.py file can be used to extract patches of size NxN from a region specified by input mask(s). There is a threshold parameter that can be tuned so that only the patches with (threshold > non\_zero\_pixels / N^2). Without threshold we get very sparse patches around the boundary of the masked region. Already extracted 16x16 patches can be found inside BoW/patches folder.
* Binary masks for extracting different regions can be found in data/masks/ . .mat masks are directly binary but nii.gz masks are nifti files. There are python libraries for these files.
* Deep features: Obtained using the unsupervised training of adversarial autoencoder. The AAC and AAC\_mix models are used to extract the features. AAC\_mix imposes a mixture of gaussians on the data and does clustering directly (see BoW section below). I would not recommend using AAC\_mix
* Bag of Words (BoW): These are histograms of size k per subject obtained by k-means clustering all the patches from all the subjects in specific metric and then counting how many patches from each cluster a specific subject has. cluster.py does this operation.
* Deep BoW: Obtained by clustering the latent representation of the patches instead of the patches itself. The files that follow naming like “Bow/BOW20\_1set\_CAE\_CCS\_AK\_March21” are Deep BoW files. These were the best performing features for binary classification task and you don’t need to extract them again
* ~~Statistical features: statistics of a specific region and metric can be extracted by using the corresponding mask. stats\_data.p contains the mean and std of the subjects for each metric (across whole brain region). This is extracted by stats\_data\_compile.py~~

Processing Ground Truth:

* The time steps are named T1, T2 and T3 on the original excel files. But not all the subjects are present through all time steps and different scoring scales are used for similar tests at different testing times. So you might want to normalize test scores before using them together
* “LUI\_mTBI\_NPwithZscores June 2018 cleaned up” has all three time steps T1, T2 and T3  
  (only 114 patients)
* This file has demographic info as well.
* “mTBI data first cycle Lui Grossman Miles 2018 review and update for combining” has only time steps T1 and T3. Remember to refer to “Combining cognitive variables across cycles” file to see how to merge different tests.
* ~~merged\_np.xlsx contains the normalized and merged test scores through all time steps together~~

~~Total 172 patients, HTI, HTN, TBI, TBN?~~

~~Under alp\_all\_MTBI/data~~

### Methods Tried:

General Framework:

The following framework algorithm had yielded good results on binary classification (see details on Deep MTBI.pdf)

Greedy feature selection algorithm:

*Extract features from raw scans (BoW, statistical features etc.)*

*Take a held-out set (around 10-30) subjects out of the data*

*create empty set selected\_features={ }*

*while size(selected\_features) < R*

*for each feature\_i in all\_features:*

*for k=1:p #we are shuffling the validation set p times*

*split training-validation remaining data*

*train a model using selected\_features set + feature\_i*

*check the validation set performance*

*performance = validation set performance mean across p shuffling*

*best\_feature = argmax(performance)*

*selected\_features.add(best\_feature)*

*train a model using both validation+train set and report performance on held\_out set*

This algorithm is implemented in BoW/SVM\_forward\_greedy\_balanced\_BOW\_14metric.py and MTBI/stats\_forward\_greedy.py

My goal was to experiment with different feature extraction methods and models within this framework while keeping the greedy selection algorithm intact.

Another method would be to directly use L1 regularization (LASSO SVM) or other feature selection methods instead of going over each one by one. l1\_svm.py is also implemented, you can try the a regression model with a different regularizer as well. However I haven’t tried any non-greedy method

As feature extraction methods I’ve tried Deep BoW and statistical features. Statistical features contained only mean and standard deviation of the features. For each metric there were two features.

As models I’ve tried SVR and logistic regression.

These experiments all yielded weak results. More details should be on google doc

Feature extraction method ideas we were going to try:

* Regional statistics. Hugh had provided us with masks that contained around 7 regions inside the brain. For each region and each metric a statistical metric can be obtained (mean, std etc)
* Regional BoW: we can create BoWs across subjects for each region
* Global BoW: Cluster all the patches within each metric and create a BoW for every metric using entire brain region.

Useful data links:

<https://nyumc.app.box.com/s/5zmxa3tfabii6exknddjvxv0c7opvgxs>

<https://nyumc.app.box.com/s/0f8ksu9y3sb8l6o1r3z14fq7lvufwuqo>

<https://nyumc.app.box.com/s/8e41bvs7oql1iypoycpwkesd5z1jvo0l>

<https://nyumc.app.box.com/s/rnmv5i4x8t2ojsy3iuqu33yoh5jf4vaa>

raw MRI scan data files should be contained inside the mat\_data.zip file.

What is the difference of the above files with your dropbox file:

<https://www.dropbox.com/s/dr3vkdmc8lvwp08/all_MTBI.zip?dl=0>

all\_MTBI.zip (3.79GB)

I see NP files and mask files and also some patch data generated

Where are the MRI data files? (do we need to go to nyumc directory if we want the raw MRI data)

In merged\_np.xls, which ones are control vs. MTBI? How did you do normalization? (zero mean and STD=1?)

New region masks provided by Hugh (7 WM region in brain)

masks.zip

<https://drive.google.com/file/d/1DQRsvvAGzpWJvQJkWU6Yar7QoZrbO1_u/view?usp=sharing>

(I see multiple files, which one is the one we should use? How to open .nii file?

Also under Alp\_all\_MTBI/data/mask

Go through the mask file with Alp, to see whether we know how to interpret the 7 regions.

If you have any questions feel free to contact me at [aa3250@nyu.edu](mailto:aa3250@nyu.edu). We can also arrange a skype talk if you are free.