April 5, 2022
Author copy. Original published in NeuroImage. DOI: 10.1016/j.neuroimage.2019.116221

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 "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 information. 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 information 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] to source modeling[3] and signal decoding.[4] 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.[5] 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.[6] 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.[7]

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.[8] 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 ℓ_1 regularization,[9] which enforces sparsity of the weight matrix, and ℓ_2 regularization,[10] which enforces a small magnitude of the individual weights. Moving beyond these approaches, imposing further restrictions that are motivated by domain information may lead to even better performance of the model. However, it is in practice very difficult to express domain information in terms of the weight matrix,[11] 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 easier to interpret (see section 2.2). While Haufe et al. (2014) focused on the computation, visualization and interpretation of the pattern matrix, they suggest that their work may have applications stretching beyond model interpretability and form the basis for a method for incorporating domain information into linear models. In the current paper, we continue this line of thought, leading to what we call the "post-hoc modification" framework.

It is often more straightforward to formulate domain information in terms of the pattern matrix than the model weights. This has been long known in the domain of electrophysiological source estimation of electroencephalography (EEG) and magnetoencephalography (MEG) data, where the pattern matrix corresponds to the leadfield (i.e., forward solution) and the weight

[1] McIntosh and Mišić, 2013

[2] Jutten and Herault, 1991; Uusitalo and Ilmoniemi, 1997; Vigario et al., 2000

[3] Gross et al., 2001; Hämäläinen and Ilmoniemi, 1994; Hauk et al., 2019; Matsuura and Okabe, 1995; Van Veen et al., 1997

[4] Grootswagers et al., 2017; Lotte et al., 2007; Tong and Pratte, 2012

[5] Hastie, 2009

[6] van Vliet et al., 2016; Parra et al., 2003; Pernet et al., 2011

[7] Huth et al., 2016; Mitchell et al., 2008

[8] Babyak, 2004; Blankertz et al., 2011

[9] Tibshirani, 1996

[10] Rifkin and Lippert, 2007

[11] Haufe et al., 2014

matrix to the inverse solution. Methods for estimating EEG/MEG source activity often formulate their domain information driven priors on the leadfield.[12] The modified leadfield is afterwards combined with a sensor-to-sensor covariance matrix and inverted to yield an inverse model that incorporates the domain information. 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.

In our framework, we decompose the weight matrix of a linear model into three subcomponents, and hence divide the problem of estimating the weight matrix into three subproblems (see section 2.2):

1. the pattern matrix of Haufe et al. (2014), associated with the subproblem of determining signal components that carry information about the desired output
2. the data covariance, associated with the subproblem of estimating the relationships between model inputs
3. the normalizer, associated with the subproblem of fine-tuning the mapping between the model output and the desired output

Inspecting these subcomponents in isolation offers an intuitive way to gain insights into the functioning of the model and possible problem points. We then proceed by modifying each component to impose new constraints and incorporate domain information, before recomposing the subcomponents back into a weight matrix. 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.[13] 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 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.[14]

2 Methods

2.1 Linear models

The post-hoc modification framework can operate on any type of linear model, regardless of function and type of data, so there are many application areas. Since our examples are in the domain of machine learning, we have chosen to adopt the general purpose terminology used in that literature[15] See Table 1 for a summary of the mathematical symbols used in this paper.

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

[12] Kohler et al., 1996; Lin et al., 2006; Trujillo-Barreto et al., 2008; Wipf and Nagarajan, 2009

[13] Neely, 1991; van Vliet et al., 2014

[14] Kutas and Federmeier, 2011; Kutas and Hillyard, 1980

[15] Hastie, 2009

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
λ	Controls the amount of $\ell 2$ regularization of the covariance matrix
α	Controls the shrinkage of the spatial component of the covariance matrix
β	Controls the shrinkage of the temporal component of the covariance matrix
μ	Controls the center of the Gaussian kernel used as a windowing function for the pattern matrix
σ	Controls the width of the Gaussian kernel used as a windowing function for the pattern matrix
ρ	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.

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, without loss of generalization, 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 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 $\mathbf{Y}$, given a certain data-fit cost function (also known as a loss function). Example cost functions are the sum of squared errors, often used in linear regression, and the logistic loss function in the case of logistic regression.

2.2 Post-hoc modification

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

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

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

In the above equations, $\widehat{\mathbf{X}}$ is the approximation of $\mathbf{X}$, $\Sigma_{\mathbf{X}}$ is the (empirical) covariance matrix of $\mathbf{X}$ and $\Sigma_{\widehat{\mathbf{Y}}}^{-1}$ is the inverse of the (empirical) covariance matrix of the output of the original decoding model (see equation 1). When we solve for $\mathbf{W}$ in equation 2, we obtain:

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

and observe that the weight matrix may be thought of as a combination of three subcomponents:

1. the covariance matrix of the data $\Sigma_{\mathbf{X}}$
2. the pattern matrix $\mathbf{P}$
3. the normalizer $\Sigma_{\widehat{\mathbf{Y}}}$

In the post-hoc framework, we replace the problem of finding the optimal weight matrix by the subproblems of finding the optimal $\Sigma_{\mathbf{X}}$, $\mathbf{P}$ and $\Sigma_{\widehat{\mathbf{Y}}}$. An initial estimate for the subcomponents can be obtained by applying a linear machine learning algorithm and decomposing its weight matrix using equation 2 (see also Figure 1). When understanding what the subcomponents represent and the subproblems they are trying to solve, the data analyst may use their domain information to refine the initial estimates at will. Afterwards, the modified subcomponents can be recomposed into an updated weight matrix (Figure 1):

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

where $\widetilde{\Sigma}_{\mathbf{X}}$ is a modified version of the data covariance, $\widetilde{\mathbf{P}}$ is a modified version of the pattern matrix, $\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}$ 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 take a closer look at the three subcomponents. For a visual explaination, see Figure 2.

In order to design a mapping from $\mathbf{X}$ to $\mathbf{Y}$, components of the data must be found that carry information that would be useful for determining the value of the decoding targets (Figure 2D, green line). Modifying the pattern matrix $\mathbf{P}$ allows for incorporating domain information on how the decoding targets $\mathbf{Y}$ are manifested in the data $\mathbf{X}$.

To paraphrase de Cheveigné and Simon (2008), the filter needs to observe all components that “contaminate” the pattern components, so as to *subtract* them. Those observations may themselves be contaminated, requiring subtraction of additional components, and so on. The filter thus uses data from all input features, even the ones that carry no information about the decoding targets, in order to cancel out any contaminants. This is achieved by transforming the data such that all correlations between the input features are eliminated and the variance of the data is equal in every dimension (Figure 2E), a process known as “whitening”. In other words, a whitening transform is a linear transformation that transforms the data from having

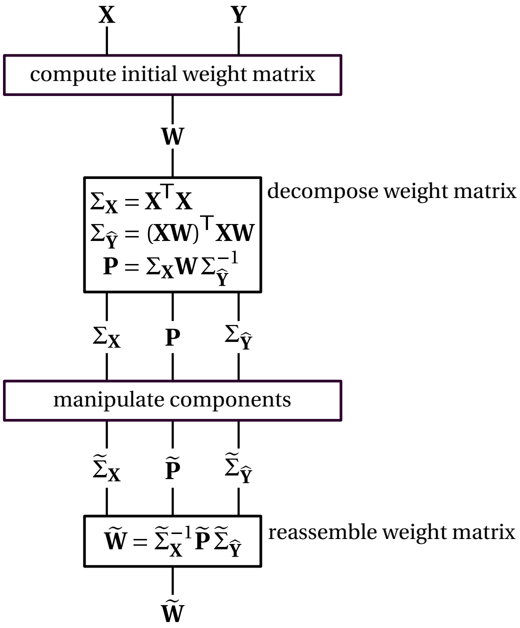
$\mathbf{X}$
$\mathbf{Y}$
compute initial weight matrix
$\mathbf{W}$
$\Sigma_\mathbf{X} = \mathbf{X}^\mathsf{T}\mathbf{X}$
$\Sigma_{\hat{\mathbf{Y}}} = (\mathbf{XW})^\mathsf{T}\mathbf{XW}$
$\mathbf{P} = \Sigma_\mathbf{X}\mathbf{W}\Sigma_{\hat{\mathbf{Y}}}^{-1}$
decompose weight matrix
$\Sigma_\mathbf{X}$
$\mathbf{P}$
$\Sigma_{\hat{\mathbf{Y}}}$
manipulate components
$\widetilde{\Sigma}_\mathbf{X}$
$\widetilde{\mathbf{P}}$
$\widetilde{\Sigma}_{\hat{\mathbf{Y}}}$
$\widetilde{\mathbf{W}} = \widetilde{\Sigma}_\mathbf{X}^{-1}\widetilde{\mathbf{P}}\,\widetilde{\Sigma}_{\hat{\mathbf{Y}}}$
reassemble weight matrix
$\widetilde{\mathbf{W}}$

Figure 1: 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 $\widetilde{\mathbf{W}}$.

covariance $\Sigma_\mathbf{X}$ to having a covariance matrix that is the identity matrix. The $\Sigma_\mathbf{X}^{-1}$ term in equation 4 represents a whitening transform that is applied to both the data and the pattern matrix (see appendix A). The signal components can now be readily extracted by projecting the whitened data unto the whitened pattern components (Figure 2F). Modifying the data covariance matrix $\Sigma_\mathbf{X}$ allows for incorporating domain information on the correlations between the input features, which is in turn used to compute the whitening transform that disentangles these correlations.

The procedure described above attempts to eliminate any components that interfere with the pattern components. However, since the whitening transform is computed using the covariance of the data, not the pattern matrix, it does not untangle the pattern components from each other, nor impose a scaling on them. In the case of $k = 1$, the whitened data is projected onto the line that is defined by the whitened pattern matrix (Figure 2E, green line). In the case of $k > 1$, the pattern matrix defines a plane. As a final step, a mapping must be made between locations along the projection line/plane and the desired target $\mathbf{Y}$. In the case of $k = 1$, this amounts to a scaling factor (Figure 2C, orange scale) and in the case of $k > 1$, the normalizer is a linear mapping between the locations on the projection plane to the model outputs $\mathbf{Y}$. Modifying the normalizer $\Sigma_{\hat{\mathbf{Y}}}$ allows for fine-tuning of the relationship between the projected data and the decoding targets $\mathbf{Y}$.

Domain information is by definition study specific, so in order to provide concrete examples, we will first introduce an example EEG study. In this study, the task of the linear model is to decode the forward association strength (FAS) between two words,[16] based on an EEG recording of a participant reading the word-pair during a semantic priming experiment. We will then explore some ways in which the subcomponents may be modified to tune the model for this specific task.

2.3 EEG recordings

The decoding performance of two linear models was 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

[16] Nelson et al., 2000

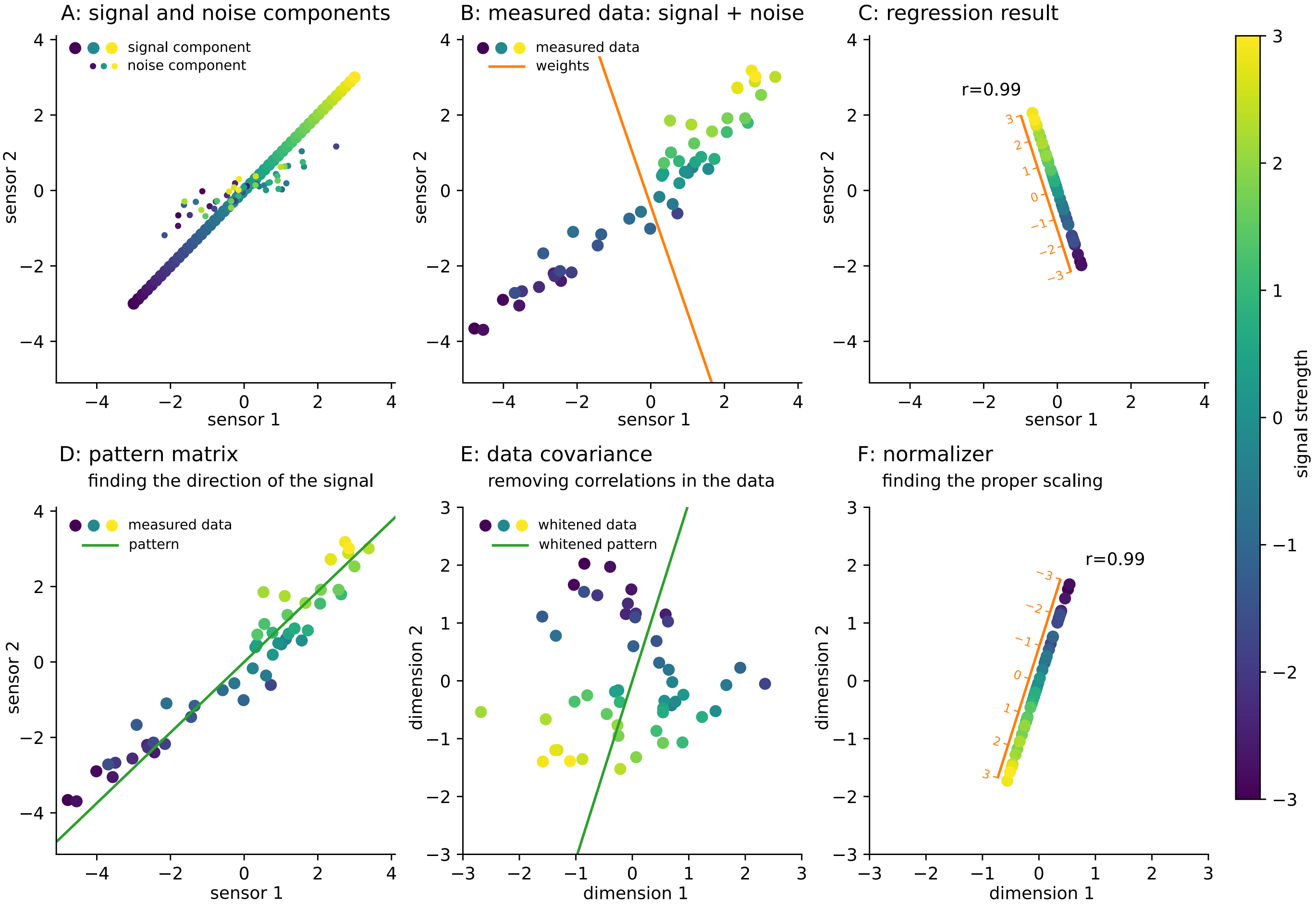

Figure 2: Visual explanation of the subcomponents of the post-hoc modification framework. This is a simulation of a signal that is being observed through two sensors. Dots represent observations of the signal and the color of the dots indicates the true signal strength during each observation.

Linear regression is used to decode the true signal strength from the observed data. In visual terms, the task of the model is to decode the color of a dot, based on its location in the graph.
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 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.
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.

In the post-hoc modification framework, the model weights (orange line) are decomposed into three subcomponents, where each subcomponent solves a part of the regression problem.
D: The pattern matrix represents the signal of interest and, like the weight matrix, can be visualized as a line (green). This line should approximate the direction of the actual signal (see panel A).
E: The data covariance matrix is used to construct a whitening operator, which removes the correlations within the data. The data is projected such that all features are of unit variance and all cross-correlations between the features are eliminated. This transformation is then also applied to the pattern matrix (green line). Performing linear regression is now equivalent to projecting the whitened data onto the whitened pattern line.
F: Finally, the normalizer (orange axis) scales the result such that the position along the projection line maps to the true signal strength.

An interactive version of this figure is available at https://aaltoimaginglanguage.github.io/posthoc, where the noise component can be manipulated to study its effect on the subcomponents.

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,[17] the participants were instructed to delay their button 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.

2.4 Stimuli

The stimuli consisted of Flemish-Dutch word pairs (see section 2.11) 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.

2.5 Data preprocessing

All data processing was performed using the MNE-Python[18] and auto-reject[19] 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 artefacts. 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.[20] Two EOG artifact elimination passes were performed

[17] van Vliet et al., 2014

[18] Gramfort et al., 2013

[19] Jas et al., 2017

[20] Perrin et al., 1989

on the data. First, the EOG channels were used to attenuate eye artefacts 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 artefacts were removed using the autoreject procedure,[21] 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.

A full report of the data preprocessing steps can be found at:
https://aaltoimaginglanguage.github.io/posthoc.

2.6 Initial linear models

In this paper, we give some examples on how to use the post-hoc modification framework to inject domain information into two general purpose machine learning models. For the regression scenario, we chose the ridge regressor as implemented in the Scikit-Learn package[22] 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)).

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.[23] 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.

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.[24] 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.[25] Both initial models (ridge and logistic regression) implement such regularization. In the following subsections, we look at the problem of overfitting not from the perspective of the weight matrix, but from that of the subcomponents as defined by the post-hoc modification framework.

2.7 Strategies for modifying the covariance matrix

The data covariance matrix $\Sigma_{\mathbf{X}}$ is the subcomponent of a linear model that describes the (linear) relationships between the input features. Overfitting of the model will occurs when the linear relationships that were inferred from the training set do not hold on the test set, either because the estimation was incorrect or because the relationships change across observations (e.g. they

[21] Jas et al., 2017

[22] Pedregosa et al., 2012

[23] van Vliet et al., 2016

[24] Babyak, 2004

[25] Hastie, 2009; Rifkin and Lippert, 2007

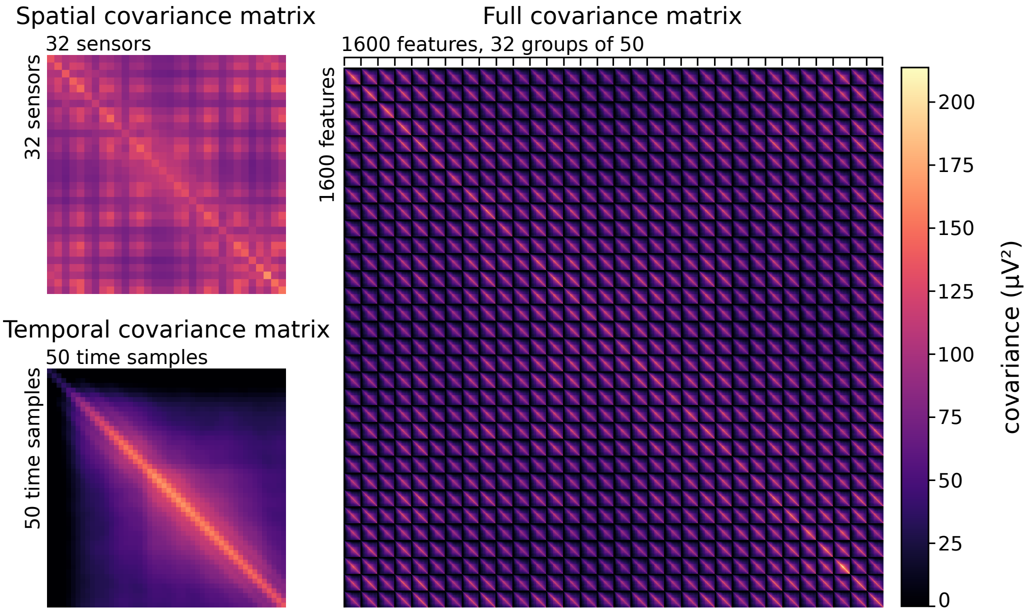

Figure 3: Shown on the right is the grand average covariance matrix. This matrix can be approximated with the Kronecker product of the grand average spatial covariance matrix (upper left) and grand average temporal covariance matrix (bottom left).

change over time due to nonstationarity of the signal). In this case, the model will benefit from de-emphasizing the relations that were estimated on the train set in favor of a conservative ground truth that is expected to hold in both the training and test sets.

The $\ell 2$ regularization that is imposed on $\Sigma_\mathbf{X}$ by the initial models (ridge and logistic regression) adds a constant value to each diagonal element of the initial 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 amount of regularization. One effect of this regularization scheme is that as λ approaches infinity, $\widetilde{\Sigma}_\mathbf{X}^{-1}$ and hence $\widetilde{\mathbf{W}}$ approach zero (equation 5). This effect is directly encoded in the optimization criterion for ridge regression.[26] However, from the point of view of the subproblem that the covariance matrix represents, a second effect becomes apparent, namely that the covariance matrix is steered towards a scaled identity matrix. This means the model is steered towards a ground truth that none of the features are linearly related, meaning any of the relationships inferred from the training set are untrustworthy. It is this second effect that provides a straightforward insight into why $\ell 2$ regularization prevents overfitting and lends itself to schemes for incorporating domain information.

An approach that has the second effect, but not the first, is "shrinkage" regularization:[27]

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

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

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 this regularization scheme, the covariance is steered towards a ground truth of no relationships between the features, without affecting the overall scaling of the matrix.

Both regularization schemes drive the covariance matrix towards a scaled identity matrix, penalizing all relationships equally in favor of the ground truth. One way of incorporating domain information is to distinguish between different kinds of relationships, and encode our belief that some may be estimated more reliably from the training data than others.

[26] Hastie, 2009; Rifkin and Lippert, 2007

[27] Blankertz et al., 2011; Engemann and Gramfort, 2015

In our EEG example, **X** was obtained by concatenating the timecourses for each sensor. Such an approach to vectorizing the input data introduces a striking regularity in the covariance matrix, see Figure 3. The covariance matrix can be approximated by the Kronecker product[28] between the spatial covariance matrix Σ_s (i.e., the linear relationship between the sensors) and temporal covariance matrix Σ_t (i.e., the linear relationship between the samples in time):[29]

$$\Sigma_{\mathbf{X}} \approx \Sigma_s \otimes \Sigma_t, \tag{9}$$

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 $\Sigma_s \otimes \mathbf{I}_t$, 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 α controls the shrinkage of the spatial component and β controls the shrinkage of the temporal component of the covariance matrix. This allows us to encode different amounts of confidence in the estimates of these two types of relationships from the training data.

2.8 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, one might expect that the model performs better if it had access to all recordings. However, in a neuroimaging setting, linear models that aim to generalize across participants are often outperformed by participant-specific models, even when the models have access to more training data.[30] Since the optimal weights depend on both the signal of interest and any interfering signals, it is often not straightforward to transfer a weight matrix from one participant to another.

The pattern matrix $\Sigma_{\mathbf{P}}$ is the subcomponent of a linear model that describes only the signal components that are informative of the targets, as opposed to other "noise" components. In some cases the pattern matrix is likely to be similar across participants. In our example study, the task was to decode FAS from the EEG signal, in which case the literature notes the N400 component of the ERP[31] as the primary signal of interest. While there are factors that affect the latency of this component, such as age,[32] the participants in our example study were drawn from a homogeneous pool (university students), so we can expect the timing of the component, as well as its distribution across sensors, to be relatively stable. Also in the case of other, similar N400 studies, the pattern matrix has been successfully transferred between participants.[33] Hence, a good strategy for improving the estimation of the pattern matrix may be to bias it towards a grand-average pattern matrix that was obtained from the recordings of other participants.

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 ρ 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 **Y** matrix[34]).

[28] Loan, 2000

[29] Bijma et al., 2005

[30] Fazli et al., 2009; Lotte et al., 2009; Reuderink et al., 2011

[31] Kutas and Federmeier, 2011; Kutas and Hillyard, 1980

[32] Kutas and Iragui, 1998

[33] van Vliet et al., 2016; van Vliet et al., 2018

[34] van Vliet et al., 2018

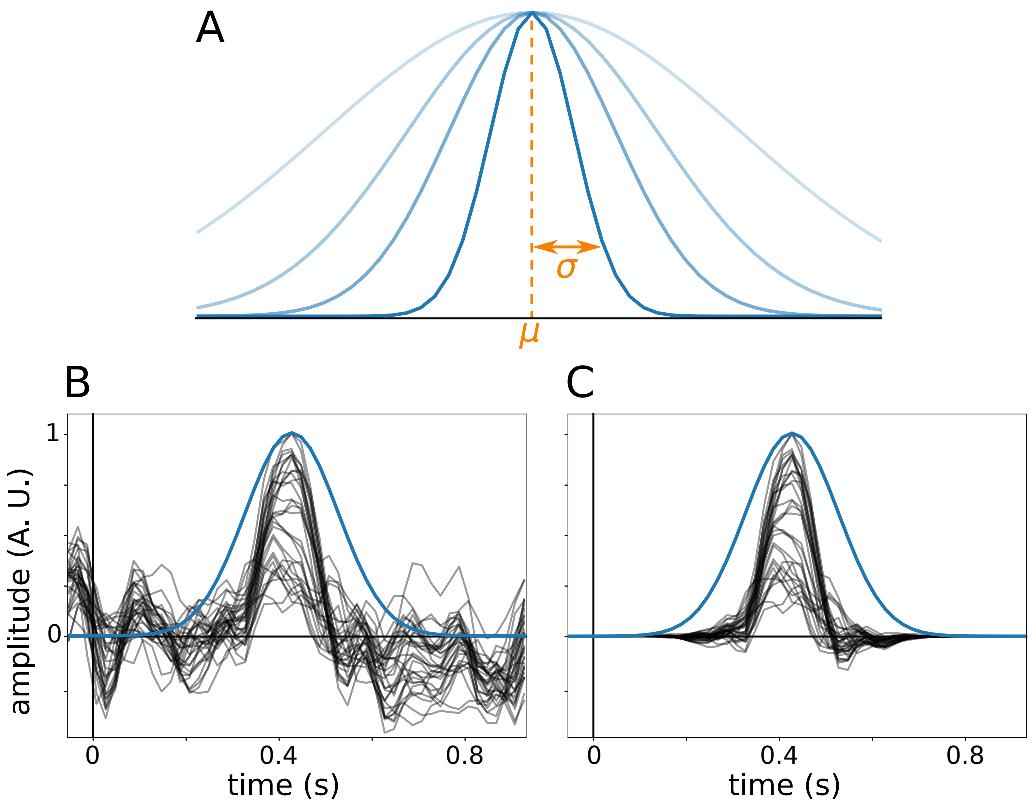

Figure 4: Example of multiplying the pattern matrix with a Gaussian kernel. **A:** Parameters μ and σ determine the position and shape of the kernel. **B:** Example of a pattern matrix, with the timecourse for each sensor drawn in black. An example Gaussian kernel is drawn in blue. For this visualization, 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.

Another approach to correcting inaccuracies in the pattern matrix is to leverage the fact that in our semantic priming study, the signal of interest (the N400) is well localized in time. One way of achieving this would be to restrict the data **X** to a time window surrounding 400 ms. However, this would deprive the model from potentially useful observations of the noise components that the model is attempting to cancel out. A good example can be found in the domain of EEG/MEG source estimation, where, even if the goal is to estimate activity at a single dipole source, it is beneficial to create a spatial filter using many sensors, and not only the sensors that are most sensitive to activity at the source dipole.[35] The post-hoc modification framework allows us to place restrictions on the pattern timecourses alone, keeping information about the noise components intact.

In our example study, we multiplied 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 μ and σ determine the center and width of the Gaussian kernel (Figure 4).

2.9 Strategies for modifying the normalizer

Modifications to the covariance and pattern matrices result in changes to the projection line ($k = 1$) or plane ($k > 1$) of the model. This means that the normalizer needs to be recomputed to re-map locations along the projection line/plane to the model outputs.

One way to compute an appropriate normalizer is to find the least-squares mapping between

[35] de Cheveigné and Simon, 2008

the output of the “raw” filter $\mathbf{X}\widetilde{\Sigma}_{\mathbf{X}}^{-1}\widetilde{\mathbf{P}}$ and $\mathbf{Y}$, through linear regression:

$$\widetilde{\Sigma}_{\widehat{\mathbf{Y}}} = (\mathbf{Y}^\top \mathbf{Y})^{-1} \mathbf{Y}^\top (\mathbf{X}\widetilde{\Sigma}_{\mathbf{X}}^{-1}\widetilde{\mathbf{P}}) \tag{13}$$

2.10 Model evaluation and automated tuning of the hyperparameters

The performance of each model was evaluated for each participant separately, using 10-fold crossvalidation. The order of the observations in the recording (the rows of **X** and **Y**) were shuffled and then assigned to ten folds. Two crossvalidation loops were used, which we will refer to as the “outer” and “inner” loops.

In the outer crossvalidation loop, nine folds were used as training data and one fold was used as test data. Normalization of **X** was performed inside the outer crossvalidation loop, 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. 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 **Y**, and in the classification scenario using the classification accuracy.

When a model incorporates data from other recordings (the “multiple subjects” and “all information” models, see section 2.8), a distinction was made between the recording for which the model was currently being evaluated and the recordings made on the other participants. During the outer crossvalidation loop, the training data was augmented with the data from the other participants, while the test data was left untouched.

Both initial models (see section 2.6) have a parameter (α) that determines the amount of ℓ_2 regularization, and throughout sections 2.7 to 2.8, we have defined several more parameters (β, ρ, μ, σ) that control various aspects of the model. These parameters can be used to impose hard constraints on the model, for example, μ and σ 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. Since searching the entire parameter space would be too time consuming, we first evaluated 100 random values for the parameters, taking the best performing parameter set as rough first estimate. This estimate was then fine-tuned using a convex optimization algorithm (Limited-memory Broyden–Fletcher–Goldfarb–Shanno with box constraints (L-BFGS-B)[36]). 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.

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 or receiver operating characteristic – area under curve (ROC-AUC), since the latter two are not differentiable. For the regression model, Pearson correlation between the leave-one-out model output and the desired output (**Y**) was used as a loss function, as this is the measure we report

[36] Byrd et al., 1995

Table 2: Models that were evaluated

Model name	Description
ridge	The initial ridge regression model (section 2.6). Employs $\ell 2$ regularization of the covariance matrix.
lm	The initial logistic regression model (section 2.6). Employs $\ell 2$ regularization of the covariance matrix.
kronecker	Employs Kronecker shrinkage of the covariance matrix (section 2.7).
multiple subjects	Employs Kronecker shrinkage of the covariance matrix and biases the pattern matrix towards the grand average pattern matrix (section 2.8).
temporal information	Employs Kronecker shrinkage of the covariance matrix and applies a Gaussian kernel to the pattern matrix (section 2.8).
all information	Employs Kronecker shrinkage of the covariance matrix and biases the pattern matrix towards the grand average pattern matrix, followed by application of a Gaussian kernel to the pattern matrix.

in the results section. This measure is closely related to the more traditional mean squared error (MSE) loss function, but is easier to interpret, as it has been normalized to range from 0 to 1.

2.11 Data and code availability

Electronic supplementary information is available at: https://aaltoimaginglanguage.github.io/posthoc. This includes a Python package that provides an implementation of the post-hoc modification framework that is compatible with Scikit-Learn.[37] The package contains optimized implementations (see appendices B and C) of all modification strategies discussed in this paper and provides an interface for implementing new ones.

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

We determined the effectiveness of the strategies for incorporating domain information by comparing the performance of the models that incorporates domain information to that of the original models. See Table 2 for an overview of the models that were evaluated.

The performance of the models was evaluated using 10-fold cross validation (the epochs were shuffled before being assigned to folds) and presented in Figure 5. For regression models, we report the Pearson correlation between the model output and the FAS of the word-pairs as the performance metric (Figure 5A). For classification models, we report the classification performance using the ROC-AUC score (Figure 5B), where the classification task was to assign each epoch to either the low-FAS or high-FAS category.

Taken individually, each manipulation strategy provided a small improvement to the performance of the initial model (for statistics, see top of Figure 5). Taken together (the "all information" model), the performance was substantially improved by using post-hoc modification to inject domain information for both the initial ridge regression (effect size: 0.088, pair-wise t-test: $t = 5.526$, $p < 0.001$) and logistic model (effect size: 0.045, $t = 6.550$, $p < 0.001$).

The post-hoc modification strategies for incorporating domain information were set up such

[37] Pedregosa et al., 2012

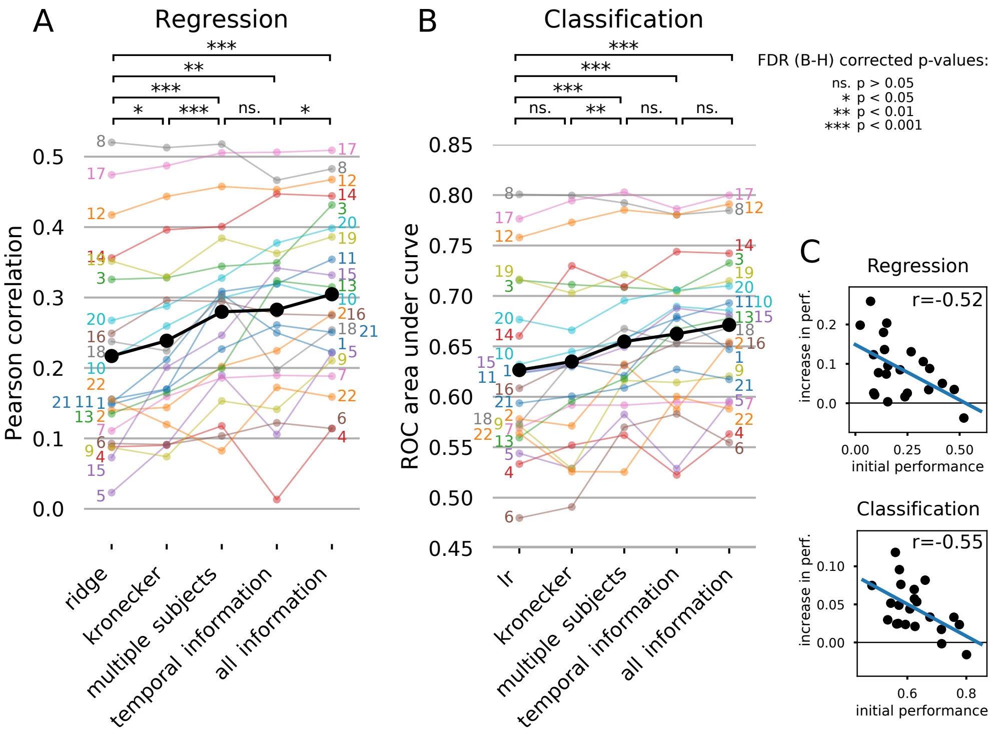

Figure 5: Performance of the linear models, before and after applying various post-hoc modification strategies. The performance for each participant is shown (colored dots and numbers), along with the mean performance across participants (black dots). Lines have been drawn between the dots in order to facilitate comparing the performance of a single participant across modification strategies. At the top, statistical comparisons between the group-level performances of the methods (paired t-tests) are shown. See the main text for an explanation of the modification strategies. **A**: Performance of the regression model. **B**: Performance of the classification model. **C**: The relationship between the performance of the initial model and the increase in performance gained by including domain information (the “all information” model).

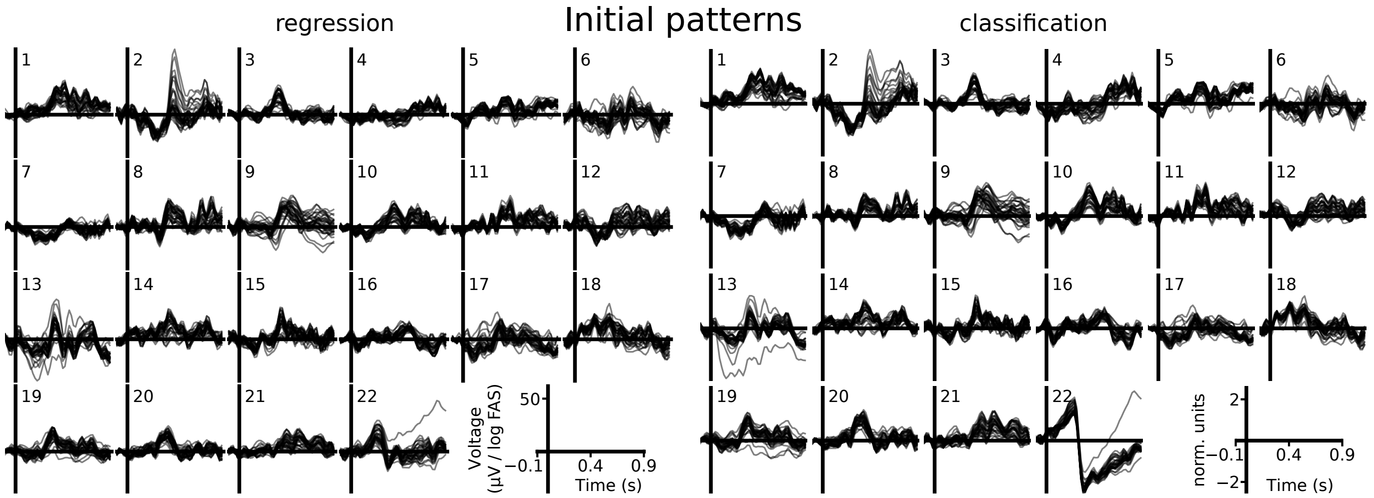

Figure 6: For each participant (1-22), the pattern that was learned by the initial linear models, for both the regression (left, ridge regression) and classification (right, logistic regression) scenarios. The timecourses of all electrodes are shown overlaid.

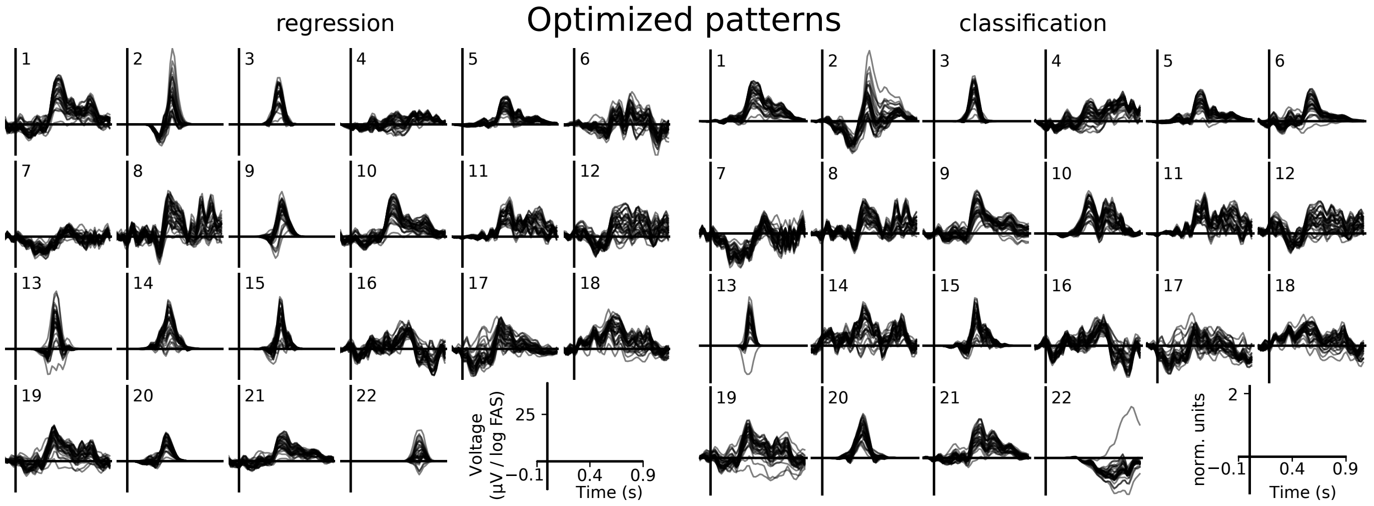

Figure 7: For each participant (1-22), the pattern that was used in the linear model that incorporates all post-hoc modifications (the “all information” model), for both the regression and classification scenarios. The timecourses for all electrodes are shown overlaid.

Table 3: Optimal parameters for the “all information” model

	Regression					Classification				
subject	α	β	ρ	μ (s)	σ (s)	α	β	ρ	μ (s)	$\sigma(s)$
1	0.97	0.11	1.00	0.77	0.96	0.60	0.07	0.50	0.42	0.24
2	0.71	0.01	0.21	0.38	0.08	0.74	0.00	0.28	0.41	0.24
3	0.90	0.03	0.15	0.37	0.07	0.66	0.02	0.24	0.37	0.07
4	0.94	0.36	0.32	0.40	0.32	0.97	0.72	0.52	0.75	0.90
5	0.70	0.00	0.64	0.41	0.24	0.80	0.00	0.65	0.41	0.24
6	0.03	0.03	0.26	0.65	0.38	0.01	0.07	0.99	0.10	0.35
7	1.00	0.35	0.22	0.30	0.78	0.99	0.00	0.00	0.30	0.78
8	0.54	0.00	0.11	0.47	0.94	0.46	0.00	0.12	0.47	0.94
9	0.82	0.00	0.23	0.38	0.07	0.92	0.00	0.62	0.22	0.95
10	0.96	0.02	0.82	0.22	0.95	0.80	0.01	0.32	0.41	0.24
11	0.24	0.02	0.36	0.66	0.37	0.22	0.02	0.19	0.66	0.37
12	0.61	0.01	0.22	0.66	0.71	0.26	0.00	0.12	0.66	0.71
13	0.64	0.00	0.31	0.37	0.07	0.64	0.00	0.23	0.37	0.07
14	0.05	0.18	0.14	0.38	0.10	0.02	0.04	0.08	0.08	0.69
15	0.86	0.08	0.00	0.38	0.08	0.75	0.05	0.03	0.38	0.09
16	0.18	0.01	0.03	0.66	0.71	0.17	0.01	0.02	0.66	0.71
17	0.53	0.04	0.29	0.24	0.39	0.32	0.01	0.27	0.08	0.69
18	0.43	0.13	0.36	0.42	0.44	0.50	0.03	0.38	0.74	0.88
19	0.48	0.02	0.31	0.47	0.94	0.30	0.01	0.19	0.47	0.94
20	0.63	0.05	0.81	0.29	0.10	0.58	0.03	0.38	0.38	0.08
21	0.90	0.17	0.61	0.00	0.82	0.85	0.06	0.46	0.40	0.31
22	0.75	1.00	0.41	0.64	0.05	0.55	0.42	0.29	0.73	0.26

that the model could always fall back to not incorporating any domain information. Hence, in theory, the models should incorporate domain information only when it is beneficial. Inspecting the optimized parameters (Table 3) reveals which types of domain information were incorporated by the model. In practice, the models optimized their parameters based on the training set only, using an inner cross-validation loop, hence can be suboptimal for the test set due to overfitting. Indeed, for participant 8, where the initial models performed best, incorporating domain information proved detrimental (Figure 5, gray lines). Generally, for recordings on which the initial models had low performance, the models had the most to gain from incorporating domain information, with diminishing returns for cases in which the initial model was already performing well (Figure 5C).

We will now look more closely into the effectiveness of the individual strategies.

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 2). By applying regularization to the covariance matrix, the estimated direction of the noise is steered towards being spherical (i.e. equal in all directions). Both initial models already apply ℓ_2 regularization. In the regression scenario, Kronecker shrinkage (Figure 5, left, “kronecker”), which controls the amount of shrinkage for the spatial and time dimensions separately, outperforms the ℓ_2 regularization approach (paired t-test: $t = 2.81$, $p < 0.05$). In the classification scenario, Kronecker shrinkage is beneficial in some cases, but detrimental in others (Figure 5, right, “kronecker”) and does not significantly outperform ℓ_2 regularization ($t = 1.48$, $p > 0.05$).

Table 3 lists the parameters chosen by the “all information” model, for each subject, fitted to the entire dataset. Heavy shrinkage is applied by most models (high values for α), however, many models made little use of shrinking the temporal component of the covariance matrix (low values for β).

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.[38] 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.

We introduced two strategies to bias the pattern matrix towards isolating the N400 component. First, a template of the N400 component was constructed by computing the grand-average pattern across participants other than the one currently being analyzed. Taken in isolation, the "multiple subjects" strategy improved the model beyond the "kronecker" model, both in the regression (paired t-test: $t = 4.89$, $p < 0.001$) and classification ($t = 3.27$, $p < 0.01$) scenarios. Second, the pattern was limited in time, allowing the model to focus on a single ERP component. Taken in isolation, the "temporal information" strategy performed equally well, both in the regression (vs. "kronecker": $t = 3.58$, $p < 0.01$ vs. "multiple subjects": $t = 0.24$, $p = 0.81$) and classification (vs. "kronecker": $t = 3.77$, $p < 0.01$ vs. "multiple subjects": $t = 1.10$, $p = 0.29$) scenarios. When both strategies were applied in tandem ("all information"), performance was increased even further, compared to the "multiple subjects" model, in both the regression ($t = 2.39$, $p < 0.05$) and classification ($t = 3.81$, $p < 0.01$) scenarios. Compared to the "temporal information" model, the "all information" model's performance was significantly better than the "temporal information" model only in the regression scenario ($t = 2.54$, $p < 0.05$).

Looking at the "all information" model, for most participants, the optimizer chose to bias the pattern matrix towards the grand-average (Table 3, high values for ρ). Then, for a selection of participants, the optimizer chose to further refine the pattern by restricting it to a narrow time window surrounding the N400 component (Table 3, μ around 400 ms and low values for σ). Overall, the optimized patterns show a much more pronounced N400 effect (Figure 7) compared to the patterns of the initial models (Figure 6), indicating that the N400 was indeed a stable feature of interest that generalizes well beyond the training set. For some participants, the initial models 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 participants 5 and 13).

4 Discussion

We have demonstrated how domain information can be incorporated into general purpose linear models with the post-hoc modification framework. When using this framework, we shift our focus away from estimating a weight matrix towards the subproblems of 1) modeling the signal of interest (the pattern matrix), 2) establishing the relationship between input features (the data covariance) and 3) performing a normalization step.

As Haufe et al. (2014) pointed out, there is a strong parallel between the pattern matrix and the concept of a leadfield or "forward solution", as used in source estimation.[39] From this perspective, the decoding targets are similar to the source dipoles and the weight matrix is similar to the inverse operator. The main difference is that the pattern matrix is not constructed by modelling volume conduction in the head, but rather through a linear machine learning algorithm. In this work, we have extended the parallel further by observing that the domain of source estimation has always approached the computation of the inverse operator (or spatial filters) as a multi-step process, where first the covariance matrix is computed on the sensor data, which is then combined with the leadfield,[40] and we may use the same approach when fitting decoding models.

From this point of view, possibilities for incorporating domain information into the model

[38] Kutas and Federmeier, 2011

[39] Hämäläinen et al., 1993

[40] Hämäläinen et al., 1993; Sekihara and Nagarajan, 2008

become obvious. In this work, we have explored a few possibilities to modify 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 (Figure 5).

The post-hoc modification framework opens up a wide range of possibilities to design strategies for incorporating domain information. Our examples aim to demonstrate the capabilities of the framework and serve as inspiration for designing new strategies for other study paradigms or recording modalities.

One may explore more informative priors for the covariance matrix than an identity matrix. For example, bandpass filtering the signal will introduce a predictable dependency between consecutive time samples, which may be used as a shrinkage target for the temporal component of the covariance matrix. Likewise, for EEG and MEG studies, volume conduction in the head will impose a predictable dependency between the signals at different sensors, which can be modeled using a leadfield.[41] Also for the pattern matrix, there are other avenues of domain information to explore. For example, the N400 potential has a well defined spatial signature[42] that may be used as a prior for the pattern matrix. Finally, there might also be opportunities to incorporate domain information through the normalizer, although we did not explore this in this study and treated the normalizer as a mere scaling of the model output. Inspiration for normalization schemes can be found in the beamformer literature.[43] For example, if the pattern matrix has been crafted to be in some measurement unit, one may wish to enforce that model output adheres to the same unit. The unit-gain constraint the of the linearly constrained minimum variance (LCMV) beamformer, $\mathbf{WP} = \mathbf{I}$, ensures that units are preserved. Using post-hoc modification, we can apply the unit-gain constraint of the LCMV beamformer to any linear model by using:

$$\widetilde{\boldsymbol{\Sigma}}_{\hat{\mathbf{Y}}} = (\widetilde{\mathbf{P}}^{\top} \widetilde{\boldsymbol{\Sigma}}_{\mathbf{X}}^{-1} \widetilde{\mathbf{P}})^{-1}. \tag{14}$$

A common approach to reducing data dimensionality is to first apply a spatial filter, followed by a temporal filter.[44] While the resulting model becomes blind to interactions happening in a different locations at different times, the reduction in dimensionality will decrease overfitting, potentially offsetting the disadvantages. Such an approach can be explored in the post-hoc framework as well, with the benefits that the choice of whether to treat space and time separately or jointly no longer has to be made model-wide, but can be done for each subcomponent separately. For example, the empirical covariance matrix can be replaced with the Kronecker product of the spatial and temporal covariance matrices, and the pattern matrix can be replaced with the outer product of a spatial and temporal pattern, for example obtained using non-negative matrix factorization.[45] As in our example modifications, a hyperparameter can be defined to scale the matrices between the full spatio-temporal forms and the reduced forms that treat space and time separately, allowing the model to dynamically seek out the most suitable approach.

In our examples, we optimized the hyperparameters (α, β, ρ, μ, σ) using only the decoding performance of the resulting model as performance metric, but one can imagine using other

[41] Hämäläinen et al., 1993

[42] Kutas and Federmeier, 2011

[43] Sekihara and Nagarajan, 2008

[44] Blankertz et al., 2008; Hoffmann et al., 2006; Rivet et al., 2009

[45] Delis et al., 2016

metrics. For example, decoding models are often employed to explore the signal of interest that was learned, in which case interpretability of the model is more important.[46] In this case, one may wish to optimize a tradeoff between sparsity of the pattern matrix (not to be confused with sparsity of the weight matrix) and decoding performance.[47]

Furthermore, 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 artefacts can be a predictor for FAS[48] 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.

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. If $\widetilde{\mathbf{P}}$ is completely fixed, the model is transformed into a beamformer[49] 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.[50]

Taking the opposite view, one may wish to use the post-hoc modification framework to steer the model away from signals that are known to be relevant for the decoding task, in order to force the model to explore as yet unknown signals. In this case, the known signals of interest may be removed from $\widetilde{\mathbf{P}}$, which will result in this signal being explicitly tagged as noise to be filtered out.

While the above examples are all in the domain of machine learning, linear models are also widely used in the domain of statistics, where applications range from familiar t-tests, through ANOVA F-tests, to more advanced multilevel models. The post-hoc framework can by applied here as well. For example, the "multiple subjects" model, which biases the pattern matrix to a group average, parallels a linear mixed-effects model which performs a similar trick to compute both a group-level slope as well as slopes for individuals.[51]

We envision the post-hoc modification framework as an iterative process, where an initial model is fitted to the data without any restrictions. This is followed by an inspection of the resulting patterns, covariance and normalizer by the data analyst, who then proceeds to place restrictions using post-hoc modification. The model is fitted again, taken the new restrictions into account and the cycle continues until finally, a model is obtained that satisfies all requirements of the study. In this manner, machine learning becomes less of a "black box" and more a dialogue between data analyst and model.

5 Conclusion

In the post-hoc modification framework, the weight matrix of a linear model is regarded as a combination of three subcomponents: a pattern matrix, a data covariance matrix, and a normalizer. The problem of computing a weight matrix can accordingly be split up into the subproblems of estimating each subcomponent. We showed how domain information can often be straightforwardly formulated in terms of these subcomponents. An initial estimate for the subcomponents can be obtained by decomposing the weight matrix as produced by a linear machine learning algorithm. In what we call "post-hoc modification" each subcomponent can then be refined at will, which provides opportunities to incorporate domain information.

[46] Haufe et al., 2014; Parra et al., 2003

[47] Kia et al., 2017

[48] see electronic supplementary information: "Decoding performance using EOG channels only", Quax2019

[49] van Vliet et al., 2016; Treder et al., 2016

[50] van Vliet et al., 2018

[51] Baayen et al., 2008

Afterwards, the modified subcomponents are re-assembled into a weight matrix, which now incorporates the injected domain information.

We have presented some strategies for incorporating domain information 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, as domain information is study specific, so are post-hoc modification strategies. While some of the presented strategies can be appropriate for other EEG studies, they mainly serve as examples of how the post-hoc modification framework 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: The relationship between $\Sigma_{\mathbf{X}}^{-1}$ and whitening

The $\Sigma_{\mathbf{X}}^{-1}$ term in equation 4 represents a whitening transform that is computed using $\mathbf{X}$ and subsequently applied to both the data $\mathbf{X}$ and the pattern matrix $\mathbf{P}$. This becomes clear when we rewrite $\Sigma_{\mathbf{X}}^{-1}$ in terms of the eigendecomposition of $\Sigma_{\mathbf{X}}$:

$$\Sigma_{\mathbf{X}}^{-1} = \mathbf{Q}\Lambda^{-1}\mathbf{Q}^{\mathsf{T}}, \tag{15}$$

where $\mathbf{Q}$ is a matrix where each row is an eigenvector of $\Sigma_{\mathbf{X}}$ and Λ is a diagonal matrix where each diagonal element is the corresponding eigenvalue. Then, the linear transformation Φ that whitens $\mathbf{X}$ is defined as:

$$\Phi = \mathbf{Q}\Lambda^{-1/2}. \tag{16}$$

Hence, $\Sigma_{\mathbf{X}}^{-1}$ can be rewritten as:

$$\begin{aligned} \Sigma_{\mathbf{X}}^{-1} &= \mathbf{Q}\Lambda^{-1/2}\Lambda^{-1/2}\mathbf{Q}^{\mathsf{T}}, && (17) \\ &= \Phi\Phi^{\mathsf{T}}. && (18) \end{aligned}$$

and we can show that that when the model is applied, it performs a whitening transformation on both the data **X** and the pattern matrix **P**:

$$\begin{aligned}\widehat{\mathbf{Y}} &= \mathbf{X}\mathbf{W} && (19)\\ &= \mathbf{X}\Sigma_{\mathbf{X}}^{-1}\,\mathbf{P}\,\Sigma_{\widehat{\mathbf{Y}}}, && (20)\\ &= (\mathbf{X}\Phi)\,(\Phi^{\top}\mathbf{P})\,\Sigma_{\widehat{\mathbf{Y}}}. && (21)\end{aligned}$$

Appendix B: 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,[52] which states that for any matrices **A**, **B**, **U**, and **V** of appropriate size, the following holds:

$$(\mathbf{A}-\mathbf{U}\mathbf{B}\mathbf{V})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}. \tag{22}$$

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

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

$$\begin{aligned}\widetilde{\mathbf{W}} &= [\alpha\gamma\mathbf{I} + (1-\alpha)(\beta\Sigma_{\mathrm{s}}\otimes\mathbf{I}_{\mathrm{t}} + (1-\beta)\mathbf{X}^{\top}\mathbf{X})]^{-1}\,\widetilde{\mathbf{P}}\,\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}, && (23)\\ &= [\alpha\gamma\mathbf{I} + (1-\alpha)\beta\Sigma_{\mathrm{s}}\otimes\mathbf{I}_{\mathrm{t}} + (1-\alpha)(1-\beta)\mathbf{X}^{\top}\mathbf{X}]^{-1}\,\widetilde{\mathbf{P}}\,\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}, && (24)\\ \mathbf{A} &= \alpha\gamma\mathbf{I} + (1-\alpha)\beta\Sigma_{\mathrm{s}}\otimes\mathbf{I}_{\mathrm{t}}, \quad \mathbf{B} = \mathbf{I}, \quad \mathbf{U} = -(1-\alpha)(1-\beta)\mathbf{X}^{\top}, \quad \mathbf{V} = \mathbf{X}, && (25)\\ \mathbf{G} &= \mathbf{A}^{-1}\mathbf{U}, \quad \mathbf{K} = \mathbf{I} + \mathbf{X}\mathbf{G}, && (26)\\ \widetilde{\mathbf{W}} &= (\mathbf{A}^{-1} + \mathbf{G}\mathbf{K}^{-1}\mathbf{X}\mathbf{A}^{-1})\widetilde{\mathbf{P}}\widetilde{\Sigma}_{\widehat{\mathbf{Y}}}. && (27)\end{aligned}$$

Appendix C: Optimizing the inner cross-validation loop

Our optimization strategy (section 2.10) depends on evaluating the leave-one-out performance of the model many times. The computationally most expensive operation in equation 27 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 **K**, which in the case of this matrix means the i'th row and column are removed. Salmen et al. (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 **X** and **Y** in the same manner. The update rule of the inverse

[52] Tylavsky and Sohie, 1986

can then be formulated as:

$$\mathbf{K}_{(i)}^{-1} = (\mathbf{K}_{(1)} + \mathbf{D})^{-1}, \tag{28}$$

$$\mathbf{D} = \mathbf{K}_{(1)} - \mathbf{K}_{(i)} = \begin{pmatrix} 0 & \cdots & k_{1,1} - k_{2,i} & \cdots 0 \\ 0 & \cdots & \vdots & \cdots 0 \\ k_{1,1} - k_{i,2} & \cdots & k_{1,i} - k_{i,i} & \cdots k_{1,n} - k_{i,n} \\ 0 & \cdots & \vdots & \cdots 0 \\ 0 & \cdots & k_{n,1} - k_{n,i} & \cdots 0 \end{pmatrix}, \tag{29}$$

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

To apply the inversion lemma (equation 22), **D** must be formulated in terms of **UBV**, which yields:

$$\mathbf{U} = \begin{pmatrix} k_{1,1} - k_{2,i} & 0 \\ k_{2,1} - k_{3,i} & 0 \\ \vdots & \vdots \\ k_{(i-1),1} - k_{(i-1),i} & 0 \\ k_{i,1} - k_{i,i} & 1 \\ k_{(i+1),1} - k_{(i+1),i} & 0 \\ \vdots & \vdots \\ k_{n,1} - k_{n,i} & 0 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{30}$$

$$\mathbf{V} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ k_{1,1} - k_{2,i} & k_{2,1} - k_{3,i} & \cdots & k_{(i-1),1} - k_{(i-1),i} & 0 & k_{(i+1),1} - k_{(i+1),i} & \cdots & k_{n,1} - k_{n,i} \end{pmatrix}. \tag{31}$$

Then, applying equation 22:

$$\mathbf{K}_{(i)}^{-1} = (\mathbf{K}_{(1)} + \mathbf{UBV})^{-1} = \mathbf{K}_{(1)}^{-1} - \mathbf{K}_{(1)}^{-1}\mathbf{U}(\mathbf{B}^{-1} + \mathbf{V}\mathbf{K}_{(1)}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{K}_{(1)}^{-1}. \tag{32}$$

References

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, *59*(4), 390–412. doi:10.1016/j.jml.2007.12.005

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

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Müller, K.-R. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. doi:10.1109/MSP.2008.4408441

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

Delis, I., Onken, A., Schyns, P. G., Panzeri, S., & Philiastides, M. G. (2016). Space-by-time decomposition for single-trial decoding of M/EEG activity. *NeuroImage*, *133*, 504–515. doi:10.1016/j.neuroimage.2016.03.043

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*, *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., Goj, R., Jas, M., Brooks, T., Parkkonen, L., & 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. S., 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., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalography - theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, *65*(2), 414–507. doi:10.1103/revmodphys.65.413

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

Hauk, O., Stenroos, M., & Treder, M. (2019). Towards an objective evaluation of EEG/MEG source estimation methods: The linear tool kit. *bioRxiv*, 672956. doi:10.1101/672956

Hoffmann, U., Vesin, J.-m., & Ebrahimi, T. (2006). Spatial filters for the classification of event-related potentials. *Neural Networks*, (April), 26–28.

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.

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

Kohler, T., Wagner, M., Fuchs, M., Wischmann, H. .-., Drenckhahn, R., & Theissen, A. (1996). Depth normalization in MEG/EEG current density imaging. *IEEE Engineering in Medicine and Biology Society, 18th Annual International Conference*, 812–813. doi:10.1155/2011/758973

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

Kutas, M., & Iragui, V. (1998). The N400 in a semantic categorization task across 6 decades. *Electroencephalography and Clinical Neurophysiology - Evoked Potentials*, *108*(5), 456–471. doi:10.1016/S0168-5597(98)00023-9

Lin, F. H., Belliveau, J. W., Dale, A. M., & Hämäläinen, M. S. (2006). Distributed current estimates using cortical orientation constraints. *Human Brain Mapping*, *27*(1), 1–13. doi:10.1002/hbm.20155

Loan, C. F. V. (2000). The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, *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. *Journal of Neural Engineering*, *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. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 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). 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

de Cheveigné, A., & Simon, J. Z. (2008). Denoising based on spatial filtering. *Journal of Neuroscience Methods*, *171*(2), 331–339. doi:10.1016/j.jneumeth.2008.03.015

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 (A. Rodriguez-Fornells, Ed.). *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

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., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & 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. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 4600–4604. doi:10.1109/IEMBS.2011.6091139

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

Rivet, B., Souloumiac, A., Attina, V., & Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: Application to brain-computer interface. *IEEE Transactions on Biomedical Engineering*, *56*(8), 2035–2043. doi:10.1109/TBME.2009.2012869

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.). Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*, *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*(1), 483–509. doi:10.1146/annurev-psych-120710-100412

Treder, M. S., Porbadnigk, A. K., Shahbazi Avarvand, F., Müller, K.-R., & Blankertz, B. (2016). The LDA beamformer: Optimal estimation of ERP source time series using linear discriminant analysis. *NeuroImage*, *129*, 279–291. doi:10.1016/j.neuroimage.2016.01.019

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

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