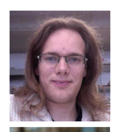
Post-hoc modification of linear models

Combining machine learning with domain information to make solid inferences from noisy data


Marijn van Vliet


Riitta Salmelin

Aalto University School of Science

Overview of the post-hoc framework

brain data → $\mathbf{X}$ behavioral data → $\mathbf{Y}$

1 Compute initial weight matrix

Take your favorite linear machine learning model (OLS, logistic regression, lSVM, ...) and fit it to the data. This yields the initial weight matrix $\mathbf{W}$.

model weights → $\mathbf{W}$

2 Decompose the weight matrix

A weight matrix is difficult to interpret. Hence, we use the "Haufe trick" [1] to decompose it into three components that are easier to work with.

covariance of $\mathbf{X}$ → $\Sigma_\mathbf{X}$ pattern matrix $\mathbf{P} = \Sigma_\mathbf{X}\mathbf{W}\Sigma_{\hat{\mathbf{Y}}}^{-1}$ → $\mathbf{P}$ covariance of model output, $\hat{\mathbf{Y}}$ → $\Sigma_{\hat{\mathbf{Y}}}$

3 Modify the components at will

Freely modify the components to inject domain information. See the example to the right for some ideas.

$\tilde{\Sigma}_\mathbf{X}$ $\tilde{\mathbf{P}}$ $\tilde{\Sigma}_{\hat{\mathbf{Y}}}$

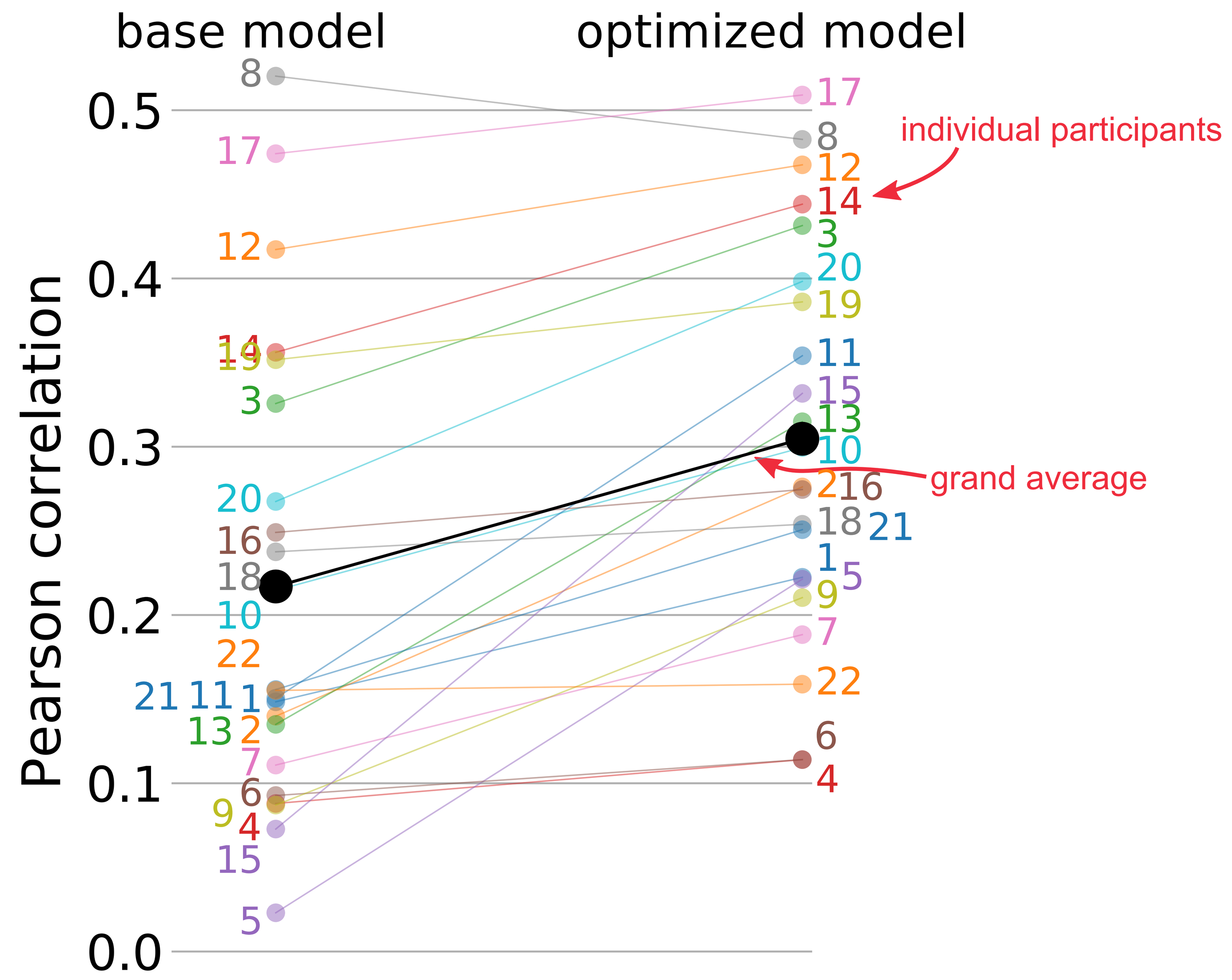
The ~ means "modified"

4 Reassemble the weight matrix

We perform the "inverse Haufe trick" to reassemble the components back into a weight matrix. This yields a new linear model that incorporates the changes we made to the components.

new model $\tilde{\mathbf{W}} = \tilde{\Sigma}_\mathbf{X}^{-1}\tilde{\mathbf{P}}\tilde{\Sigma}_{\hat{\mathbf{Y}}}$ → $\tilde{\mathbf{W}}$

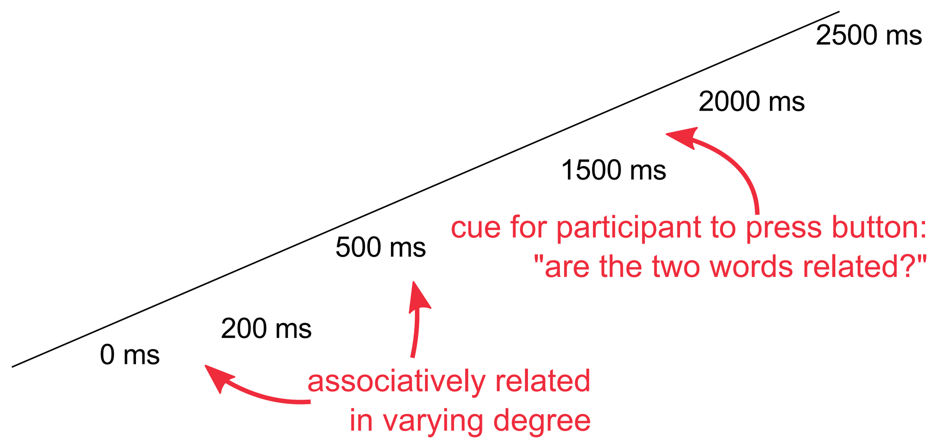
Model performance



Application to example study

- EEG was recorded from 22 participants, reading 200 sequentially presented word-pairs (BEAR-HONEY)
- Each word-pair has a forward association strength (FAS), which is a measure of relatedness, derived from a huge norm study [2].
- Decoding task: infer FAS from the EEG
- Evaluation metric: correlation between leave-one-out model output and ground-truth FAS
- We started with a ridge regression model [3], then injected domain information using the post-hoc framework.

Stimulus presentation



Meet the three components!

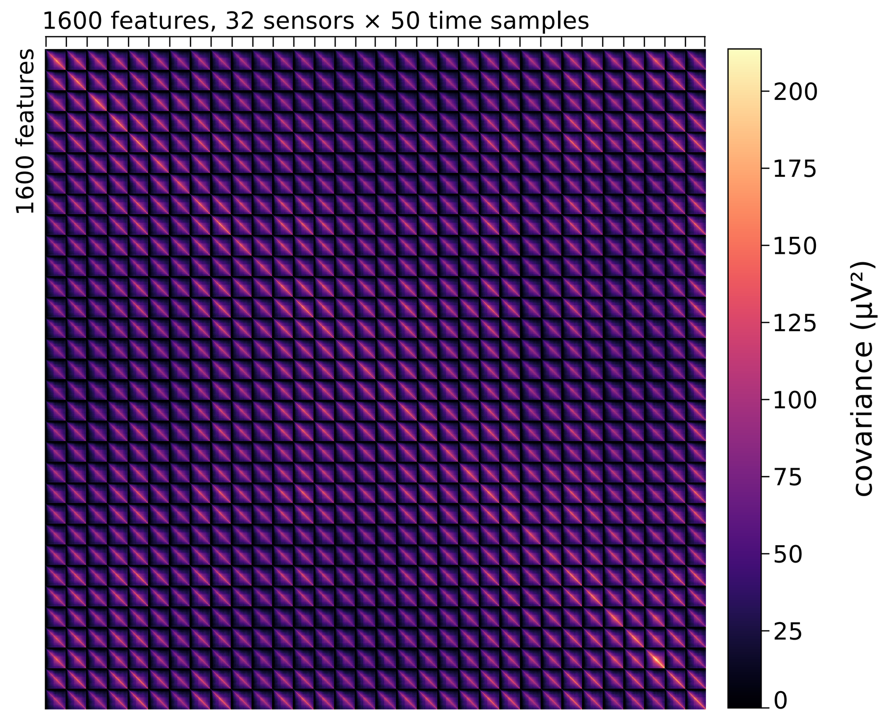
Instead of thinking about a linear model in terms of the weight matrix, we invite you to think about it in terms of three matrices:

the covariance, the pattern and the normalizer.

The three components each model a different aspect of the data.

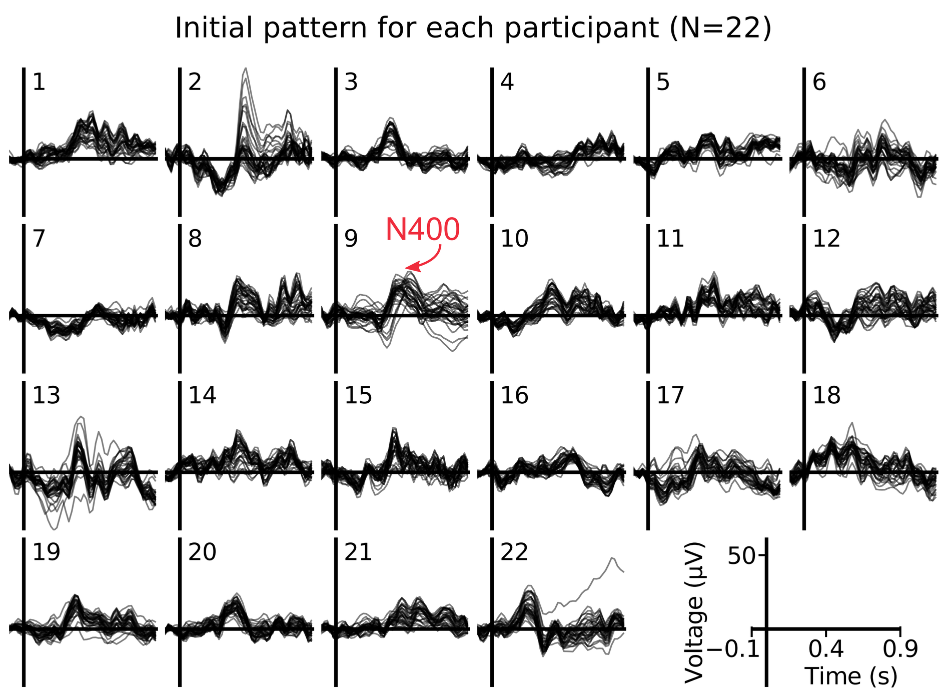
$\Sigma_\mathbf{X}$: the data covariance matrix

Models the relationship between the inputs



$\mathbf{P}$: the pattern matrix

Models the signal part of the data



$\Sigma_{\hat{\mathbf{Y}}}$: the normalizer

Models the relationship between the outputs

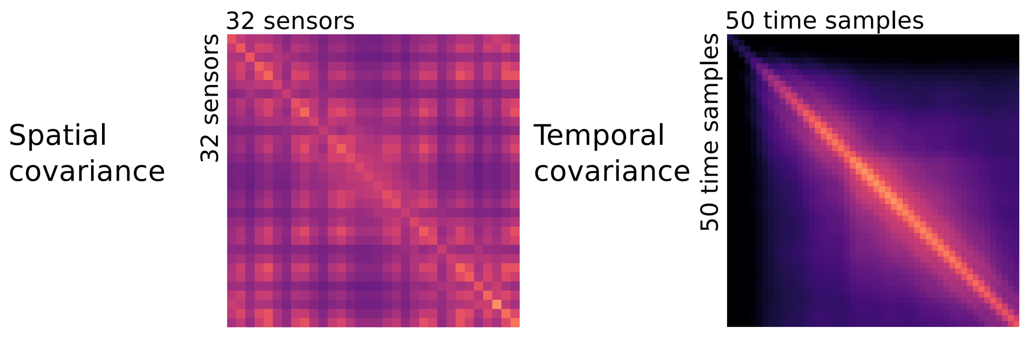
0.7

in our example, a single number indicating the scaling of the output

Modifying the data covariance

idea: "Kronecker shrinkage"

Looking at the figure above, the full covariance matrix can be approximated using the Kronecker product between the spatial and temporal covariance matrices:



This is because our data is spatio-temporal in nature. We can leverage this domain information when applying shrinkage (a common way of regularizing the model) by applying shrinkage to each component separately.

Modifying the normalizer

idea: apply "weight normalization" idea from LCMV beamformers

When we modify the covariance and/or the pattern, the scaling of the output of the model will change. The normalizer can be used to impose a standardized scaling. For example, by setting the normalizer to the following:

$$\tilde{\Sigma}_{\hat{\mathbf{Y}}} = (\tilde{\mathbf{P}}^\mathsf{T}\tilde{\Sigma}_\mathbf{X}^{-1}\tilde{\mathbf{P}})^{-1}$$

we ensure that the pattern always passes through our model with unit gain.

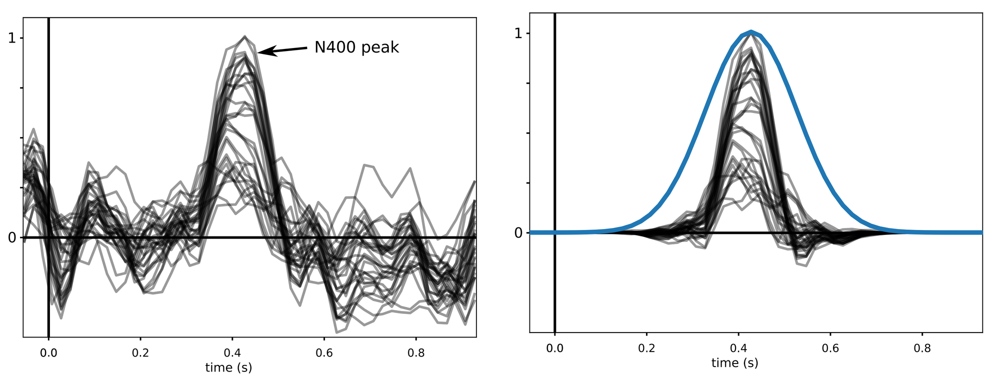
Modifying the pattern matrix

Informing the model of the n400 ERP component

The pattern matrix offers an intuitive way to inform the model about the signal of interest. In our example study on semantic priming, we bias our model to home in on the N400 potential [4].

idea 1: restrict pattern in time

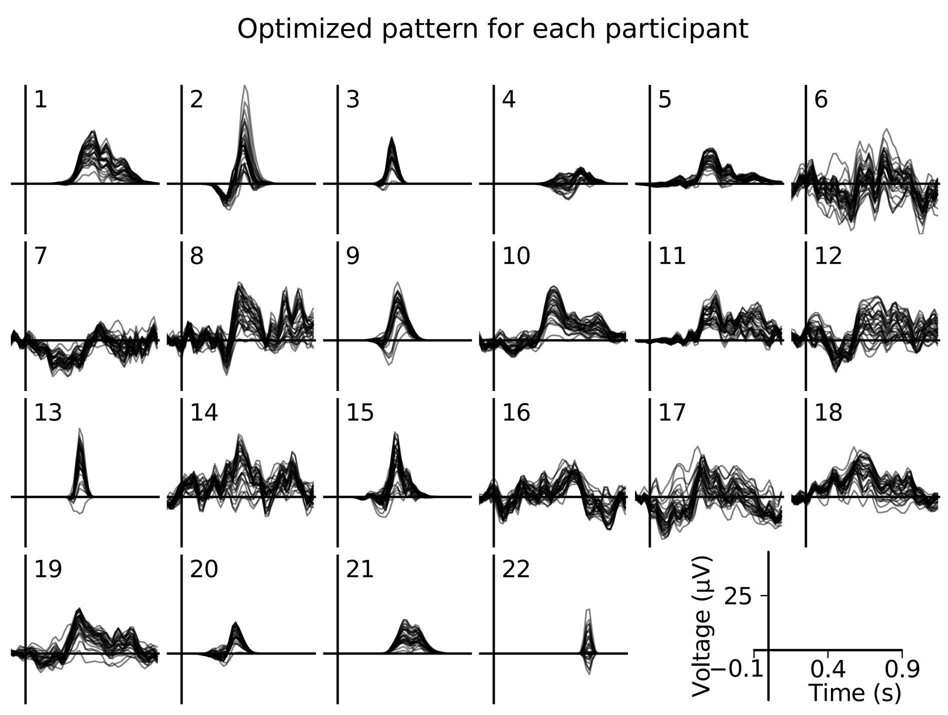
We multiply the pattern with a Gaussian kernel to emphasize the region around 400 ms. after stimulus onset. Width of the kernel becomes a hyperparameter.



idea 2: bias the pattern towards the grand average

$$\tilde{\mathbf{P}} = \rho\bar{\mathbf{P}} + (1-\rho)\mathbf{P}$$

grand-average pattern; how much to bias towards the average



References

[1] Haufe et al. 2014, "On the interpretation of weight vectors of linear models in multivariate neuroimaging" https://doi.org/10.1016/j.neuroimage.2013.10.067

[2] De Deyne & Storms 2008, "Word associations: network and semantic properties" https://doi.org/10.3758/BRM.40.1.213

[3] Rifkin & Lippert 2007, "Notes on regularized least squares" http://cbcl.mit.edu/publications/ps/MIT-CSAIL-TR-2007-025.pdf

[4] Kutas & Federmeier 2011, "Thirty years and counting: finding meaning in the N400 component of the event-related potential(ERP)" https://doi.org/10.1146/annurev.psych.093008.131122

code and full paper