**Can We read Your Mind:  
Detecting Emotional States FROm EEGS Using BIG Data**

The fields of bioengineering, electrical engineering and computer science are converging to allow many biomedical signals to be manipulated and understood by computer. Though the technology is a long way from being ready for everyday use, systems are emerging that claim to be able to understand your thoughts from non-invasive measurements taken from the surface of your body, such as electrodes placed on your head. In this laboratory, we will specifically explore the correlation between thought and electroencephalography (EEG) signals. This lab will introduce you to three important engineering skills often associated with electrical and computer engineering:

* **Signal Transduction:** measuring EEG signals using surface-mounted electrodes located along your scalp;
* **Computer Programming:** programming simple pattern recognition algorithms in Python;
* **Pattern Recognition:** mapping information extracted from an EEG signal to your mental state.

The broader goal for this laboratory is to introduce you to two rapidly growing fields: machine learning and big data. [*Machine learning*](http://en.wikipedia.org/wiki/Machine_learning), a branch of artificial intelligence (computer science) that incorporates closely-related fields of statistics (mathematics) and signal processing (electrical and computer engineering), involves the study of systems that can [learn](http://en.wikipedia.org/wiki/Learning) from data. For example, a machine learning system could be trained to recognize when you are relaxed or confused by measuring and processing EEG signals and modeling their typical behavior for each emotion. Machine learning approaches, however, require large amounts of data to learn how such signals vary many dimensions of variability including variations in emotional state and subject population. There are, after all, billions of people in the world who believe they are truly unique in the way they think :)

[*Big Data*](http://en.wikipedia.org/wiki/Big_data) is a blanket term for any collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. Every day, we create [2.5 quintillion bytes of data](http://www.ibm.com/software/data/bigdata/what-is-big-data.html) — so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data.

The fields of machine learning and big data together allow us to discover trends and to predict behavior. For example, most sports teams today employ statisticians that analyze vast amounts of data from sporting events to predict an athlete’s success. This concept was popularized in the movie [Moneyball](http://www.imdb.com/title/tt1210166/) and the book [The Signal and the Noise](http://www.amazon.com/The-Signal-Noise-Many-Predictions/dp/159420411X). How would you predict who will win the World Cup? who will win American Idol? who the next president will be? There are a number of web sites devoted to such topics, including [fivethirtyeight.com](http://fivethirtyeight.com).

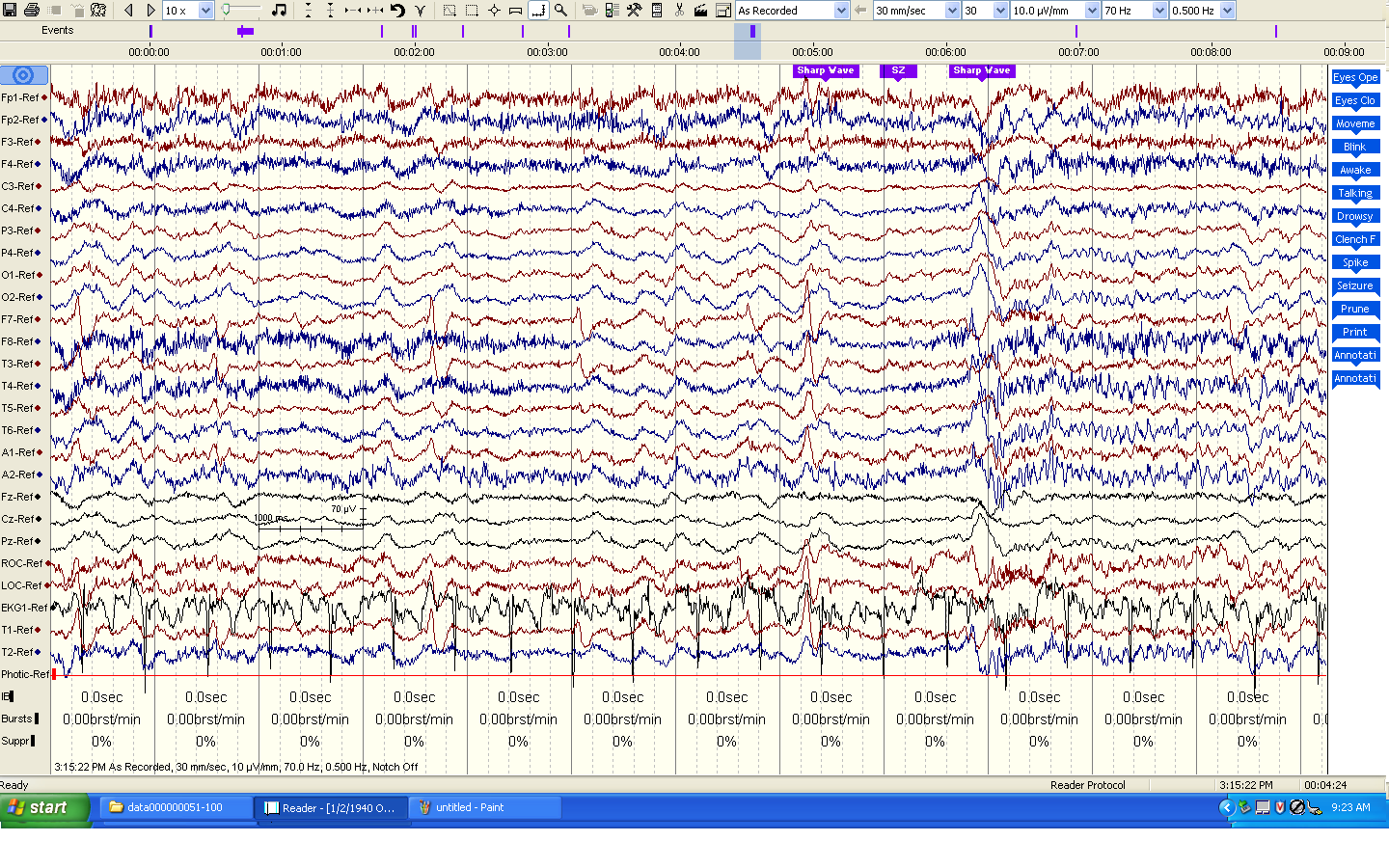
The laboratory presented here is part of a larger effort at Temple University to develop the data resources essential to driving technology development in bioengineering. The [Neural Engineering Data Consortium](http://www.nedcdata.org)) is currently focused on the development of the [world’s largest publicly released EEG database](http://www.isip.http://www.isip.piconepress.com/projects/tuh_eeg) that is being used to develop technology to predict diseases from EEG data. This laboratory will introduce you to the steps involved in this process. This laboratory is funded from a grant provided by the [National Science Foundation](http://www.nsf.gov) to explore the development of big data bioengineering resources. (You should consider getting involved in such externally funded research programs as undergraduates.)

During this laboratory, we will assist you in working through each step. We don’t expect you to become experts in any of these technologies after one short lab. Our goal is to help you become acquainted with many engineering disciplines that contribute to our understanding and manipulation of human anatomy and physiology. Remember that you should not be afraid to ask for help if you get stuck on something – we all have been there and have experienced similar problems. No question is too basic.

**Phase I: Signal Transduction**

An EEG measures spontaneous electrical activity of the brain over a short period of time, typically 30 minutes, as shown in Figure 1. When you go to a hospital for an EEG test, you will be wired up with anywhere from 20 to 128 electrodes and told to relax while a technician collects data. The entire session, including the time required to affix sensors to a patient’s scalp, requires one to two hours. In more extreme cases, you might be asked to stay in the hospital for several days while you are continuously monitored. Patients are asked to lie still in a prone position, and are periodically requested to perform limited movements (e.g., breath, blink). The electrical signals are presented in a waveform display shown in Figure 2. EEG specialists review these waveforms and develop a diagnosis.

EEGs traditionally have been used to diagnose epilepsy and strokes. Other common clinical uses have been for diagnoses of coma, encephalopathies, brain death and sleep disorders. EEGs and other forms of brain imaging such as fMRI are increasingly being used to diagnose head-related trauma injuries and Alzheimer’s disease. A board certified EEG specialist currently is required by law to interpret an EEG and produce a diagnosis. It takes several years of additional training post-medical school for a physician to qualify as a clinical specialist. The interpretation of an EEG is a serious matter since a diagnosis of a brain-related illness such as epilepsy is a life-altering event. Physicians will make decisions such as whether to perform brain surgery on a patient (yes, that means cutting your brain open and removing sections of it) based on these EEG measurements. Therefore, accurately diagnosing an illness is critical.

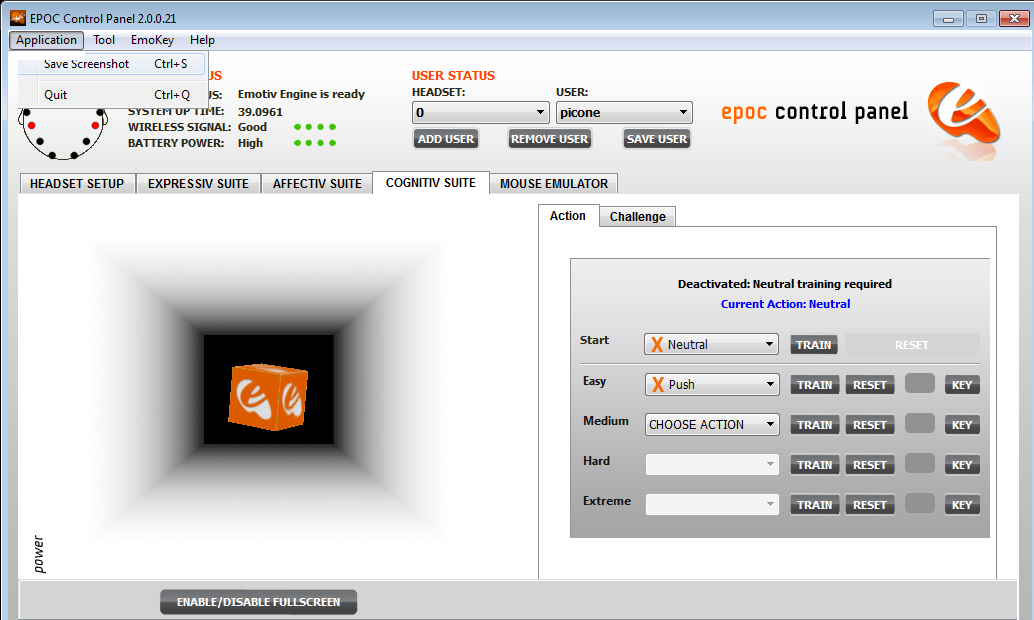


1. Figure 2. EEG data, which consists of multi-channel recordings, is displayed.

[](http://www.eleccircuit.com/wp-content/uploads/2008/07/dc-to-ac-inverter-by-ic-555-and-tip41-tip42.gif)

1. Figure 1. An EEG measures electrical activity along the scalp.

Familiarize yourself with the [Emotiv EPOC headset](http://www.emotiv.com/apps/epoc/299/) that we will use to collect data. We will demonstrate how to affix this headset to your head (are you having a bad hair day? it might get a little worse :) and collect data using a simple software tool shown in Figure 3. This is the Emotiv EPOC Control Panel. It includes a display that shows the status of the electrodes, and several demonstrations of what you can do with the technology. Using the “Headset Setup” tab, make sure the headset is transmitting data properly and the electrodes are working (highlighted in green). Then select the “Mouse Emulator” screen and observe the use of the gyroscope sensors. Notice that this demonstration works fairly well because the gyroscopes are very reliable signals compared to the brainwave signals we will attempt to measure.

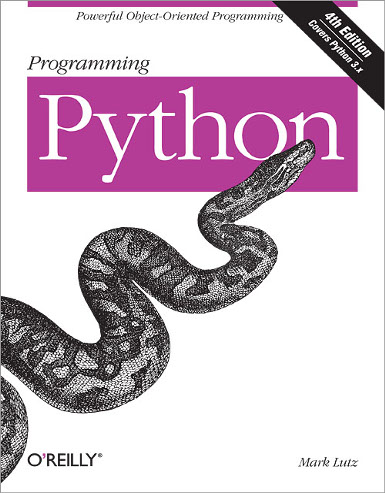


1. Figure 3. EEG data collected from the Emotiv EPOC is displayed using an interactive tool.

Next, select the “COGNTIV SUITE” tab. This demonstration is very similar to what we are attempting to accomplish in this lab. In this demonstration, you can train the system to recognize your thought patterns when you are imagining certain movements. The end result is your ability to move the cube with your thoughts rather than physical actions. The demonstration will lead you through a process of training the system when you provide certain stimuli, and then it will allow you to attempt to move the cube when you provide the same stimuli. With a little practice, you can make the system work fairly well.

Some useful background on EEG signals and their manipulation can be found in this [presentation](http://www.isip.piconepress.com/publications/presentations_misc/2013/isip/eeg_minterp/) and these [related resources](http://www.isip.piconepress.com/projects/auto_eeg/html/publications.shtml).

**Phase II: Computer Programming**

[](http://shop.oreilly.com/product/9780596158118.do)

1. Figure 4. O’Reilly books on are always good places to begin when you want to study a programming language in detail.

[Python](http://en.wikipedia.org/wiki/Python_(programming_language)) is a widely used general-purpose, high-level programming language. Its main feature is that it is a scripting language. This means you can type instructions as plain text, store them in a file, and run them directly using a Python interpreter. Though you sacrifice some efficiency, you gain much in terms of flexibility and reduced software development time. There is a wealth of information available on [how to learn Python](https://www.python.org). It is easy to use if you are familiar with basic programming concepts. In this phase of the lab, we will focus on learning the basics of Python.

The most important thing to remember is that any time you need to figure out how to do something in Python, just do a Google search. You will undoubtedly find that someone has written an entire web page devoted to solving your problem.

As preparation for the lab, we encourage you to review these excellent resources:

* [*The Python Tutorial*](https://docs.python.org/2/tutorial/): walks you through the basics of Python programming;
* [*Python Tutorial for Beginners*](https://www.youtube.com/watch?v=29mq1Bn52GY): a short YouTube video demonstrating how to write a short program;
* *Learn Python*: a nice interactive web site that teaches you the basics.

Python is a popular choice for a language today because so many people are using it. There are millions of web pages devoted to its use and you can generally find a package that was developed for about any task. You can write many different types of code using Python. In this lab, we will focus on doing some simple things like reading EEG data from a file and doing some simple computations.

**Task 1: Hello World**

Write a simple program to display the text “Hello World” on your computer screen. Open a cmd window and change directories to the location of the tools using the cd command. Open up a plain text editor and follow [these instructions](http://www.learnpython.org/en/Hello,_World!), saving the file to we2\_hello.py. To run this program, type:

../python/python.exe we2\_hello.py

This is how we run python from the command line. If you feel particularly adventuresome, run the application called [Eclipse](http://www.eclipse.org) and try this from within Eclipse. Eclipse is an integrated development environment (IDE). It lets you develop, debug, run and maintain code from a single user interface.

**Task 2: Opening and Reading Files**

The next step is to open your EEG files and read the data in them. We will use a package known as [edfplus](https://github.com/breuderink/eegtools/blob/master/eegtools/io/edfplus.py). Again, using your text editor, open the file names “we2\_average.py”. Execute this code using the following command:

../python/python.exe we2\_average.py data\r\_01.edf

The output should look something like this:

../python/python.exe we2\_average.py data\r\_01.edf

2 : channel[ AF3 ]: avg = 4322.81 std = 26.5019

3 : channel[ F7 ]: avg = 3822.92 std = 25.1258

...

13 : channel[ F4 ]: avg = 4276.88 std = 48.9342

14 : channel[ F8 ]: avg = 3815.15 std = 42.9944

15 : channel[ AF4 ]: avg = 4531.23 std = 31.8881

Read the code and study each step, particularly the line containing the function call “load\_edf.”

**Task 3: Looping and Program Control**

The example program contains two lines that do the key calculation of computing the average and standard deviation of each channel. The code is shown below:

for i in range(first\_channel, last\_channel + 1):

# compute the average and standard deviation

#

avg = numpy.mean(signal[i]);

std = numpy.std(signal[i]);

# display the results

#

print i, ": channel[", physical\_channels[i], "]: avg = ", avg, " std = ", std

The “for” statement tells the program to process each channel one at a time. The function calls “numpty.mean” and “numpy.std” compute the average and standard deviation of the signal.

Modify this code to do the following:

(1) Subtract the average value from every sample of the signal.

(2) Compute the standard deviation of the new signal.

We call this signal a debiased signal. Bias sometimes appears in signals due to the way they are collected. For example, the electronics used to process the signal might not be properly grounded, or the measurements might drift over time as the electronics heat up. Most often bias is an artifact that must be removed before the data is processed further.

The modification you made to the code is a typical example of the type of software an engineer specializing in signal processing might write. Signal processing involves the application of well-known mathematical concepts to manipulate signals in ways that are beneficial to users. For example, [pacemakers](http://www.nhlbi.nih.gov/health/health-topics/topics/pace/) measure electrical signals collected from the heart and generate control signals that allow your heart to beat more regularly. Inside the small device inserted into your chest is a processor which runs code very similar to what you wrote above.

**Phase III: Pattern Recognition**

Next, we are going to attempt to put this all together and run a pattern recognition experiment. We will train statistical models and then use that data to predict your mental state. There are two steps:

* *Training*: generating statistical models of the relaxed and busy states;
* *Evaluation*: compare each of your data files to both models and choose the best match.

For each task, we will provide you with code to help you learn how to do this, and lots of hands-on instruction. By the end of the lab, you should be in a position to implement your own ideas.

**Task 1: Training**

Run the training module, *we2\_train.py*, on each set of 10 files to generate two models. This command will compute some simple statistics over all of the data. From the command line, execute this command:

../python/python.exe we2\_train.py r\_model.txt data\_r.list

This will generate a model file called r\_model.txt (the first argument). Repeat this process for the “busy” files (using filenames b\_model.txt and data\_b.list) to generate a second model file called b\_model.txt. These model files contain the [average and standard deviation](http://www.wikihow.com/Calculate-Mean,-Standard-Deviation,-and-Standard-Error) of each channel in each training example, averaged over the 10 files that you provided. You can view these model files by typing “more r\_model.txt” from the command line.

**Task 2: Evaluation**

Let’s now evaluate your data against your models. Execute the following command:

../python/python.exe we2\_evaluate.py r\_model.txt b\_model.txt data\_all.list

This command processes each of your EEG files and outputs a score for each model. Your display will look something like this:

file: r\_01.edf relaxed: 27.45 busy: 50.36

...

|  |  |  |
| --- | --- | --- |
| GUESS: | | |
| TRUTH: | relaxed: | busy: |
| relaxed: | 80.0 ( 8/10) | 20.0 ( 2/10) |
| busy: | 0.0 ( 0/10) | 100.0 (10/10) |

The lower the score the better. In this case, the system guessed that the relaxed model was a better fit than the busy model, so it guessed correctly. The table summarizes the choices made for the 20 files that were processed.

Now that you have an idea of how these basic tools work, try modifying the evaluation and training scripts to add your debiasing code to the system and see how it influences performance.

We have demonstrated in this phase the basic approach we use to developing computer software to recognize EEG signals. The actual mathematics implemented in the training and evaluation programs are referred to as algorithms and can be quite complex. A popular approach today is based on [deep learning](http://deeplearning.net) and [neural networks](http://en.wikipedia.org/wiki/Artificial_neural_network). The technology we use to perform such tasks for EEGs is very similar to the technology Internet providers such as Google and Netflix use to predict what you advertisements you want to see, or what Siri uses to understand what you are saying. A very nice demonstration of the range of capabilities of the Emotiv EPOC system can be found in this [TED Talk](http://www.ted.com/talks/tan_le_a_headset_that_reads_your_brainwaves).