January 11, 2019
This is a preprint that has not undergone peer review yet.

# Post-hoc modification of linear models: combining machine learning with domain information to make solid inferences from noisy data

Marijn van Vliet[1*] and Riitta Salmelin[1]

[1]Department of Neuroscience and Biomedical Engineering, Aalto University
[*]Corresponding author: marijn.vanvliet@aalto.fi

## Abstract

Linear machine learning models are a powerful tool that can "learn" a data transformation by being exposed to examples of input with the desired output, forming the basis for a variety of powerful techniques for analyzing neuroimaging data. However, their ability to learn the desired transformation is limited by the quality and size of the example dataset, which in neuroimaging studies is often notoriously noisy and small. In these cases, it is desirable to fine-tune the learned linear model using domain information beyond the example dataset. To this end, we present a framework that decomposes the weight matrix of a fitted linear model into three subcomponents: the data covariance, the identified signal of interest, and a normalizer. Inspecting these subcomponents in isolation provides an intuitive way to inspect the inner workings of the model and assess its strengths and weaknesses. Furthermore, the three subcomponents may be altered, which provides a straightforward way to inject prior information and impose additional constraints. We refer to this process as "post-hoc modification" of a model and demonstrate how it can be used to achieve precise control over which aspects of the model are fitted to the data through machine learning and which are determined through domain knowledge. As an example use case, we decode the associative strength between words from electroencephalography (EEG) reading data. Our results show how the decoding accuracy of two example linear models (ridge regression and logistic regression) can be boosted by incorporating information about the spatio-temporal nature of the data, domain knowledge about the N400 evoked potential and data from other participants.

*Highlights:*

- We present a framework to decompose any linear model into three subcomponents that are straightforward to interpret.
- By modifying the subcomponents before re-assembling them into a linear model, prior information and further constraints may be injected into the model.
- As an example, we boost the performance of a linear regressor and classifier by injecting knowledge about the spatio-temporal nature of the data, the N400 evoked potential and data from other participants.

*Keywords:* multivariate analysis, linear model, prior knowledge, event-related potentials, N400, EEG

## 1 Introduction

Linear models are the workhorse behind many of the multivariate analysis techniques that are used to process neuroimaging data,[1] with applications ranging from signal decomposition[2 3 4] to source modeling[5 6 7 8] and signal decoding.[9 10 11] Even though they may serve very different purposes, the data transformation performed by all linear techniques can be mathematically described by a single matrix multiplication between the input data and a "weight matrix". From this point of view, the key difference between the various techniques is how the weight matrix is computed.

Supervised linear machine learning algorithms compute the weight matrix based on examples of the input data and the desired output.[12] This class of algorithms have advanced the analysis of neuroimaging data on two important fronts. First, by learning what is signal and what is noise, the signal can be projected away from noise sources, which provides an alternative method to increase signal-to-noise ratio (SNR) to signal averaging. This makes it for example possible to perform single-subject and even single-trial analysis.[13 14 15] Second, by focusing on patterns rather than individual data points, there is no longer a requirement for a one-to-one correspondence between the experimental manipulation and a change in the signal at a certain location, time, or frequency, which enables more ambitious neuroimaging studies.[16 17]

The success of machine learning algorithms to find the desired transformation is for a large part dependent on the ratio between the number of parameters that need to be estimated and the number of provided training examples. In general, the more parameters that need to be estimated, the more training data is needed to prevent overfitting of the model.[18 19] Unfortunately, it is common in neuroimaging studies for the data dimensionality to exceed the number of trials in a recording, in which case restrictions need to be placed on the model in order to force a unique solution. Especially in these cases, it is desirable to inspect the data transformation that was "learned" by the algorithm to understand what aspects of the data contribute to the output of the model, identify possible problems, and possibly impose further restrictions on the model if the transformation was unsatisfactory.

In linear models, there are some effective general purpose approaches to place restrictions on the learned data transformation, notably $\ell_1$ regularization,[20] which enforces sparsity of the weight matrix, and $\ell_2$ regularization,[21] which enforces a small magnitude of the individual weights. Moving beyond these approaches, imposing further restrictions that are motivated by domain knowledge may lead to even better performance of the model. However, it is in practice very difficult to express domain knowledge in terms of the weight matrix,[22] since interpreting this matrix is not straightforward when there are co-linearities in the data, which is almost always the case in neuroimaging.

To facilitate the interpretation of linear models, Haufe et al. (2014) introduced a way to transform the weight matrix into a pattern matrix, which is more straightforward to interpret (see section 2.2). It is therefore also easier to formulate domain knowledge in terms of the pattern than the weights. For example, in a source estimation setting, the pattern matrix is the leadfield (i.e., forward solution) and the weight matrix is the inverse solution. Bayesian methods for computing the inverse solution formulate their domain knowledge driven priors on the leadfield, rather than the inverse solution.[23 24] In this paper, we combine the insight of Haufe et al. (2014) that a pattern matrix can be computed for any linear model, with the insight from source estimation methods that priors that are formulated on the pattern matrix can be translated into priors on the weight matrix.

We propose a framework that decomposes the weight matrix of a linear model into three

1 McIntosh and Mišić, 2013
2 Uusitalo and Ilmoniemi, 1997
3 Jutten and Herault, 1991
4 Vigario, Sarela, Jousmäki, Hamalainen, and Oja, 2000
5 Hämäläinen and Ilmoniemi, 1994
6 Matsuura and Okabe, 1995
7 Van Veen, van Drongelen, Yuchtman, and Suzuki, 1997
8 Gross et al., 2001
9 Grootswagers, Wardle, and Carlson, 2017
10 Lotte, Congedo, Lecuyer, Lamarche, and Arnaldi, 2007
11 Tong and Pratte, 2012
12 Hastie, 2009
13 Pernet, Sajda, and Rousselet, 2011
14 Parra et al., 2003
15 van Vliet et al., 2016
16 Mitchell et al., 2008
17 Huth, Heer, Griffiths, Theunissen, and Jack, 2016
18 Babyak, 2004
19 Blankertz, Lemm, Treder, Haufe, and Müller, 2011
20 Tibshirani, 1996
21 Rifkin and Lippert, 2007
22 Haufe et al., 2014
23 Wipf and Nagarajan, 2009
24 Trujillo-Barreto, Aubert, and Penny, 2008

subcomponents: 1) the data covariance, 2) the pattern matrix, and 3) the normalizer (see section 2.3). Inspecting these subcomponents in isolation offers an intuitive way to gain insights into the functioning of the model and possible problem points. Importantly, we may modify each component to impose new constraints and incorporate domain knowledge, before recomposing the subcomponents back into a weight matrix. By doing so, we are not only able to improve the performance of linear analysis techniques, but also gain a deeper insight into the similarities and differences between different linear models, as seemingly unrelated techniques appear in this framework as variations of the same underlying theme (see section 4).

Since the decomposition-modification-recomposition cycle of the weight matrix takes place after the initial model has been constructed through a conventional machine learning algorithm, we refer to this process as "post-hoc modification".

While the framework is agnostic to the methods by which the initial linear model was constructed, and is hence applicable to a wide variety of data analysis techniques, we will use linear regression as an example throughout this paper to provide context to our procedures and equations. To provide practical examples, we demonstrate several ways in which the framework may be used to combine machine learning with domain information to decode the associative strength between words from an EEG recording, following a semantic priming protocol.[25][26] We explore a regression scenario with a ridge regressor as a base model, and also a classification scenario with a logistic regressor. Using the post-hoc modification framework, these two general purpose models were modified to incorporate 1) the fact that there is a both a dependency between EEG sensors and time samples, 2) data recorded from the other participants, and 3) the timing of the N400 component of the event-related potential (ERP), which occurs around 400 ms after the onset of the second word stimulus.[27][28]

[25] Neely, 1991

[26] van Vliet et al., 2014

[27] Kutas and Hillyard, 1980

[28] Kutas and Federmeier, 2011

## 2 Methods

### 2.1 Linear models

Since the post-hoc modification framework can operate on any type of linear model, regardless of function and type of data, we have chosen to adopt the general purpose terminology used in the machine learning literature.[29] See Table 1 for a summary of the mathematical symbols used in this paper.

[29] Hastie, 2009

We will refer to a data instance, for example a single epoch of EEG data or a single functional magnetic resonance imaging (fMRI) image, as an "observation". An observation consists of $m$ "features", for example the voltage at each sensor and each each time point of an epoch, or the beta weight for each voxel in an fMRI image. In this manner, a single observation is described by row vector $\mathbf{x} \in \mathbb{R}^{1\times m}$ and an entire data set, consisting of $n$ observations, by matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$.

A linear model transforms the input data by making a linear combination of the $m$ features to produce output data with $k$ dimensions, referred to as "targets". In machine learning, the desired transformation is deduced by exposing the algorithm to an example input data set $\mathbf{X}$ along with the desired output $\mathbf{Y} \in \mathbb{R}^{n\times k}$. This process is referred to as "training" the model.

To simplify the equations, it is assumed w.l.o.g. that the columns of both $\mathbf{X}$ and $\mathbf{Y}$ have zero mean. In practice, this can be achieved by removing the column-wise mean from $\mathbf{X}$ and $\mathbf{Y}$ before entering them into the model and adding the removed offsets back to the output. Under the zero-mean assumption, the data transformation that is performed by a linear model can

| | |
|---|---|
| $k$ | Number of targets |
| $m$ | Number of features describing an observation |
| $n$ | Number of observations in a dataset |
| $\mathbf{x}$ | A row vector of length $m$ that describes a single observation |
| $\mathbf{X}$ | A dataset consisting of $n$ observations |
| $\widehat{\mathbf{X}}$ | An approximation of $\mathbf{X}$ |
| $\Sigma_{\mathbf{X}}$ | The empirical covariance matrix of $\mathbf{X}$ |
| $\widetilde{\Sigma}_{\mathbf{X}}$ | A modified version of the empirical covariance matrix of $\mathbf{X}$ |
| $\mathbf{y}$ | A row vector of length $k$ that describes the desired output for a single example observation |
| $\mathbf{Y}$ | The desired output of a model for $n$ example observations |
| $\widehat{\mathbf{Y}}$ | The actual output of a model, here an approximation of $\mathbf{Y}$ |
| $\Sigma_{\widehat{\mathbf{Y}}}$ | The empirical covariance matrix of $\widehat{\mathbf{Y}}$, also referred to as the normalizer |
| $\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}$ | A modified normalizer |
| $\mathbf{W}$ | The weight matrix describing a linear transformation from $\mathbf{X}$ to $\widehat{\mathbf{Y}}$ |
| $\widetilde{\mathbf{W}}$ | The updated weight matrix obtained by combining $\widetilde{\Sigma}_{\mathbf{X}}$, $\widetilde{\mathbf{P}}$ and $\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}$ |
| $\mathbf{P}$ | The pattern matrix describing a linear transformation from $\widehat{\mathbf{Y}}$ to $\widehat{\mathbf{X}}$ |
| $\widetilde{\mathbf{P}}$ | A modified pattern matrix |
| $\mathbf{I}$ | An identity matrix of appropriate size |
| $\lambda$ | Controls the amount of $\ell 2$ regularization of the covariance matrix |
| $\alpha$ | Controls the shrinkage of the spatial component of the covariance matrix |
| $\beta$ | Controls the shrinkage of the temporal component of the covariance matrix |
| $\mu$ | Controls the center of the Gaussian kernel used as a windowing function for the pattern matrix |
| $\sigma$ | Controls the width of the Gaussian kernel used as a windowing function for the pattern matrix |
| $\rho$ | Controls the weighting between the pattern matrix for the current recording and the grand-average pattern matrix across all other recordings |

**Table 1:** A summary of the mathematical symbols used in this paper.

be represented by a multiplication between $\mathbf{X}$ and a weight matrix $\mathbf{W} \in \mathbb{R}^{m \times k}$:

$$\widehat{\mathbf{Y}} = \mathbf{X}\mathbf{W}, \tag{1}$$

where $\widehat{\mathbf{Y}} \in \mathbb{R}^{n \times k}$ denotes the output of the model. In the case of machine learning, $\mathbf{W}$ is chosen such that $\widehat{\mathbf{Y}}$ approximates $\mathbf{Y}$.

## 2.2 The pattern matrix

Haufe et al. (2014) showed the relationship between a linear model $\mathbf{W}$ that transforms $\mathbf{X}$ into $\widehat{\mathbf{Y}}$ and a linear model $\mathbf{P} \in \mathbb{R}^{m \times k}$ that does the opposite and approximates $\mathbf{X}$ given $\widehat{\mathbf{Y}}$:

$$\mathbf{P} = \Sigma_{\mathbf{X}}\mathbf{W}\Sigma_{\widehat{\mathbf{Y}}}^{-1}, \tag{2}$$

$$\widehat{\mathbf{X}} = \widehat{\mathbf{Y}}\mathbf{P}^{\mathsf{T}}. \tag{3}$$

In the above equations, $\Sigma_{\mathbf{X}}$ is the (empirical) covariance matrix of $\mathbf{X}$, $\Sigma_{\widehat{\mathbf{Y}}}^{-1}$ is the inverse of the (empirical) covariance matrix of $\widehat{\mathbf{Y}}$ and $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times m}$ is an approximation of $\mathbf{X}$. We refer to $\mathbf{P}$ as the pattern matrix and it can be interpreted in many ways. For example, $\mathbf{P}$ can be interpreted as the inverse of the linear transformation performed by $\mathbf{W}$. In a source estimation setting, $\mathbf{P}$ can be regarded as the leadfield and $\mathbf{W}$ as the inverse solution. Alternatively, if $\mathbf{W}$ is a decoding model, then $\mathbf{P}$ is the corresponding encoding model. Another useful interpretation is to regard $\mathbf{P}$ as the signal of interest and $\mathbf{W}$ as a filter that isolates this signal from the rest of the data (see Figure 1 for a visual guide).

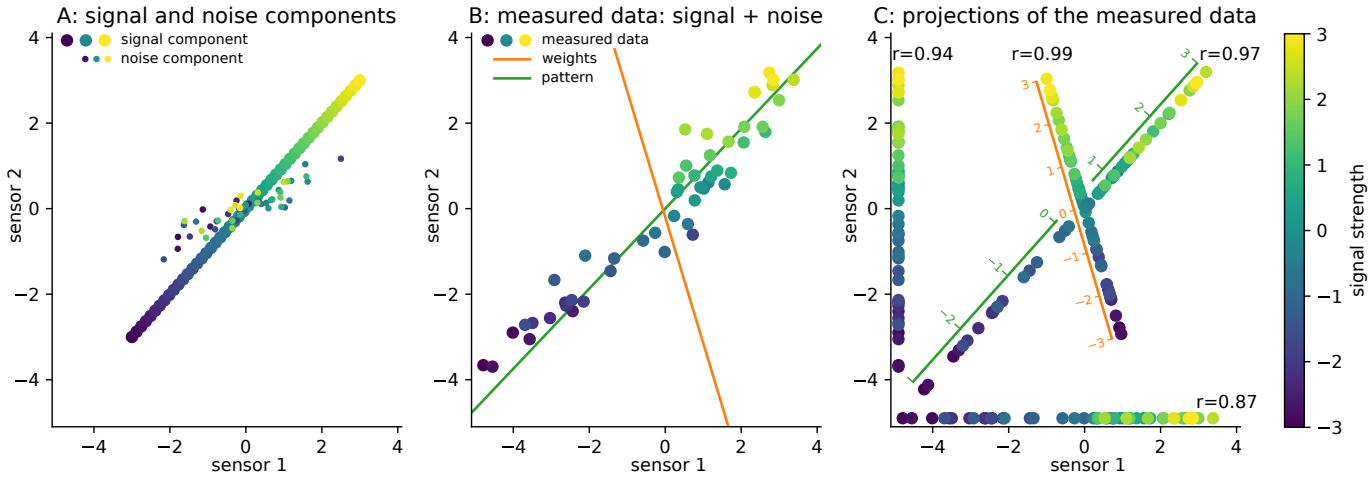


**Figure 1: Visual explanation of the filter/signal interpretation of the weight and pattern matrices.** This is a simulation of a signal that is being observed through two sensors. Dots represent observations of the signal and the color of the dots indicates the true signal strength during each observation. Linear regression is used to decode the true signal strength from the observed data.
**A:** The simulated data consists of two components. The first component (large dots) dictates how the signal is measured by the sensors (i.e. the encoding model). In this simulation, there is a one-to-one relationship between the true signal strength and the measurements at both sensors. The second component (small dots) is simulated using random numbers drawn from a two-dimensional Gaussian distribution and is a simulation of noise that is unrelated to the strength of the signal.
**B:** The data that is recorded by the sensors (large dots) is the summation of both the signal and noise components. A linear regression model was trained on these observations, with the true signal strength as target, to determine the optimal linear transformation to map the measured data to signal strength. In visual terms, the task of the model is to decode the color of a dot, based on its location in the graph. In this two-dimensional example, the model's weights can be visualized as a line (orange line). We see that the direction of the regression line is dictated by the noise rather than the signal component, which is why the weight matrix is so hard to interpret. In a similar fashion, the corresponding pattern (computed using equation 2) can also be visualized as a line (green), which approximates the signal component.
**C:** Applying the linear regression to the data is equivalent to projecting the measured data onto the regression line (orange axis). By projecting the data orthogonal to the noise, a near perfect reconstruction of the signal strength can be obtained, which explains why the direction of the regression line is dictated by the noise component. For comparison, the result of projecting the measured data onto several other lines is also shown: the pattern line (green axis) in the direction of the signal component, the horizontal axis and the vertical axis. For each projection, the Pearson correlation between the signal strength and the position along the projection line is provided.

An interactive version of this figure is available at https://users.aalto.fi/~vanvlm1/posthoc, where the noise component can be manipulated to study its effect on the direction of the regression line.

### 2.3 Post-hoc modification

When we solve for $\mathbf{W}$ in equation 2, we obtain:

$$\mathbf{W} = \Sigma_{\mathbf{X}}^{-1} \mathbf{P} \Sigma_{\hat{\mathbf{Y}}}, \tag{4}$$

and observe that a weight matrix may be constructed from three subcomponents:

1. the covariance matrix of the data $\Sigma_{\mathbf{X}}$ which describes the scale of the features and their relationship
2. the pattern matrix $\mathbf{P}$ which describes the signal of interest
3. the normalizer $\Sigma_{\hat{\mathbf{Y}}}$ which, in the case of $k = 1$ describes the scale of the result, or in the case of $k > 1$ also the relationship between the targets

Since the above subcomponents are easier to interpret and manipulate than the weight matrix, we propose a framework in which the weight matrix is first decomposed into these subcomponents, which may then be manipulated at will, and finally recomposed into an updated weight

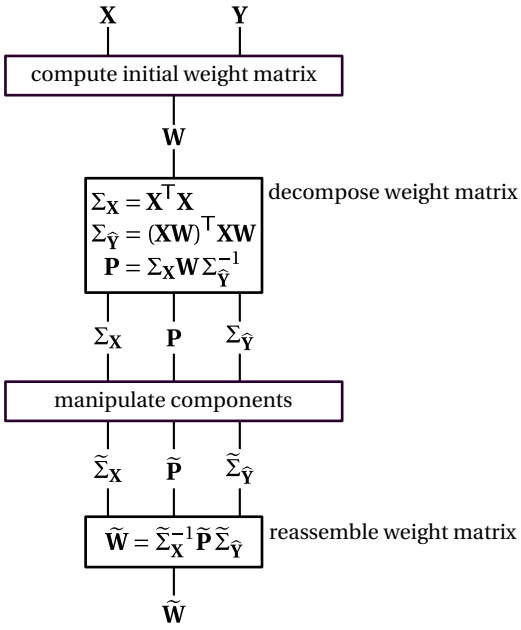


**Figure 2: The post-hoc modification framework.** First, the initial linear model $\mathbf{W}$ is constructed. This can for example be done with a general purpose linear machine learning algorithm. Then, using equation 2, $\mathbf{W}$ is decomposed into data covariance $\Sigma_{\mathbf{X}}$, pattern $\mathbf{P}$ and normalizer $\Sigma_{\hat{\mathbf{Y}}}$. These subcomponents can then be manipulated at will to impose further restrictions on the model or inject prior information. Finally, the modified subcomponents $\widetilde{\Sigma}_{\mathbf{X}}$, $\widetilde{\mathbf{P}}$ and $\widetilde{\Sigma}_{\hat{\mathbf{Y}}}$ are reassembled into an updated linear model $\widetilde{\mathbf{W}}$.

matrix (Figure 2):

$$\widetilde{\mathbf{W}} = \widetilde{\Sigma}_{\mathbf{X}}^{-1} \widetilde{\mathbf{P}} \widetilde{\Sigma}_{\hat{\mathbf{Y}}}, \tag{5}$$

where $\widetilde{\Sigma}_{\mathbf{X}}$ is a modified version of the data covariance, $\widetilde{\mathbf{P}}$ is a modified version of the pattern matrix, $\widetilde{\Sigma}_{\hat{\mathbf{Y}}}$ is a modified version of the normalizer, and $\widetilde{\mathbf{W}}$ is the updated weight matrix that reflects the changes made to the subcomponents.

We will now go over some examples of how the subcomponents may be modified to improve the performance of a linear model in a neuroimaging setting. In these examples, the task of the linear model is to decode the forward association strength (FAS) between two words,[30] based on an EEG recording of a participant reading the word-pair during a semantic priming experiment.

[30] Nelson, McEvoy, and Dennis, 2000

## 2.4 EEG recordings

The decoding performance of two linear models were evaluated on an EEG dataset, which was recorded with 24 participants (7 female, aged 22–38, mixed handedness and all native speakers of Flemish-Dutch). Two recordings were dropped from the study: one was dropped due to problems with the stimulus synchronization signal and the other due to excessive sensor impedance. Participants signed an informed consent form prior to participating. Ethical approval of these studies was granted by an independent ethical committee ("Commissie voor Medische Ethiek" of the UZ Leuven, Belgium). These studies were conducted according to the most recent version of the declaration of Helsinki.

The participants read a series of sequentially presented words, organized in *prime–target* pairs, and pressed one of two mouse buttons to indicate whether the two words of a word-pair were related or not. The hand used to hold the mouse and the assignment of buttons to "yes"/"no" responses was counterbalanced across participants.

The prime word was presented for 200 ms and the target word for 2000 ms with a stimulus onset asynchrony (SOA) of 500 ms. Words were presented in white on a black background, rendered in the Arial font. Since a speeded button response task will generate ERP components that can mask N400 modulations,[31] the participants were instructed to delay their button

[31] van Vliet et al., 2014

response until the target word turned yellow, which happened 1000 ms after the onset of the target word. The participants had 1000 ms to respond, or else a non-response code would be logged for the trial.

In addition to capturing the button response of the participant, EEG was recorded continuously using 32 active electrodes (extended 10–20 system) with a BioSemi Active II System, having a 5th order frequency filter with a pass band from 0.16 Hz to 100 Hz, and sampled at 256 Hz. An electro-oculogram (EOG) was recorded simultaneously using the recommended montage outlined by Croft and Barry (2000). Two final electrodes were placed on both mastoids and their average was used as a reference for the EEG.

## 2.5 Stimuli

The stimuli consisted of Flemish-Dutch word pairs (see section 2.12) with varying FAS between the two words in each pair, as measured by a large-scale norming study performed by De Deyne and Storms (2008). In this norm dataset, FAS is defined as the number of participants, out of 100, that wrote down the target word in response to the prime word in a free association task.

The stimuli used in the experiment were the top 100 word-pairs with highest FAS in the norm dataset and 100 word-pairs with an assumed FAS of zero that were matched in length, frequency and in-degree. Each word-pair with a high FAS consisted of words with a length of 3 to 10 letters, with no restrictions on frequency or in-degree. To construct the low FAS pairs, for each word in the high FAS condition, a random word was selected with equal length, frequency and in-degree (or, if no such word existed, a word that matched these as close as possible), and random pairings were made from the resulting words.

## 2.6 Data preprocessing

All data processing was performed using the MNE-Python[32] and auto-reject[33] software packages. The EEG was bandpass filtered offline between 0.1 Hz and 50 Hz by a 4th order zero-phase Butterworth filter to attenuate large drifts and irrelevant high frequency noise, but retain eye movement artifacts. Individual epochs were obtained by cutting the continuous signal from 0.2 s before the onset of each target stimulus to 1 s after. Baseline correction was performed using the average voltage in the interval before the stimulus onset (−200 ms to 0 ms) as baseline value. The random sample consensus (RANSAC) algorithm was used to detect excessively noisy channels, and those signals were subsequently replaced by interpolating the signals from nearby sensors using spherical splines.[34] Two EOG artifact elimination passes were performed on the data. First, the EOG channels were used to attenuate eye artifacts from the EEG signal using the regression method outlined in Croft and Barry (2000). Second, the data was decomposed using independent component analysis (ICA) and any components that correlated strongly with one or more EOG channels were removed. Next, the signal was band pass filtered further using a tight passband around the frequency range in which the N400 component was found, namely between 0.5 Hz and 15 Hz, by a 4th order zero-phase Butterworth filter and downsampled to 50 Hz to reduce the dimensionality of the data. Further artifacts were removed using the autoreject procedure,[35] which flags and interpolates noisy channels in each individual epoch by measuring how well data from other epochs predicts the data of the epoch currently under consideration. While autoreject can also flag and remove noisy epochs, this functionality was disabled to ensure no epochs were dropped from the data.

[32] Gramfort et al., 2013

[33] Jas, Engemann, Bekhti, Raimondo, and Gramfort, 2017

[34] Perrin, Pernier, Bertrand, and Echallier, 1989

[35] Jas et al., 2017

A full report of the data preprocessing steps can be found at:
https://users.aalto.fi/~vanvlm1/posthoc.

## 2.7 Initial linear models

In this paper, we give some examples on how to use the post-hoc modification framework to inject domain knowledge into two general purpose machine learning models. For the regression scenario, we chose the ridge regressor as implemented in the Scikit-Learn package[36] as the base model, and for the classification scenario the logistic regressor from the same package was chosen. These two particular models were chosen because they are widely used in neuroimaging and their performance on our example datasets is equal or better than other commonly used linear models (e.g. shrinkage linear discriminant analysis (LDA) or linear support vector machine (lSVM)).

[36] Pedregosa et al., 2012

Each epoch of the recording served as a single observation for the model, and the corresponding row-vector $\mathbf{x}$ was obtained by concatenating the timecourses recorded at all EEG sensors. The resulting vectors formed the rows of input matrix $\mathbf{X}$, resulting in $\mathbf{X} \in \mathbb{R}^{200\times1600}$. In the regression scenario, the desired output of the model, $\mathbf{Y} \in \mathbb{R}^{200\times1}$, was specified as the log-transformed FAS of the word-pair presented during each epoch.[37] In the classification scenario, $\mathbf{Y}$ was formed by specifying 1 if the word-pair presented during the epoch consisted of two associatively related words, and $-1$ otherwise.

[37] van Vliet et al., 2016

To evaluate the overall performance of a model, 10-fold crossvalidation was used, where nine folds were used as training data and one fold was used as test data. By repeating this ten times, such that each fold has been used as test data once, and collecting the output of the model for each run, the full matrix $\widehat{\mathbf{Y}}$ was constructed, containing the crossvalidated model output for each epoch. The performance of the model, $p$, was then quantified in the regression scenario using the Pearson correlation between $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$, and in the classification scenario using the classification accuracy.

Normalization of $\mathbf{X}$ was performed inside the crossvalidation loop as well, such that the mean and standard deviation of each feature across observations was computed on the training data only, and subsequently used to normalize the features of the test data.

## 2.8 Strategies for modifying the covariance matrix

The input data $\mathbf{X}$ that is given to the linear model is a set of preprocessed EEG epochs. Each epoch is encoded as a row vector $\mathbf{x}$ by concatenating the timeseries (consisting of 50 samples), as recorded at each of the 32 sensors, resulting in $m = 50 \times 32 = 1600$ features. Because we have a maximum of $n = 200$ epochs available for each participant, the problem of estimating 1600 weights from the data of a single participant is massively underspecified and the model will overfit.[38]

[38] Babyak, 2004

A common way to alleviate overfitting in linear models is to introduce regularization when estimating the covariance matrix during the training of the model. For example, with $\ell 2$ regularization, a trade-off is made between maximizing the fit between $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$ and minimizing the absolute value of the weights $\|\mathbf{W}\|$, which prevents the model from placing too much emphasis on a single feature.[39][40] The initial ridge regression model we use as a base model (section 2.7) already implements such regularization.

[39] Rifkin and Lippert, 2007

[40] Hastie, 2009

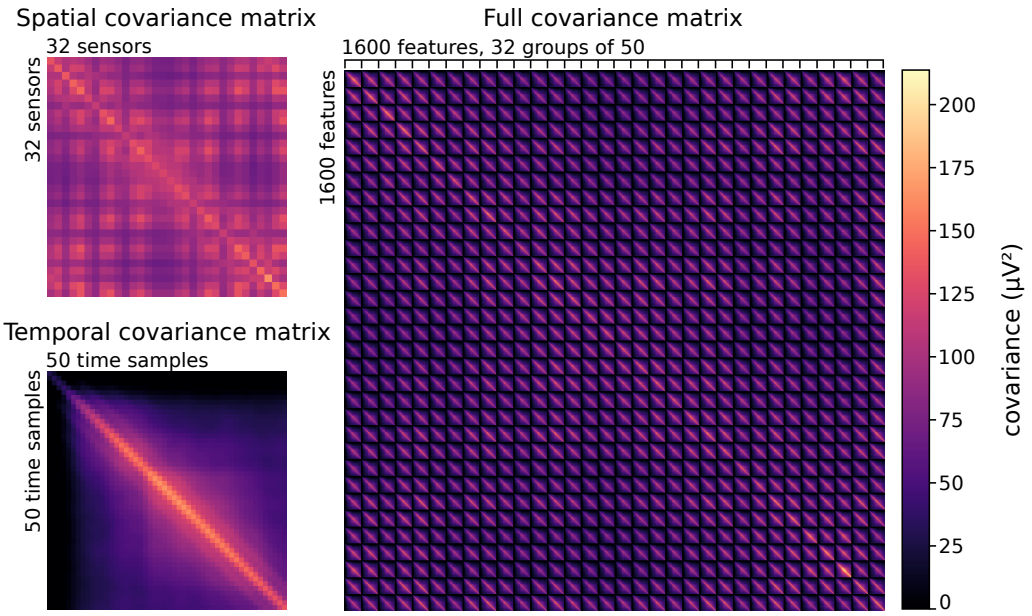


**Figure 3:** Shown on the right is the grand average covariance matrix. This matrix can be approximated with the Kronecker product of the grand average spatial covariance matrix (upper left) and grand average temporal covariance matrix (bottom left).

The post-hoc modification framework can be used to impose $\ell 2$ regularization on any linear model by adding a constant value to each diagonal element of the covariance matrix $\Sigma_{\mathbf{X}}$:

$$\widetilde{\Sigma}_{\mathbf{X}} = \lambda \mathbf{I} + \Sigma_{\mathbf{X}}, \tag{6}$$

where $\mathbf{I}$ is an identity matrix of the appropriate size and $\lambda \in [0 \ldots \infty)$ is a parameter that controls the tradeoff between the fit between $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$ and the scaling of the weights $\|\mathbf{W}\|$. As $\lambda$ approaches infinity, $\widetilde{\Sigma}_{\mathbf{X}}^{-1}$ and hence $\widetilde{\mathbf{W}}$ approach zero (equation 5).

Another approach is to apply "shrinkage" regularization,[41] [42] which drives the covariance matrix towards a (scaled) identity matrix:

$$\gamma = \frac{\text{trace}(\Sigma_{\mathbf{X}})}{m}, \tag{7}$$

$$\widetilde{\Sigma}_{\mathbf{X}} = \alpha\gamma\mathbf{I} + (1-\alpha)\Sigma_{\mathbf{X}}, \tag{8}$$

where $\alpha \in [0 \ldots 1]$ controls the amount of shrinkage and $\gamma\mathbf{I}$ is an identity matrix that is scaled by the mean of the diagonal elements of the empirical covariance matrix. In the shrinkage scheme, we choose $\alpha$ according to how certain we are that the empirical estimate of the covariance matrix is a good approximation of the true covariance matrix. When the data dimensionality is low and the number of available training examples high, the empirical estimate is likely a good approximation, so we choose $\alpha$ close to zero. In the inverse case, where the empirical estimate is likely to be biased, it may be beneficial to choose $\alpha$ close to one, in which case the linear model will disregard any relationship between the features.

In our EEG example, $\mathbf{X}$ was obtained by concatenating the timecourses for each sensor, which introduces a striking regularity in the covariance matrix, see Figure 3. The covariance matrix can be approximated by the Kronecker product[43] between the spatial covariance matrix $\Sigma_{\text{s}}$ (i.e., the linear relationship between the sensors) and temporal covariance matrix $\Sigma_{\text{t}}$ (i.e., the linear relationship between the samples in time):[44]

$$\Sigma_{\mathbf{X}} \approx \Sigma_{\text{s}} \otimes \Sigma_{\text{t}}, \tag{9}$$

where $(\otimes)$ denotes the Kronecker product. With this in mind, we propose a variation of the shrinkage approach that we call "Kronecker shrinkage". First, we shrink of $\Sigma_{\mathbf{X}}$ towards $\Sigma_{\text{s}} \otimes \mathbf{I}_{\text{t}}$,

[41] Blankertz et al., 2011

[42] Engemann and Gramfort, 2015

[43] Loan, 2000

[44] Bijma, De Munck, and Heethaar, 2005

where $\mathbf{I}_t$ denotes an identity matrix of the same dimensionality as the temporal covariance matrix. Then, we substitute the result into equation 8 instead of $\Sigma_{\mathbf{X}}$:

$$\widetilde{\Sigma}_{\mathbf{X}} = \alpha\gamma\mathbf{I} + (1-\alpha)(\beta\Sigma_s \otimes \mathbf{I}_t + (1-\beta)\Sigma_{\mathbf{X}}), \tag{10}$$

where $\alpha$ controls the shrinkage of the spatial component and $\beta$ controls the shrinkage of the temporal component of the covariance matrix.

## 2.9 Strategies for modifying the pattern matrix

The root problem that causes overfitting of the model is a lack of available training data. Therefore, for datasets that include multiple participants or recording sessions, we would expect the model to perform better if it had access to all recordings. However, transferring a weight matrix from one participant to another is often non-optimal, because it is unlikely to be similar across participants. Since this matrix represents a filter that isolates the signal of interest from all signals of non-interest (Figure 1), it depends on all signal components. This is why, in a neuroimaging setting, linear models that aim to generalize across participants are often outperformed by participant-specific models, even when the general models have access to more training data.[45,46,47]

In contrast, the pattern matrix is more likely to be similar across participants, because it represents only the signal of interest, and has been successfully transferred between participants.[48,49] In the post-hoc modification framework, the pattern matrix can be straightforwardly steered towards the grand average. Let $\overline{\mathbf{P}}$ be the average of the pattern matrices for all recordings, excluding the recording currently under consideration. Then:

$$\widetilde{\mathbf{P}} = \rho\overline{\mathbf{P}} + (1-\rho)\mathbf{P}, \tag{11}$$

where $\rho$ controls how much the pattern matrix is steered towards the grand average. This operation can be beneficial if the model has difficulty identifying the signal of interest during the training phase (e.g, due to noisy data, lack of training data, or absence of a $\mathbf{Y}$ matrix[50]).

Another approach to correcting inaccuracies in the pattern matrix is to leverage domain knowledge of the signal of interest. In our example case, the task is to decode FAS from the EEG signal, in which case the literature notes the N400 component of the ERP[51,52] as the primary signal of interest. We can instruct the model to look for a specific ERP component, by multiplying the timecourses in the pattern matrix with a Gaussian kernel (Figure 4):

$$\widetilde{\mathbf{P}}(c,t) = e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}\mathbf{P}(c,t), \tag{12}$$

where $c$ iterates over all channels, $t$ iterates over all time samples, and $\mathbf{P}(c,t)$ denotes the element of $\mathbf{P}$ that corresponds to channel $c$ at time $t$. Parameters $\mu$ and $\sigma$ determine the center and width of the Gaussian kernel (Figure 4).

Of course, the above operation is specific to our example case, in which we have intimate knowledge of the signal of interest. But even when the signal of interest is unknown, most event-related studies are designed such that each epoch is preceded by a baseline period that should be devoid of the signal of interest. In these cases, setting the values in the pattern matrix that correspond to the baseline period to zero may be beneficial. Note that this is not the same as discarding the baseline data. Rather, this operation flags the data as containing only noise, which is beneficial for designing the filter, which, as we have seen, depends on accurate

45 Reuderink, Farquhar, Poel, and Nijholt, 2011
46 Lotte, Guan, and Ang, 2009
47 Fazli et al., 2009
48 van Vliet et al., 2016
49 van Vliet, Van Hulle, and Salmelin, 2018
50 van Vliet et al., 2018
51 Kutas and Hillyard, 1980
52 Kutas and Federmeier, 2011

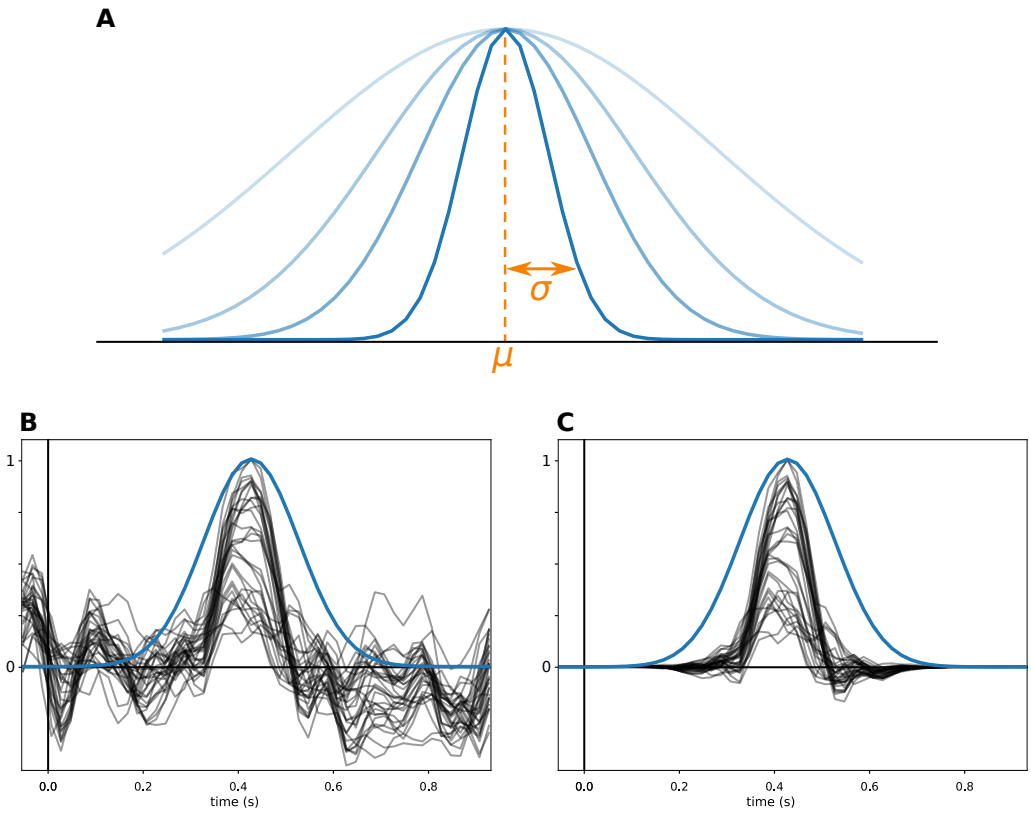


**Figure 4:** Example of multiplying the pattern matrix with a Gaussian kernel. **A:** Parameters $\mu$ and $\sigma$ determine the position and shape of the kernel. **B:** Example of a pattern matrix, with the timecourse for each sensor drawn in black. An example Guassian kernel is drawn in blue. For this visualization, the pattern was normalized to have a maximum amplitude of 1 to have the same visual scale as for the kernel. **C:** The result of multiplying the pattern matrix with the Gaussian kernel.

modelling of both signal and noise.

## 2.10 Strategies for modifying the normalizer

Modifying the covariance and pattern matrices also influences the matrix norm of the weight matrix and hence the magnitude of the output of the linear model. Depending on the application of the model, this may be problematic. For example, setting elements of the pattern matrix to zero (section 2.9) may decrease the error as measured by a scale-free metric such as Pearson correlation. However, as it will reduce the magnitude of the output, it may increase the error as measured by a scale-dependent metric such as the commonly used mean squared error (MSE). In such cases, it may be desirable to modify the normalizer to enforce a standardized scaling of the output.

Inspiration for such normalization schemes can be found in the beamformer literature.[53] For example, the basic formulation of a linearly constrained minimum variance (LCMV) beamformer[54] constrains the result such that $\widetilde{\mathbf{W}}\widetilde{\mathbf{P}} = \mathbf{I}$ and is computed as follows:

[53] Sekihara and Nagarajan, 2008

[54] Van Veen et al., 1997

$$\mathbf{W} = \frac{\Sigma_{\mathbf{X}}^{-1}\mathbf{P}}{\mathbf{P}^{\mathsf{T}}\Sigma_{\mathbf{X}}^{-1}\mathbf{P}}. \tag{13}$$

In the post-hoc modification framework, we can use the same scaling strategy by using the denominator of equation 13 as a normalizer:

$$\widetilde{\Sigma}_{\widehat{\mathbf{Y}}} = (\widetilde{\mathbf{P}}^{\mathsf{T}}\widetilde{\Sigma}_{\mathbf{X}}^{-1}\widetilde{\mathbf{P}})^{-1}. \tag{14}$$

## 2.11 Tuning of the parameters

Both initial models (see section 2.7) have a parameter ($\alpha$) that determines the amount of $\ell_2$ regularization, and throughout sections 2.8 to 2.9, we have defined several more parameters

($\beta, \rho, \mu, \sigma$) that control various aspects of the model. These parameters can be used to impose hard constraints on the model, for example, $\mu$ and $\sigma$ limit the time-range in which the model will search for the signal of interest. Alternatively, they can be treated as parameters that need to be learned, just like the model weights.

In our example analysis, we used an inner leave-one-out cross-validation loop to learn these parameters during the training phase, using a general convex optimization algorithm (Limited-memory Broyden–Fletcher–Goldfarb–Shanno with box constraints (L-BFGS-B)[55]). This algorithm searches for the optimal parameters by alternating between two phases: 1) estimating the direction of maximum performance gain by making tiny changes to each parameter and measuring the effect on the leave-one-out performance of the model, followed by 2) updating the parameters in the direction of maximum positive effect on the performance. This process is repeated until no changes to the parameters can be found that improve the leave-one-out performance.

[55] Byrd, Lu, Nocedal, and Zhu, 1995

The optimization approach employed by the L-BFGS-B algorithm requires that the chosen model performance evaluation function is continuous and differentiable. This is why, for the classification model, we used the logistic loss function rather than classification accuracy, since the latter is not differentiable. For the regression model, the Pearson's correlation coefficient between the leave-one-out model output and the desired output (**Y**) was used.

### 2.12 Data and code availability

Electronic supplementary information is available at: https://users.aalto.fi/~vanvlm1/posthoc. This includes a Python package that provides an implementation of the post-hoc modification framework that is compatible with Scikit-Learn.[56] The package contains optimized implementations (see appendices) of all modification strategies discussed in this paper and provides an interface for implementing new ones.

[56] Pedregosa et al., 2012

The consent form that was signed by the participants stated that the raw data would not be shared publicly without limitations. This data can be obtained upon request from the corresponding author, for reasons such as replication, assessment of other analysis methods, or aid in future studies on semantic processing.

All nonsensitive data can be found in the electronic supplementary information, including the grand-average pattern matrices, the preprocessing reports for the data of each participant, the output of the models and the stimulus list.

## 3 Results

The performance of the initial ridge regression and logistic regression models (section 2.7) were evaluated using 10-fold cross validation (the epochs were shuffled before being assigned to folds). For the ridge regression model, we report the Pearson correlation between the model output and the FAS of the word-pairs as the performance metric (Figure 5, left, "ridge"). For the logistic regression model, we report the classification performance using the receiver operating characteristic – area under curve (ROC-AUC) score (Figure 5, right, "logistic"), where the classification task was to assign each epoch to either the low-FAS or high-FAS category.

The performance of the models varied substantially between recordings performed on different participants. One factor that influences the performance of the model is the amount of noise and the ability of the model to accurately determine the "direction" of the noise (see Figure 1). By applying regularization to the covariance matrix, the estimated direction of the noise is

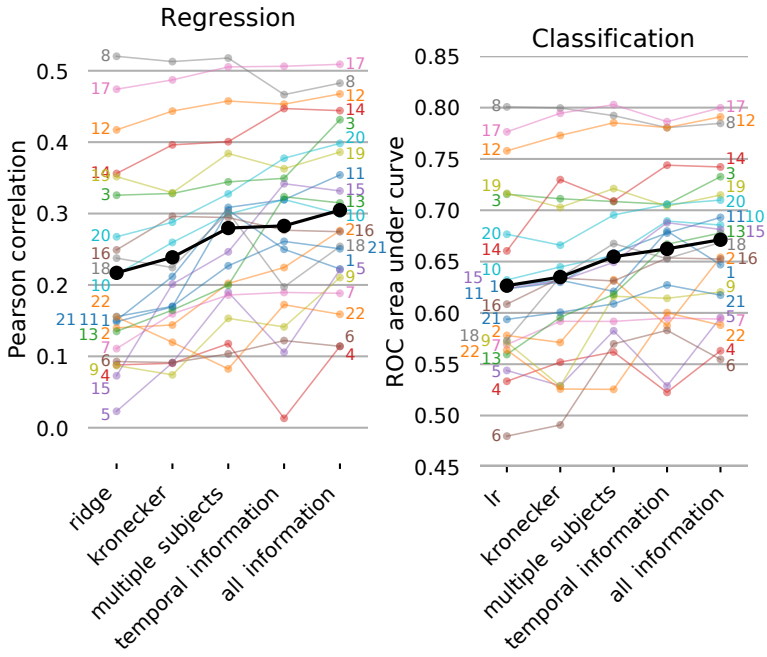


**Figure 5:** Performance of the linear model, before and after applying various post-hoc modification strategies. Regression performance (left) is measured as the Pearson correlation between the model output and the log-transformed FAS of the word-pairs. Classification performance (right) is measured as the ROC-AUC, where the classification task was to distinguish between word-pairs with either a low or high FAS. The performance for each participant is shown (colored dots and numbers), along with the mean performance across participants (black dots). Lines have been drawn between the dots in order to facilitate comparing the performance of a single participant across modification strategies. See the main text for an explanation of the modification strategies.

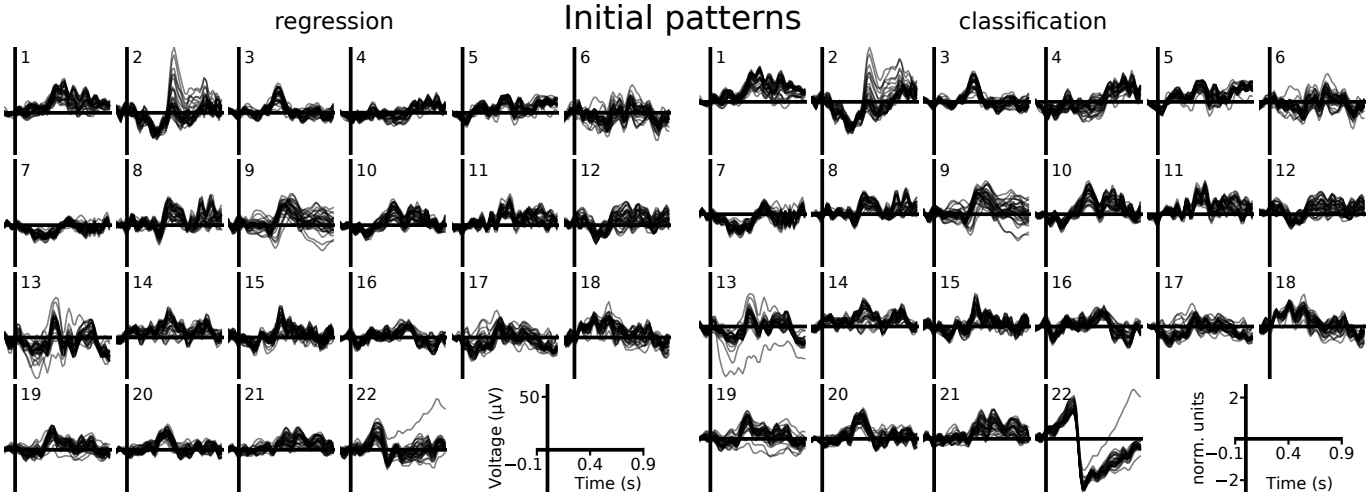


**Figure 6:** For each participant (1-22), the pattern that was learned by the initial linear models, for both the regression (left, ridge regression) and classification (right, logistic regression) scenarios. The timecourses of all electrodes are shown overlaid.

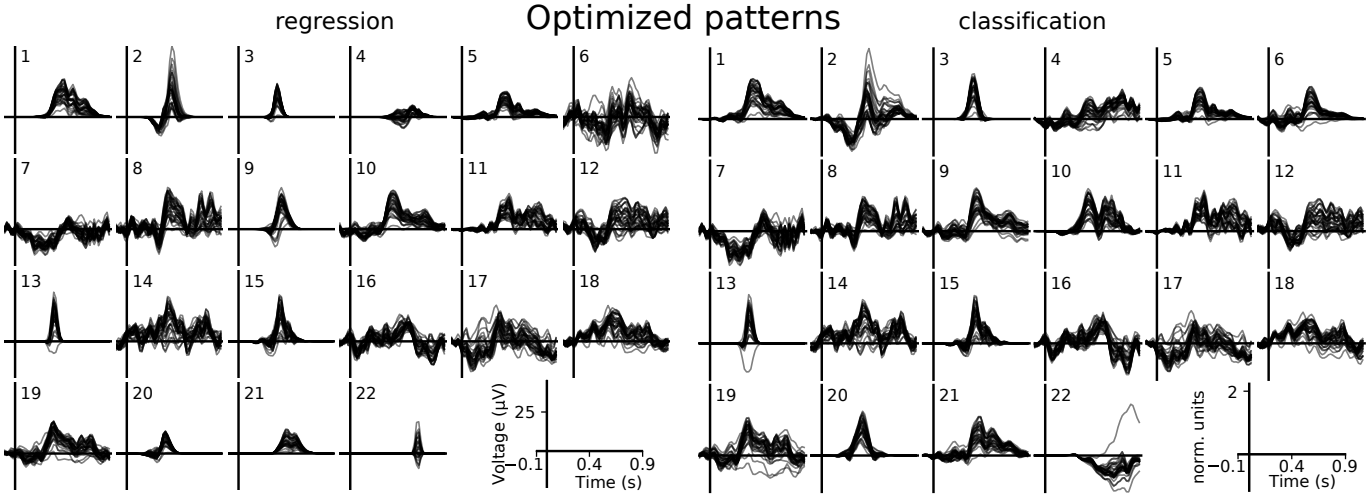


**Figure 7:** For each participant (1-22), the pattern that was used in the linear model that incorporates all post-hoc modifications (the "all information" model), for both the regression and classification scenarios. The timecourses for all electrodes are shown overlaid.

steered towards being spherical (i.e. equal in all directions). Both initial models already apply $\ell_2$ regularization. In the regression scenario, Kronecker shrinkage, which controls the amount of shrinkage for the spatial and time dimensions separately, outperforms the $\ell_2$ regularization approach (Figure 5, left, "kronecker"). In the classification scenario, Kronecker shrinkage is beneficial in some cases, but detrimental in others (Figure 5, right, "kronecker").

Inspecting the pattern matrices (Figure 6), computed with equation 2, reveals another contributing factor that influences the performance of the models. The N400 component is a prominent signal of interest for determining FAS from EEG data.[57] In some patterns (e.g, participants 3 and 20), the N400 is clearly visible as a peak at around 400 ms. However, in almost all patterns, there are other peaks, indicating that the model has learned other signals of interest as well. The question is how well these features generalize beyond the training set.

[57] Kutas and Federmeier, 2011

By taking a weighted average between the participant-specific pattern and the grand-average pattern, signals of interest that are present for all participants can be given priority. Furthermore, if too much noise is present, the model may not be able to accurately learn a signal of interest from the training data alone, in which case using the grand-average template can be beneficial. This procedure increases the performance of both the ridge regression and logistic regression models (Figure 5, "multiple subjects").

Another approach is to put the focus on the N400 potential by limiting the pattern in time (Figure 4), as this signal of interest is likely to generalize beyond the training set. This procedure also increases the performance of both the ridge regression and logistic regression models (Figure 5, "temporal information").

Finally, the result of combining all the above modification strategies was evaluated, yielding the best performing regression and classification models (Figure 5, "all information"). Inspecting the modified pattern matrices (Figure 7) reveals that the N400 potential is a much more prominent signal of interest for the best performing model.

## 4 Discussion

We have demonstrated how general purpose models can be made more task specific with the post-doc modification framework. In our example, we have modified a ridge regression and logistic regression model to:

1. employ Kronecker shrinkage that takes the spatio-temporal nature of EEG into account
2. use the grand-average pattern across multiple recordings as a prior for the current model
3. use information about the temporal characteristics of the N400 potential as a further prior

The resulting models show a remarkable improvement over the initial general purpose models. However, we also note a diminishing return when too many parameters are introduced, since it increases the risk of overfitting. The final "all information" model should theoretically never perform worse than the "temporal information" or "multiple subjects" models, since $\sigma$ can be set to a large value to disable restricting the pattern in time, and $\rho$ can be set to zero to disable incorporating any data from other recordings. Yet, due to overfitting, the optimal settings were not always chosen during the inner-cross validation loop.

The presented modification strategies are only a few EEG-specific examples, meant to demonstrate the capabilities of the post-hoc modification framework. The post-hoc modification framework opens up a wide range of possibilities to design such modification strategies, and we hope that our examples can serve as inspiration. For example, the N400 potential has a well

defined spatial signature[58] that may be used in addition to a temporal prior. Alternatively, one may define more informative priors for the covariance matrix, for example, a recording of the empty measurement room, without any participant, is often used in magnetoencephalography (MEG) studies. For other decoding tasks, wholly new modification strategies can be designed that are appropriate for the characteristics of the noise and signal of interest.

In our examples, we focused on optimizing the decoding performance of the model, but this may not always be the main criterion for such models. Decoding models are often employed to explore the signal of interest that was learned, in which case interpretability of the model is more important.[59][60][61] In this case, the pattern matrix is a valuable tool,[62] which contains the signals of interest that were learned by the model. However, the fact that a signal is useful for a decoding task does not necessarily mean that it is of interest to the study. For instance, in our example EEG study, eye artifacts are a good predictor for FAS and, despite the preprocessing steps to attenuate them, are likely still present in the pattern matrices (e.g., Figure 6, participant 22). Furthermore, given that most models in neuroimaging are overfitting, due to the ratio of number of features versus the size of the training set, the pattern matrix can be noisy and/or biased.

We propose to combine the pattern matrix that was learned from the data with a prior that is based on domain information. Using the decoding performance of the model as a guide, the initial pattern matrix may be pushed towards the prior, attenuating some of the noise. In our example study, the initial models sometimes failed to find a signal that clearly resembles the N400 potential, yet when a template N400 signal was mixed in with the pattern matrix, the decoding accuracy increased, which suggests that the N400 potential was present in the EEG of the participant after all (e.g., compare Figure 6 and Figure 7 for participant 5).

If the goal of the analysis is to study a specific signal of interest, it may be desirable to fix aspects of the pattern. For example, if the goal is to measure the timing of the N400 potential, we may explicitly set the pattern matrix to a time-shifted version of a suitable N400 template. Restricting the pattern allows for precise control over which aspects are "learned" from the data and which are dictated by the researcher. This enables an interative process, where first some initial restrictions are placed by the data analyst, then the model is fitted to the data, followed by an inspection of the resulting pattern matrices, leading to more refined restrictions, until finally, a model is obtained that satisfies all requirements.

If $\tilde{\mathbf{P}}$ is completely fixed, the model is transformed into a beamformer[63] and no ground truth ($\mathbf{Y}$) is required to train the model. For example, it is possible to train a model on a dataset for which a ground truth is available, and transplant the resulting pattern matrix into a new model that is fitted to a dataset for which no ground truth is (yet) available.[64]

[58] Kutas and Federmeier, 2011

[59] Haufe et al., 2014

[60] Kia, Pedregosa, Blumenthal, and Passerini, 2017

[61] Parra et al., 2003

[62] Haufe et al., 2014

[63] van Vliet et al., 2016

[64] van Vliet et al., 2018

## 5 Conclusion

The post-hoc modification framework can be used to decompose the weight matrix of any linear model into three components: a covariance matrix, a pattern matrix and a normalizer. Domain knowledge can often be straightforwardly incorporated into the model by modifying these components and re-assembling them into a modified weight matrix.

We have presented some strategies for incorporating domain knowledge and demonstrated their effectiveness on an example EEG dataset, where the task of the linear model was to predict, given a single epoch, the associated relatedness between the two words that were presented during the epoch. Through post-hoc modification of two general purpose models, a ridge regression and logistic regression model, information was incorporated about the spatio-

temporal nature of EEG data, the recordings performed on other participants and the N400 potential. The resulting domain specific models achieved an increase in decoding performance compared to the initial, general purpose models.

However, the presented modification strategies merely serve as examples. Post-hoc modification offers many possibilities to implement modification strategies to suit the many different purposes of linear models in neuroimaging and other fields.


## 6 Acknowledgements

The EEG data was recoded at the KU Leuven, Department of Neurosciences, under the supervision of Marc Van Hulle.

MvV was supported by the Interuniversity Attraction Poles Programme – Belgian Science Policy (IUAP P7/11) and a grant from the Aalto Brain Centre (ABC), and is currently supported by the Academy of Finland (grant 310988). RS is supported by the Academy of Finland (grant 315553) and the Sigrid Jusélius Foundation.


### Appendix A: Optimizing covariance computation

Computing the empirical covariance matrix $\Sigma_{\mathbf{X}}$ and its inverse $\Sigma_{\mathbf{X}}^{-1}$ can be time consuming, given the number of features in EEG and especially MEG epochs. Typically, however, the number of features far exceeds the number of epochs, which allows us to compute equation 5 efficiently by applying the matrix inversion lemma,[65] which states that for any matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{U}$, and $\mathbf{V}$ of appropriate size, the following holds:

[65] Tylavsky and Sohie, 1986

$$(\mathbf{A}-\mathbf{UBV})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1} - \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}. \tag{15}$$

This allows us to reformulate $\mathbf{X}^{\mathsf{T}}\mathbf{X}$, which is for our example EEG dataset a $1600 \times 1600$ matrix, in terms of $\mathbf{X}\mathbf{X}^{\mathsf{T}}$, which is in our example a $200 \times 200$ matrix.

For example, in the case of Kronecker shrinkage, equation 5 may be computed as:

$$\widetilde{\mathbf{W}} = [\alpha\gamma\mathbf{I} + (1-\alpha)(\beta\Sigma_{\mathrm{s}} \otimes \mathbf{I}_{\mathrm{t}} + (1-\beta)\mathbf{X}^{\mathsf{T}}\mathbf{X})]^{-1}\,\widetilde{\mathbf{P}}\,\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}, \tag{16}$$

$$= [\alpha\gamma\mathbf{I} + (1-\alpha)\beta\Sigma_{\mathrm{s}} \otimes \mathbf{I}_{\mathrm{t}} + (1-\alpha)(1-\beta)\mathbf{X}^{\mathsf{T}}\mathbf{X}]^{-1}\,\widetilde{\mathbf{P}}\,\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}, \tag{17}$$

$$\mathbf{A} = \alpha\gamma\mathbf{I} + (1-\alpha)\beta\Sigma_{\mathrm{s}} \otimes \mathbf{I}_{\mathrm{t}}, \quad \mathbf{B} = \mathbf{I}, \quad \mathbf{U} = -(1-\alpha)(1-\beta)\mathbf{X}^{\mathsf{T}}, \quad \mathbf{V} = \mathbf{X}, \tag{18}$$

$$\mathbf{G} = \mathbf{A}^{-1}\mathbf{U}, \quad \mathbf{K} = \mathbf{I} + \mathbf{XG}, \tag{19}$$

$$\widetilde{\mathbf{W}} = (\mathbf{A}^{-1} + \mathbf{GK}^{-1}\mathbf{XA}^{-1})\widetilde{\mathbf{P}}\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}. \tag{20}$$

### Appendix B: Optimizing the inner cross-validation loop

Our optimization strategy (section 2.11) depends on evaluating the leave-one-out performance of the model many times. The computationally most expensive operation in equation 20 is computing $\mathbf{K}^{-1}$. However, this matrix only needs to be computed once, whereafter the leave-one-out case where one observation $i$ is left out can be obtained efficiently by only computing the change caused by leaving one observation out, instead of re-computing the matrix from scratch. Let $\mathbf{K}_{(i)}$ denote the leave-one-out version of $\mathbf{K}$, which in the case of this matrix means the $i$'th row and column are removed. Salmen, Schlipsing, and Igel (2010) have devised an efficient updating algorithm for this case, using the matrix inversion lemma.

Begin by computing $\mathbf{K}_{(1)}$ and $\mathbf{K}_{(1)}^{-1}$ in a conventional manner. Then, $\mathbf{K}_{(i)}$ can be constructed for $i > 1$ by replacing the $(i-1)$'th row and column of $\mathbf{K}_{(1)}$ with the first observation. Note that this results in a non-standard ordering of the rows and columns of $\mathbf{K}_{(i)}$, so care must be taken to

order the leave-one-out versions of $\mathbf{X}$ and $\mathbf{Y}$ in the same manner. The update rule of the inverse can then be formulated as:

$$\mathbf{K}_{(i)}^{-1} = (\mathbf{K}_{(1)} + \mathbf{D})^{-1}, \tag{21}$$

$$\mathbf{D} = \mathbf{K}_{(1)} - \mathbf{K}_{(i)} = \begin{pmatrix} 0 & \cdots & k_{1,1} - k_{2,i} & \cdots 0 \\ 0 & \cdots & \vdots & \cdots 0 \\ k_{1,1} - k_{i,2} & \cdots & k_{1,i} - k_{i,i} & \cdots k_{1,n} - k_{i,n} \\ 0 & \cdots & \vdots & \cdots 0 \\ 0 & \cdots & k_{n,1} - k_{n,i} & \cdots 0 \end{pmatrix}, \tag{22}$$

where $k_{i,j}$ refers to the element at row $i$ and column $j$ of the original matrix $\mathbf{K}$ and $n$ is the total number of observations in $\mathbf{K}$.

To apply the inversion lemma (equation 15), $\mathbf{D}$ must be formulated in terms of $\mathbf{UBV}$, which yields:

$$\mathbf{U} = \begin{pmatrix} k_{1,1} - k_{2,i} & 0 \\ k_{2,1} - k_{3,i} & 0 \\ \vdots & \vdots \\ k_{(i-1),1} - k_{(i-1),i} & 0 \\ k_{i,1} - k_{i,i} & 1 \\ k_{(i+1),1} - k_{(i+1),i} & 0 \\ \vdots & \vdots \\ k_{n,1} - k_{n,i} & 0 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{23}$$

$$\mathbf{V} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ k_{1,1} - k_{2,i} & k_{2,1} - k_{3,i} & \cdots & k_{(i-1),1} - k_{(i-1),i} & 0 & k_{(i+1),1} - k_{(i+1),i} & \cdots & k_{n,1} - k_{n,i} \end{pmatrix}. \tag{24}$$

Then, applying equation 15:

$$\mathbf{K}_{(i)}^{-1} = (\mathbf{K}_{(1)} + \mathbf{UBV})^{-1} = \mathbf{K}_{(1)}^{-1} - \mathbf{K}_{(1)}^{-1}\mathbf{U}(\mathbf{B}^{-1} + \mathbf{V}\mathbf{K}_{(1)}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{K}_{(1)}^{-1}. \tag{25}$$

**References**


Babyak, M. A. (2004). What you see may not be what you get: A brief, nontechnical introduction to overfitting in regression-type models. *Psychosomatic Medicine*, *66*(3), 411–421. doi:10.1097/01.psy.0000127692.23278.a9

Bijma, F., De Munck, J. C., & Heethaar, R. M. (2005). The spatiotemporal MEG covariance matrix modeled as a sum of Kronecker products. *NeuroImage*, *27*(2), 402–415. doi:10.1016/j.neuroimage.2005.04.015

Blankertz, B., Lemm, S., Treder, M. S., Haufe, S., & Müller, K.-R. (2011). Single-trial analysis and classification of ERP components - A tutorial. *NeuroImage*, *56*(2), 814–825. doi:10.1016/j.neuroimage.2010.06.048

Byrd, R., Lu, P., Nocedal, J., & Zhu, C. (1995). A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, *16*(5), 1190–1208. doi:10.1137/0916069

Croft, R. J., & Barry, R. J. (2000). Removal of ocular artifact from the EEG: A review. *Neurophysiologie Clinique*, *30*(1), 5–19. doi:10.1016/S0987-7053(00)00055-1

De Deyne, S., & Storms, G. (2008). Word associations: Network and semantic properties. *Behavior research methods*, *40*(1), 213–231. doi:10.3758/BRM.40.1.213

Engemann, D. A., & Gramfort, A. (2015). Automated model selection in covariance estimation and spatial whitening of MEG and EEG signals. *NeuroImage*, *108*, 328–342. doi:10.1016/j.neuroimage.2014.12.040

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., & Grozea, C. (2009). Subject-independent mental state classification in single trials. *Neural Networks*. Brain-Machine Interface, *22*(9), 1305–1312. doi:10.1016/j.neunet.2009.06.003

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., … Hämäläinen, M. S. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, *7*(December), 1–13. doi:10.3389/fnins.2013.00267

Grootswagers, T., Wardle, S. G., & Carlson, T. A. (2017). Decoding Dynamic Brain Patterns from Evoked Responses: A Tutorial on Multivariate Pattern Analysis Applied to Time Series Neuroimaging Data. *Journal of Cognitive Neuroscience*, *29*(4), 677–697. doi:10.1162/jocn_a_01068

Gross, J., Kujala, J., Hämäläinen, M., Timmermann, L., Schnitzler, A., & Salmelin, R. (2001). Dynamic imaging of coherent sources: Studying neural interactions in the human brain. *Proceedings of the National Academy of Sciences*, *98*(2), 694–699. doi:10.1073/pnas.98.2.694

Hämäläinen, M. S., & Ilmoniemi, R. J. (1994). Interpreting magnetic fields of the brain: Minimum norm estimates. *Medical & Biological Engineering & Computing*, *32*(1), 35–42. doi:10.1007/BF02512476

Hastie, T. (2009). *Elements of Statistical Learning: Data mining, inference, and prediction* (2nd edition). Springer.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., & Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, *87*, 96–110. doi:10.1016/j.neuroimage.2013.10.067

Huth, A. G., Heer, W. A. D., Griffiths, T. L., Theunissen, F. E., & Jack, L. (2016). Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, *532*(7600), 453–458. doi:10.1038/nature17637

Jas, M., Engemann, D. A., Bekhti, Y., Raimondo, F., & Gramfort, A. (2017). Autoreject: Automated artifact rejection for MEG and EEG data. *NeuroImage*, *159*, 417–429. doi:10.1016/j.neuroimage.2017.06.030

Jutten, C., & Herault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, *24*(1), 1–10. doi:10.1016/0165-1684(91)90079-X

Kia, S. M., Pedregosa, F., Blumenthal, A., & Passerini, A. (2017). Group-level spatio-temporal pattern recovery in MEG decoding using multi-task joint feature learning. *Journal of Neuroscience Methods*, *285*, 97–108. doi:10.1016/j.jneumeth.2017.05.004

Kutas, M., & Federmeier, K. D. (2011). Thirty years and counting: Finding meaning in the N400 component of the event related brain potential (ERP). *Annual Review of Psychology*, *62*(1), 621–647. doi:10.1146/annurev.psych.093008.131123

Kutas, M., & Hillyard, S. A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science (New York, N.Y.) 207*(4427), 203–205. doi:10.1126/science.7350657

Loan, C. F. V. (2000). The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*. Numerical Analysis 2000. Vol. III: Linear Algebra, *123*(1), 85–100. doi:10.1016/S0377-0427(00)00393-9

Lotte, F., Congedo, M., Lecuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain-computer interfaces. *J Neural Eng*, *4*(2), R1–R13. doi:10.1088/1741-2560/4/2/R01

Lotte, F., Guan, C., & Ang, K. K. (2009). Comparison of designs towards a subject-independent brain-computer interface based on motor imagery. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 4543–4546). doi:10.1109/IEMBS.2009.5334126

Matsuura, K., & Okabe, Y. (1995). Selective minimum-norm solution of the biomagnetic inverse problem. *IEEE Transactions on Bio-Medical Engineering*, *42*(6), 608–615. doi:10.1109/10.387200

McIntosh, A. R., & Mišić, B. (2013). Multivariate Statistical Analyses for Neuroimaging Data. *Annual Review of Psychology*, *64*(1), 499–525. doi:10.1146/annurev-psych-113011-143804

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M. M. K., Malave, V. L., a. Mason, R., & Just, M. A. (2008). Predicting Human Brain Activity Associated with the Meanings of Nouns. *Science*, *320*(5880), 1191–5. doi:10.1126/science.1152876

Neely, J. H. (1991). Semantic priming effects in visual word recognition: A selective review of current findings and theories. In D. Besner & G. W. Humphreys (Eds.), *Basic processes in visual word recognition* (pp. 264–323). Hillsdale, NJ: Lawrence Erlbaum Associates.

Nelson, D. L., McEvoy, C. L., & Dennis, S. (2000). What is free association and what does it measure? *Memory & Cognition*, *28*(6), 887–899. doi:10.3758/BF03209337

Parra, L. C., Alvino, C., Tang, A., Pearlmutter, B., Yeung, N., Osman, A., & Sajda, P. (2003). Single-trial detection in EEG and MEG: Keeping it linear. *Neurocomputing*, *5254*, 177–183. doi:10.1016/S0925-2312(02)00821-4

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2012). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830. doi:10.1007/s13398-014-0173-7.2

Pernet, C. R., Sajda, P., & Rousselet, G. A. (2011). Single-trial analyses: Why bother? *Frontiers in Psychology*, *2*(NOV), 1–2. doi:10.3389/fpsyg.2011.00322

Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. F. (1989). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, *72*(2), 184–187. doi:10.1016/0013-4694(89)90180-6

Reuderink, B., Farquhar, J., Poel, M., & Nijholt, A. (2011). A subject-independent brain-computer interface based on smoothed, second-order baselining. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 4600–4604). doi:10.1109/IEMBS.2011.6091139

Rifkin, R. M., & Lippert, R. A. (2007). *Notes on regularized least squares*. MIT: Computer Science and Artificial Intelligence Laboratory. Massachusetts.

Salmen, J., Schlipsing, M., & Igel, C. (2010). Efficient update of the covariance matrix inverse in iterated linear discriminant analysis. *Pattern Recognition Letters*, *31*(13), 1903–1907. doi:10.1016/j.patrec.2010.03.001

Sekihara, K., & Nagarajan, S. S. (2008). *Adaptive spatial filters for electromagnetic brain imaging* (J. H. Nagel, Ed.). Berlin, Heidelberg: Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*. Series B, *58*(1), 267–288. doi:10.1.1.35.7574

Tong, F., & Pratte, M. S. (2012). Decoding patterns of human brain activity. *Annual Review of Psychology*, *63*, 483–509. doi:10.1146/annurev-psych-120710-100412

Trujillo-Barreto, N. J., Aubert, E., & Penny, W. D. (2008). Bayesian M/EEG source reconstruction with spatio-temporal priors. *NeuroImage*, *39*(1), 318–335. doi:10.1016/j.neuroimage.2007.07.062

Tylavsky, D. J., & Sohie, G. R. L. (1986). Generalization of the matrix inversion lemma. *Proceedings of the IEEE*, *74*(7), 1050–1052. doi:10.1109/PROC.1986.13587

Uusitalo, M. A., & Ilmoniemi, R. J. (1997). Signal-space projection method for separating MEG or EEG into components. *Medical & Biological Engineering & Computing*, *35*(2), 135–140. doi:10.1007/BF02534144

Van Veen, B. D., van Drongelen, W., Yuchtman, M., & Suzuki, A. (1997). Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. *IEEE Transactions on Biomedical Engineering*, *44*(9), 867–880. doi:10.1109/10.623056

van Vliet, M., Chumerin, N., De Deyne, S., Wiersema, J. R., Fias, W., Storms, G., & Van Hulle, M. M. (2016). Single-trial ERP component analysis using a spatiotemporal LCMV beamformer. *IEEE Transactions on Biomedical Engineering*, *63*(1), 55–66. doi:10.1109/TBME.2015.2468588

van Vliet, M., Manyakov, N. V., Storms, G., Fias, W., Wiersema, J. R., & Van Hulle, M. M. (2014). Response-related potentials during semantic priming: The effect of a speeded button response task on ERPs. *PLoS ONE*, *9*(2), E87650. doi:10.1371/journal.pone.0087650

van Vliet, M., Van Hulle, M. M., & Salmelin, R. (2018). Exploring the organization of semantic memory through unsupervised analysis of event-related potentials. *Journal of Cognitive Neuroscience*, *30*(3), 381–392. doi:10.1162/jocn_a_01211

Vigario, R., Sarela, J., Jousmäki, V., Hamalainen, M., & Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE Transactions on Biomedical Engineering*, *47*(5), 589–593. doi:10.1109/10.841330

Wipf, D., & Nagarajan, S. (2009). A unified Bayesian framework for MEG/EEG source imaging. *NeuroImage*, *44*(3), 947–966. doi:10.1016/j.neuroimage.2008.02.059