Representational similarity analysis with multiple models and cross-validation in magnetoencephalography

Gustaf Lönn

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 4.5.2017

Thesis supervisor:

Prof. Lauri Parkkonen

Thesis advisor:

D.Sc. (Tech.) Linda Henriksson

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Gustaf Lönn

Title: Representational similarity analysis with multiple models and cross-validation in magnetoencephalography

Date: 4.5.2017 Language: English Number of pages: 7+58

Department of Neuroscience and Biomedical Engineering

Professorship: Neuroimaging Methods

Supervisor: Prof. Lauri Parkkonen

Advisor: D.Sc. (Tech.) Linda Henriksson

Due to the increased availability of computational resources, more complex analysis methods taking advantage of the inherent high dimensionality of the data can be employed in functional brain imaging, allowing for development and assessment of intricate models. Models are utilized for both explanatory and predictive purposes and permits generalization from individual brain responses to the functioning principles of the brain. Representational similarity analysis (RSA) is a framework allowing evaluation of the performance of models by comparison to imaging data via the use of representational distance matrices (RDMs). This type of analysis also enables finding the linear combination of models that best explains the imaging data, something that successfully has been applied to functional magnetic resonance imaging (fMRI) data. In this thesis, RSA is applied to magnetoencephalography (MEG) data on the sensor level using a spatiotemporal searchlight approach. The method is validated through simulations based on the forward-inverse modelling framework of MEG, where complete control over the source activation is exerted. Non-negative least squares fitting of a linear combination of multiple models is carried out, with an additional option of performing leave-k-out cross-validation to prevent overfitting to the simulated dataset. Finally, the method is applied to real MEG data.

Keywords: magnetoencephalography, representational similarity analysis, cross-validation, spatiotemporal searchlight

AALTO-UNIVERSITETET
HÖGSKOLAN FÖR TEKNIKVETENSKAPER

SAMMANDRAG AV
DIPLOMARBETET

Författare: Gustaf Lönn

Titel: Representationslikhetsanalys med flera modeller och korsvalidering inom magnetoencefalografi

Datum: 4.5.2017 Språk: Engelska Sidantal: 7+58

Institutionen för neurovetenskap och biomedicinsk teknik

Professur: Neuroimaging Methods

Övervakare: Prof. Lauri Parkkonen

Handledare: TkD Linda Henriksson

Tack vare den ökade tillgången till beräkningsresurser kan mer komplexa analysmetoder som utnyttjar den inneboende höga dimensionaliteten hos datan användas inom funktionell neuroradiologi, vilket möjliggör utveckling och utvärdering av invecklade modeller. Modeller kan användas i både förklarande och förutsägande syfte och tillåter generalisering av individuella hjärnresponser till hjärnans funktionsprinciper. Representationslikhetsanalys (RSA) är ett ramverk som möjliggör utvärdering av modellers prestanda genom jämförelse med radiologidata via användning av representationsavståndsmatriser (RDM). Den här typen av analys gör det också möjligt att bestämma den linjära kombination av modeller som bäst förklarar radiologidatan, något som redan framgångsrikt använts för funktionell magnetresonanstomografidata. I det här arbetet tillämpas RSA på magnetoencefalografidata (MEG) på sensornivå med hjälp av ett spatiotemporalt sökljus. Metoden valideras genom simulationer baserade på ramverket för forward-inverse-modellering för MEG, inom vilket fullständig kontroll över källaktivationen kan utövas. Anpassning av en linjär kombination av modeller görs med hjälp av den icke-negativa minsta kvadrat-metoden och innefattar också ett alternativ att utföra utelämna-*k* korsvalidering för att förhindra överanpassning till den simulerade datan. Slutligen tillämpas metoden på verklig MEG-data.

Nyckelord: magnetoencefalografi, representationslikhetsanalys, korsvalidering, spatiotemporalt sökarljus

Contents

Notation

A	matrix
A^T	transposed matrix
$\mathbf{x}$	column vector
$(x_1, \ldots, x_n)$	column vector
$[a : b : c]$	vector of the values from a to c in steps of b
$\mu_{\mathbf{x}}$	mean of vector $\mathbf{x}$
$\sigma_{\mathbf{x}}$	standard deviation of vector $\mathbf{x}$
$\tilde{\mathbf{x}}$	the vector $\mathbf{x}$ normalized
$\|\|\mathbf{x}\|\|$	L^2-norm of the vector $\mathbf{x}$
$\|\|\mathbf{x}\|\|_A$	L^2-norm of the vector $\mathbf{x}$ weighted by the matrix A
$\mathbb{R}^n$	set of real column vectors of length n
$\mathbb{R}^{n \times m}$	set of real matrices with n rows and m columns
$\mathrm{corr}(\mathbf{x}, \mathbf{y})$	correlation between the vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathrm{cov}(\mathbf{x}, \mathbf{y})$	covariance of the vectors $\mathbf{x}$ and $\mathbf{y}$
$d_{\mathrm{Euc}}(\mathbf{x}, \mathbf{y})$	Euclidean distance between the vectors $\mathbf{x}$ and $\mathbf{y}$
$d_{\mathrm{corr}}(\mathbf{x}, \mathbf{y})$	correlation distance between the vectors $\mathbf{x}$ and $\mathbf{y}$
τ_a	Kendall's rank correlation coefficient
ρ	Spearman's rank correlation coefficient
R	Pearson's correlation coefficient
w_1, w_2	component model weights in fitted model
$n!$	factorial
$\binom{n}{k}$	binomial coefficient
$p(X = i)$	probability of the stochastic variable X obtaining the value i
$E[X]$	expected value of the stochastic variable X
$\mathtt{a}$	letter in an alphabet
$O(N)$	computational complexity class

Abbreviations

AP	action potential
BEM	boundary element model
CV	cross-validation
EEG	electroencephalography
EMEG	referring to both electroencephalography and magnetoencephalography
EOG	electrooculography
EPSP	excitatory postsynaptic potential
EVC	early visual cortex
fMRI	functional magnetic resonance imaging
ICP	iterative closest point
IPSP	inhibitory postsynaptic potential
IT	inferior temporal cortex
LSQ	linear least squares
MEG	magnetoencephalography
MNE	minimum norm estimation
MR	magnetic resonance
MRI	magnetic resonance imaging
MSD	mean squared deviation
PSP	postsynaptic potential
RDM	representational distance matrix
RMS	root mean square
RSA	representational similarity analysis
SD	standard deviation
SNR	signal-to-noise ratio
SQUID	superconducting quantum interference device
ssRSA	spatiotemporal searchlight representational similarity analysis
SVM	support vector machine

1 Introduction

While the fundamental challenge of neuroscience is to understand the anatomy and physiology of nervous systems in general, a large focus lies on the study of the human brain and corresponding animal models (Manger et al., 2008) and especially on the human cerebral cortex. For this purpose, noninvasive neuroimaging techniques are used and have in recent decades made large leaps forward, both in instrumentation and data analysis. For overviews of different modalities, see e.g., the works by Haacke and colleagues (1999), by Huettel and colleagues (2004) and by Baillet and colleagues (2001). To examine the flow and processing of information occurring in the cortex, anatomical images obtained via for example magnetic resonance imaging are not adequate. A temporal dimension and some type of measurement of activity must be included. Two prominent methods of functional brain imaging are functional magnetic resonance imaging (fMRI) and magnetoencephalography (MEG), both relying on quite different physiological and physical phenomena for determining brain activity. fMRI uses nuclear magnetic resonance to identify changes in the blood oxygen level shown to be related to neural activity (Logothetis et al., 2001), while MEG detects extracranial magnetic fields arising from the electrical activity in the brain (Hämäläinen et al., 1993). MEG is therefore a more direct measure of the actual neuronal activity.

One common problem in working with these imaging modalities is that they generate huge amounts of high-dimensional data. Many common methods of analysis, e.g., event-related fields in MEG and the general linear model in fMRI, do not exploit the high dimensionality fully. The assumption that the data contain more interesting information than what is extracted by these methods does therefore not seem unfounded. As more computing power becomes available, complex analysis methods that previously were not viable can now be used (see e.g., the works by Cichy and colleagues (2014) and by Cox and colleagues (2003)). Computationally expensive statistical and machine learning inspired methods can be used to extract information and high-dimensional patterns invisible to the researcher. This in an ongoing effort in many aspects, and this thesis reviews an existing framework called representational similarity analysis (RSA) utilizing up to the full dimensionality of the data (Kriegeskorte et al., 2008). It has successfully been used on fMRI data (Kriegeskorte and Kievit, 2013), but published applications on MEG data are few.

The emphasis of the RSA framework lies not only on the high dimensionality, but also on the inherent ability to utilize models. Models can be can be used both to explain and predict brain responses as a function of stimuli. The eventual goal is however to draw more broad conclusions about how certain subsystems in the brain work based on individual brain responses. RSA is very general and can be applied to in principle any type of measurement where different stimuli give rise to different quantitative observations, eliminating the need of finding matching units between modalities. It transforms both models and measured data to data of the same type, allowing direct comparison and fitting of model parameters, no matter how the model is constructed or how the measurement is conducted. To add complexity, models can under certain circumstances be linearly combined and a least-squares fit can be applied to find the best weights for the linear combination (Khaligh-Razavi and Kriegeskorte, 2014). This allows for multiple differently modelled phenomena to explain the observed data. Both the weights and the correspondence of the fitted model to the data can be of interest. As with all fitting, there is a risk of overfitting to the observed data. To tackle this issue, cross-validation can be performed.

In testing a novel data analysis method or in applying it to data of a novel type, simulations are useful for determining how well the method works and for identifying possible issues. They give the researcher maximal control over the data, not only enabling identification of boundaries for parameters of the analysis method itself, but also aiding in the design of upcoming experiments. Owing to the comprehensive mathematical framework of MEG (Hämäläinen et al., 1993), simulations can easily be conducted. The desired activation region and type can be selected and the measured signals can be modelled on the basis of these. Using the same framework, the underlying brain activation can also be inferred from the MEG signal.

In this thesis, we will examine possibilities and issues in fitting multiple models both with and without cross-validation to MEG data within the RSA framework. The fits are done at the sensor level, i.e. no modelling of the actual brain activity is performed. Due to the cost of MRI scans required for accurate source modelling, both money and time could be saved if only the MEG data by itself can be used to model and at least crudely localize neural activity.

2 Background

In this chapter, we will review the principles of two different topics: measurement and data analysis. In the case of measurement, we will discuss a specific measurement technique for observing brain activity. In the case of data analysis, we will on the other hand present a framework that can be used for analyzing data obtained by a vast range of measurement techniques.

2.1 Magnetoencephalography

In magnetoencephalography (MEG), miniature fluctuations in the magnetic field close to the outside of the skull are measured by sensitive detectors. It is notable that MEG is a noninvasive imaging procedure, which makes it suitable for studying the human brain. A comprehensive overview of most aspects of MEG can be found in Hämäläinen and colleagues (1993).

Another closely related measurement technique is electroencephalography (EEG), where a grid of electrodes is attached to the scalp to measure changes in the electric potential. The methods of processing and analyzing EEG data are highly similar to those for MEG data.

2.1.1 Neuroanatomy and -physiology

The neuroanatomy and -physiology required to follow this thesis will shortly be reviewed here. For a deeper overview of the subject, consult e.g., Bear and colleagues (2016).

The nowadays widely accepted neuron doctrine states that the nervous system consists of discrete building blocks, neurons. The whole human brain is estimated to contain on average 86 billion neurons (Azevedo et al., 2009) and only in the cerebral cortex, the evolutionary most recent part of the brain, 0.15 quadrillion connections between these (Pakkenberg et al., 2003). The very high density of neurons imposes limits on the accuracy with which we can observe neuronal activity noninvasively.

Each neuron is a biological cell, whose main parts are the body, the axon and dendrites. The membrane of the cell separates the inside, containing a fluid called *cytoplasm*, from the outside of the cell and is interspersed with channel proteins, allowing passage of certain ions (mainly K^+, Na^+ and Cl^-) depending on the properties of the electrochemical environment. This possible passage of ions make neurons *excitable cells*, which means that they can be polarized. Due to ions pumps located in the membrane maintaining electrochemical gradients between the inside and outside of the cell, the resting neuron has a negative membrane potential, called the *resting potential*.

Neurons can communicate with each other via electrical or chemical means. The dendrites can be thought of as the inputs and the axon as the output. The axon forms connections, *synapses*, onto the dendrites of other neurons. There is no direct physical connection between the axon and a dendrite, but a narrow synaptic cleft between the neurons. When observing a specific synapse, the neuron on the axon end of the synapse is referred to as the presynaptic neuron and the neuron on the dendrite end of the synapse is referred to as the postsynaptic neuron.

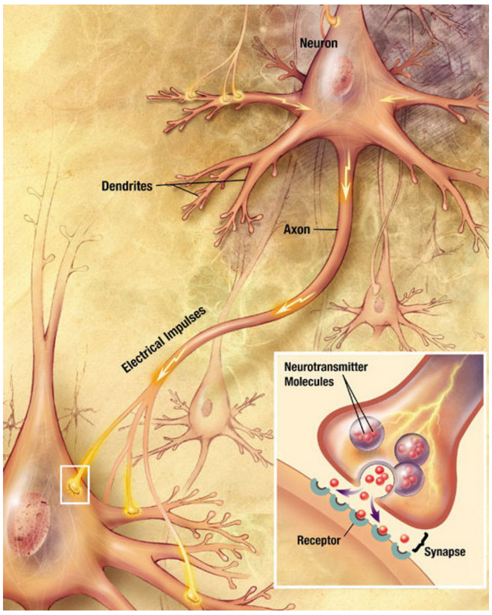

Figure 1: Illustration of two neurons. The AP travels along the axon to the synapse. Inset: Magnification of the synapse where neurotransmitter molecules are released onto receptors of the postsynaptic neuron.

An inflow of positively charged ions, *depolarization*, can occur as a result of activated channel proteins. If this depolarization reaches a certain threshold, an *action potential* (AP) is generated. The AP is an unstoppable rapid depolarization and can be seen as a sharp spike in electrical recordings. It travels across the membrane along the axon of the cell to finally reach the synapse. At the synapse, the arrival of the electrical AP results in the release of chemical compounds, *neurotransmitters*, from the presynaptic neuron into the synaptic cleft. The neurotransmitters diffuse over the cleft and bind to receptors on the postsynaptic neuron, resulting in channel openings/closings or more complex behavior. The communication between neurons is therefore of chemical and not electrical nature (although there are exceptions to this rule). This activity induces changes in the electrochemical composition of the cytoplasm of the postsynaptic neuron, and can lead to a local polarization. This is called a *postsynaptic potential* (PSP), and can be either excitatory (EPSP, depolarization) or inhibitory (IPSP, hyperpolarization). If the EPSP is strong enough, it can trigger an AP in the postsynaptic neuron. Usually, however, spatial and/or temporal summation of EPSPs must occur to elicit an AP.

The cerebral cortex is a thin folded sheet consisting of six different layers. The main type of neurons present are the pyramidal neurons, distinguished by a pyramidal cell body and far-reaching apical dendrites. The orientation of the dendrites is orthogonal to the actual cortex itself and the dendrites of different neurons run in parallel.

The cortex can be divided into different regions. A broad division can be done into the temporal lobe, occipital lobe, parietal lobe and frontal lobe. Different lobes

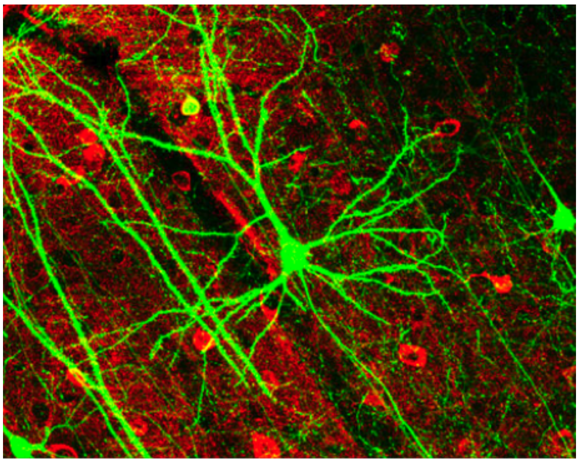

Figure 2: A pyramidal neuron stained with green fluorescent protein (Lee et al., 2005).

are crudely put specialized at different functions, e.g., the primary visual cortex responsible for the rudiments of vision is located in the occipital lobe while higher cognitive functions like planning are located in the frontal lobe.

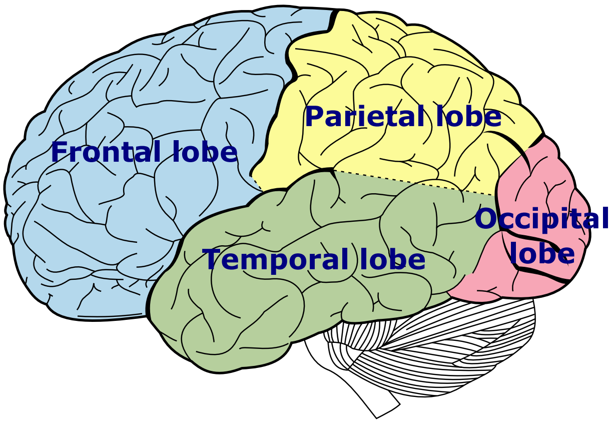

Figure 3: The lobes of the cerebral cortex illustrated.

2.1.2 The physical basis of the MEG signal

As explained by Biot-Savart's law, electrical currents give rise to a magnetic field. Therefore, the flow of ions omnipresent in neuronal activity results in changes in the magnetic field both inside and outside the brain. There are two main events the give rise to currents: the AP and the PSP.

When the AP is travelling down the axon, the area in front of the spike is quickly depolarized, while the area behind the spike is more slowly repolarized. This behavior can be described by two oppositely aligned current dipoles, forming a current quadrupole (Hämäläinen and Hari, 2002). On the other hand, the PSP is the result of ion flow over the postsynaptic membrane, which corresponds to a current dipole. The magnetic field generated by a quadrupole decreases faster with distance (proportional to $1/r^3$) than for a dipole (proportional to $1/r^2$). In addition, the time course for the AP is tens of times shorter than for the PSP, resulting in more prominent temporal summation for PSPs. Combining these insights leads to the conclusion that the changes in the magnetic field detected mostly arise from PSPs.

Murakami and colleagues (2006) estimate using simulations that the current dipole strength associated with a single PSP of a pyramidal cell is on the order of $0.29-0.90$ pAm. Based on this data, they approximate that about 50000 synchronized neurons are needed to generate a measurable signal. As the cortex is highly folded, there might be cancellation of the magnetic field generated by one part of the cortex by another part, which means that depending on the geometry, even more neurons might need to be active to generate a measurable signal.

2.1.3 Measurement device

MEG is as previously mentioned noninvasive, which means that all measurements are external to the body. The only preparation required is removing magnetic objects and attaching positioning coils to the head of the subject. Electrooculography (EOG) can also be measured at the same time to make removal of artifacts easier. The MEG device used for gathering the data used for analysis in this work is a 306-channel device produced by Elekta Oy. The lower part of the device is formed like a helmet to have the sensors as close to the skull as possible.

The sensors are superconducting quantum interference devices (SQUIDs). Due to their low operating temperature, they have to be cooled down by liquid helium. By using sets of three SQUIDs arranged in a certain pattern (see Figure 4 for a 2D version) in 102 locations, the gradients of the magnetic field can be measured in two directions and the magnitude in one. The gradiometers are not sensitive to uniform distortions of the magnetic field, since they determine the change in the magnetic field over two points. The magnetometers however measure the magnitude in one point and are therefore more susceptible to external noise.

As the currents involved in neuronal activity are quite small and the magnetic field of the PSPs decreases as $1/r^2$, the resulting electromagnetic signals outside of the head are on the order of 1 fT-1 pT. In comparison, the magnitude of the earth's magnetic field is on the order of 10 μT. Due to the magnitude of the earth's magnetic field and other disturbances, the device is located inside a magnetically shielded room.

2.1.4 Theoretical framework for analysis

Maxwell's equations form the basis for understanding MEG measurements. Due to the slow temporal dynamics, a quasi-static approximation can be used, where the temporal dependence is ignored (Hämäläinen et al., 1993). The approach we will be using here is called distributed source modelling. We assume that the brain activity can be modelled by a certain number of current dipoles in certain locations. Using

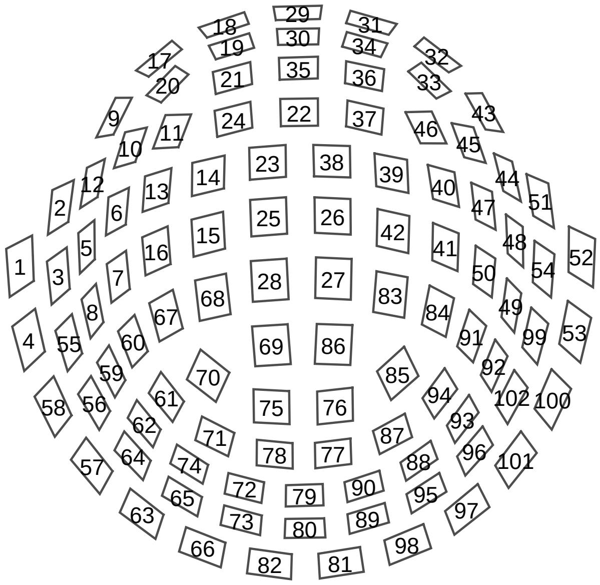

Figure 4: The spatial layout of channel locations for the flattened MEG helmet. Channels are later on referred to by their indices in some visualizations of results.

a spherical model for the head, the calculations involved become quite simple and can be solved analytically. One problem is that radial sources cannot be detected. Of more practical interest, employing a more realistic geometry obtained from MR images, the equations can be solved numerically using the boundary element model (BEM). Using this method, we can determine what activation pattern the SQUIDs will detect when a certain source is active. However, the number of sources are usually much larger than the number of sensors, which means that the inverse problem (estimating the source activations from the detected magnetic signals) is ill-posed. There are a number of tricks that can be used to constrain the solution; here we will focus on minimum norm estimation (MNE). In this way, MEG can, as e.g., fMRI, be used to localize brain activity. The spatial resolution is however much lower than for fMRI, but the temporal resolution is significantly better (Baillet et al., 2001).

We will here describe the variables and equations used in the MNE solution (Hämäläinen and Ilmoniemi, 1994; Wang et al., 1992). We start by assuming that we have N spatially distributed current dipole sources inside the brain. The changes in the magnetic field generated by these sources are detected by M sensors in the MEG helmet. As previously noted, we can numerically determine the pattern seen by the sensors as a function of the activation strength for each single source. This information can be summarized in the gain matrix $A \in \mathbb{R}^{M \times N}$, where the element a_{ij} is the proportionality constant between the signal seen by sensor i and the amplitude of source j. Assuming a linear summation of sources, we can write

$$\mathbf{y} = A\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^M$ is the resulting signal pattern, $\mathbf{x} \in \mathbb{R}^N$ is the source activation and $\mathbf{n} \in \mathbb{R}^M$ is the sensor noise. Equation 1 is called the *forward model*. Once the gain matrix for a system has been determined, the forward model can straightforwardly be applied.

In practice, the source activation is what interests us and is unknown. Also, the number of assumed sources is larger than the number of sensors, i.e. $N > M$. The inverse problem is underdetermined in this case, since there are more variables than equations. To formulate the MNE problem, we also have to introduce the source covariance matrix $C_x \in \mathbb{R}^{N \times N}$ and the noise covariance matrix $C_n \in \mathbb{R}^{M \times M}$. C_n can be calculated by measuring an empty room, but C_x cannot be determined by measurement and is usually assumed to be diagonal. The MNE solution is described by the optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} ||\mathbf{x}||^2_{C_x} \quad \text{when} \quad ||\mathbf{y} - A\mathbf{x}||^2_{C_n} < \delta, \tag{2}$$

where the L_2-norm with regard to a matrix is defined as

$$||\mathbf{a}||^2_A = \mathbf{a}^T A^{-1} \mathbf{a}.$$

In other words, we want to minimize the norm of the source activation with respect to C_x, with the additional constrain that the forward modelled signal must be very close to the measured signal. Using the technique of Lagrange multipliers, it can be shown that the solution to Equation 2 is given by

$$\hat{\mathbf{x}} = C_x A^T \left(A C_x A^T + \lambda^2 C_n \right)^{-1} \mathbf{y}, \tag{3}$$

where λ is a regularization parameter. This is called the *inverse model*. The formula shows that the inverse problem using MNE also is linear.

2.1.5 Common methods of analysis

Although MEG is a much more recent technique than EEG, the data acquired is similar and can be exposed to similar analysis methods. Therefore, MEG data analysis has inherited both terminology and methods from EEG data analysis. Except for the source modelling presented in the previous section, there are two main areas of interest: evoked responses and frequency analysis.

Usual post-processing procedures include noise removal by low-pass and/or high-pass filtering, signal space separation and more sophisticated techniques to remove artifacts arising from different sources (Haumann et al., 2016). Due to the spatial distribution of the sensors, brain activity can crudely be localized even without performing any source modelling.

An evoked response is the MEG signal during a time window following some kind of stimulus. The time for the presentation of the stimulus itself is recorded to provide zero point and the subsequent activation can later be analyzed channel by channel. The time series of the signal usually shows some characteristic features, like consistent peaks at certain times, depending on the nature of the stimulus. The different peaks can be associated with different neural processes (Coles and Rugg, 1996).

Frequency analysis involves observing the dynamics of the frequency content of the MEG signal by applying the Fourier transform to it. In the resting and working brain, oscillations at different frequencies are naturally present. Using MEG, the source of oscillations can be localized and connectivity between different regions of the cortex established. (Jenson et al., 2014)

Many aspects of the methods described above rely on a channel-by-channel approach, which means that the dimensionality of the data is reduced to only one dimension. As noted in the introduction, working with the full dimensionality of the data could lead to new insights. There are recent studies utilizing machine learning methods applied to the high-dimensional MEG data (see e.g., the work by Cichy and colleagues (2014)).

2.2 Representational similarity analysis

Representational similarity analysis (Kriegeskorte et al., 2008) is a framework taking advantage of patterns in the high-dimensional data obtained in neuroimaging. One notable strength of the framework is that comparisons can be made across individuals, species and even experimental methods. It also provides a natural way of evaluating and fitting models to the often limited amount of brain imaging data, something that can be arduous using other methods.

2.2.1 Background

Neuroimaging data is inherently high-dimensional. In the case of MEG, the number of dimensions is determined by the number of channels in the MEG device. However, the underlying activation in the brain requires a much higher dimensionality to be complete characterized. Generally, a stimulus gives rise to a certain spatiotemporal activation pattern in the brain. This pattern can be thought of as the *representation* of the stimuli by the brain and can momentarily be regarded as a point in a high-dimensional space (Kriegeskorte and Kievit, 2013). This “real” activation pattern could be envisioned as a point with the same number of dimensions as the number of neurons. The pattern we observe in imaging has a reduced dimensionality, but preserves some of the structure of the original pattern.

The representations of other stimuli correspond to other points in this space. The similarity (or dissimilarity) of the representations of two different stimuli can be measured by calculating the distance between the corresponding points. There are many distance functions that can be used, but we will focus on the correlation distance, since it ignores the actual amplitudes and determines similarity on the basis of the overall pattern. The correlation distance is defined as $1-r$, where r is Pearson’s correlation coefficient. In other words, a correlation distance of 0 indicates perfectly correlated vectors, 1 indicates no correlation and 2 indicates perfectly anti-correlated vectors. It can be shown (see Appendix A) that the correlation distance is equivalent to the squared Euclidean distance for normalized patterns, which will turn out to be a useful property (Khaligh-Razavi and Kriegeskorte, 2014).

The pair-wise distances can be calculated for all representations of stimuli we are interested in and summarized in a representational distance matrix (RDM). An RDM has two important properties. Firstly, since the distance metric used is commutative, the RDM is a symmetric matrix, and secondly, since the distance between a stimulus and itself is zero, the diagonal consists of zeroes. An example of an RDM and the underlying data it represents can be seen in Figure 5.

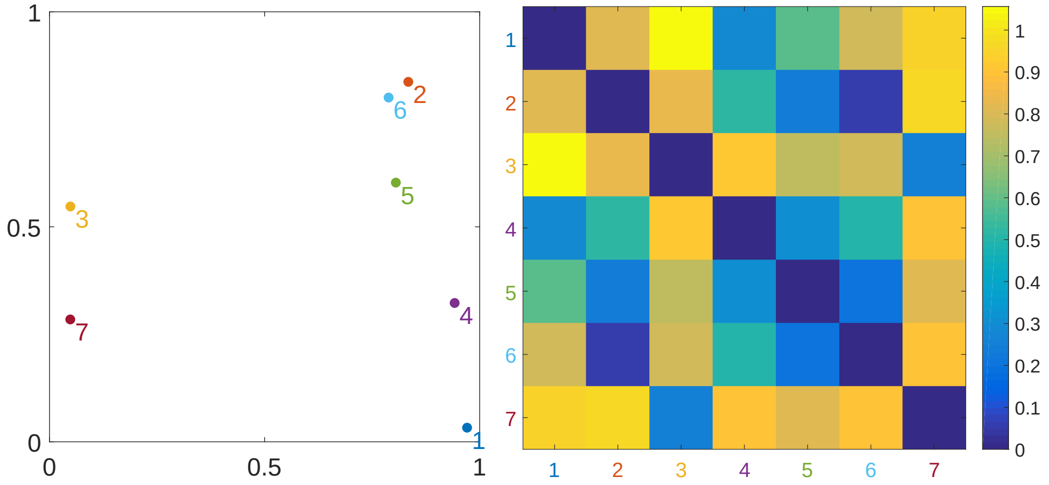

Figure 5: *Left:* Points in a two-dimensional space. *Right:* RDM calculated using the Euclidean distance for the points seen to the left. By visually inspecting rows of the RDM, one can determine that e.g., point 7 lies far away from all points, except for point 3. In this case, where points are only two-dimensional, this could as well be seen from the left image, but in a high-dimensional case, this kind of visualization becomes more useful.

The strength of RSA is that any type of measurement involving different stimuli can be summarized in an RDM. In this way, we can abstract away from the specifics of a method and avoid the question on how to define a mapping between units of different modalities. RDMs can also be calculated for models, which allows for a natural way to determine how well models explain the observed representations (see e.g., the work by Mur and colleagues (2013)) and to fit model parameters, no matter how the models are constructed. Another way of utilizing RDMs is by using multi-dimensional scaling. This allows the distance between RDMs to be visualized in two dimensions, which helps to create an intuitive picture of how similar different measurements and/or models are to each other.

RDMs are compared to each other using some kind of distance metric. Usually, the Spearman rank correlation coefficient is used, since it does not require the relationship between the RDMs to be linear to yield a good result. For cases where there are many tied ranks, Kendall's τ_a is recommended (Nili et al., 2014). We will shortly look at the technical details of both coefficients.

An overview of the whole procedure as applied to MEG data is shown in Figure 6. This example shows using the signal from two different sensors (shown in red on the flattened MEG helmet) when determining the RDM. The data from all channels used are concatenated into a long vector, and the distances (usually correlation distances as mentioned before) between these concatenated vectors arising from different stimuli are put in the RDM. A model RDM is also representing the dissimilarity between all pairs of stimuli can be correlated against the data RDM to arrive at a measure of correspondence. More advanced types of analysis, like fitting parameters for a computational model or combining multiple models (more about this in Section 2.2.4), can also be performed.

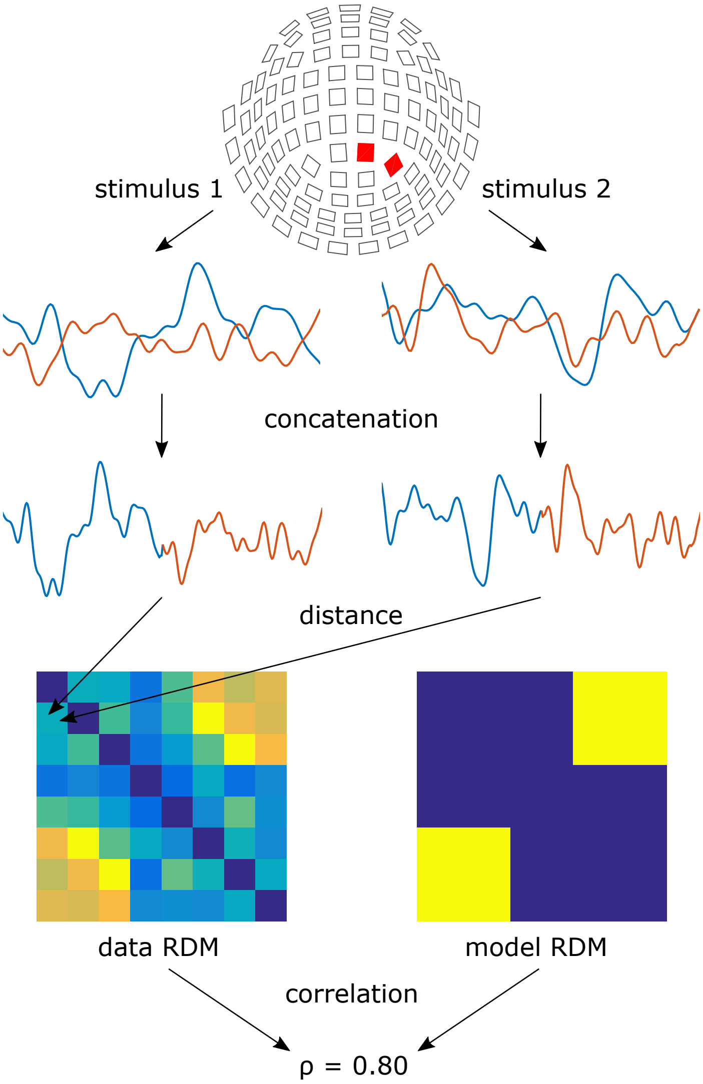

Figure 6: An illustration of the application of RSA to MEG data on the channel level. See the text for more details.

2.2.2 Correlation coefficients

Spearman's rank correlation coefficient is essentially Pearson's correlation coefficient applied to the ranks of the data. Determining the rank involves sorting, which results in a complexity of $O(N \log N)$ for the whole calculation. If there are repeated values in the data, ranks become tied, which adds an element of arbitrariness in the sense that tied ranks can be permuted. This problem is solved by using fractional ranks (Hays, 1973). For example, in the case of ranking the vector [3 3 2 4], we first perform ranking normally, which could give [2 3 1 4] or [3 2 1 4] depending on the sorting algorithm used. For repeated values, the average rank is used instead, resulting in [2.5 2.5 1 4]. This is equivalent to averaging over all possible permutations of ranks (see Appendix D).

To compute Kendall's τ_a, pairs (x_a, y_b) and (x_c, y_d) of points from both datasets are compared to each other (Nelsen, 2011). Two pairs are said to be *concordant* if $x_a > x_c$ and $y_b > y_d$ (increasing function) and *discordant* if $x_a > x_c$ and $y_b < y_d$ (decreasing function). Denote the number of concordant pairs by c and the number of discordant pairs by d. Kendall's τ_a is defined as

$$\tau_a = \frac{c-d}{\binom{N}{2}} = \frac{c-d}{n(n-1)/2},$$

where n is the number of points. The value in the numerator is the total number of pairs. Here, it can be noted that if all pairs are concordant, we have that $\tau_a = 1$, and if all pairs are discordant, $\tau_a = -1$, in complete agreement with Pearson's and Spearman's correlation coefficients. Due to the forming of pairs, the complexity of calculating Kendall's τ_a is $O(N^2)$.

2.2.3 Spatiotemporal searchlight

Brain activity related to a specific stimulus is often localized. Therefore, we are not interested in including signal from locations irrelevant to the task at hand in our analysis. Also, using all available data for determining the RDM dismisses spatial information. Instead, we can utilize a searchlight. This procedure was introduced in the work by Kriegeskorte and colleagues (2006), where a type of multivariate effect statistic was used instead of RDMs. For the fMRI data used, a spherical searchlight with a radius of 4 mm seemed to be optimal. In the case of RDMs, the distances between representations are calculated using a subset of all dimensions. In the case of fMRI, this refers to a subset of voxels and in MEG a subset of either sensors or sources.

Su and colleagues (2012) applied a similar searchlight to EMEG data (referring to both EEG and MEG), adding a temporal aspect, called *spatiotemporal searchlight* (ssRSA). They proposed performing source modelling and applying the searchlight in the source space. Spatially, hexagonal cortical patches with a radius of 20 mm were used, and temporally, overlapping time windows with a length 20 ms and step of 5 ms. Further experiments employing this method for tonotopic mapping are described in Su and colleagues (2014) and are described more in detail in Section 2.2.7.

2.2.4 Multiple models

In the original version of RSA, RDMs are compared pair-wise. Models used for brain activity data are however usually too simple by themselves to explain all of the variance in the data. Combining multiple models might therefore lead to improved results. To do this, we express the target RDM as a weighted sum of model RDMs. There are however some theoretical considerations that must be taken into account when doing this.

The weighting of features should be done in the original space, and not on the RDM level (Khaligh-Razavi and Kriegeskorte, 2014). This is due to the fact that different distance metrics will bend the space in different ways, which cannot be accounted for in the linear combination. However, using the squared Euclidean distance when calculating the RDMs abolishes this problem. This is not a serious limitation, since the correlation distance and the squared Euclidean distance are equivalent when using normalized patterns (see Appendix A).

The squared Euclidean distance has two useful properties. First, as long as the features can be assumed to be orthogonal to each other, adding a dimension corresponds to adding a term. Assume that $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and that $\hat{\mathbf{x}} = [\mathbf{x}\ x_{n+1}]^T \in \mathbb{R}^{n+1}$ and $\hat{\mathbf{y}} = [\mathbf{y}\ y_{n+1}]^T \in \mathbb{R}^{n+1}$. Now we have that

$$d^2_{\text{Euc}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = d^2_{\text{Euc}}(\mathbf{x}, \mathbf{y}) + (x_{n+1} - y_{n+1})^2.$$

Note that the Euclidean distance itself does not have this property because of the square root.

The second useful property is that it does not matter if weighting is done in the original space or on the distance level. Say that the extra dimension introduced into $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ is weighted by $w \in \mathbb{R}$. We then get

$$d^2_{\text{Euc}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = d^2_{\text{Euc}}(\mathbf{x}, \mathbf{y}) + (wx_{n+1} - wy_{n+1})^2 = d^2_{\text{Euc}}(\mathbf{x}, \mathbf{y}) + w^2(x_{n+1} - y_{n+1})^2.$$

In other words, weighting at the level of RDMs is exactly the same as weighting at the level of features. We can see that distance-wise, the added dimension is weighted by w^2. This implies that fitting must be performed using a non-negative linear least squares method.

Fitting multiple models is especially valuable when working with MEG data on the channel level, i.e. when no source modelling is made. Contrasting the situation to fMRI data, where the dimensions easily can be spatially separated to account for different anatomically defined brain areas, illuminates the problem: the signal picked up by one MEG sensor is spatially limited only by the position of the sensor in relation to the head of the subject. In other words, one MEG channel contains a mix of signal from areas close to the channel. Combining models when analyzing MEG data is a natural way of accounting for this diverse activity, while separate models straightforwardly can be tested for separate brain areas using fMRI.

2.2.5 Cross-validation

The process of fitting a model to data is usually mechanical. However, the resulting fit must also be evaluated in some way, since there are some issues that can arise (Alpaydin, 2014). The model used might be underfitting the data, which means that it does not have the power to accurately explain the underlying phenomenon generating the data. This is a problem with the model that can only be fixed by making necessary corrections to it. If the model is powerful enough (or too powerful) to explain the data, we may on the other hand face the risk of overfitting. This means that the fit looks good for our dataset, but the generalization performance is poor.

In machine learning, when e.g., training a classifier, this problem is usually solved by artificially dividing the original data into two different sets, a training set and a test set. The training set is used to train the classifier, while the performance of the classifier is evaluated on the test set. During learning, performance on the training set continues to increase, but at some point, performance on the test set will start to decrease. After this point, overfitting occurs. Using some type of regularization can remedy this problem.

Cross-validation (CV) refers to methods systematically and repeatedly dividing the available data into training and test sets. One example is k-fold CV. The data is divided into k folds of equal size and $k-1$ of the folds are used for fitting, while the performance is tested on the left-out fold. This procedure is repeated leaving out each fold in turn and the average performance over folds is determined.

Cross-validation can also be employed when fitting RDMs (Khaligh-Razavi and Kriegeskorte, 2014), and pseudocode for the algorithm is presented as Algorithm 3 in Section 3.3.3. When fitting multiple model RDMs to a target RDM, a specified number of stimulus are left out when fitting. The fitted model is then evaluated for these left-out stimuli, and the resulting values are put in the CV-fitted RDM. This procedure is repeated until the whole CV-fitted RDM is filled. In this way, most of the entries in the fitted RDM will result from fits with different weights, so this is a procedure for determining a CV-fitted RDM, not the weights.

2.2.6 Application to fMRI data

RSA has successfully been applied to fMRI data (see Kriegeskorte and Kievit (2013) for a review). We will present a few more recent examples here of how it has been employed, focusing on using multiple models.

Khaligh-Razavi and Kriegeskorte (2014) used previously obtained RDMs for the inferior temporal (IT) cortex associated with object recognition both from monkeys and from humans and compared them to a wide range of model RDMs. The tested models included neurally realistic models, filters, more complex models used in computer vision and even the different layers of a deep convolutional network. Fitting with cross-validation was performed, showing that a combination of models explained the IT RDM in both the human and monkey case significantly better than individual models.

Jozwik and colleagues (2016) let subjects judge the category membership and absence or presence of features on a set of images. Separate RDMs were then created for each category and for each feature. All of these RDMs (total of 234) could then be fitted to the data RDM (human IT, EVC and similarity judgements) using non-negative linear least squares with cross-validation. The fitted RDMs had significant correlations with the data RDM in the case of IT and similarity judgements, but not high enough to be classified as the true underlying model.

2.2.7 Application to MEG data

As previously mentioned, there has not been many studies employing RSA on MEG data. One previously mentioned example is the work by Su and colleagues (2014) where tonotopic mapping was examined using ssRSA. Here, subjects listened to spoken words in English while having their EEG and MEG measured. Source modelling was performed, and ssRSA was employed to hexagonal cortical patches of a radius of 20 mm and a sliding time window with length 30 ms and step 10 ms. For the modelling part, the words were filtered using a Gammatone filter bank, to produce a representation better corresponding to the cochlear representation than the original sound. The hearing range was divided into 16 frequency bands and RDMs were calculated for each band and for each time window based on the average power of the stimulus. These 16 RDMs per time window were then fitted to the corresponding data RDM using a general linear model. Each weight corresponds to the contribution of a certain frequency band, and so, a Gaussian was fit to the weights. This allowed determination of the center frequency and selectivity (standard deviation) for each searchlight.

Wingfield and colleagues (2016) used a similar kind of analysis as described above, comparing representations in speech recognition performed by humans and machines. A general linear model was used fitting phonetic model RDMs to the data RDMs. In this way, the activation for different groups of phonemes could be localized in the brain.

Tyler and colleagues (2013) studied the flow of information in listening to locally syntactically ambiguous sentences, which are characterized by that the meaning of the sentence becomes fully clear at some point. They used functionally defined regions of interest in the source space for calculating RDMs, a very similar procedure to how RSA generally is used in fMRI studies. Every time point included data from a 50 ms window. The resulting RDMs were correlated against models and time series of correlations were examined in order to draw conclusions about the flow of information.

Wardle and colleagues (2016) showed circular images with differently oriented elements to subjects. The data RDMs were calculated separately for each time point using the total MEG signal. Time-series of correlations between the data RDM and various model RDMs (simple feature models and ratings) were determined. This study also employed linear SVMs to determine the decodability of pairs of stimuli. Decodability of the MEG signal is also done in the works by Cichy and colleagues (2014), by Carlson and colleagues (2013), by Redcay and colleagues (2015) and by Peters and colleagues (2016). Note that the analysis in these studies is done in sensor space and no inverse modelling is conducted.

Ramkumar and colleagues (2014) combined the spatiotemporal searchlight with decoding in studying scene perception, but did not use RSA per se. They showed grayscale natural images to the subjects while recording MEG. Source modelling was performed and the parameters for ssRSA were spatial neighborhoods consisting of 25 sources each and time windows of 20 ms. For each searchlight, linear SVMs were using for decoding. Cross-validation was employed in determining the classifier accuracy.

Even though some studies previously have used the RSA approach for MEG data at the sensor level, and even fewer employed a spatiotemporal searchlight in source space, a fusion of these methods into a spatiotemporal searchlight in sensor space has to our knowledge previously only been used in the work by Henriksson and colleagues (2016).

3 Methods

Most of the simulations and analysis were run on a desktop computer in MATLAB R2016a (The MathWorks, Inc.) on CentOS 6 using the MNE software package (Gramfort et al., 2014) for forward and inverse modelling. For some more computation intensive simulations (especially those containing CV), a distributed computing cluster was used. The actual parameters used for individual simulations are presented in connection with the corresponding results in Chapter 4.

3.1 Overview of simulations

A significant part of this work consisted of validating the RSA approach for MEG data at the sensor level. While the validation was performed using both simulated and measured data, more emphasis was put on the simulations, since they allowed maximum control over the experimental situation. In addition to validation of the main goal of this thesis, fitting multiple models, more basic properties like correlation dependence on activation region depth and model RDM to signal RDM correlations were also examined. This is done to both validate the approach used in the work by Henriksson and colleagues (2016) and further advance the understanding and possibilities of using a spatiotemporal searchlight in sensor space.

The basis for the simulations were the forward and inverse models presented in Section 2.1.4. While the forward model is more of theoretical interest when analyzing data, it has a prominent role in simulations. Usage of the forward model allowed the exact location, extent and type of activation in the brain to be preset and the corresponding sensor patterns to be generated. The activations in the simulations were always located in the cerebral cortex and quite often in the occipital lobe, where the sensors of the MEG helmet are very close to the brain.

The simulations were based on data from 10 subjects. This data included the forward and inverse operators for the MEG data, a noise covariance matrix and a distribution of sources based on previously acquired MR images. The resulting data were always averaged over subjects and single subject data were never explicitly examined, as this also is the usual approach in experiments of this kind.

The work flow of simulations can be divided into two main parts: *data generation* and *correlation and fitting*. The substeps of these parts will be explained in the following sections. A general overview is also shown in Figure 7, where the major steps in the simulation process are visualized. An RDM can be determined for every step and for example have a model (not necessarily the generating model) evaluated against it or a set of models fitted to it.

3.2 Data generation

The data generation part of most simulations are more or less variations on the same basic pattern:

1. Selection of regions (sources) of the cortex to be activated.

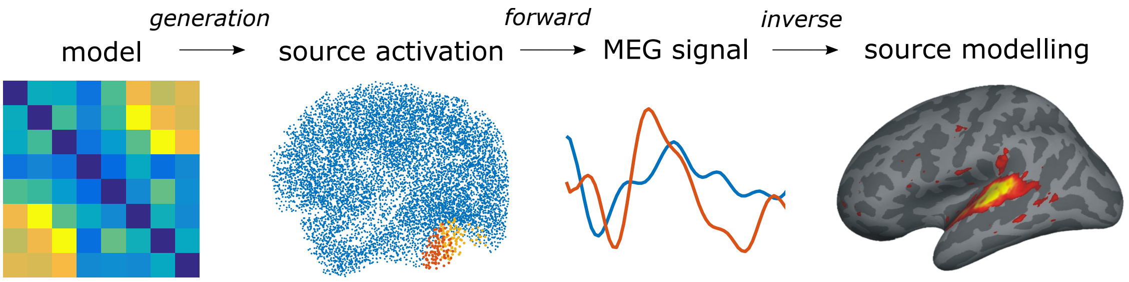

Figure 7: The data generation part of the simulation process. An RDM can be determined for every step depending on the purpose of the specific simulation. Part of figure from the work by Woods and colleagues (2010).

2. Selection of activation model(s).
3. Simulation of source activation according to the activation model(s).
4. Generation of MEG signal data using the forward model.
5. Modelling of source activation using the inverse model.

3.2.1 Definition of activation regions

MR images had previously been acquired for the subjects used for the simulations. Based on these, the cortices had been triangularized with a mean of 294000 patches (SD = 23800) per subject. The distribution of sources used for modelling was sparser than this and the actual number of sources was on average 11500 (SD = 911) per subject.

Since the simulations were based on real anatomical data from subjects, all source distributions were different. To more precisely make sure that an activation occurred in the same location relative to the MEG helmet for all subjects, the source distributions were all registered to the source distribution of a particular subject. This was done using the iterative closest point (ICP) algorithm, presented as Algorithm 1. We used an affine transformation, i.e. a translation vector $t \in \mathbb{R}^3$ and an arbitrary transformation matrix $R \in \mathbb{R}^{3\times 3}$. Although registration of image data usually is confined to rotation and translation, scaling and shearing was also allowed here to fulfil the above mentioned goal. It proved necessary to employ all these degrees of freedom here, since the cortices of the subjects were of different shapes and sizes. Registration was done by

$$X_{\text{reg}} = RX + t,$$

where $X, X_{\text{reg}} \in \mathbb{R}^{3\times n}$ are the points to be registered and the registered points respectively and t is replicated n times in the row direction. To be able to determine the optimal parameters using linear least squares, we concatenated R and t into

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

and add a row of ones to X. In Algorithm 1, the unit transformation on row 2 stands for the transformation not affecting the points at all, i.e.

$$T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The notation TX on row 5 refers to matrix multiplication and is the set of points transformed by T. Finally, on row 9 the optimal 12 parameters for T that minimize the total Euclidean distance between the points in X and the corresponding closest points in *points* are determined. The algorithm eventually converges on a set of closest points and the distance is unchanged after this.

Algorithm 1 Iterative closest point

```
Require: a set of points $X$ to be registered and a set of points $Y$ to register to
 1: procedure ICP
 2:     $T \leftarrow T_0$                                   ▷ Unit transformation
 3:     repeat
 4:         $dist \leftarrow 0$
 5:         for $x$ in $TX$ do
 6:             $y \leftarrow \text{closestpoint}(Y, x)$
 7:             $dist \leftarrow dist + \text{distance}(x, y)$
 8:             $points_x \leftarrow y$
 9:         $T \leftarrow \text{leastsquares}(X, points)$
10:     until $dist$ is unchanged
```

The region of activation was determined in one of two different ways: by using previously anatomically labelled regions (only for areas located in the visual cortex) or by systematically selecting sources according to some rule. One goal of the systematic selection was to allow selection of small regions throughout the cortex without having to rely on any labelling. By doing this, the number of activated sources could be controlled for and the effects of depth on the MEG signal determined. In some simulations, two regions were activated at the same time, both exhibiting different types of activity.

The registered sources were used for defining activation regions. One early definition method was based on generating a cubic grid and selecting a cubic or spherical volume of sources within a certain distance of every grid point. This method however proved to provide quite unstable results, since the number of sources in a region could vary much between subjects, both due to the inevitable registration error and the different source distributions of the subjects. Differences also occurred between regions of the cortex due to variation in the overall density of the sources. To remedy these problems, a fixed number of the sources closest to a grid point was instead used to define an activation region. An example of what a region like this might look like is shown in Figure 8.

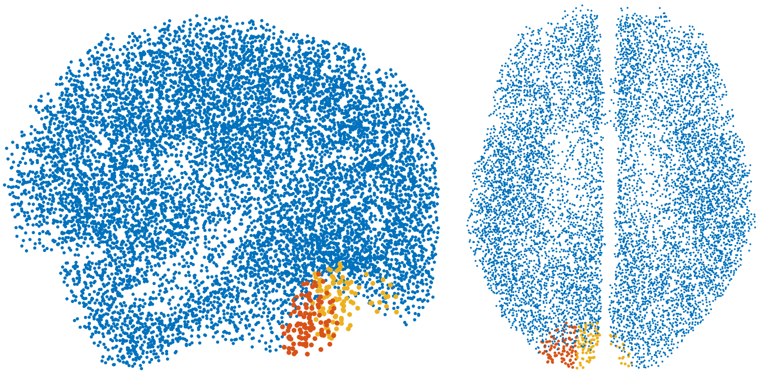

Figure 8: Example of two activation regions (yellow and red) determined by the closest sources to a given point. In this case, the region has originally been defined as one continuous region and then split into two along the coronal plane. This also illustrates how systematic region selection can result in unwanted gaps in the activation region; here the activation "jumps" to the other hemisphere. Every point in the image is a source. *Left:* View from behind and left. *Right:* View from above, the two hemispheres clearly visible.

The depth of an activation region was calculated in a way very similar to in which the distance is calculated in Algorithm 1. Rows 5-7 of the algorithm were used (with $T = T_0$) between the sets of source locations and sensor locations, after which the resulting distance was divided by the number of sources in the activation region to arrive at a mean distance between the sources and the sensors.

One notable weakness of the automated methods is that they ignore neurophysiological realities. Sources are included in the activation region only based on their locations without taking their orientation in regard in any way. This might lead to naturally improbable results, where the activation can jump over sulci, over functionally defined regions or even to the other hemisphere (see Figure 8). Ignoring orientation in region selection also leads to sources cancelling each other out in varying degrees, introducing an artificial source of variation in the results. Relying on anatomically or functionally labelled regions could have remedied problems like these, but would not have been able to provide as systematic a division.

3.2.2 Source activation

In MEG studies on human subjects, the underlying source activation is always unknown, but of great interest and therefore estimated from the signal using different methods (see Section 2.1.4). In the simulations, the activation was generated based on a model RDM, which will be explained more in details a bit later. Specifically, the amplitude of a source is set to a constant plus a vector pseudorandomly generated using a multivariate normal distribution with zero mean and covariance 1−RDM, i.e. a high dissimilarity between stimuli results in a low covariance between the corresponding activations. In this procedure, every source receives a different amplitude for every stimulus, but the overall pattern of all the activated sources is what is of interest. The RDM of the activated sources (using the correlation distance) will be approximately equal to the original model RDM (in the limit of an infinite number of sources, equality holds) when using this pseudorandom technique. The randomness inherent in this procedure introduces variance at the source level, but results between runs were found to be consistent to the degree that this variance could be ignored. Notice that the covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ must be a positive semi-definite matrix. This means that the inequality

$$\mathbf{x}^T \Sigma \mathbf{x} \geq 0$$

must apply for all $\mathbf{x} \in \mathbb{R}^n$. An equivalent condition is that all eigenvalues of Σ are positive.

A number of classes of model RDMs were used for generating the activation. One particularly simple one and highly structured is the *categorical RDM*. Stimuli are grouped into a number of categories with a fixed number of stimuli per category. Stimuli belonging to the same category produce exactly the same source activation, while stimuli belonging to different categories produce maximally dissimilar source activation. There can also be some variations on this, where e.g., some stimuli do not belong to any category. Examples of the categorical RDMs used in this work (although in smaller versions) can be seen in Figure 9. The block models were usually used together in distinguishability tests, since they are structurally similar.

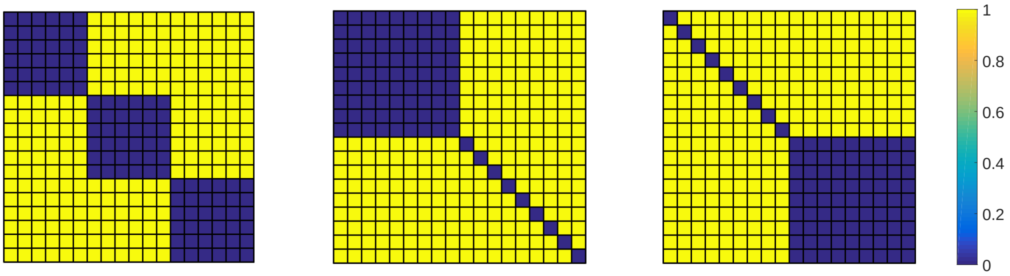

Figure 9: Examples of categorical RDMs. These three examples are the ones most often used in the simulations. The grid is added to allow for better visualization of the actual rows and columns of the RDM. *Left:* Categorical model with 3 categories and 5 stimuli per category. *Middle:* Block model, referred to as `[0 1; 1 1]`. *Right:* Block model, referred to as `[1 1; 1 0]`.

In contrast to categorical RDMs, a random RDM contains little or no structure. Due to the constraints imposed on an RDM (symmetric and diagonal zero), a completely random matrix cannot be used. In addition, to use the RDM as a covariance matrix, 1−RDM must be positive semi-definite, which in itself implies symmetry. In this work, three different methods of producing random RDMs were used. The first two methods are very similar and are described in detail in Algorithm 2.

Algorithm 2 Random positive semi-definite RDM generation

```
Require: The size $n$ of the RDM.
 1: procedure RANDRDM
 2:     $rdm \leftarrow$ random matrix of size $n \times n$, elements uniformly drawn from $[0, 1]$
 3:     execute row 4 or 5
 4:     $rdm \leftarrow rdm$ + transpose($rdm$)                          ▷ Method 1
 5:     $rdm \leftarrow rdm \times$ transpose($rdm$)                     ▷ Method 2
 6:     diagonal($rdm$) $\leftarrow 1$
 7:     $\lambda \leftarrow$ minimum eigenvalue of $rdm$
 8:     if $\lambda < 0$ then
 9:         diagonal($rdm$) $\leftarrow$ diagonal($rdm$) - $\lambda$     ▷ Make eigenvalues positive.
10:         $rdm \leftarrow rdm/rdm[1, 1]$                              ▷ Normalize diagonal to 1.
```

Note that Algorithm 2 contains a choice between two different generation methods to be made on row 3. Row 5 of the algorithm actually produces a positive semi-definite matrix (since $\mathbf{x}^T \Sigma^T \Sigma \mathbf{x} = ||\Sigma \mathbf{x}||_2^2 \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$), but setting the diagonal elements to 1 breaks this property. This can be remedied by making the eigenvalues positive (row 9).

Another method of producing random positive semi-definite RDMs is by random sampling from the Wishart distribution, which can be done using the function `wishrnd` in MATLAB. Examples of typical RDMs arising from the different generation methods are shown in Figure 10.

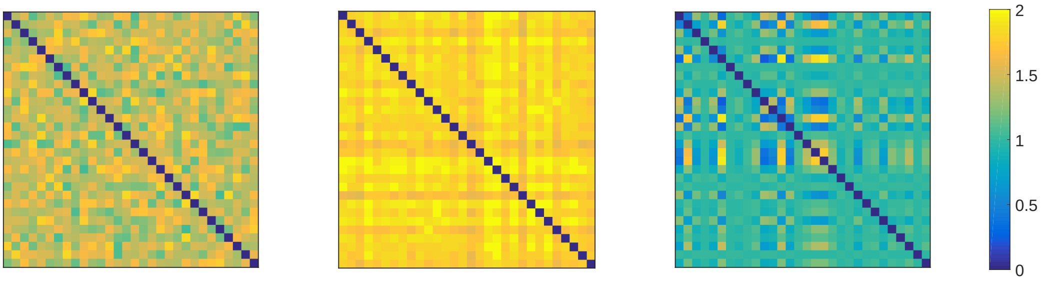

Figure 10: Typical examples of random RDMs used as models in some simulations. From left to right: RDM generated using row 4 in Algorithm 2, RDM generated using row 5 in the same algorithm and RDM sampled from the Wishart distribution with unit covariance matrix and *df* = 1. Notice that the level of perceivable structure differs between generation methods.

To be able to test the performance of the fitting of multiple models, combinations of models were also used for generating source activations. A linear combination of two models was the one mostly used,

$$M = w_1 M_1 + w_2 M_2 \; (+ c), \tag{4}$$

where M_1, M_2 are two different models called *component models*, w_1, w_2 are the weights applied to them and M is the resulting model. Without loss of generality, it was assumed that $w_1 + w_2 = 1$. The simulations were usually run for $w_1 = 0 : 0.05 : 1$.

3.2.3 Application of forward model and inverse model

The application of the forward model is straightforward according to Equation 1 in Section 2.1.4. We explained how to obtain the source activation $\mathbf{x}$ in the previous section. The gain matrix A had previously been determined for all subjects. The noise covariance matrix C_n was also known, meaning that the noise $\mathbf{n}$ could be simulated by sampling from a multivariate normal distribution with C_n as the covariance matrix. The signal to noise ratio could also be varied by weighting $\mathbf{n}$ appropriately, but for most simulations, it was set at $\mathrm{SNR} = 3$.

Modelling the source activation was done using the MNE solution presented in Equation 2 in Section 2.1.4. The inverse operator required the parameter λ, which was set to SNR^{-2}.

3.3 RDMs and fitting

The second part of the simulation relates to examining the results using RDMs. The steps used in this part depends very much on the phenomenon being analyzed.

3.3.1 RDMs

RDMs can be calculated for the source activation, the MEG signal and for the inversely modelled source activation. These RDMs can then be compared to each other and to the activation model(s).

In early testing, the signal RDM was compared to the activated source RDM instead of the activation model RDM. This was to combat effects of the randomness in source activation generation. However, the relationship between these correlations was always monotonic and therefore we focused on the correlation between the signal RDM and the activation model RDM, since one neurophysiologically also could expect noise in the EPSPs while there is an underlying “true” model these EPSPs are based on.

Magnetometer data was ignored in all experiments performed in this thesis, due to the fact that they are more prone to noise. The signal RDM can be calculated for the total MEG signal, using all channels. To perform localization, we instead used the signal from *channel neighborhoods*, which corresponds to the spatial aspect of the spatiotemporal searchlight. For each channel location, all channels within a certain radius are selected to form a channel neighborhood. This is done since one single channel rarely contains enough information to actually get sensible RDMs. Therefore, what is referred to as the signal RDM actually consists of 102 separate RDMs, one for each channel. All of these RDMs are then one by one correlated to the target RDM to create a map of correlations. For simplicity and where the spatial pattern of the correlations is not important, maximum and mean correlations over channel neighborhoods are usually reported.

As a technical detail, the symmetry of the RDMs allows for using less memory by representing them as vectors. In addition, since the distance between a stimulus and itself always is 0, the diagonal does not need to be represented. A $n \times n$ RDM can be converted into a vector of length

$$s = (n-1) + (n-2) + \ldots + 1 = \frac{n(n-1)}{2} \tag{5}$$

by using the procedure seen in Figure 11. To convert a vector back to a matrix, the size of the matrix can be determined from the length of the vector by solving Equation 5 for n, which gives us

$$\begin{aligned} n &= \frac{-\left(-\frac{1}{2}\right) \pm \sqrt{\left(-\frac{1}{2}\right)^2 - 4 \cdot \frac{1}{2} \cdot (-s)}}{2 \cdot \frac{1}{2}} \\ &= \frac{1}{2} \underset{(-)}{+} \sqrt{\frac{1}{4} + 2s} \\ &= \frac{1 + \sqrt{1 + 8s}}{2}. \end{aligned}$$

A method for converting matrix indices to vector indices is presented in Appendix B.

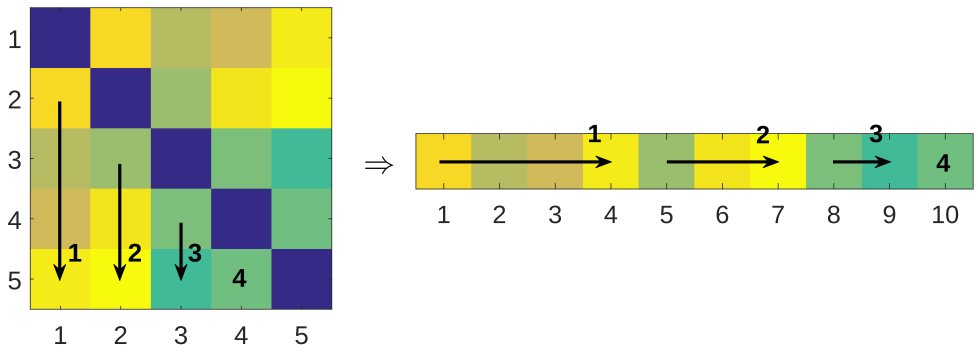

Figure 11: Representing an RDM as a vector. *Left:* RDM represented as a matrix. The arrows denote the order in which elements are stored in the vector version. *Right:* RDM represented as a vector. This representation cuts the memory requirements in half.

3.3.2 Fitting multiple models

As explained in the previous section, fitting multiple models to the signal RDM actually consists of making 102 separate fits, one per channel neighborhood. Fitting is based on Equation 4 in Section 3.2.2 with the constant included in some cases. As mentioned in Section 2.2.4, the weights of the linear combination must be positive. Therefore, the function `lsqnonneg` in MATLAB was used for fitting. The function uses an algorithm presented in Lawson and Hanson (1974).

The usage of linear least squares fitting with forced positive coefficients is not always problem-free. In cases where the component models are not suitable for explaining the signal RDM, the optimal weights might be negative. This might force `lsqnonneg` to return $w_1 = w_2 = 0$ and c as the mean of the RDM. This especially causes problems in determining the cross-validated RDM, as this results in the algorithm performing mean interpolation (see Appendix F).

It is notable that linear least squares fitting is based on minimizing the squared Euclidean distance between the linear combination of component model RDMs and the target model RDM. This procedure does not guarantee that the correlation between the fitted RDM and the data will be the maximum possible. For example, there is a possibility that there is a different linear combination that would have a higher correlation with the data RDM or even that one of the component RDMs has a higher correlation with the data RDM than the fitted RDM has.

To evaluate the performance of the fit, the ratio between the fitted weights was compared to the ratio between the original weights. A direct comparison was not possible, since the fitted weights (corresponding to the signal RDM) might be of a completely different order of magnitude then the original ones (corresponding to the activation model RDM). Nevertheless, for the weights to contain any information, we expected that their ratio should be preserved. Note that the extreme cases $w_1 = 1$ and $w_2 = 0$ or vice versa cannot be studied this way, since they result in an infinite or zero ratio. The fitted weight ratios were calculated for all $w_1 = 0.05 : 0.05 : 0.95$ and correlated to the original weight ratios using Pearson's r. Also, a mean squared logarithmic deviation (MSD) from a one-to-one correspondence between original and fitted weight ratios was used as a performance measure.

3.3.3 Cross-validation

The cross-validation procedure was implemented separately and is included as Algorithm 3. The actual implementation in MATLAB (see Appendix B) uses vectors instead of matrices to increase speed and lower memory requirements. This requires some indexing tricks to easily be able to leave out stimuli from fitting. The implemented version also checks whether the choice of stimuli to be left out actually allows for the evaluation of any previously undetermined elements of the CV-fitted RDM, and select a new leave-out set if not.

As can be seen in rows 6 and 7 in Algorithm 3, the CV-fit is also based on a non-negative linear least squares algorithm. When fitting (row 6, see Equation 4 in Section 3.2.1), the constant is included. However, when evaluating the fit (row 7), one question examined was whether to include the constant or not.

While a standard linear least squares fit naturally returns weights, this implementation of a CV-fit does not. The actual weights used for determining values of the CV-fitted RDM are calculated at row 6 and depend on which stimuli has been left out. To be able to return weights by this procedure, the weights used for the determination of each value of the CV-fitted RDM were saved and the average (over RDM positions) of these weights was returned. Also note that elements of the CV-fitted RDM might be evaluated several times and overwritten. The last value (and its corresponding weights) written is always the one that is kept.

Algorithm 3 Cross-validated RDM fit

```
Require: The number of stimuli to leave out $k$, the target RDM $target$ and a
    collection of model RDMs $models$.
procedure CVFIT
    $n \leftarrow \text{size}(target)$
    while $fit$ has undefined elements do
        $leaveout \leftarrow k$ random integers from $\{1, \ldots, n\}$
        $keep \leftarrow$ elements in $\{1, \ldots, n\}$, but not in $leaveout$
        $fitpar \leftarrow \text{nonnegativelsq}(target[keep], models[keep])$
        $fit[leaveout] \leftarrow \text{evaluate}(models[leaveout], fitpar)$
```

3.4 Application to real data

The dataset used for validation and testing is described in more detail in the works by Ölander (2015) and by Henriksson and colleagues (2016). In short, MEG and eye-movements were measured simultaneously while the subjects viewed grayscale photographs. Since the subjects were instructed to freely fixate on the images, the MEG data is contaminated with artifacts from saccades. Therefore, only data from the time period before the first saccade has been used for each trial and analysis was only performed for 125 ms post-stimulus.

No MR images had been acquired of the subjects, so source modelling was not possible. The goal was however to apply the spatiotemporal searchlight at the channel level. For every channel location, a channel neighborhood of a certain radius was determined, and RDMs were calculated for each neighborhood. Only data from gradiometers were used. Temporally, the data was divided into four windows of a length of 25 ms. This procedure has been applied to the same dataset in the work by Henriksson and colleagues (2016), but that study only used single models. The models used in this thesis are also the same: one model based on low-level image features (Gabor-wavelet pyramid model), one related to the content of target of the first saccade, one related to the spatial length of the first saccade, one related to the distance between the fixation point and the first saccade and one for saccade scanpaths. To fulfil the theoretical assumptions of the fitting routine, the original distances used in the study were dismissed and correlation distances were used for RDMs where possible.

3.5 Workflow

The first simulations were performed to see how RSA enabled the study of basic properties of the forward and inverse models. After this, the performance of the different types of activation models and linear combinations of these was evaluated. This provided the necessary basis for testing fitting.

In fitting, both weights and correlations can be examined. Starting with the standard LSQ-fit, weights are examined, both for the total MEG signal and per channel neighborhood. Correlations are then examined. The same approach is used for the CV-fit. Once the properties of the CV-fit has been established, multiple activation regions with different activation models are tested. Finally, the CV-fit is applied to the real data.

4 Results

4.1 Correlation coefficients

As noted in Section 2.2.1, Kendall's τ_a is to be preferred over Spearman's rank correlation coefficient. Due to the coefficients belonging to different complexity classes, the calculation time varies greatly for large matrices. In the experiment using real data, the RDM was of size 199 × 199. On a laptop (Intel Core i5 520M, 2.4 GHz), the speed of calculation was 210 coefficients per second for Spearman's ρ and 0.70 coefficients per second for Kendall's τ_a. Applying the spatiotemporal searchlight (102 channel neighborhoods and 4 time windows), resulted in a speed of about 2 seconds per model when using Spearman's ρ and about 10 minutes for Kendall's τ_a. The significant difference in speed plus some additional simulations (see Appendix C) motivated us to employ Spearman's ρ in all simulations. Whenever dubious results were attained, the correlations were double-checked using Kendall's τ_a, but this never caused any qualitative changes in the results.

4.2 Validation of methods used in simulations

4.2.1 Activation region depth

As a first test that all procedures were working as expected, the effect of depth of the activated region was examined. Here, 150 activation regions (defined by a cubic grid with step 0.02 cm) each containing 50 sources were examined. The activation model was a categorical model with 3 categories and 10 stimuli per category. The activation region depth was compared to the maximum MEG signal and signal RDM to model RDM correlation. Results can be seen in Figure 12. The maximum signal refers to the maximum RMS value of the signal over gradiometer pairs, averaged across subjects and stimuli. Note that channel neighborhoods are not used here. For the correlation, channel neighborhoods were used, and the plotted correlation value is the maximum over channel neighborhoods.

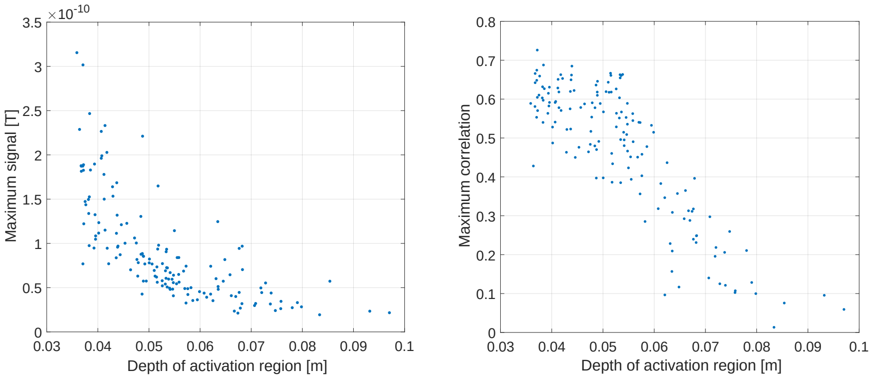

Figure 12: *Left:* The maximum MEG signal as a function of activation region depth, $\rho = -0.8407$. *Right:* The maximum correlation between the signal RDM and the model RDM as a function of activation region depth, $\rho = -0.7673$.

The spread of the data points in Figure 12 (left) can perhaps be explained by the automated region selection not taking source direction into account. We can see that deep activation regions always generate a weak signal, but the signal generated by shallow ones varies more. The right panel in the same figure shows that the relationship between depth and signal RDM to model RDM correlation is quite linear, indicating that a region might elicit a low maximum MEG signal, but still a significant correlation.

4.2.2 Forward-inverse modelling activation spread

To further evaluate the possibilities of using RSA, the activation spread in forward-inverse source modelling was examined. Again, a categorical model with 3 categories and 10 stimuli per category was used. A total of 334 activation regions (grid step 0.015 cm) of 50 sources each was used. This density resulted in overlaps between neighboring regions. In one run, a specific region was activated and inverse modelling was used to determine the source activation in all other regions. The modelled source activation in the non-active regions was correlated with the activation model. Results of this simulation are shown in the somewhat complex Figure 13.

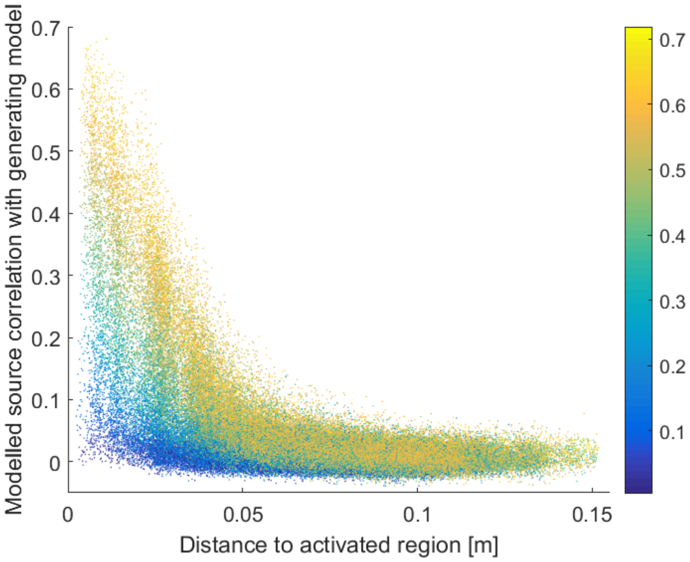

Figure 13: Spread of correlation in forward-inverse modelling. Each point corresponds to the correlation between the modelled source activation in a non-activated region and the activation model. The color denotes the correlation between the modelled source activation in the activated region and the activation model.

Focusing on the yellowish data points (corresponding to activation regions were the correlation between the region itself and the activation model was high) of Figure 13, we can see a boomerang-shaped pattern. The closer a region lies to the activated region, the higher the correlation of the modelled source activation in the non-activated region to the activation model is. Activated regions with a lower correlation (bluer colors) show a similar behavior, but with a slower rise when the distance decreases. There are at least two factors at play here: firstly, there are limits on how accurately inverse modelling can locate the activation, which becomes spread out, and secondly, the grid was so dense that some regions are overlapping.

4.2.3 Random model performance

Simulations were also conducted to evaluate the behavior of random activation models. Here, the activated region consisted of 150 sources in the occipital lobe. The simulation was run 20×20 times, meaning that 20 random models were generated using each method and the results for each random model were averaged over 20 runs. The categorical model used had 3 categories and 10 stimuli per category and the random models were of the same size, i.e. 30×30. Separate runs were done with and without channel noise. Channel neighborhoods were used and the correlation between the signal RDM (per neighborhood) and the source activation RDM was calculated. The results can be seen in table 1. The mean correlation included in part to show that the maximum correlation really stands out.

Table 1: Mean and maximum correlation (over channel neighborhoods) for noise/no noise conditions using random models generated by the two methods described in Algorithm 2 and a categorical model.

	No noise		Noise	
Model	Mean ρ	Max ρ	Mean ρ	Max ρ
Random (Method 1)	0.0788	0.1195	0.0184	0.0776
Random (Method 2)	0.2002	0.2951	0.0286	0.1502
Categorical	0.8107	0.8144	0.1204	0.5380

The random models do indeed perform differently as seen in table 1. The categorical model, which has a well-defined structure, is the easiest one to detect (highest correlation) of the tested models. The less structure a model has (models generated randomly using Method 2 at least visually seems to have more structure than those generated using Method 1, refer to Figure 10), the less the correlation between the activation model and the signal is. It can also be noted that the presence of noise makes the correlation more localized. In the case of the categorical model without channel noise, there was almost no difference between the average and the maximum correlation. With added noise, the difference was more than 4-fold.

4.3 Multiple models

4.3.1 Linear combination

Another preparatory validation simulation used a linear combination M of two component models M_1 and M_2 as the activation model and correlated the resulting signal to both the real activation model M and to the component models M_1 and

M_2. The activation region was 150 sources in the occipital lobe and the simulation was run 20 times for each weight-pair combination. Two different pairs of models were tested: the block models `[0 1; 1 1]` and `[1 1; 1 0]` of size 30 × 30, and a random (Method 1) model and a categorical model with 3 categories and 10 stimuli per category. Results are shown in Figure 14.

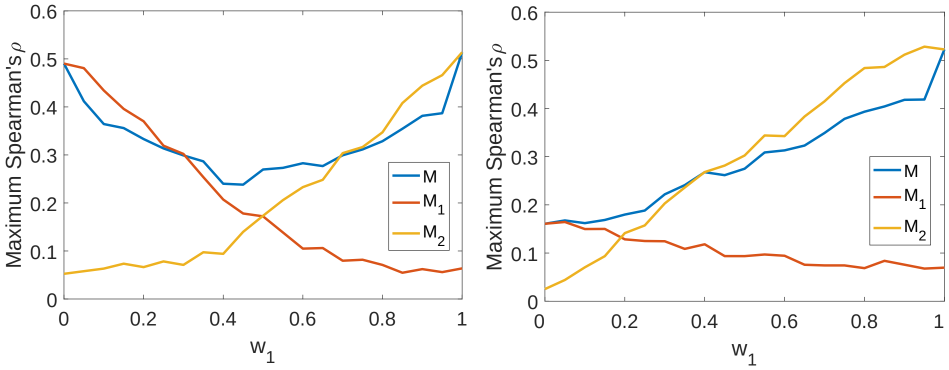

Figure 14: Maximum correlation (over neighborhoods) of the signal to the activation model M and its components M_1 and M_2. *Left:* Comparison of the block models `[0 1; 1 1]` (w_1) and `[1 1; 1 0]`. *Right:* Comparison of a categorical model (w_1) and a random model (Method 1).

Note that while the graphs for the block models that have a similar structure intersect at $w_1 = 0.5$, the graphs of the categorical and random model intersect already at $w_1 = 0.2$. This indicates that the categorical model is detected already at lower weights. One possible explanation for this is that the values of the models are not normalized in any way. An RDM generated randomly by Method 1 usually has values in the range of $[0.6, 1]$, while a categorical RDM only consists of the values 0 and 1. Since 1−RDM is used for the covariance matrix used for sampling the source activation, the categorical model contributes with either 0 or 1, while the random model contributes with a value in $[0, 0.4]$, clearly a lower influence. Figure 14 also illustrates that a component model actually can have a higher correlation to the signal than the real activation model.

4.3.2 Fitting using all channels

As a first step in evaluating fitting, the signal RDM was calculated using the total gradiometer signal instead using separate channel neighborhoods. The activation model was a combination of two models, a categorical one (w_1) with 3 categories and 10 stimuli per category and a random one (w_2, Method 1). 150 sources in the occipital lobe were activated and the simulation was run 100 times. In the conditions with includes noise, SNR was set to 3. All results are averages over runs and subjects and can be seen in Figure 15. Fitted weight ratios versus original weight ratios (left) as well as actual weights are shown (right). When channel noise is included, the ratio of the fitted weights is shifted away from the one-to-one line when w_1 is high, but not as much when w_1 is low. The shift was more symmetric when using a combination of models more similar to each other, e.g., two Wishart sampled models or the block models `[1 1; 1 0]` and `[0 1; 1 1]` (see Figure 19).

Figure 15 shows that the weights of the components model can be determined under optimal conditions with no noise. The dimensionality of the signal is much lower than that of the sources, but the information is still carried over by the forward model. It can of course be argued that since we're only using 150 sources, the dimensionality of the signal actually is higher than that of the sources. In practice, noise hinders us from obtaining fitted weight ratios this perfect. Fitting without a constant to noisy data (middle row) results in the fitted weight ratios not containing any sensible information. Adding a constant (bottom row) shifts the fitted weight ratio curve much closer to the one-to-one correspondence, but significant errors are still being made when the categorical model dominates.

4.3.3 Fitting using channel neighborhoods

The next step was to actually use channel neighborhoods and examine how the weights behave both in locations far away from and close to the point of activation. These simulations were run concurrently with the ones described in the previous section and have the same parameters. Although these kinds of results usually are visualized on the flattened MEG helmet, we have chosen a different approach of putting the indices of the channel on the horizontal axis in a standard scatter plot. This both makes the visualization of all weight ratios possible in one figure and also better indicates the deviations from the original weight ratio. Refer to Figure 4 for the location of the channels based on channel index. The results can be seen in Figure 16. The process was repeated using two random models from the Wishart distribution as component models with similar results (not shown).

The pattern visible in Figure 15 can also be seen here when focusing on the high-performing channel neighborhoods. In the absence of noise, channels in the index range 50-97 perform almost optimally. With noise present and focusing on channels 71-82, we can see that the correspondence between original and fitted weights is very good when $0.2 < w_1 < 0.5$, but that performance degrades with higher w_1. We also evaluated the fitted weight ratios of the neighborhood of channel 82, the best-performing channel in this case, and found that $R = 0.9924$ and $\mathrm{MSD} = 0.0943$, a notably better result than when calculating the RDM based on all channels (see the lower left panel in Figure 15). Qualitatively, the resulting plot looked similar as the one using all channels (not shown).

As mentioned in Section 3.3, the weights for the fit might be set to zero in some cases. For the data shown in the lower panel of Figure 16, this occurred for on average 16% of channel neighborhoods, while the corresponding value when using component models from the Wishart distribution was 23%. This phenomenon usually occurs in channel neighborhoods registering mostly noise, located far from the activated source region. Ignoring the zero values when plotting the average ratios does not result in visible differences.

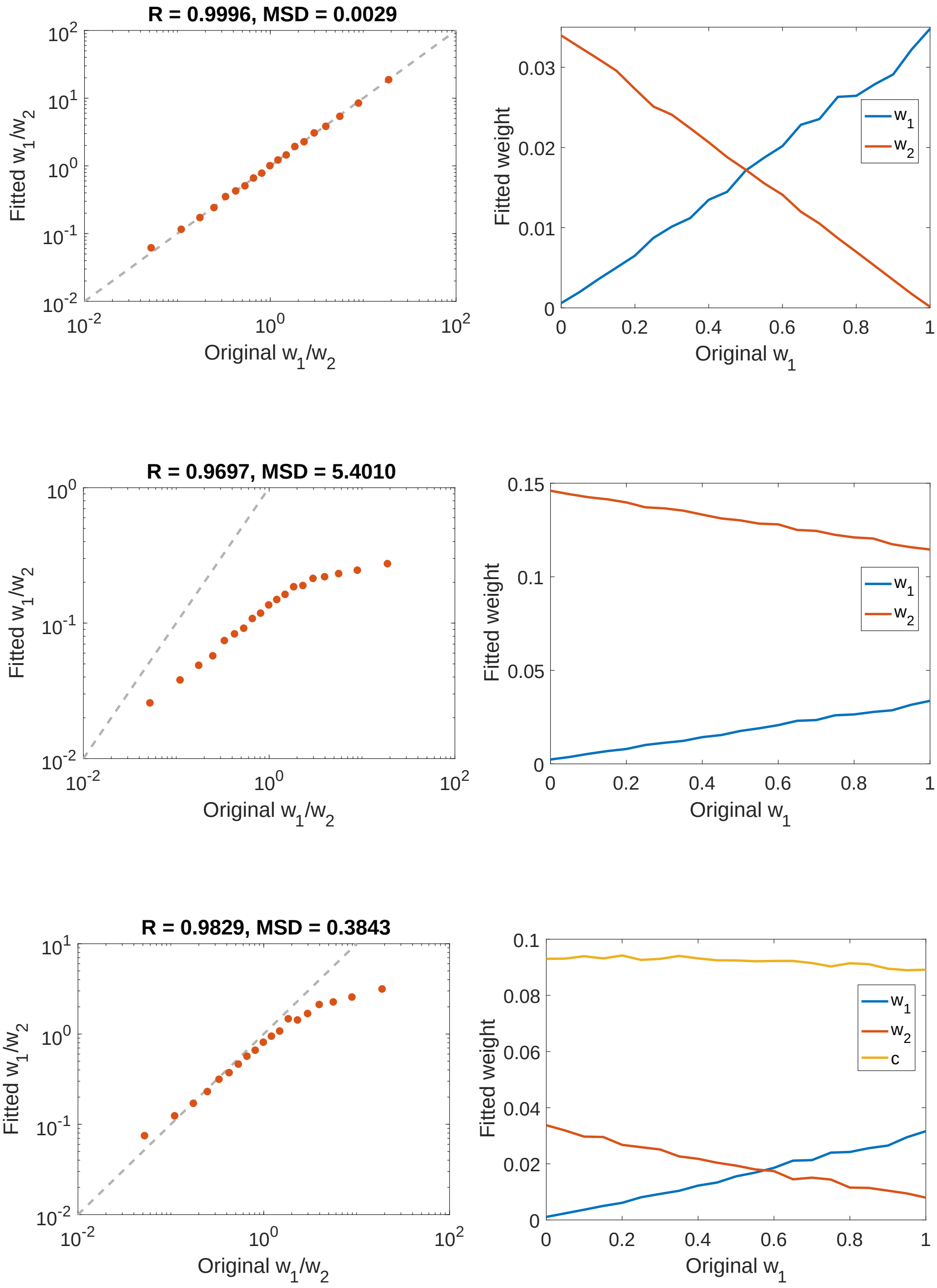

Figure 15: Behavior of weights when fitting multiple models. R is Pearson's R, MSD is mean squared logarithmic deviation from the one-to-one line, w_1 corresponds to a categorical model with 3 categories and 10 stimuli per category, w_2 corresponds to a random model (Method 1). Note the double logarithmic axes in the plots to the left. A low ratio w_1/w_2 corresponds to a low w_1, so the directionality is the same in the left and right plots. *Top row:* No channel noise, no constant used (nor needed) in fitting. *Middle row:* Channel noise added. *Bottom row:* Channel noise added, fitting using constant.

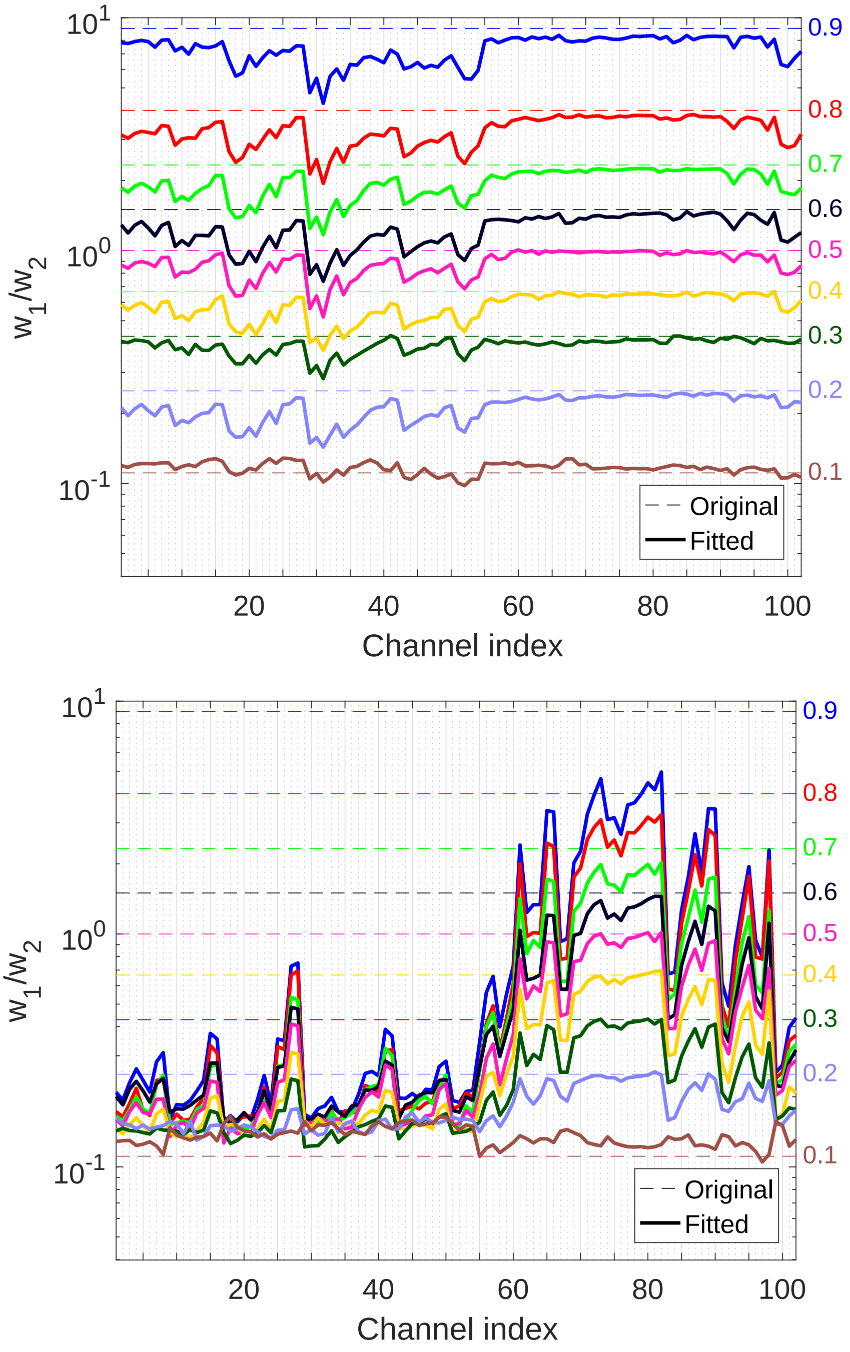

Figure 16: Fitting to channel neighborhood RDMs. The channel index is the index of the central channel in each neighborhood. The colored number to the right are the values of w_1 (categorical model with 3 categories and 10 stimuli per category). The other component model (w_2) was a random model (Method 1). Dashed lines signify the original weight ratios. *Upper:* No channel noise. *Lower:* Channel noise included.

4.3.4 Correlations

In addition to examining how the fitted weights behave, we were also interested in how the fitted RDM correlates with the activation model RDM. In simulations very similar to the previous ones, two RDMs from the Wishart distribution were used as component models. Both for maximum and average (over channel neighborhoods), the correlation between the fitted RDM and the data RDM was systematically higher than the correlation between the underlying model RDM and the data RDM for all weight pairs. For example, over the range $w_1 = 0 : 0.05 : 1$, the average correlation between model and data RDM was in the range $[0.02, 0.03]$ while the correlation between fitted and data RDM was in the range $[0.06, 0.07]$. This result calls for the use of cross-validation.

4.4 Cross-validation

4.4.1 Initial problems

Preliminary testing using CV-fitted RDMs conducted in the same manner as the simulations above showed both very low and highly negative correlations between the CV-fitted RDM and the signal RDM. Including the constant of Equation 4 in Section 3.2.2 when performing the CV-fit seemed to shift correlations downwards by in some cases as much as -0.75. To further analyze these issues, a signal RDM with a highly problematic correlation was examined (there were plenty of these, one was arbitrarily chosen). The channel neighborhood used and the RDM itself can be seen in Figure 17. The source activation was in the occipital lobe, so this area only had minor influences of the real activation, whose generating model was a combination of two random models from the Wishart distribution. The best fit of the component models to the signal RDM of this channel neighborhood had negative weights and a non-negative linear least squares resulted in weights of 0. This situation occurred in 23.4% of channel neighborhoods in the simulation overall.

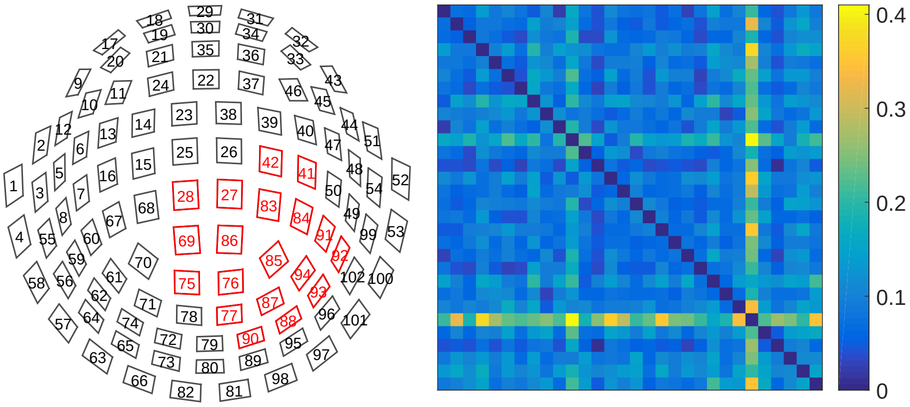

Figure 17: Channel neighborhood and signal RDM from one subject used to evaluate problems in CV-fitting.

A CV-fit was performed on the same signal RDM, both with and without the constant when evaluating the fit with $k = 2$. The results can be seen in Figure 18. In the CV-fit without the constant, 93% of the elements are 0 due to weights of zero being returned by the non-negative linear least squares algorithm. Increasing k led to more filled RDMs, but the variability between runs was much higher. Including the constant resulted in very high negative correlations. This can be explained by the algorithm mostly performing mean interpolation in this case, since an RDM calculated by using mean interpolation was visually inseparable from the right RDM in Figure 18. All of these problems were found to be alleviated by selecting a higher k and not using the constant when evaluating the fit.

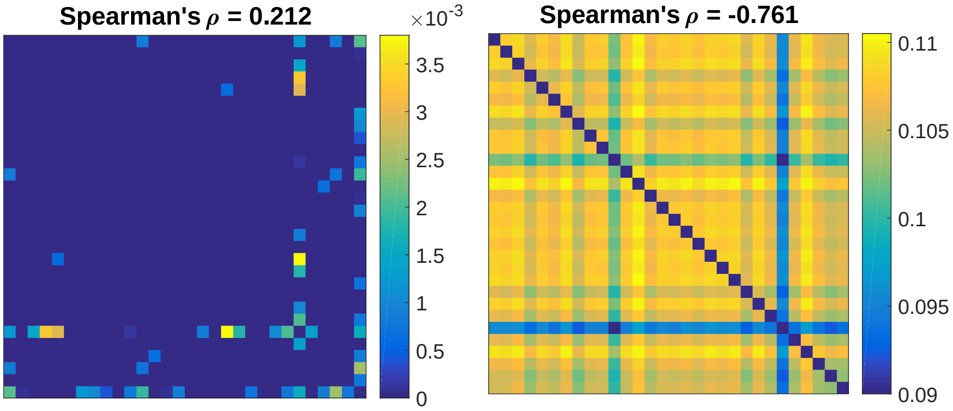

Figure 18: CV-fitted RDMs to the signal RDM shown in Figure 17. The component models were two random models from the Wishart distribution. *Left:* Constant not used when evaluating fit. *Right:* Constant used when evaluating fit.

4.4.2 Weights

Next, the average weights for determining the CV-fitted RDM were compared the to weights given by non-negative least-squares fitting. The activation was based on a linear combination of the two block models `[0 1; 1 1]` and `[1 1; 1 0]` of size 30×30 and comprised 150 sources in the occipital lobe. The simulation is an average of 5 runs. The situation was first examined using by averaging weights over all channel neighborhoods before calculating the ratio. Results of this can be seen in Figure 19. The results per channel are shown in Figure 20. The difference between weight ratios given by LSQ and CV is barely noticeable in these cases. It is worth observing that the simulation with only 5 runs leads to quite much variation in Figure 19, even though each run includes 10 subjects. This illustrates the importance of repeated trials.

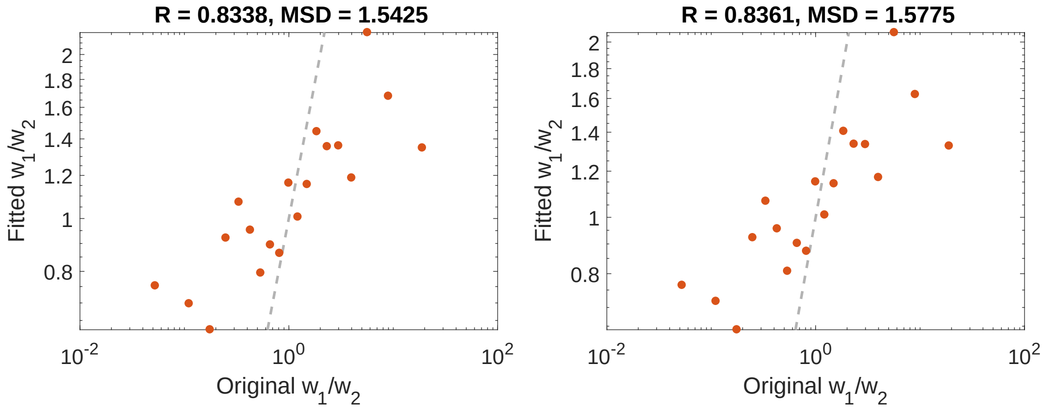

Figure 19: Fitted weight ratios versus original weight ratios for a linear combination of the block models `[0 1; 1 1]` and `[1 1; 1 0]`. *Left:* Using non-negative linear least squares. *Right:* Using average CV-fitted RDM weights.

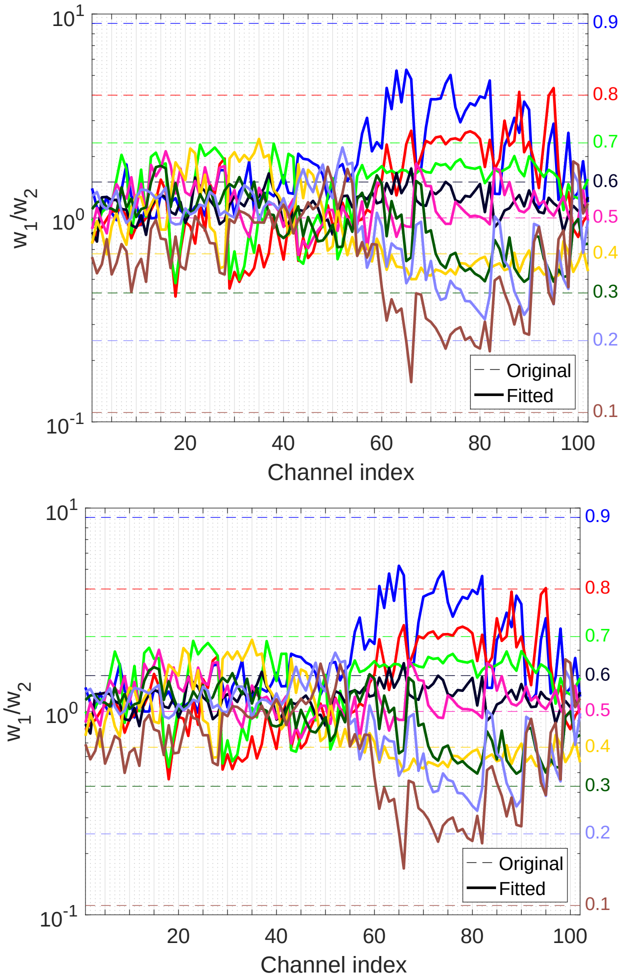

Figure 20: Fitted and original weights ratios per channel illustrated. The activation model is a linear combination of the block models `[0 1; 1 1]` and `[1 1; 1 0]`. *Left:* Using non-negative linear least squares. *Right:* Using average CV RDM weights.

4.4.3 Correlations

After this, focus was put on examining the correlations between the activation model RDM and the CV-fitted RDM. As previously noted, the LSQ-fitted RDM in many cases correlated higher with the signal RDM than the actual underlying model RDM did; this can be attributed to overfitting. In Figure 21, the correlations per channel neighborhood are visualized on the flattened MEG helmet. 150 sources in the occipital lobe were activated and the component models used are the block models `[1 1; 1 0]` and `[0 1; 1 1]` of size 150 × 150 with weights $w_1 = w_2 = 0.5$. The results are averages of 5 runs and $k = 15$ was used for the CV-fit. Note that even though the colorbar scales are different for each subplot, it visually looks like the LSQ-fit spreads out the correlations more than the CV-fit, which seems to better correspond to the correlations of the model RDM to the signal RDM (although they are a bit lower). The behavior is further visualized in Figure 22, where each data point corresponds to the correlation of the signal RDM of one channel neighborhood to the fitted RDM. While only the case of $w_1 = w_2 = 0.5$ is shown, the qualitative behavior for the correlations was similar for all weights $w_1 = [0 : 0.05 : 1]$ and $w_2 = 1 - w_1$. We can see that there is a substantial offset in the correlation between the LSQ-fitted RDM and the signal RDM and the correlation between the CV-fitted RDM and the signal RDM. This difference can be attributed to simple LSQ overfitting the RDM to the signal RDM.

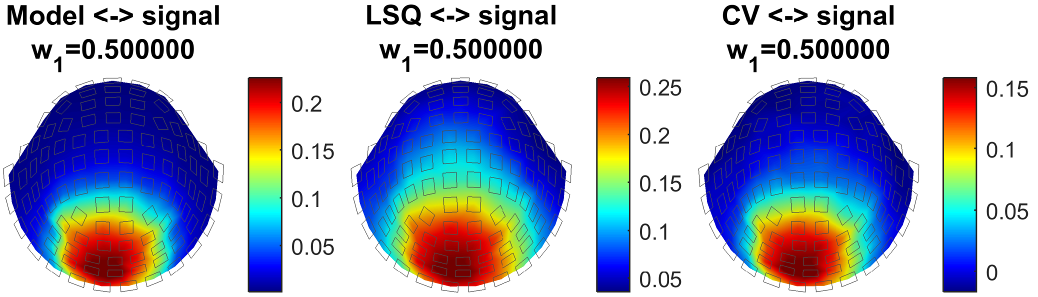

Figure 21: Correlations between the signal RDM and the model, LSQ-fitted and CV-fitted ($k = 15$) RDMs shown per channel neighborhood on the flattened MEG helmet. The activation models are the block models `[0 1; 1 1]` and `[1 1; 1 0]` of size 150 × 150.

The results of simulation with another choice of component models, a random model (Method 1) and a categorical model with 3 categories and 10 stimuli per category, can be seen in Figure 23. This particular case was chosen to illustrate a potential problems in CV-fitting. The activation model is 30% a categorical model and 70% a random model. We can see that the correlation of the LSQ-fitted RDM to the signal RDM corresponds quite well to the model RDM correlation to the signal RDM. However, when looking at the correlations between the CV-fitted RDM and the signal RDM, we notice that they are very small and spread out in a pattern not resembling the previous ones. This behavior can probably be attributed to the

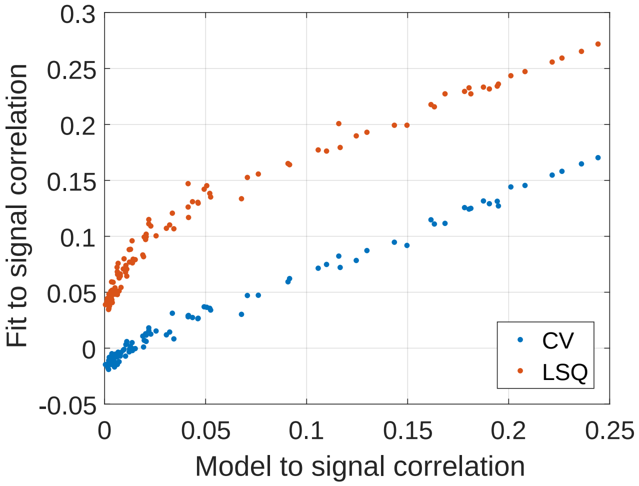

Figure 22: Comparing LSQ-fitted and CV-fitted RDM signal correlations to the model RDM signal correlation. The data are the same as in Figure 21, but visualized in another way.

randomness of the model and will be discussed further in Chapter 5.

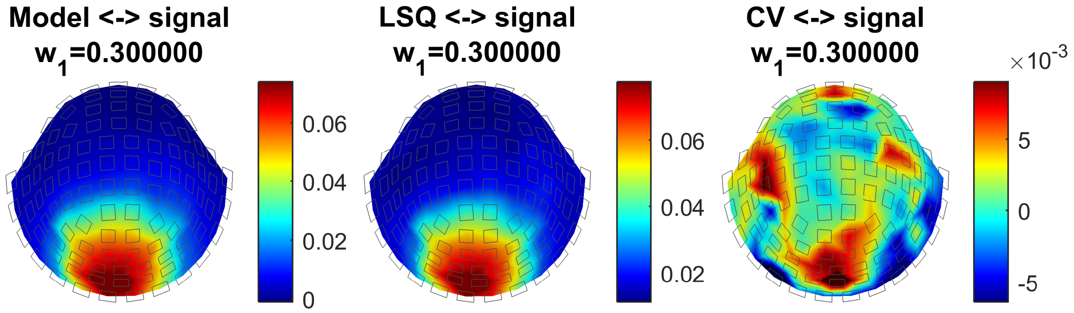

Figure 23: Correlations between the signal RDM and the model, LSQ-fitted and CV-fitted RDMs shown per channel neighborhood on the flattened MEG helmet. The activation models are a random model (Method 1, w_1) and a categorical model with 3 categories and 10 stimuli per category.

Another aspect of the two previous simulations is shown in Figure 24. Here, the correlation between the signal RDM and the activation model, LSQ-fitted and CV-fitted RDMs are shown for different weights. We can see that in the case of one random and one categorical component model (right plot), the maximum CV-fitted correlation is mostly flat for low w_1 (categorical model) and starts rising at $w_1 = 0.4$. This is probably due to the randomness of the model mentioned in the previous paragraph. The same figure also illustrates how the correlation of the LSQ-fit to the signal RDM is higher than the correlation between the activation model RDM and the signal RDM, indicating overfitting and calling for CV.

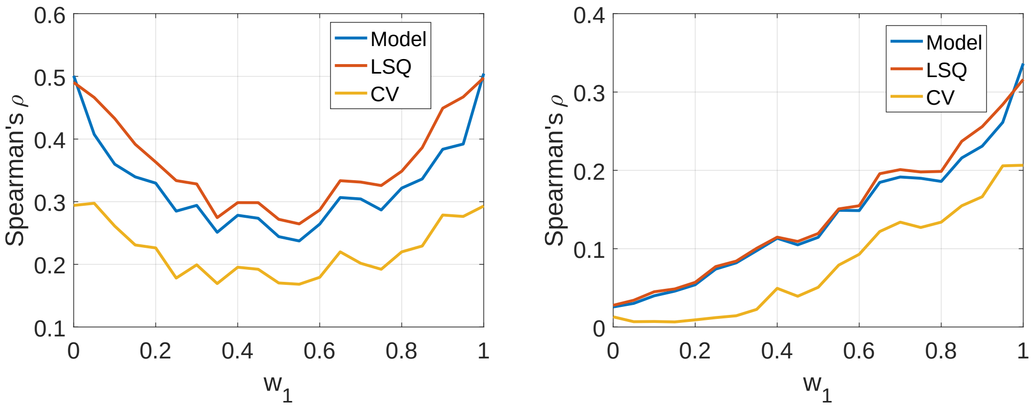

Figure 24: Maximum correlations over the channel neighborhood as a function of the weights of the component models, $w_2 = 1 - w_1$. *Left:* Block models `[0 1; 1 1]` (w_1) and `[1 1; 1 0]` (w_2) of size 30 × 30. *Right:* Random model (Method 1, w_1) and a categorical model (w_2) with 3 categories and 10 stimuli per category.

4.5 Multiple activations

In this section, results from simulations where two regions are active simultaneously, but activated using different models, are presented. 100 sources in the occipital lobe are used and they are divided into two separate regions of 50 sources each. The division is done straight along a given dimension (sagittal, transverse, coronal). See Figure 8 for an illustration. One of the regions is then activated by the model `[0 1; 1 1]` while the other one is activated by `[1 1; 1 0]`, both models of size 30 × 30. The fits were done using CV ($k = 10$) and average weights were calculated. The ratio of the weights of the models can be seen in Figure 25. Component model correlations in the specific case of splitting the activation region in the sagittal direction is illustrated in Figure 26.

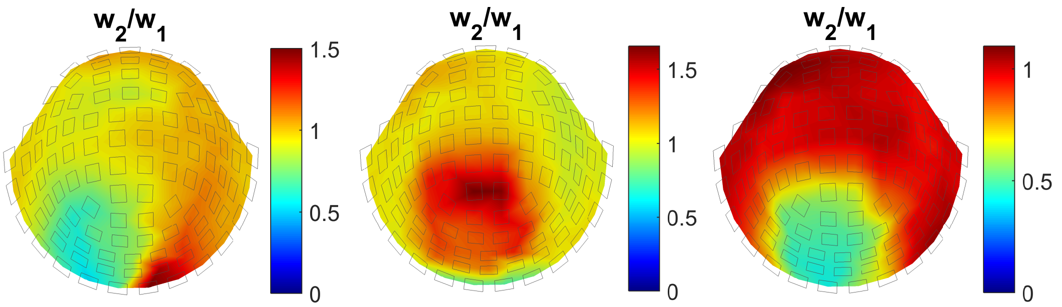

Figure 25: Ratios of CV-fitted weights per channel neighborhood shown on the flattened MEG helmet. The activation region has been split in the (from left to right) sagittal, coronal and transverse dimension and one part is activated by the block model `[0 1; 1 1]` while the other one is activated by `[1 1; 1 0]`.

The activation of two separate (but neighboring) regions using different models is reflected in the weight ratios in a quite expected way. The left-most panel of Figure

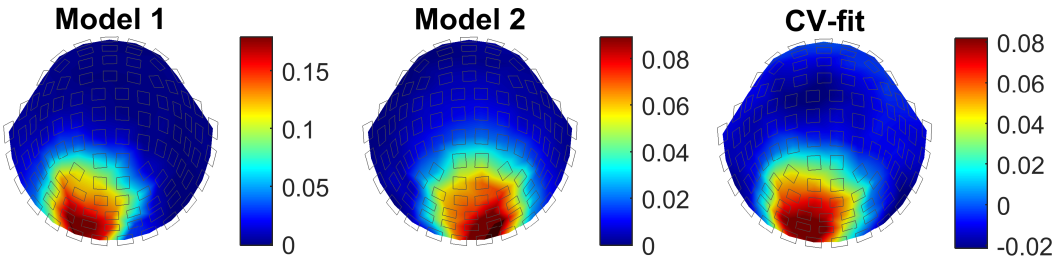

Figure 26: Correlations of component model RDMs and the CV-fitted RDM to the signal RDM. The activation region was split in the sagittal direction and the models used were 30 × 30 block models `[0 1; 1 1]` and `[1 1; 1 0]`.

25 shows a clear left-right separation most prominent at the location close to the activated region (in the occipital lobe). In the middle panel, the weight w_2 seems to be higher than weight w_1 almost everywhere, but close inspection shows that the channels in the back reach a weight ratio as low as 0.75. For the third panel, the split in the coronal direction leads to a deep and a shallow activation region. The shallow one naturally corresponds to w_1, as it conceals the activity of the deeper one. The correlations of the component models themselves to the signal RDM in the case of the sagittal split are also shown in 26, where we can see that it is possible to separate them spatially.

4.6 Application to real data

Since single model correlation already have been performed in the work by Henriksson and colleagues (2016), it was not repeated here. Instead, the difference between a LSQ-fit using only the Gabor-wavelet pyramid model and using all models mentioned in Section 3.4 was determined. The statistical significance of the difference was tested using the one tailed t-test and spatial cluster permutation described in the study. The difference between the fits for the different time windows can be seen in Figure 27. In these tests, a constant was not used when fitting, since it caused correlations in all location and for all time windows to be regarded significant. To determine weights for the models however, a constant was used, since the LSQ-routine otherwise would attribute a weight of almost 1 to the first model. The weights of different models for the last time window examined can be seen in Figure 28. An average of the weights over channels can also be plotted as a function of time for each model, but this visualization was not very informative in this case, since the Gabor-wavelet pyramid model was much stronger than all other models and the time span analyzed was short (due to saccade artifacts).

Applying CV to the fit did not yield any interesting results. When only using the Gabor-wavelet pyramid model, the maximum correlation occurred in the time window 75-100 ms and was 0.027. Using all models, the maximum correlation occurred in the same time window and was 0.028. Note that this is the maximum correlation and not the difference of correlations as shown in Figure 27. The results were obtained using $k = 20$. Other values of k were tested, but did not give any better results.

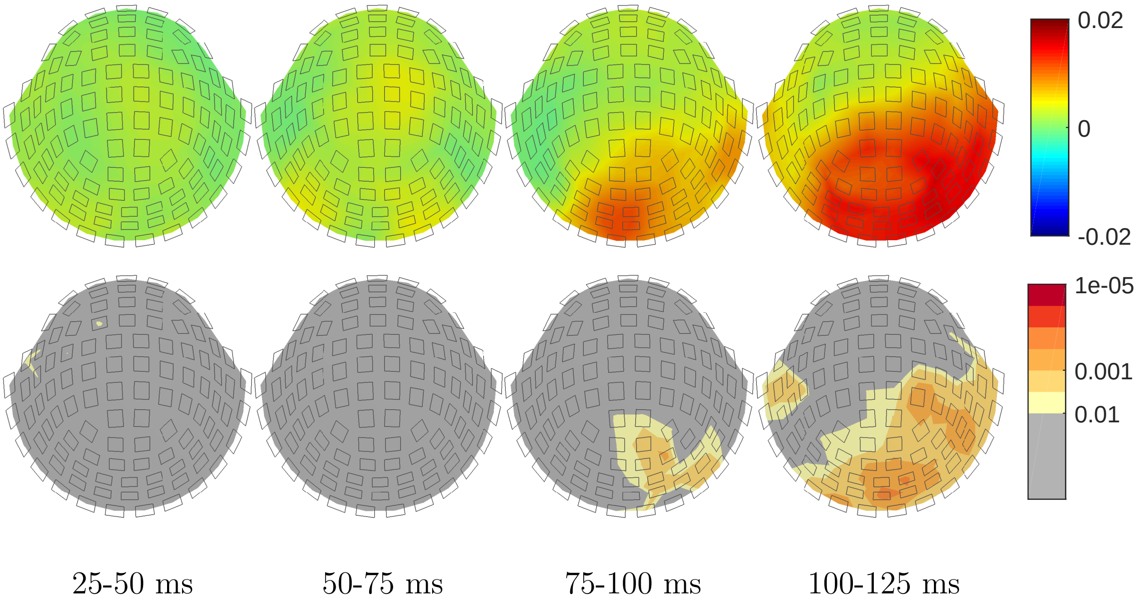

Figure 27: Application of LSQ-fitting to real data. *Top row:* Difference between correlations of a LSQ-fitted RDM with only the Gabor-wavelet pyramid model and all available models. *Bottom row:* Significance tested using cluster permutation.

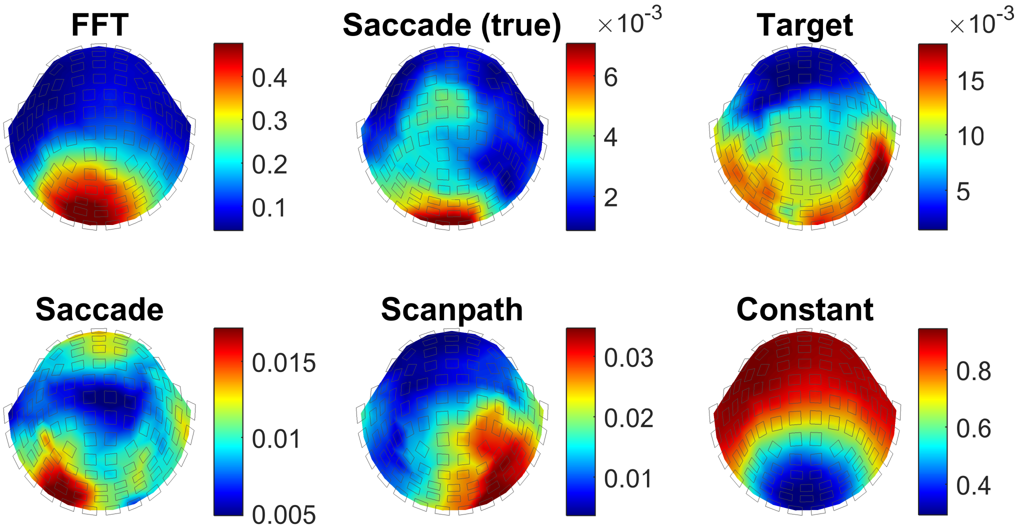

Figure 28: Weights assigned to the different models in the LSQ-fit with constant for the time window 100-125 ms. Note the that each panel has a different color scale.

5 Discussion

This work sought to both validate and extend a method for analyzing MEG data used in the work by Henriksson and colleagues (2016). As a preliminary step, it was shown that separate models can be detected via correlation even when the generating model is a combination of two models. The correlation between the signal RDM and a component model RDM monotonically increased as the weight of the component model in the generating model increased, as one intuitively would expect. This result gave a strong indication that fitting multiple models might give sensible results.

One interesting question was whether the forward model would preserve weight ratios of models from the source space to the sensor space. Although the fitted weight ratios (signal) not always corresponded to the original weight ratios (source), the dependence of these on each other usually seemed to be monotonic. The deviation from a one-to-one correspondence was complex and varied with different models. As opposed to the situation in simulations, there is in principle only a single set of underlying weights in the analysis of real data. Fitting therefore returns a single set of weights, but no data on how these depend on the underlying weights. By using simulations, we can however get a picture of how the weights of the models we are using behave theoretically. Using this information, the the real weight ratio can be estimated more accurately. This result might seem discouraging, but as absence of noise resulted in an almost perfect correspondence between original and fitted weight ratios, taking measures to raise the SNR will no doubt help.

Since one of the goals was to be able to use a spatiotemporal searchlight, the behavior of weight ratios on a channel neighborhood level was also examined. As expected and seen in Figure 16, adding noise made the detectability much more concentrated and the fitted weight ratios approach the real ones only for channel neighborhoods that are close to the activated region. When it comes to real data, the situation is similar to the one described in the previous paragraph.

It was determined that CV worked best by excluding the coefficient when fitting and using a k not too low. The method was also able to return weights that did not notably differ from those returned by a LSQ-fit. Therefore, if only the weights are of interest, LSQ can well be used instead, especially since CV is considerably slower (see Appendix E for details). A potential issue in using CV to determine correlations is shown in Figure 23, where the LSQ-fit very well corresponds to the actual model RDM to signal RDM correlation, while the situation for the CV-fit is completely the opposite. This phenomenon can be explained by referring to the structure of the models used. When performing leave-k-out CV, we assume that there are some kind of information about the left-out data retained in the kept data. Otherwise, the values of the left-out stimuli could not be estimated form a fit involving only the kept stimuli. In a truly random matrix, there is no dependence between the elements, so this assumption is therefore invalidated in such a case. In this particular case, even though the model to 30% consists of a categorical model (plus 70% random), this apparently is not enough to enable CV-fitting.

In the application to real data, the models used originally had relatively low correlations with the signal. The Gabor-wavelet pyramid model had a much greater explanatory power than the rest of the models in the original study, but still, a significant difference in correlation was attained by combining all models. Using CV did not return any meaningful results. Similar behavior of the CV-fit was also observed during testing multiple activations, where fitting only one model using CV (results not shown) resulted in negligible correlations although the standard LSQ-fit did not. In the real data, correlations were relatively low using LSQ, no repeated trials were performed and there was no clear categorical structure of the stimuli. All these factors probably contributed to the fact that the CV-fit was unsuccessful.

Overall, using CV seemed to result in too conservative correlation estimates, which is evident partly because of the failures mentioned above and partly in Figure 24. Here, the LSQ-fit is almost always overfitted, since it for most weights returns a higher correlation with the signal RDM than the actual model RDM does. The CV-fit however returns a noticeably lower correlation than the actual model RDM. Although conservative, correlations of the CV-fit deemed significant will indeed be that. Relying on LSQ is problematic if we want to report absolute correlations and determine their significance. The best solution in this case is to rely on CV. However, results between different LSQ-fits can be thought to be comparable. Especially, the difference in correlation of separate LSQ-fits can be subjected to ordinary methods for testing statistical significance. This is exactly what was done for the real data in this work. We did not report absolute correlations, but examined whether the correlations significantly increased when adding models to the fit.

Most of the tools required to perform multiple model fitting using non-negative linear least squares are provided by MATLAB and do not require implementation of any additional functions. A function for determining CV-fitted RDMs and their corresponding weights taking advantage of the indexing tricks presented in Appendix B is available from the author. It has been shown that using RSA for MEG on the sensor level and fitting multiple models to measured data is possible and can also be used to crudely localize activity corresponding to a given model. However, as discussed in the previous paragraph, the current recommendation is to use LSQ for fitting and apply CV only when the type of results demanded require it.

6 Summary

The main goal of this thesis was to examine fitting multiple models using linear least squares (LSQ) both with and without cross-validation (CV) within the representational similarity analysis (RSA) framework to magnetoencephalography (MEG) data. RSA was also used to illustrate activation spread in inverse modelling. By simulating source activations based on different types of models, we have shown that activity crudely can be localized. When using multiple models, the ratio of the weights returned by a LSQ-fit of models to the signal is under realistic conditions (i.e. noisy) not equivalent to the ratio of the weights of the underlying models, but simulations can help understand the behavior of the weights when fitting to real data. Weight ratios behave as expected when there are two neighboring activation regions activated by different models, that is, the ratio is biased towards the model whose activation lies closest to the point examined. Using CV prevents overfitting and the average weights returned are virtually identical to the weights given by a simple LSQ-fit. However, if the underlying model contains too little structure or if the correlations are very low, employing CV can result in negligible correlations. This phenomenon was observed both in simulated data and in the application to real data, where LSQ-fitting showed a significant increase in correlation when adding more models, while CV-fitting overall returned very low correlations.

References

Alpaydin, E. (2014). *Introduction to machine learning.* MIT press.

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541.

Baillet, S., Mosher, J. C., and Leahy, R. M. (2001). Electromagnetic brain mapping. *IEEE Signal Processing Magazine*, 18(6):14–30.

Bear, M. F., Connors, B. W., and Paradiso, M. A. (2016). *Neuroscience: Exploring the Brain.* Wolters Kluwer.

Carlson, T., Tovar, D. A., Alink, A., and Kriegeskorte, N. (2013). Representational dynamics of object vision: The first 1000 ms. *Journal of Vision*, 13(10):1.

Cichy, R. M., Pantazis, D., and Oliva, A. (2014). Resolving human object recognition in space and time. *Nature Neuroscience*, 17(3):455–462. Article.

Coles, M. and Rugg, M. (1996). *Event-related brain potentials: an introduction*, pages 1–26. Oxford University Press.

Cox, D. D. and Savoy, R. L. (2003). Functional magnetic resonance imaging (fMRI) “brain reading”: detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2):261–270.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., and Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. *NeuroImage*, 86:446–460.

Haacke, E. M., Brown, R. W., Thompson, M. R., Venkatesan, R., et al. (1999). *Magnetic resonance imaging: physical principles and sequence design*, volume 82. Wiley-Liss New York:.

Haumann, N. T., Parkkonen, L., Kliuchko, M., Vuust, P., and Brattico, E. (2016). Comparing the performance of popular MEG/EEG artifact correction methods in an evoked-response study. *Computational Intelligence and Neuroscience*, 2016.

Hays, W. L. (1973). *Statistics for the social sciences.* Holt, Rinehart and Winston, Inc.

Henriksson, L., Olander, K., and Hari, R. (2016). Cortical dynamics of saccade-target selection during free-viewing of natural scenes. *bioRxiv.* URL: http://biorxiv.org/content/early/2016/09/19/075929.

Huettel, S. A., Song, A. W., and McCarthy, G. (2004). *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland.

Hämäläinen, M. and Hari, R. (2002). *Magnetoencephalographic characterization of dynamic brain activation: Basic principles and methods of data collection and source analysis*, pages 227–253. Elsevier Science, 2nd edition edition.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65:413–497.

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

Jenson, O., Spaak, E., and Zumer, J. M. (2014). Human brain oscillations: From physiological mechanisms to analysis and cognition. In Aine, C. J. and Supek, S., editors, *Magnetoencephalography: From Signals to Dynamic Cortical Networks.*, pages 359–404. Springer.

Jozwik, K. M., Kriegeskorte, N., and Mur, M. (2016). Visual features as stepping stones toward semantics: Explaining object similarity in IT and perception with non-negative least squares. *Neuropsychologia*, 83:201–226. Special Issue: Functional Selectivity in Perceptual and Cognitive Systems - A Tribute to Shlomo Bentin (1946-2012).

Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLOS Computational Biology*, 10(11):1–29.

Kriegeskorte, N., Goebel, R., and Bandettini, P. (2006). Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America*, 103(10):3863–3868.

Kriegeskorte, N. and Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in Cognitive Sciences*, 17(8):401–412.

Kriegeskorte, N., Mur, M., and Bandettini, P. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Lawson, C. and Hanson, R. (1974). *Solving Least Squares Problems.* Prentice-Hall.

Lee, W.-C. A., Huang, H., Feng, G., Sanes, J. R., Brown, E. N., So, P. T., and Nedivi, E. (2005). Dynamic remodeling of dendritic arbors in gabaergic interneurons of adult visual cortex. *PLOS Biology*, 4(2).

Logothetis, N. K., Pauls, J., Augath, M., Trinath, T., and Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412(6843):150–157.

Manger, P., Cort, J., Ebrahim, N., Goodman, A., Henning, J., Karolia, M., Rodrigues, S.-L., and Strkalj, G. (2008). Is 21st century neuroscience too focussed on the rat/mouse model of brain function and dysfunction? *Frontiers in Neuroanatomy*, 2:5.

Mur, M., Meys, M., Bodurka, J., Goebel, R., Bandettini, P. A., and Kriegeskorte, N. (2013). Human object-similarity judgments reflect and transcend the primate-it object representation. *Frontiers in Psychology*, 4:128.

Murakami, S. and Okada, Y. (2006). Contributions of principal neocortical neurons to magnetoencephalography and electroencephalography signals. *The Journal of Physiology*, 575(3):925–936.

Nelsen, R. B. (2011). Kendall tau metric. In *Encyclopedia of Mathematics*. Springer. URL: https://www.encyclopediaofmath.org/. Accessed: 19.12.2016.

Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., and Kriegeskorte, N. (2014). A toolbox for representational similarity analysis. *PLoS Comput Biol*, 10(4):1–11.

Pakkenberg, B., Pelvig, D., Marner, L., Bundgaard, M. J., Gundersen, H. J. G., Nyengaard, J. R., and Regeur, L. (2003). Aging and the human neocortex. *Experimental Gerontology*, 38(1–2):95–99. Proceedings of the 6th International Symposium on the Neurobiology and Neuroendocrinology of Aging.

Peters, B., Bledowski, C., Rieder, M., and Kaiser, J. (2016). Recurrence of task set-related MEG signal patterns during auditory working memory. *Brain Research*, 1640, Part B:232–242. Auditory Working Memory.

Ramkumar, P., Hansen, B. C., Lee, A., Lanphier, S., Pannasch, S., and Loschky, L. C. (2014). A high resolution neural portrait of natural scene processing. SUNw: Scene Understanding Workshop, Columbus, OH.

Redcay, E. and Carlson, T. A. (2015). Rapid neural discrimination of communicative gestures. *Social Cognitive and Affective Neuroscience*, 10(4):545–551.

Su, L., Fonteneau, E., Marslen-Wilson, W., and Kriegeskorte, N. (2012). Spatiotemporal searchlight representational similarity analysis in EMEG source space. In *Proceedings of the 2012 Second International Workshop on Pattern Recognition in NeuroImaging*, PRNI '12, pages 97–100, Washington, DC, USA. IEEE Computer Society.

Su, L., Zulfiqar, I., Jamshed, F., Fonteneau, E., and Marslen-Wilson, W. (2014). Mapping tonotopic organization in human temporal cortex: representational similarity analysis in EMEG source space. *Frontiers in Neuroscience*, 8:368.

Tyler, L., Cheung, T., Devereux, B., and Clarke, A. (2013). Syntactic computations in the language network: Characterizing dynamic network properties using representational similarity analysis. *Frontiers in Psychology*, 4:271.

Wang, J. Z., Williamson, S. J., and Kaufman, L. (1992). Magnetic source images determined by a lead-field analysis: the unique minimum-norm least-squares estimation. *IEEE Transactions on Biomedical Engineering*, 39(7):665–675.

Wardle, S. G., Kriegeskorte, N., Grootswagers, T., Khaligh-Razavi, S.-M., and Carlson, T. A. (2016). Perceptual similarity of visual patterns predicts dynamic neural activation patterns measured with MEG. *NeuroImage*, 132:59–70.

Wingfield, C., Su, L., Liu, X., Zhang, C., Woodland, P., Thwaites, A., Fonteneau, E., and Marslen-Wilson, W. D. (2016). Relating dynamic brain states to dynamic machine states: human and machine solutions to the speech recognition problem. *bioRxiv*. URL: http://biorxiv.org/content/early/2016/09/12/074799.

Woods, D., Herron, T., Cate, A., Yund, E. W., Stecker, G. C., Rinne, T., and Kang, X. (2010). Functional properties of human auditory cortical fields. *Frontiers in Systems Neuroscience*, 4:155.

Ölander, K. (2015). Eye movements and early magnetoencephalographic brain responses induced by faces in natural scenes. Master's thesis, Aalto University, Finland.

A On distance metrics

Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ be real vectors of length n. We will show that the correlation distance of these vectors is proportional to the squared Euclidean distance of the corresponding normalized vectors (with mean 0 and standard deviation 1). The Euclidean distance is defined as

$$d_{\mathrm{Euc}}(\mathbf{x}, \mathbf{y}) := \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

Normalizing the vectors results in

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} \quad \text{and} \quad \tilde{\mathbf{y}} = \frac{\mathbf{y} - \mu_{\mathbf{y}}}{\sigma_{\mathbf{y}}},$$

where μ_a and σ_a denote the mean and standard deviation respectively of a. The squared Euclidean distance of the normalized vectors is

$$\begin{aligned} d^2_{\mathrm{Euc}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) &= \sum_{i=1}^{n} \left(\frac{x_i - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} - \frac{y_i - \mu_{\mathbf{y}}}{\sigma_{\mathbf{y}}} \right)^2 \\ &= \sum_{i=1}^{n} \left(\left(\frac{x_i - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} \right)^2 - 2\frac{(x_i - \mu_{\mathbf{x}})(y_i - \mu_{\mathbf{y}})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} + \left(\frac{y_i - \mu_{\mathbf{y}}}{\sigma_{\mathbf{y}}} \right)^2 \right). \qquad (A1) \end{aligned}$$

Let us now analyze one of the squared terms in the sum. Using the definition of the standard deviation, we have that

$$\sum_{i=1}^{n} \left(\frac{x_i - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} \right)^2 = \frac{\sum_{i=1}^{n} (x_i - \mu_{\mathbf{x}})^2}{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{\mathbf{x}})^2} = n.$$

Substituting this result into Equation A1, we get

$$\begin{aligned} d^2_{\mathrm{Euc}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) &= 2n - 2\sum_{i=1}^{n} \frac{(x_i - \mu_{\mathbf{x}})(y_i - \mu_{\mathbf{y}})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \\ &= 2n \left(1 - \sum_{i=1}^{n} \frac{\frac{1}{n}(x_i - \mu_{\mathbf{x}})(y_i - \mu_{\mathbf{y}})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \right) \\ &= 2n \left(1 - \frac{\mathrm{cov}(\mathbf{x}, \mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \right), \qquad (A2) \end{aligned}$$

where cov stands for the sample covariance of the vectors. The correlation distance is defined as

$$d_{\mathrm{corr}}(\mathbf{x}, \mathbf{y}) := 1 - \mathrm{corr}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathrm{cov}(\mathbf{x}, \mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}. \qquad (A3)$$

By comparing Equations A2 and A3, we can see that

$$d_{\mathrm{corr}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2n} d^2_{\mathrm{Euc}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}).$$

B Matrix- and vector-based indexing

In this section, we will derive a formula for converting matrix indices to corresponding vector indices in accordance with the transformation discussed in Section 3.3.1. Since RDMs are symmetric and have zeroes on the diagonal, we only need to take the lower triangular part below the diagonal into account. We will start by examining the 6×6 matrix

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 1 & \times & \times & \times & \times & \times \\ 2 & 6 & \times & \times & \times & \times \\ 3 & 7 & 10 & \times & \times & \times \\ 4 & 8 & 11 & 13 & \times & \times \\ 5 & 9 & 12 & 14 & 15 & \times \end{bmatrix},$$

where the crosses indicate uninteresting elements. The values of the matrix correspond to the indices of the corresponding vector. We now wish to find a function that would allow us to convert row-column-pairs (r, c) into vector indices i. Once we have found the index of the first element in a column ($1, 6, 10, 13$ or 15 in the matrix above), it is easy to take the required number of steps forward to arrive at the correct row. The difference between the indices of the first elements of the columns is always reduced by 1 when moving to the right. This tells us that the relation between the column number and the index can be explained by a second-degree polynomial function.

In the general case of $A \in \mathbb{R}^{n \times n}$, the indices of the first elements of the first three columns are $1, s+1$ and $2s$, where $s = n - 1$. We will fit the second-order polynomial function $i(c) = a_1 c^2 + a_2 c + a_3$ to these values. We arrive at the equation system

$$\begin{cases} a_1 + a_2 + a_3 = 1 \\ 4a_1 + 2a_2 + a_3 = s + 1 \\ 9a_1 + 3a_2 + a_3 = 2s \end{cases},$$

that can be solved to give

$$\begin{cases} a_1 = -\frac{1}{2} \\ a_2 = s + \frac{3}{2} \\ a_3 = -s \end{cases}.$$

This gives us

$$i(c) = -\frac{1}{2}c^2 + c\left(s + \frac{3}{2}\right) - s.$$

We extend this function to also take the row index into account. The number of steps to walk downwards from the first element of a column to reach the correct row is $r - c - 1$. This results in

$$\begin{aligned} i(r, c) &= \frac{1}{2}c^2 + c\left(s + \frac{3}{2}\right) - s + r - c - 1 \\ &= \frac{1}{2}c\left(2s - c + 1\right) + r - s - 1. \end{aligned}$$

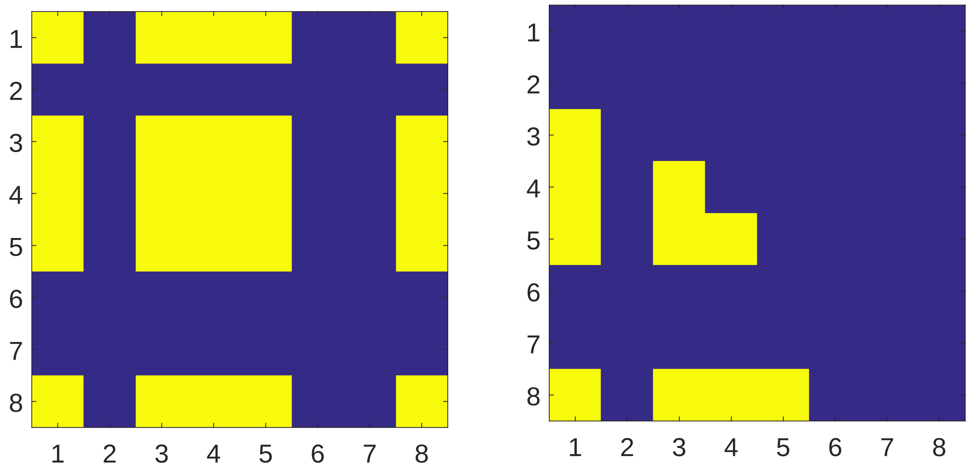

Figure B1: *Left:* The submatrix sought when extracting rows and columns 1, 3, 4, 5 and 8 shown in yellow. *Right:* The actual elements that need to be extracted when working with an RDM due to symmetry and zeroes on the diagonal.

This formula will return incorrect values if trying to retrieve the index for a diagonal element or for an element in the upper right part of the matrix.

An usual application in this thesis is the extraction of a submatrix. This operation is illustrated in Figure B1 and an implementation in MATLAB is shown as Code 1. Rows 4 and 5 generate index matrices that correspond to the yellow part of the left image in Figure B1. Row 7 will generate erroneous output for all elements except for those shown in the right image of Figure B1. Since we only need the elements shown in this image, we can remove all indices where the original row index was larger than the column index (row 8).

Code 1: Converts matrix indices to vector indices for submatrix extraction. The input *useInds* is a vector of columns/rows to be extracted (refer to Figure B1) and *numStimuli* is the size of the original matrix.

```
1  function inds = convInds(useInds, numStimuli)
2
3  s = numStimuli - 1;
4  c = repmat(useInds, length(useInds), 1);
5  r = c';
6
7  inds = .5 * c .* (1 - c + 2 * s) + r - s - 1;
8  inds = inds(c < r);
```

C Spearman's ρ and Kendall's τ_a

Due to the computation of Kendall's τ_a having a higher complexity than the computation of Spearman's ρ, the behavior of these rank correlation coefficients was evaluated to examine whether Kendall's τ_a provided meaningful additional information in our simulations.

To test a case with many tied ranks, a categorical model C with 3 categories and 10 items per category was used as the real model. This model was contaminated with noise by linearly combining it with a random (uniformly sampled from $[0, 1]$) model R in different proportions, so that the observed model was given by

$$O = w_1 C + w_2 R,$$

where $w_1 + w_2 = 1$. Rank correlation coefficients between O and C were calculated for different values of w_1 and the results can be seen in Figure C1. Every point is the average of results from using 50 different random models. The ratio between Spearman's ρ and Kendall's τ_a is constant (1.84) for all weights except for $w_1 = 1$. In this case, the real model is correlated against itself, and $\rho = 1$ as expected. As stated in Section 2.2.1, Kendall's τ_a only includes the number of concordant and discordant pairs, ignoring the pairs for which either $x_a = x_b$ or $y_c = y_d$. The pairs arise frequently when comparing a categorical model (containing only the values 0 and 1) to itself. Therefore, τ_a remains low, even though there is a perfect match between the correlated vectors. The conclusion is that Kendall's τ_a provides no more information than Spearman's ρ in this particular case.

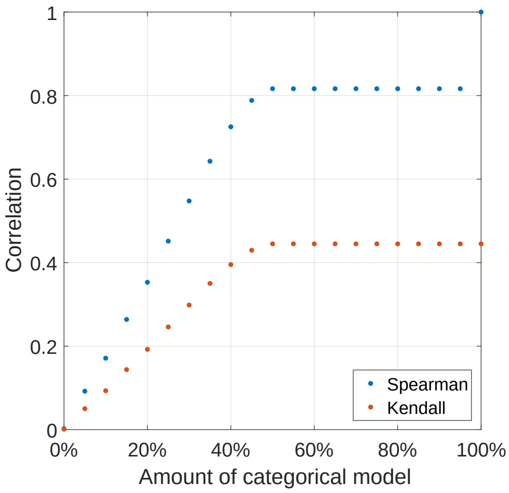

Figure C1: Values of Spearman's ρ and Kendall's τ_a when correlating a model to itself with added noise. See the text for more details.

D Fractional ranking

Here we will show that using tied ranks is equivalent to averaging over all possible permutations of ranks (refer to the discussion in Section 2.2.2). We will start by considering permutations of the set $\{k, \dots, n+k-1\}$, which contains n elements and therefore $n!$ permutations. We can list all of these permutations row-wise; an example for $k = 1$ and $n = 3$ is shown below.

$$\begin{array}{cccc} & 1 & 2 & 3 \\ & 1 & 3 & 2 \\ & 2 & 1 & 3 \\ & 2 & 3 & 1 \\ & 3 & 1 & 2 \\ + & 3 & 2 & 1 \\ \hline & 12 & 12 & 12 \end{array}$$

The sum of every column $S(k, n)$ is the same, since each element has the same number of occurrences in every column. The sum is obtained by multiplying the average of a row with the total number of rows,

$$\begin{aligned} S(k,n) &= n! \left(\frac{(n+k)(n+k-1)}{2} - \frac{k(k-1)}{2} \right) \Big/ n \\ &= \frac{1}{2} n!(n+2k-1) \end{aligned}$$

The mean $m(k, n)$ of each column when averaging over rows is therefore

$$m(k,n) = \frac{1}{2}(n+2k-1),$$

the same as the mean of the numbers $\{k, \dots, n+k-1\}$.

Denote the non-ties of a ranking by r_i, where $i = 1, \dots, n_r$ and n_r is the number of non-ties. Also denote the ties by $t_{i,j}$, where $i = 1, \dots, n_g$ and $j = 1, \dots, n_{t,i}$, where n_g is the number of groups of ties and $n_{t,i}$ is the number of ties per group. An example of a ranking in this notation is

$$\underset{1}{r_1}, \underset{2}{r_2}, \underset{3}{t_{1,1}}, \underset{4}{t_{1,2}}, \underset{5}{t_{1,3}}, \underset{6}{r_4}, \underset{7}{t_{2,1}}, \underset{8}{t_{2,2}}, \underset{9}{r_5}.$$

The number of possible permutations is $\prod_{i=1}^{n_g} n_{t,i}!$. Non-ties are unaffected when averaging over permutations, since they are constant. When listing all permutations, we notice that for each permutation of the group t_b, all possible permutations of t_a are repeated. This increases the number of rows by a factor of $n_{t,b}$, but does not change the means of the columns for t_a. The same reasoning applies to the behavior of any tie group when considering permutations of all the other tie groups. Therefore,

$$\bar{t}_{a,1} = \dots = \bar{t}_{a,n_{t,a}} = \frac{m(t_{a,1}, t_{a,n_{t,a}}) \prod_{i=1}^{n_g} n_{t,i}!}{\prod_{i=1}^{n_g} n_{t,i}!} = m(t_{a,1}, t_{a,n_{t,a}})$$

for all $a = 1, \dots, n_g$, where the bar symbol denotes the mean. This result is the mean of the numbers $\{t_{a_1}, \dots, t_{n_{t_a}}\}$, i.e. the same as fractional ranking of ties.

E Computational cost of cross-validation

We examine the expected value of the number of steps required to fully construct a vector of length n when using leave-k-out cross-validation with a random choice in every step. For $k = 1$, we will derive an analytical expression, but for $k > 1$, we will rely on simulations.

We start by deriving a formula $s(l, n)$ for determining the number of possible strings of length l using n characters where every character must appear at least once. As an example, we will consider $n = l = 4$ and the alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. The number of possible strings without any constraint is 4^4. We start by removing all strings not containing the character `a`, of which there are 3^4. Repeating this for all characters, we get $4^4 - 4 \cdot 3^4$. In doing this, we have removed the strings containing e.g., only the characters `a` and `b` twice, since they are missing both `c` and `d`. Therefore, we must add back the number of strings not containing `c` and `d` and repeat this for all possible pairs, of which there are $\binom{4}{2}$. This gives us $4^4 - 4 \cdot 3^4 + \binom{4}{2} \cdot 2^4$. Now, we have compensated too much, since also the strings `aaaa`, `bbbb`, `cccc` and `dddd` have been added back. We ultimately arrive at

$$\begin{aligned} s(4,4) &= 4^4 - 4 \cdot 3^4 + \binom{4}{2} \cdot 2^4 - 4 \\ &= \binom{4}{0} \cdot 4^4 - \binom{4}{1} \cdot 3^4 + \binom{4}{2} \cdot 2^4 - \binom{4}{3} \cdot 1^4 \\ &= \sum_{j=0}^{3} (-1)^j \binom{4}{j} (4-j)^4. \end{aligned}$$

This can be generalized to

$$s(l, n) = \sum_{j=0}^{n-1} (-1)^j \binom{n}{j} (n-j)^l. \tag{E1}$$

We can illustrate the leave-out choice history by a string of numbers. For example `113` means that data point 1 was left out during the first and second run and data point 3 was left out during the third run. For a history x of length l to be *complete*, every number must appear at least once in the whole string, but every number cannot appear in the subhistory $x[1 : (l-1)]$. The complete histories of length l correspond to the CV finishing in l steps.

We look at the case of $n = 4$, where n is the number of data points, and try to determine to number of complete histories as a function of their length l. If the last number of a complete history of length l is `4`, all others numbers in the history must belong to $\{\mathtt{1}, \mathtt{2}, \mathtt{3}\}$; otherwise, the process would have finish earlier. The number of complete histories ending with `4` is therefore $s(l-1, 3)$. This reasoning should be repeated every other number, resulting in a total of $4 \cdot s(l-1, 3)$ complete histories of length l. In the general case, there are $n \cdot s(l-1, n-1)$ complete histories of length l.

Let the stochastic variable X_n be the length of the complete history when there is n data points. To determine the expected value of X_n, we need to calculate the probability of a complete history of a certain length occurring. The total number of possible histories is n^l. This gives us

$$p(X_n = l) = \frac{n \cdot s(l-1, n-1)}{n^l} = \frac{s(l-1, n-1)}{n^{l-1}}.$$

For the expected value, we get

$$\begin{aligned} \mathrm{E}[X_n] &= \sum_{l=n}^{\infty} l \cdot p(X_n = l) = \sum_{l=n}^{\infty} \frac{l \cdot s(l-1, n-1)}{n^{l-1}} \\ &= \sum_{l=n}^{\infty} \frac{l}{n^{l-1}} \sum_{j=0}^{n-2} (-1)^j \binom{n-1}{j} (n-1-j)^{l-1}, \end{aligned}$$

where the summing starts from n, since no complete history can have a length less than n. The sum can be evaluated exactly using a CAS (in our case, Wolfram Mathematica 11 was used). Experiments showed that the expected value can be approximated by

$$\mathrm{E}[X_n] \approx c_1 n \log n + c_2 n + c_3. \tag{E2}$$

Using $n = [2 \quad 7 \quad 12]$ with corresponding $\mathrm{E}[X_n] = [3 \quad \frac{363}{20} \quad \frac{86021}{2310}]$ and performing a least-squares fit resulted in the coefficients $c = [0.9940 \quad 0.5978 \quad 0.4266]$. The errors for $n = 1000$ and $n = 2000$ were 0.28% and 0.31% respectively.

For $k > 1$, deriving a formula turned out to be tedious, since the alphabet consists of k-tuplets of numbers which are highly dependent on each other in determining the completeness of a history. Therefore, we relied on simulations instead. The expected number of steps required was calculated as the average number of steps needed over 10000 runs for each combination of n and k. Results are shown in Figure E1. Equation E2 was again used for fitting curves. Using the data points shown in the figure and examining $n = 5000$, the fit resulted in the errors 0.24%, 1.6% and 0.20% for $k = 2$, $k = 5$ and $k = 10$.

Another point of view perhaps more practically motivated is to look a how changing k affects the expected number of steps for a fixed n. Results of this are shown in Figure E2, where each point again is the average of 10000 runs. These curves can be approximated by a function of the form

$$s = \frac{c_1}{k} + c_2,$$

where s is the expected number of steps needed.

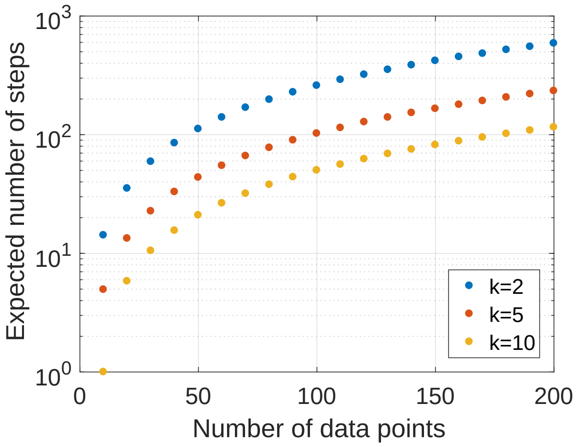

Figure E1: The expected number of steps required as a function of n, the number of data points, shown for different values of k, the number of left-out data points per step.

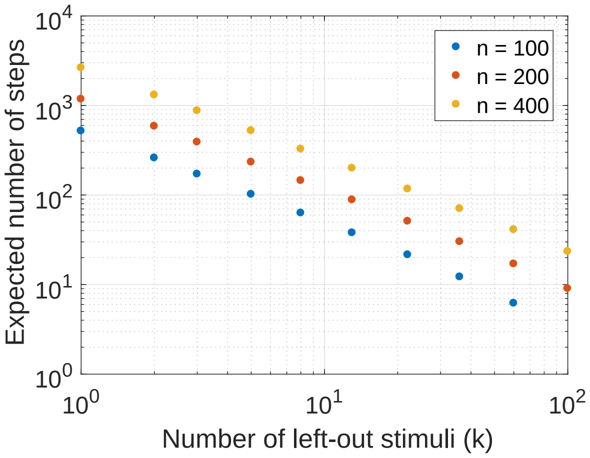

Figure E2: The expected number of steps required as a function of k, the number of left-out data points per step, shown for different values of n, the number of data points. Note the double logarithmic axes.

F Mean interpolation

In this section, we will show that a vector constructed from another vector using mean interpolation results in perfect anti-correlation between the vectors. A variant of this arises when performing CV and including the constant when evaluating the fit (see Section 4.4.1 for details). If the weights are zero (due to the non-negative weight constrain and bad data/models), the constant will be the mean of the training data, since this is the best solution in the least squares sense. The left-out stimuli will then acquire this mean value.

Let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ be given and let $\mu_{\mathbf{x}}$ be the mean of $\mathbf{x}$. We will construct $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) \in \mathbb{R}^n$ by letting

$$\tilde{x}_i = \frac{1}{n-1} \sum_{k \neq i} x_k, \tag{F1}$$

i.e. letting $\tilde{x}_i$ be the mean of $\mathbf{x}$ with x_i left out. We can rewrite Equation F1 as

$$\begin{aligned} \tilde{x}_i &= \frac{\sum_{i=1}^n x_i}{n-1} - \frac{x_i}{n-1} \\ &= \frac{n \cdot \frac{1}{n} \sum_{i=1}^n x_i}{n-1} - \frac{x_i}{n-1} \\ &= \frac{n\mu_{\mathbf{x}}}{n-1} - \frac{x_i}{n-1} \\ &= a + bx_i, \end{aligned}$$

where a and b are both independent of i and $b = -1/(n-1)$, showing that $\tilde{x}_i$ is the same decreasing linear function of x_i for all i. We can therefore write $\tilde{\mathbf{x}} = a + b\mathbf{x}$. Since $b < 0$, this implies that

$$\operatorname{corr}(\mathbf{x}, \tilde{\mathbf{x}}) = -1.$$