6/30/2016

Today's Goal Implement and test Approximate Entropy. Look into method of feature evaluation.

P-values are deceptively small for my ANOVAs, but this seems to be because I have an extremely large sample size with well separated means. Autocorrelation also had a high F-value, but lower than the other features. So, it seems that it is not as good as the other features. I will use a similar test for Approximate Entropy.

Calculating Approximate Entropy for a time series u(i) i=[1…N]:

Choose parameters m (length of vectors) and r (filtering level). Generally, m=2 is theoretically the right choice and r is subject to one's own data.

For a sequence of vectors x(1), x(2), …, x(N-m+1) that exist in R^2 defined by x(i)=[u(i), u(i+1), …, u(i+m-1)]

Use the x(i) sequence to calculate C_i=(number of x(j) such that d[x(i),x(j)]<r)/(N-m+1) for all i in [1,…,N-m+1]

where d(a,b)=max|u(a)-u(b)| component wise

Note: since j takes all values, the case when i=j will be counted. Calculate:

$$\Phi^m(r) = \frac{\sum_{i=1}^{N-m} \ln(C_i^m(r))}{N-m}$$

which gives you:

$$\mathrm{ApEn} = \Phi^m(r) - \Phi^{m+1}(r)$$

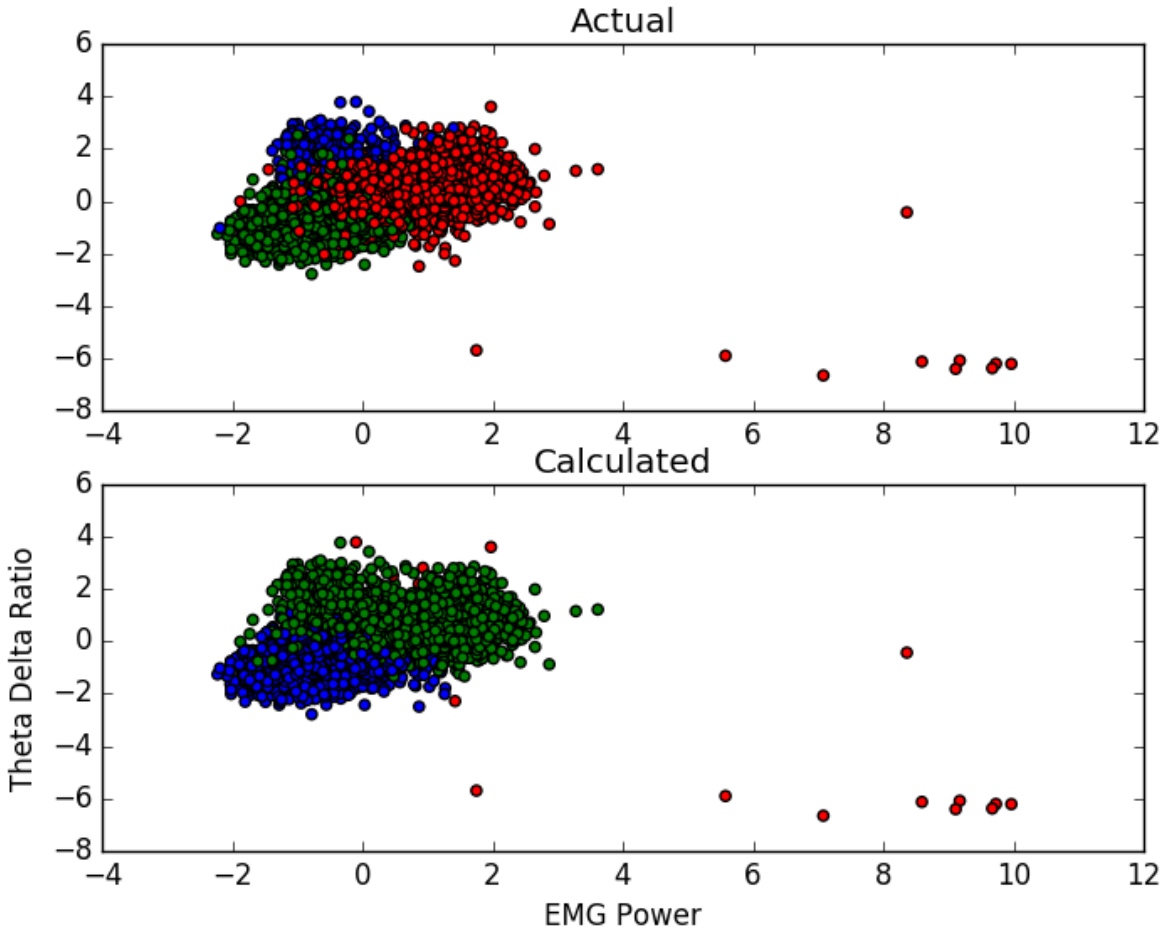
I will use m=2 and r=0.2*SD(epoch) since that is what theoretically gives good results (see: "Approximate Entropy as an Electroencephalographic Measure of Anesthetic Drug Effect during Desflurane Anesthesia")

This could be useful, because REM shows signs of regularity while the other two states do not.

I implemented this algorithm. Unfortunately, it seems like it will be too computationally inefficient to implement in real-time.

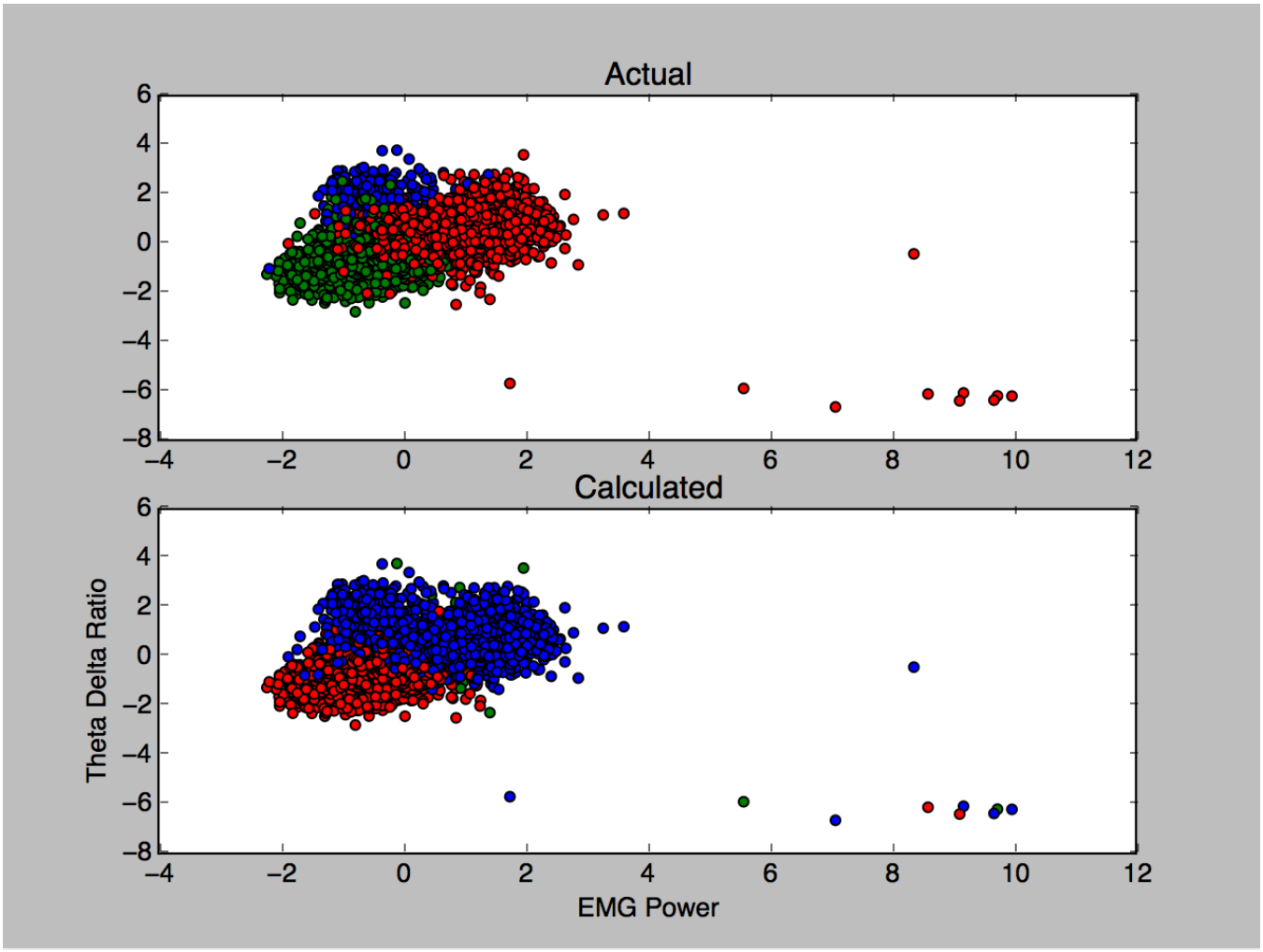
Features that I have right now seem pretty good, so for now I will be looking at more clustering algorithms to see if any work better.

I ran a Gaussian Mixture Model clustering. Below are the results (subject 1032):



The clustering for NREM and Wake went overall well, but the REM clustering was very thrown off by the outliers. I need to ignore these outliers in the data.

I implemented a way to ignore outliers with SD's>4 and classifying them to random states. Below is the Gaussian Mixture Model clustering result (subject 1032):



The cluster size of REM is extremely small (Wake=2346, NREM=2040, REM=16). I have to find a way to set the minimum size of the REM cluster size so that the algorithm will start to gravitate towards classifying more REM points.

Density Based Clustering can do this. I can also calculate new features to make more defined clusters or run a Principle Components Analysis on the entire signal to find the best features.