

# Analyzing EEG Data By Dmitry Batenkov

One of the main challenges of brain computer interface (BCI) signals is to extract useful information from the extremely complicated brainwave signal. The particular piece of information might be very simple, such as “make a left turn.” The standard approach to this problem involves algorithms from signal processing and machine learning [1]. In this tutorial we will try to learn the major steps of working with BCI signals (not including the capturing process) with the help of the open source toolbox BioSig.

## Prerequisites

The working environment is either MATLAB or GNU Octave. BioSig toolbox [2] (specifically, the biosig4octmat package) for working with EEG data should be downloaded and installed according to the instructions [BIOSIG].

## Get the Data

We will be working with one data set from BCI Competition IV [BCI1]. During the experiment, the subject was asked to imagine certain movements such as “left” or “right.” The task of the “analyst” (that’s us) was to correctly classify the type of the movement from the measured EEG potentials. Several test sets (with known labels) are provided for learning, while others are initially unlabeled.

Please download data set 2b named “motor imagery” (you will need to provide a valid email address). Please also download the true labels for the test cases from [http://www.bbc1.de/competition/iv/results/ds2b/true\\_labels.zip](http://www.bbc1.de/competition/iv/results/ds2b/true_labels.zip). Unpack both archives into the same directory (/path/to/data).

It is time to fire up the environment (Octave/MATLAB) and load the data:

```
fname = '/path/to/data/B0101T.gdf';
[s,HDR] = sload(fname);
```

The `s` matrix contains the raw electrode potentials measured with the rate of 250 Hz, while various metadata

is stored in the HDR structure. It can be visually examined using the SigViewer tool [SGV]. We see that there are six different channels, three of them are EEG-related and the other three are EOG-related (electrooculographic, i.e. coming from eye movements).

## Preprocessing

The raw data is usually contaminated with noise coming from various sources. Uncorrelated noise can be dealt with by smoothing, while correlated noise (e.g. coming from eye movements) can be estimated by what is called “artifact removal” techniques. BioSig implements several such techniques (see the 250\_ArtifactPreProcessingQualityControl subdirectory). For example, a particular EOG removal can be done as follows:

```
eogchan = identify_eog_channels(fname);
eegchan = find(HDR.CHANTYP=='E');
R = regress_eog(s, eegchan, eogchan);
s = s*R.r0;
```

## Extracting Features

Recall that our main task is to discriminate between “left” and “right” EEG signals. A frequency domain representation of the signal is usually used. The simplest such representation is probably the Power Spectral Density, but more suitable parameterizations have been developed [1]. Many of those are available in BioSig’s 300\_FeatureExtraction subdirectory. For example, we can calculate the power in a particular frequency band:

```
bands = [8,30]; % can be multiple bands
win = 1; % smoothing window in sec.
all_freq = bandpower(s,HDR.
SampleRate,bands,win);
```

Usually, not all of the recording is useful for analysis. The time instants at which the subject is presented with instruction are in fact “anchors” around which the interesting data appears. The list of those “event onset times” are

available in the HDR.TRIG field. For each event, we will extract a small portion of the (transformed) signal between 7.5 and 8 seconds from the onset time. (This interval is probably not the best one for this particular subject. There is a convenient method findclassifier() that can test several intervals and select the best one.) We will consider only the three EEG channels. Each “feature” will therefore consist of three numbers representing the PSD at a particular time instant.

```
start = 7.5*HDR.SampleRate;
final = 8*HDR.SampleRate;
[tmpm, tmps] = trigg(all_freq(:,eegchan),
HDR.TRIG,start,final,0);
features = tmpm';
```

## Classification

The event labels can be read as follows:

```
clear classlabel;
load('/path/to/data/B0101T.mat');
n = length(start:final);
labels = repmat(classlabel',n,1);
labels = labels(:);
```

(The reason for duplicating the labels is that the actual training data will have `n` samples for every event.)

There are plenty of classifiers to choose from, such as Linear Discriminant Analysis, Radial Basis Functions, Support Vector Machines and what not. All of these can be used interchangeably:

```
MODE = 'RBF'; % see full list in train_sc.m
CC = train_sc(features,labels,MODE);
```

The `cc` structure contains all which is needed to evaluate the classifier on new data. So let us run it on another data set obtained from the same subject. Repeat all the above instructions, excluding the training step (last snippet), replacing the file name ‘B0101T.gdf’ with ‘B0104E.gdf’. Then write:

```
[R] = test_sc(CC,features1,MODE,labels1);
```

Here `labels1` are the true labels, while the evaluated ones are to be found in `R.Classlabels`. The `test_sc` function also evaluates the performance of our classifier using several statistical tests such as the Cohen's kappa coefficient [2] available in `R.kappa`.

### Conclusion

Our results may not be so good, but this is just a toy example, disregarding many options for improvement. You can look at the solutions of the top ranking algorithms at <http://www.bbc.de/competition/iv/results/index.html#dataset2b> to get some additional ideas.

### References

[1] Bashashati, A., Fatourechi, M., Ward, R.K., and Birch, G.E. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, 4(2) [2007], R32.

[2] Schlogl, A., and Brunner, C. Biosig: A free and open source software library for BCI research. *Computer*, 41, 10 [2008], 44–50.

[BIOSIG] BioSig project <http://biosig.sourceforge.net>

[BCI1] BCI Competition IV <http://www.bbc.de/competition/iv/>

[SGV] SigViewer <http://sigviewer.sourceforge.net>

[EEG] EEGLab <http://sccn.ucsd.edu/eeglab/> - MATLAB toolbox for EEG

[DAT] EEG/ERP data for public download [http://sccn.ucsd.edu/~arno/fam2data/publicly\\_available\\_EEG\\_data.html](http://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html)

[OPENEEG] OpenEEG project <http://openeeg.sourceforge.net>

[ICA] ICA Toolbox Tutorial <http://sccn.ucsd.edu/~scott/tutorial/index.html>

[ART] Artifact Removal Tutorial <http://sccn.ucsd.edu/~jung/artifact.html>

© 2011 ACM 1528-4972/11/0900 \$10.00

Figure 1: A typical recording, viewed with SigViewer tool.

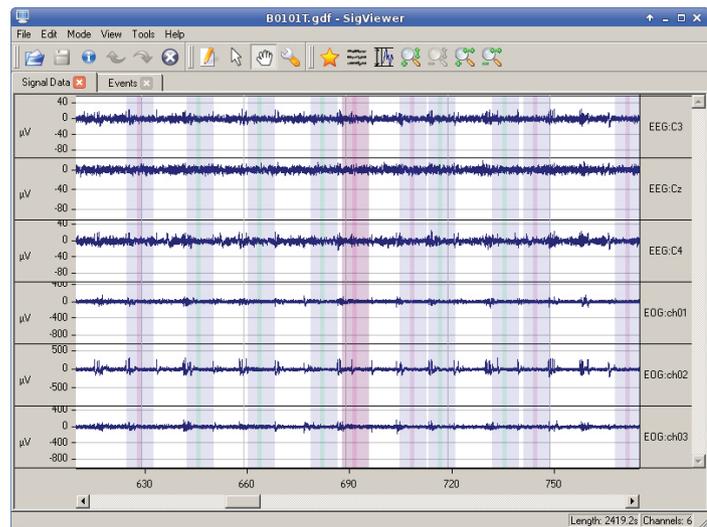


Figure 2: Raw signal portion near an event.

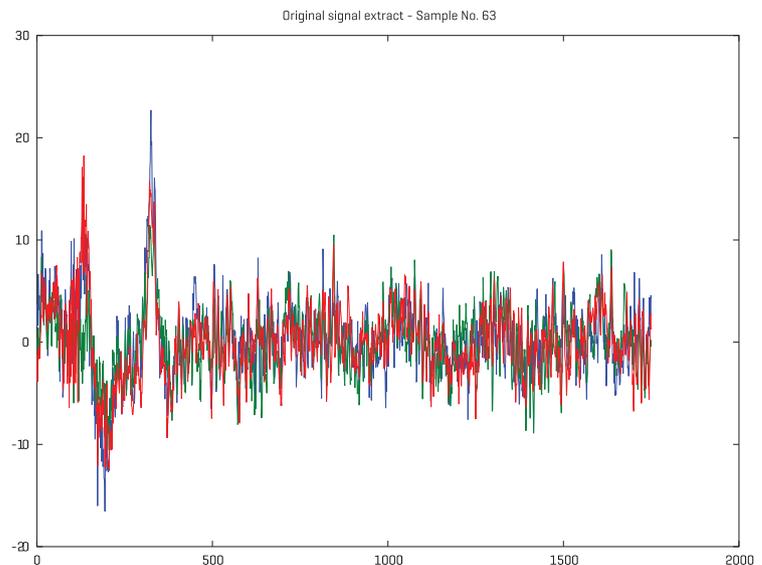


Figure 3: Features used for classification.

