# Marrying Encoding and Decoding Models

Saket Saurabh

Institut für Informatik, Universität Stuttgart, Stuttgart, Germany

**ARTICLE HISTORY**
Compiled August 31, 2021

**ABSTRACT**
There are broadly two categories of EEG analyses: Decoding, `(f(brain) = stimulus)` and encoding `(f(stimulus) = brain)`. In this project we will try to combine the benefits of both methods based on the analysis-approach of back2back regression, which in some sense encompasses both.

**KEYWORDS**
back-to-back regression; encoding; decoding; eeg; saliency;

## 1. Introduction

### *1.1. The* **Encoding** *models*

In encoding models [1], the neural activity is modelled as a weighted sum of stimulus features. The choice of the stimuli and input feature space are critical and have a strong influence on the interpretation of the encoding model. Often, these features are complex and allows us to study the brain under conditions observed in the real world. This can be modelled into a linear regression problem:

$$\texttt{Activity} = SW + e$$

Where, S is the stimulus matrix where each row corresponds to a timepoint of the response, and the columns are the feature values at that timepoint and time-lag. w is a vector of model weights, and e is a vector of random noise at each timepoint.

### *1.2. The* **Decoding** *Models*

Decoding models [1] can be used to infer the stimulus that gave rise to a distributed patterns of neural activity by fitting a weight to each neural signal. The weights can operate on a multi-dimensional neural signal by considering the joint activity across multiple channels (e.g., electrodes or voxels) around the same time.

$$s = Xw + e$$

where s is a vector of stimulus feature values recorded over time, and X is the channel activity matrix where each row is a timepoint and each column is a neural feature. w is a vector of model weight, and e is a vector of random noise at each timepoint.

### *1.3. Difference between Encoding and Decoding Models*

Encoding models are often called Forward models, reflecting the direction of time from stimulus to neural activity. Conversely, decoding models are often called Backward or Inverse models, as they move "backwards" in time in a traditional sensory experiment.

In the experiment by Mesgarani and Chang (2012) and Pasley et al. (2012), one model was fitted for each stimulus feature that is being decoded. This corresponds to reversing the terms in the standard regression equations:

$$weights_{encoding} = (X^T X)^{-1} X^T y$$

$$weights_{decoding} = (Y^T Y)^{-1} Y^T x$$

## 2. Dataset

We use the subject 45 and subject 48 from the study conducted by Anna et al. [2] In this study, the high-density EEG is recorded from a group of participants in two main contrast conditions. The conditions differ based on position of the fixation points – in free viewing, the participants were allowed to do free eye movements whereas in lab conditions the participants maintained the central fixation while looking at the faces and objects. The observed ERP difference can be seen in Figure 1.It was observed that there is a high correlation in the N170 effect between the participants in contrast with experiment conditions. In our project, we focus on variables SAC_AMPLITUDE, and SAC_VMAX. The SAC_AMPLITUDE represents the amplitude of the eye saccades where SAC_VMAX represents maximum velocity of the saccades. Furthermore, we plotted the distribution plots for them Figure 2 and Correlation Matrix Figure 3 to understand their relation with other variables.

## 3. Implementation

### *3.1. Simulating Events*

We generate the events from a multivariate Gaussian distribution based on number of trials, event ids and a correlation matrix, which defines the event relation with each other. The columns in the resultant events matrix can represent a combination of stimulus and conditions. Our condition values fall into two categories: present and absent whereas the stimulus can take any continuous values. Furthermore, we use dummy encoding to represent any addition categorical conditions and preset the intercept value at one to construct a Design Matrix from the events matrix.

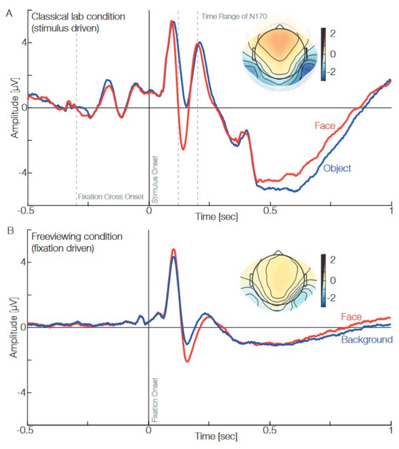
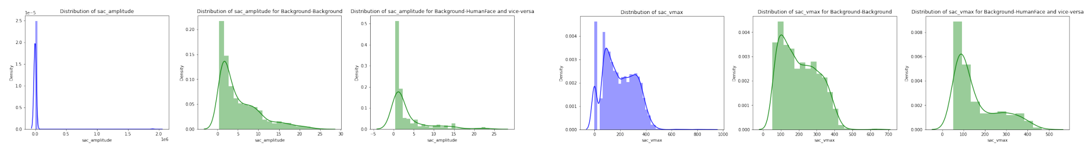


**Figure 1.** Plot above shows the traditional ERP in lab conditions and Plot below shows ERP in wild from paper [2]



(a) Distribution of sac_amplitude events (b) Distribution of sac_vmax events

**Figure 2.** Distribution of Events

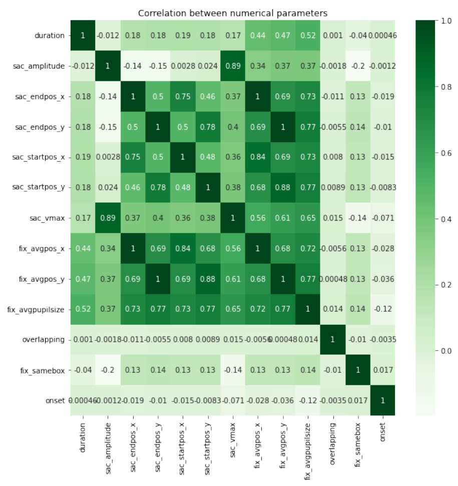


**Figure 3.** Correlation between the events

## *3.2. Simulating Evoked response*

### *3.2.1. Base Signal*

To construct our underlying evoked response, we compared hanning window aka cosine bell and barlett-hann windows, which is a weighted sum of the Bartlett and Hann window. Finally, we picked hanning windows because the barlett-hann windows seems to have too much ripple at the ends [1] in the frequency domain and use of hanning windows are widely supported in EEG literature [3]

### *3.2.2. Noise*

We generate two types of noise – White Noise and Pink Noise. The white noise is generated from a univariate "normal" (Gaussian) distribution of mean zero and variance one and the Pink noise or 1/f noise is generated from the amplitudes of a sine wave with random phase by giving a minimum and a maximum frequency. We distributed the outputs of the above noise functions to vary linearly based on the number of channels required in our experiment. Therefore, the channels with high index values will have more weights as compared to channels with low index values. Finally, we then generate the evoked response using the algorithm 1

**Algorithm 1** Simulate EEG evoked response

```
Require: $ntime \vee nchannels \vee events \vee alphas$
  $basicfunc \leftarrow hanning(ntime, padding)$
  $ntrials \leftarrow dim(events, 1)$
  $\hat{y} \leftarrow events \times (repeat(basisfunc', dim(events, 2)) \times alphas)$
  $noise \leftarrow noise\_generator(nchannels, ntimes, ntrials)$
  $beta \leftarrow repeat(\hat{y}, nchannels) + noise$
  return $beta$
```

**Algorithm 2** Simulate Pink Noise

```
Require: $ntime \vee nchannels \vee events$
  $min\_freq \leftarrow 30$
  $max\_freq \leftarrow 150$
  $steps \leftarrow 30$
  $freq \leftarrow range(min\_freq, max\_freq, step)$
  function $sin\_amp = (\theta, amp) \leftarrow amp * \sin(2\pi\theta + 2\pi random\_phase)$
  for ch in nchannels do
      $c \leftarrow random(1, 2)$
      for fi in size(freq, 1) do
          $amp = \frac{1}{freq[fi]^c}$
          for t in ntims do
              $noise[ch, t, :] = noise[ch, t, :].+[sin\_amp(freq[fi]*t, amp)$ for tr in ntrials]
          end for
      end for
  end for
  return $noise$
```

### *3.2.3. Unfold toolbox*

Unfold [4] is a toolbox written in Julia and provides tools for performing deconvolution of overlapping EEG (Pupil, LFP etc.) signals and (non)-linear modelling Saliency.

Unfold can extract features like how much is the power or extract one frequency band of a continuous predictor using all time points from the EEG data. In our project

[1]Window_function

this continuous predictor is saccade amplitude. The design matrix columns can be then filled by this continuous predictor. The subsequent columns of the design matrix are then filled by incrementing the values of this predictor by one unit. Thus, giving us a time shifting effect. Thereafter, regularization is added.

In our project, we did not include the basis function – it is useful for cases when we try to fit the predictor (saccade amplitude) to account for non-linear effect. The basis functions categorize our continuous predictors by fitting one effect for each interval. This gives us a step effect while fitting on the predictor.

The Unfold.jl toolbox[2] provides many solver for mass-univariate analysis, and for our project we used the back-to-back regression [5] solver. Since, the toolbox does not contain import function for EEG data, we had to use our own function. We also implemented regularized versions (L1, L2, and Elastic)[2] of solvers and a single layer Neural Net[2] solver.

### 3.2.4. *Saliency Maps*

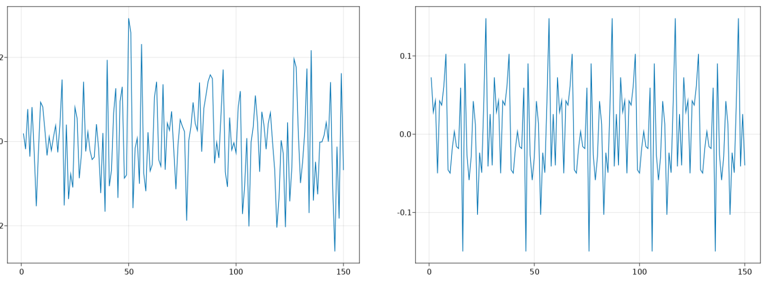
The saliency maps are generated using edge splatting technique. In splatting multiple textures with different alpha maps over a mesh. In our project we augmented the method devised by Krueger et al [3].

- In the first step, a Gaussian kernel of fixed standard deviation (10% of max-pixel width) with maximum value taken from saccades amplitude is applied to all the fixation points. The values at the overlapping pixels are added.
- In the second step the density values are mapped to a color gradient.

## 4. Observations

### 4.1. *Noise*

We generated both the noise functions but in section below we will only discuss our results based on using white noise.



(a) Plot for white noise signal (b) Plot for pink noise signal

**Figure 4.** Comparison between white and pink noise. Pink noise has periodicity

[2]https://github.com/ssaket/eeg-b2b-regression/blob/main/src/utils.jl

## *4.2. Without Regularization*

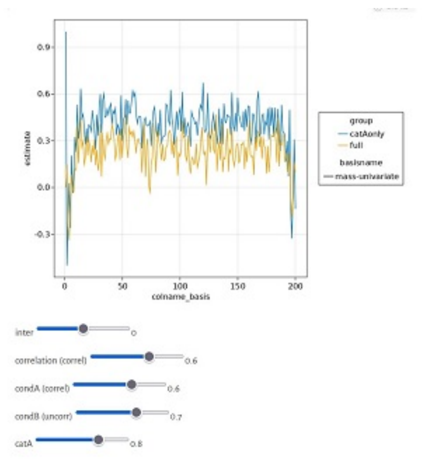
We ran simulation for 200 time points, 100 trials and 30 channels using Pluto notebook, `DataSimulation.jl`[3]

We then use the following equation to generate betas using Back-to-Back regression based on `StatsModel.jl`[4] formula.

$$y \leftarrow 1 + \texttt{catA} \tag{1}$$

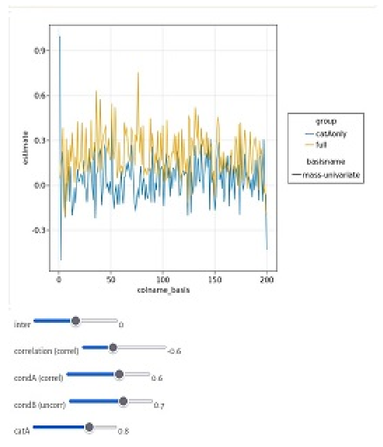
$$y \leftarrow 1 + \texttt{catA + condA + condB} \tag{2}$$

In the equations above, `catA` represents continuous predictor saccad_amplitude. `condA` represents continuous predictor saliency which correlates with `saccad_amplitude`. `condB` represents categorical predictor `face-effect`.



inter 0
correlation (correl) 0.6
condA (correl) 0.6
condB (uncorr) 0.7
catA 0.8

(a) Positive correlation between saccad_amplitude and saliency



inter 0
correlation (correl) -0.6
condA (correl) 0.6
condB (uncorr) 0.7
catA 0.8

(b) Negative correlation between saccad_amplitude and saliency

**Figure 5.** Comparison between betas using positive and negative correlations

We can see that, the back-to-back regression can even decode the negative correlation between the predictors. Though, we are not sure why and to what degree. we also observed that if the correlation has a high negative value, the margin of separation between the predictors become less.

## *4.3. With Regularization*

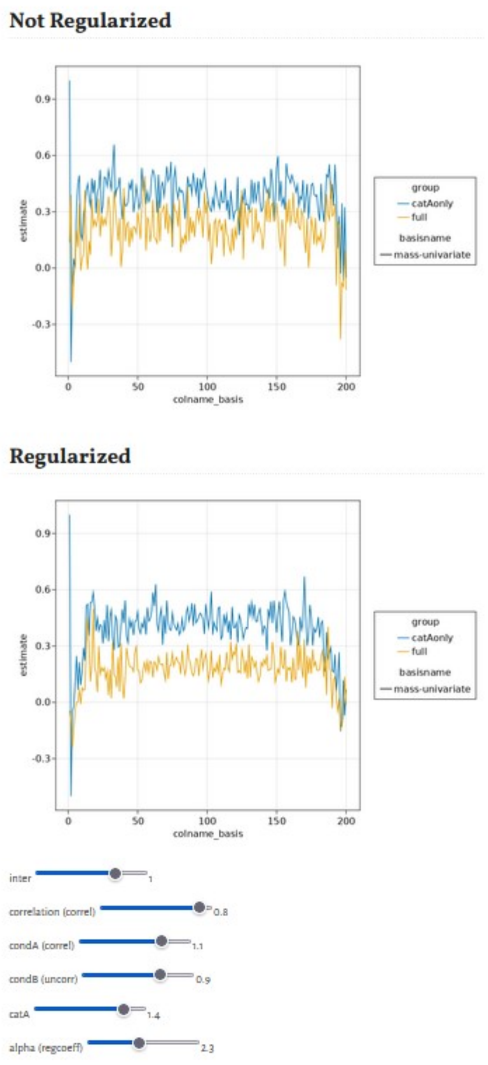
To apply regularization, we first do scaling of our features. This is required because regularization penalizes large values of all parameters equally. Hence, it is best to scale all features, otherwise a feature would be penalized much more than another feature. To perform feature scaling we use Z-score. Standardisation gives the advantage of interpreting the model as we would do with unregularised OLS regression.

[3]https://github.com/ssaket/eeg-b2b-regression/blob/main/DataSimulation.jl
[4]https://github.com/JuliaStats/StatsModels.jl/blob/master/docs/src/formula.md

$$\texttt{Z-score} = \frac{\texttt{X - mean}}{\texttt{standard deviation}}$$

### *4.3.1. Positive Correlation*

From figure 6, we can observe that the regularized the version gives better separation between the conditions, hence, can decode the the predictors very well. The *aplha* (regularized penalty) values can be further increased or decreased to achieve more or less regularization penalty. At `alpha = 0`, we see similar result as unregularized version.

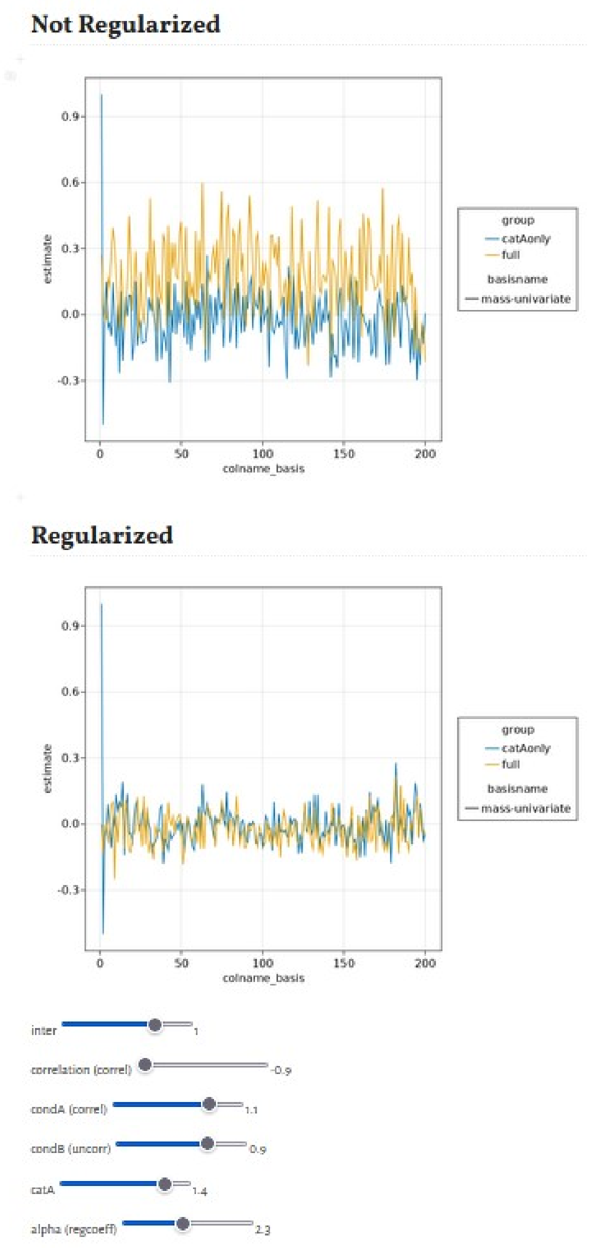


inter 1
correlation (correl) 0.8
condA (correl) 1.1
condB (uncorr) 0.9
catA 1.4
alpha (regcoeff) 2.3

**Figure 6.** Comparison between regularized and not regularized beta coefficients from back-to-back regression where the predicts have positive correlation

### *4.3.2. Negative Correlation*

Similarly, from figure 7, we can observe that the regularized version gives correct result when we have high negative correlation between the predictors whereas the un-regularized version still decodes some signal.



inter 1

correlation (correl) -0.9

condA (correl) 1.1

condB (uncorr) 0.9

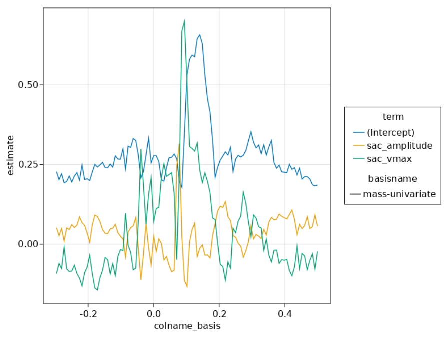
catA 1.4

alpha (regcoeff) 2.3

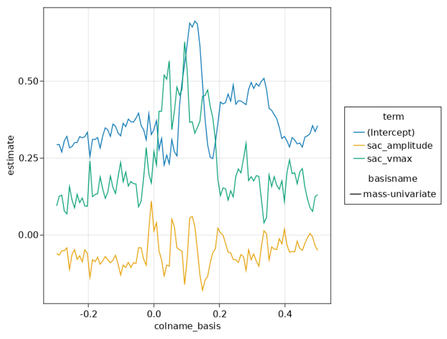
**Figure 7.** Comparison between regularized and not regularized beta coefficients from back-to-back regression where the predicts have negative correlation

### 4.4. Back-to-back regression on our dataset

We ran the back-to-back regression on subject 45 and subject 48, and we observed that it successfully disentangled (figure 8) the effect of sacc_amplitude and sacc_vax predictor with high correlation values from our EEG data.



(a) Back-to-back regression on subject 45



(b) Back-to-back regression on subject 48

**Figure 8.** Disentangling the effects of correlated continuous predictors

### 4.5. Effect of Saliency

We ran our saliency script[5] to generate the saliency maps using `fix_avgpos_x` and `fix_avgpos_y` and `sac_amplitude`. Figure 9 and Figure 10 shows the saliency maps for subject 45 and subject 48 on Pic ID 1, 2. Figure 11 shows the overall saliency score based on MSE error.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

where, n and m are the dimensions of image I and K

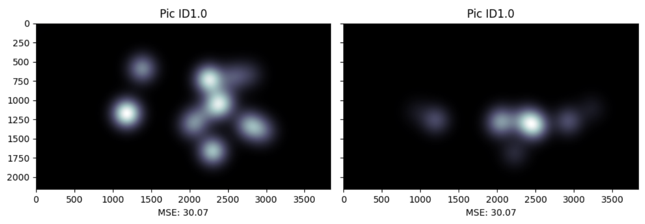


**Figure 9.** Comparison between Subject 45 and Subject 48

[5]https://gist.github.com/ssaket/e431b2d5b7aad65ec1b49ae49652cc86

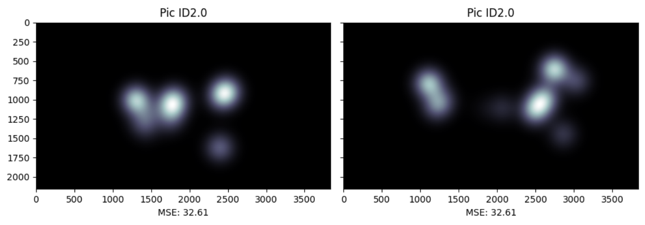


**Figure 10.** Comparison between Subject 45 and Subject 48

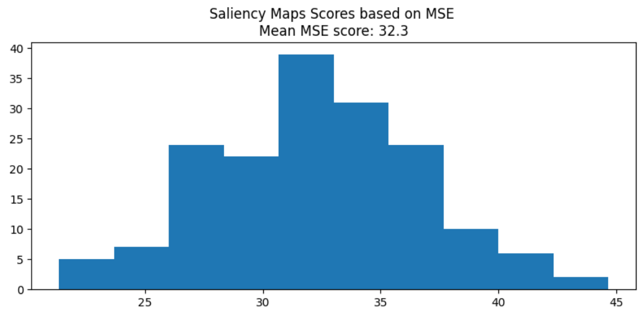


**Figure 11.** Overall Saliency score distribution

## 5. Limitations

- In our project, we use Gaussian Noise for generation predictors value based on the correlations whereas in pragmatic world this is not sufficient.
- We use MSE scores for comparing saliency maps, however literature shows that MSE is not very robust against comparing hidden patterns/structure in the images.
- While using back-to-back regression for decoding analysis, it is possible that we might have missed or overlooked something.

## Acknowledgement(s)

I would like to thank Jun.-Prof. Dr.Benedikt Ehinger for his guidance and support.

## References

[1] Holdgraf CR, Rieger JW, Micheli C, Martin S, Knight RT, Theunissen FE. Encoding and Decoding Models in Cognitive Electrophysiology. Front Syst Neurosci. 2017 Sep 26;11:61. doi: 10.3389/fnsys.2017.00061. PMID: 29018336; PMCID: PMC5623038.

[2] Gert, Anna & Ehinger, Benedikt & Timm, Silja & Kietzmann, Tim & König, Peter. (2021). Wild lab: A naturalistic free viewing experiment reveals previously unknown EEG signatures of face processing. 10.1101/2021.07.02.450779.

[3] Khairalseed, Mawia. (2015). A Comparison between Windowing FIR Filters for Extracting the EEG Components. Journal of Biosensor Bioelectronics ISSN: 2155-6210. 6. 2: 6. 10.4172/2155-6210.1000191.

[4] Ehinger BV & Dimigen O, Unfold: An integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis, peerJ, https://peerj.com/articles/7838/

[5] King JR, Charton F, Lopez-Paz D, Oquab M. Back-to-back regression: Disentangling the influence of correlated factors from multivariate observations. Neuroimage. 2020 Oct 15;220:117028. doi: 10.1016/j.neuroimage.2020.117028. Epub 2020 Jun 27. PMID: 32603859.