

Sparse EEG Imaging

Sofie Therese Hansen

DTU



Kongens Lyngby 2013
IMM-M.Sc.-2013-3

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc.-2013-3

Summary, English

In this thesis a new algorithm is examined with respect to its application to electroencephalography (EEG) source reconstruction and its potential use in EEG biofeedback. The novel technique is named the *variational Garrote* (VG) and was suggested by Kappen et al. in a not yet published article. The algorithm makes two key assumptions; the problem at hand is linear, and it has a sparse solution. The latter is obtained by including a binary switch for each input variable in the linear model that determines whether a variable is relevant or not. The solution is found using Bayesian inference. The assumptions potentially make the algorithm well-suited for solving the highly underdetermined EEG inverse problem. Main contributions of this thesis include verifying VG in EEG settings and expanding the algorithm to the time domain. Publications of findings are submitted to the International Conference on Acoustics, Speech, and Signal Processing 2013 and the IEEE International Winter Workshop of Brain-Computer Interface 2013. The algorithm's performance, as described by Kappen et al., was confirmed initially. Reformulations of the VG problem reducing computation complexity using the Kailath Variant relation and a dual representation, respectively, were compared to applying the least absolute shrinkage and selection operator (LASSO) and to a sparse Bayesian model with a linear basis. Here, a forward field matrix was used as input while the source distribution was synthetically created. The dual formulation of the VG algorithm was found to be superior and was expanded from the time instantaneous formulation. Under the assumption that activity in a source is present for a certain but possibly short amount of time, the individual binary switches were assumed to have constant modes (on or off) across 20-25 time samples, corresponding to 100 ms in EEG settings. The time-expanded version of the dual VG formulation was validated using synthetic data and EEG data with the

visual stimuli paradigm described by Henson et al. (2003). The resulting source distribution was comparable to that presented in studies of the response measured by EEG as well as by other modalities. The VG algorithm is suggested to be further expanded to perform online tracking of brain activity by reducing the computation complexity further and to include spatial smoothness.

Keywords: Sparsity, EEG, Real-time imaging, Bayesian inference, Variational Garrote, LASSO, ARD.

Summary, Danish

I denne afhandling undersøges en ny algoritme med hensyn til dens anvendelse i elektroencefalografi (EEG) kilde lokalisation og dens potentielle brug i EEG biofeedback. Den nye teknik kaldes *variational Garrote* (VG) og blev foreslået af Kappen et al. i en endnu upubliceret artikel. Algoritmen foretager to vigtige antagelser: problemet der skal løses er lineært, og det har en sparse løsning. Det sidstnævnte opnås ved inkludering af en binær parameter for hver inputvariabel i den lineære model, der bestemmer om en variabel er relevant eller ej. Løsningen findes via Bayesian inferens. Antagelserne gør potentielt VG velegnet til at løse det stærkt underbestemte inverse EEG problem. Hovedbidrag i denne afhandling inkluderer verifikation af VG i EEG sammenhænge og udvidelse af algoritmen til tidsdomænet. Publikationer af fund er indsendt til International Conference on Acoustics, Speech, and Signal Processing 2013 og IEEE International Winter Workshop of Brain-Computer Interface 2013. Algoritmens evner, som beskrevet af Kappen et al., blev bekræftet i et forstudie. Reformuleringer af VG problemet blev udført for at reducere beregningskompleksiteten ved hjælp af Kailath Variant relationen og en dual repræsentation. Begge blev sammenlignet med least absolute shrinkage and selection operator (LASSO) og med en sparse Bayesian model med lineær basis. En forward field matrix blev her brugt som input, mens kildefordelingen var syntetisk dannet. Den duale formulering af VG problemet blev fundet overlegen og blev udvidet fra den momentane formulering. Under antagelsen om at aktivitet i en kilde er til stede i et vist, muligvis kort, tidsrum, blev de enkelte kilders binære variabel modelleret til at være konstant tændt henholdsvis slukket inden for 20-25 tidssamples, svarende til 100 ms i EEG sammenhænge. Den udvidede tidsversion af den duale VG-formulering blev valideret via syntetisk data og via EEG data med det visuelle stimuli paradigme beskrevet af Henson et al. (2003). Den resulterende kilde-

fordeling var sammenlignelig med den fundet i litteraturen, både fra studier af responset målt med EEG og målt med andre billedmodaliteter. Det foreslås at udvide VG-algoritmen til brug af online tracking af hjerneaktivitet ved at reducere beregningskompleksiteten yderligere og inkludere spatial glathed.

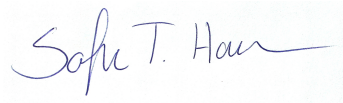
Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfillment of the requirements for acquiring an M.Sc. in Medicine and Technology.

The thesis includes machine learning tools applied to physiological processes. More specifically a Bayesian inferring model is examined with respect to its ability to recover brain activity. The performed experiments expand on the knowledge of sparsity enforcing algorithms applied to the ill-posed problem of locating the generators of the potential field measured at the scalp.

The project was carried out in the period from September 2012 to February 2013.

Lyngby, 11-February-2013

A handwritten signature in blue ink, reading "Sofie T. Hansen". The signature is fluid and cursive, with the first name "Sofie" and last name "Hansen" clearly legible.

Sofie Therese Hansen
s072331

Acknowledgements

I would like to thank my supervisors, Professor Lars Kai Hansen and Postdoc Carsten Stahlhut at Department of Informatics and Mathematic Modelling, for guidance, valuable discussions and motivating ideas for direction of study.

This thesis included experiments on several algorithms which were partly enabled by the open source MATLAB toolboxes of several authors. These include K. Sjöstrand with *SpaSM* [Sjö05], M. Tipping with *SparseBayes* [Tip09b] and the Distributed Representations group at Donders Institute for Brain, Cognition and Behaviour with *DMLT* [Dis12].

Finally I would like to thank Martin C. Axelsen for proofreading this thesis and to my fellow students in the project room for keeping up motivation.

Abbreviations

ARD	Automatic relevance determination
BCI	Brain computer interface
BEM	Boundary element method
COH	Coherence
DMLT	Donders machine learning toolbox
EEG	Electroencephalography
EP	Evoked potential
EPSP	Excitatory postsynaptic potential
ERP	Event-related potential
FEM	Finite element method
(f)MRI	(Functional) Magnetic resonance imaging

ICA	Independent component analysis
IPSP	Inhibitory postsynaptic potential
LARS	Least angle regression
LASSO	Least absolute shrinkage and selection operator
LORETA	Low resolution brain electromagnetic tomography
MCE	Minimum current estimate
MEG	Magnetoencephalography
MNE	Minimum norm estimate
(n)MSE	(Normalized) Mean squared error
MSP	Multiple sparse prior
OLS	Ordinary least squares
PET	Positron emission tomography
PSP	Postsynaptic potential
RVM	Relevance vector machine
SBL/M	Sparse Bayesian learning/model
SNR	Signal to noise ratio
SPM	Statistical Parametric Mapping
SVM	Support vector machine
VG	Variational Garrote
VG-dual	Dual formulation of variational Garrote
VG-KV	Kailath Variant formulation of variational Garrote

Nomenclature

$\text{diag}(\mathbf{v})$	Diagonal matrix with the vector \mathbf{v} in the diagonal
\mathbf{V}_{diag}	Vector of diagonal elements in matrix \mathbf{V}
$ a $	The absolute value of the scalar a
$E[a]$	Expected value of a
\bar{v}	Mean of \mathbf{v}
∇	Gradient operator
δ	Kronecker delta
\oslash	Element-wise division
\odot	Element-wise multiplication
\cdot^2	Element-wise square
σ^2	Variance
β	Precision

n	Number of sources/input variables; index i
p	Number of electrodes/output samples; index μ
T	Number of time samples; index t
\mathbf{y} (and \mathbf{Y})	Samples of response/EEG potentials in electrodes (across time)
\mathbf{w} (and \mathbf{W})	EEG sources (across time)
\mathbf{X}	Input/transposed forward field matrix
χ	Input covariance matrix
ξ	Noise component
F	Variational free energy
s_i	Binary switch in VG
m_i	Activation, probability of $s_i = 1$ in VG
γ	Sparsity level in VG
λ	Regularization parameter in LASSO
α_i	hyperparameter/weight decay in ARD and SBM
$\phi(\cdot)$	Basis function
F_s	Source retrieval index

Contents

Summary, English	i
Summary, Danish	iii
Preface	v
Acknowledgements	vii
Abbreviations	ix
Nomenclature	xi
1 Introduction	1
1.1 The Structural and Functional Brain	2
1.1.1 Macro structure and organization	2
1.1.2 Physiology of neurons	3
1.2 Principles of EEG	4
1.2.1 Generation of EEG signal	4
1.2.2 Recording EEG	6
1.2.3 Application of EEG	6
1.2.4 Example of EEG data	8
1.3 Motivation of Thesis	10
1.3.1 General considerations	10
1.3.2 The forward model	11
1.3.3 Head models	13
1.3.4 Linearity of the forward problem	14
1.3.5 Sparsity of the EEG sources	14

2	Linear Regression Theory	17
2.1	Ordinary Least Square	18
2.2	Least Absolute Shrinkage and Selection Operator	19
2.2.1	Performance of LASSO	21
2.3	Non-negative Garrote	23
2.4	Automatic Relevance Determination and Sparse Bayesian Models	24
2.4.1	Performance of a sparse Bayesian model	26
2.5	Variational Garrote	27
2.5.1	Performance of VG	31
2.5.2	Reformulation using Kailath Variant	34
2.5.3	Dual formulation	36
2.5.4	Time-expanded dual formulation	37
3	Experimental Design	43
3.1	Sparse Algorithms in Single Time	44
3.1.1	Synthetic data	44
3.1.2	Experiment 1.1: Stability in number of cross-validation folds	45
3.1.3	Experiment 1.2: Initialization of γ and \mathbf{m} in VG-dual	48
3.2	Time-expanded VG-dual	49
3.2.1	Multimodal face-evoked response data set	50
3.2.2	Experiment 2.1: Performance on synthetic data	51
3.2.3	Experiment 2.2: Performance on differential ERP	51
3.2.4	Experiment 2.3: Performance on single face epoch	52
4	Results	53
4.1	Sparse Algorithms in Single Time	53
4.1.1	Experiment 1.1: Stability in number of cross-validation folds	53
4.1.2	Experiment 1.2: Initialization of γ and \mathbf{m} in VG-dual	59
4.2	Time-expanded VG-dual	60
4.2.1	Experiment 2.1: Performance on synthetic data	60
4.2.2	Experiment 2.2: Performance on differential ERP	62
4.2.3	Experiment 2.3: Performance on single face epoch	66
5	Discussion	71
5.1	Sparse Algorithms in Single Time	71
5.2	Time-expanded VG-dual	75
5.3	General Reflections	78
6	Conclusion and Perspectives	81
A	Derivation of VG in Primal Space	83
B	Details of VG-code	93

C	Extensions to VG Kailath Variant Formulation	97
D	Extensions to Dual Formulation of VG	101
E	Extensions to Time-expanded VG-dual	105
F	Selected MATLAB Implementations	111
F.1	Two-level Cross-validation	111
F.2	Kailath Variant Formulation of VG	114
F.3	Time-expanded VG-dual	115
G	Submission to ICASSP2013	119
H	Submission to IEEE BCI2013	125
	Bibliography	129

CHAPTER 1

Introduction

In this chapter the general problems of investigating the functional brain are introduced. Electroencephalography (EEG) represents one of the earliest attempts of looking inside the brain. This technique exploits the electric fields generated by neuronal activity which are measurable at the scalp. Later additions to neuroimaging includes positron emission tomography (PET) and functional magnetic resonance imaging (fMRI), both of which use the hemodynamics of the brain as an expression of brain activity [HFT00]. PET and fMRI have higher spatial resolution than EEG, however EEG is superior in respect to time resolution. The goal of this thesis is to achieve online tracking of brain activity, i.e. to investigate time dependent conditions of the brain. EEG is therefore the most suitable modality for the current application. Further adding to its use in this setting is its portability and low cost compared to many other neuroimaging modalities.

Backtracing from the measured EEG potentials at the scalp to the actual generators in the brain is a highly underdetermined task. Research in this area is therefore intense and ongoing. Numerous mathematical models have been suggested, and many of these approach the problem by introducing sparseness into the solution [SHH⁺97, UHS99, DET06, SSL06, FHD⁺08, HNZ⁺08, HTD⁺11, DPO⁺12]. The aim of this thesis is to explore the sparsity-inducing algorithm; *variational Garrote* (VG), presented by Kappen et al. (2012) [KG12]. Especially its performance in solving the inverse EEG problem is investigated along with the possi-

bilities of using the algorithm for EEG biofeedback. VG is further described in section 2.5.

First an overview of the brain's organization is given, on a larger as well as on a smaller scale. Next the mechanisms behind EEG are explained and finally the difficulties and considerations of using the measured EEG to create a 3D reconstruction of the brain are reviewed.

1.1 The Structural and Functional Brain

The ability to make realistic interpretations of signal measured from the brain largely includes understanding the building blocks and mechanisms of the brain [NS06]. Misunderstandings of what produces e.g. EEG data is an obstacle in translating the signal and using it in clinical applications.

This section briefly explains the elements of the brain's composition which are important for the problem at hand. The further technical description of the generation and use of EEG signal is left for the next head section.

1.1.1 Macro structure and organization

The cerebrum is the most relevant part of the brain in EEG contexts. It consists most importantly of neurons, whose orientation and location of structural parts are what create the white and gray matter of the brain [NHW03]. The cell bodies, or somas, of the neurons are located in the gray matter. The gray matter is found at the surface of the brain, where it forms the cortex, and embedded in the white matter where it is known as the nuclei [NHW03]. The white matter consists of the neurons' myelinated axons. These carry information between neurons in related areas, to the brain stem or spinal cord, or to the other hemisphere. It is these approximately 10^{14} interconnections that make up 'intelligence' [NHW03, NS06].

The 10^{10} neurons of the cerebral cortex are among other things responsible for motor function, perception, visual function, language and communication between these brain processes [SST08]. The cerebrum is functionally divided into lobes, which are areas with specialized functions, see figure 1.1.

The nuclei of the brain include the thalamus (responsible for passing sensory information to the correct cortex area), the hypothalamus (known for monitoring

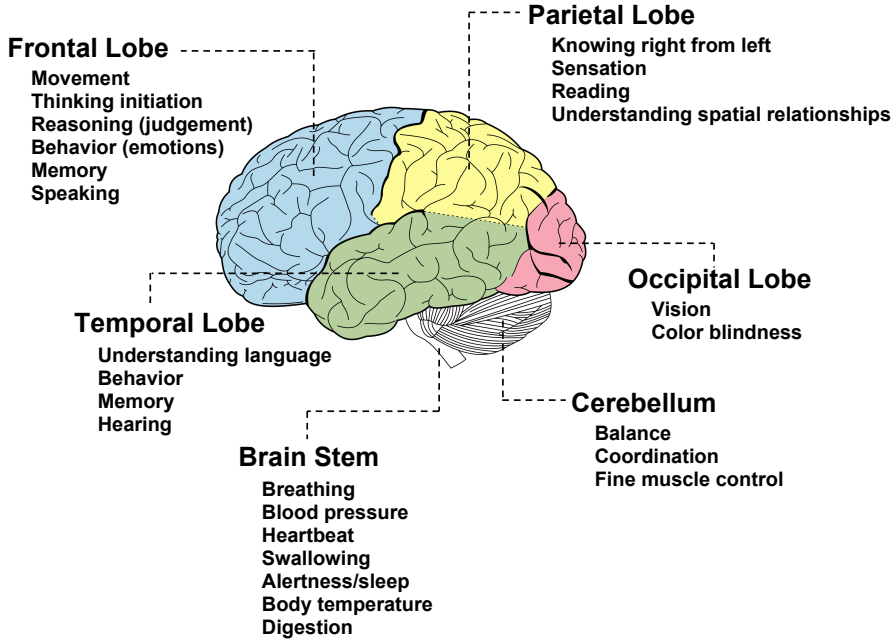


Figure 1.1: The division of the cerebrum into lobes is visualized together with the brain stem and cerebellum. The main functions of these structures are presented. Modified from [Gra06].

the water balance and temperature), and the basal ganglia [SST08]. The latter act in processing of voluntary movement and in many mental functions [SST08]. The basal ganglia includes the largest nucleus of the brain; the substantia nigra.

One of the most common diseases of the brain is Parkinson's disease [MWDD05]. The majority of the symptoms produced are caused by deterioration of the dopamine producing neurons of the substantia nigra [MWDD05, TT08]. As mentioned, this area plays a role in movement, and this function is thus affected by the degeneration [TT08]. The etiology is largely unknown and treatment options are sparse, indicating the complexity of the disease and of the brain.

1.1.2 Physiology of neurons

The typical neuron consists of a cell body from where an axon, or nerve fiber, along with several dendrites extend [SST08]. The dendrite is the place of signal reception from other neurons through their respective axons [HVG⁺07]. The

signal propagates along the axon as action potentials, and this at a speed of 15-100 m/s [SST08].

In the cerebrum a signal is transmitted between neurons through a chemical synapse. At the terminal of the axon, neurotransmitters are released following an action potential. When these neurotransmitters reach a dendrite of the postsynaptic neuron, ligand gated ion channels open [SST08]. This can either result in depolarization or hyperpolarization of the postsynaptic neuron, i.e. create an excitatory postsynaptic potential (EPSP) or an inhibitory postsynaptic potential (IPSP), respectively [HVG⁺07]. An IPSP will push the postsynaptic neuron further away from triggering an action potential, while the EPSP will do the opposite. The reaction of the neuron is a summation of all of the postsynaptic potentials (PSPs) it receives at its dendrites [HVG⁺07, SST08].

The pyramidal neurons are in EEG applications the most interesting [NS06, Tep02]. They have one axon but many dendrites. The dendrites of neighboring neurons in the cortex are highly aligned and perpendicular to the scalp [HVG⁺07]. The electric fields generated by the influx or efflux of ions at the dendrites are thus from a distance summed to create a field measurable with EEG [HVG⁺07]. Although the axons also carries ions, their generated potential difference is shorter in duration, 0.3 ms compared to 10-20 ms, and often not summable between neighboring axons as these are not aligned [HVG⁺07]. It is assumed that primarily the neurons in the cortex give rise to the electrical field measured by scalp electrodes, as potential fields generated at the nuclei are most likely too far away to be easily detected [BML01].

1.2 Principles of EEG

Measuring EEG is relatively simple. Understanding the origin of the signal and extracting information from it, is not as straightforward. This section introduces EEG with respect to how it is produced and how it is used.

1.2.1 Generation of EEG signal

The origin of EEG signal was briefly explained in the previous section and will be elaborated in the following.

Most signal recorded by EEG is the result of PSPs at cortical pyramidal neurons. The PSPs occur at the dendrites of the neuron and create sources or sinks,

depending on the nature of the PSP, i.e. inhibitory or excitatory, respectively. To conserve electrical neutrality a sink/source arises in a different place of the neuron [PM09]. An example where an EPSP depolarizes the neuron by letting in Na^+ , is seen in figure 1.2.

Compared to the distance from the cortex to the EEG electrodes at the scalp, the distance between the source and sink of the neuron is very small. The neuron is thus comparable to a current dipole [PM09]. The current illustrated by the fat arrow in figure 1.2 is called the primary current and runs inside the neuron. To close the circuit, extracellular cations flow along the membrane towards the sink of the dipole, as removal of cations from this area makes it less positive than the surroundings [NS06]. On the intracellular side of the membrane the current moves in the opposite direction (away from the sink) and exits at another part of the membrane (the source), and thus the current loop across the membrane is connected, creating an intra- as well as extracellular potential [NS06]. It is the latter that can be measured by EEG [HVG⁺07].

Predominately the neurons near the scalp produce a measurable signal. For deep dipoles to show up on EEG they have to be of high magnitude as the potential is

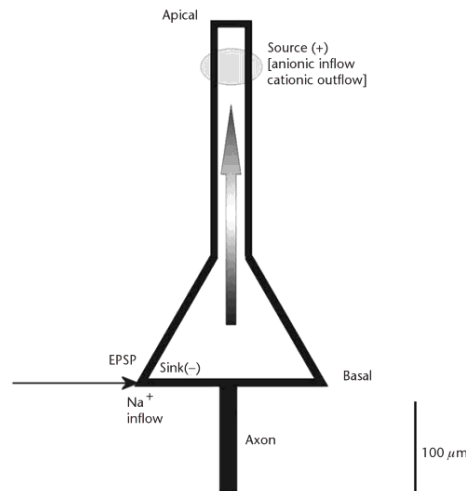


Figure 1.2: Current dipole representation of a cortical pyramidal neuron undergoing EPSP. The thin arrow indicates position of cation inflow, and thus the place of the sink. The fat arrow indicates the movement of cations from the sink towards the source, this current is called the primary current. Outside the neuron, current flows in the opposite direction. From [PM09].

inversely proportional to the squared distance, approximately [NS06]. However if many synchronous neurons are at play, this relation overestimates the effect distance has on the potential field.

1.2.2 Recording EEG

EEG can be recorded using scalp electrodes or intracranial electrodes. The latter is, as the name implies, a very invasive approach, and is mostly used to locate epileptic foci. The current application is the non-invasive scalp EEG and the description of the procedure in measuring EEG will thus be focused on this type of recording.

EEG contains potential differences measured between pairs of electrodes, which are fed into an amplifier that boosts the signal preparing it for analog-to-digital conversion [NS06, Tep02]. The common-mode potential, consisting mainly of artifacts from power lines, is sought removed [NS06]. Often all electrodes utilized for EEG registration are placed on the scalp, and by using the difference between their measured potentials non-brain produced scalp potentials are calculated out. One specific electrode can be chosen to work as reference for the other electrodes. The recording can later be re-referenced to another electrode or to an average reference [NS06]. A bipolar montage is also sometimes applied, here the potential difference is measured between two neighboring electrodes [NS06]. Furthermore non-scalp EEG electrodes can be used as reference [Tep02].

Additional processing of the recorded data is performed. This includes filtering to reduce artifacts e.g from blinking and employing band pass filters to remove noise [Tep02].

1.2.3 Application of EEG

The recorded EEG signal can be divided into spontaneous potentials and evoked potentials (EPs)/event-related potentials (ERPs)[NS06]. The spontaneous potentials are subdivided according to their frequency content; delta (1-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (13-20 Hz) and gamma (>20 Hz) bands [NS06]. The potentials generated by the neurons of the brain are usually a mix of these. EPs are the direct response to a specific stimulus while ERPs require a higher order of processing of the presented stimulus. To clearly see the EP/ERP, many repetitions of the stimuli are often necessary in order to facilitate an averaging out of the spontaneous potentials as well as to remove noise. The EP/ERP normally produces a waveform of a specific appearance, maximum 0.5 seconds

after presentation of stimulus [NS06]. The N170 complex is an example of a component found in the ERP specific to the face-evoked responses [HG^{GG}+03]. This ERP shows peak activity around 170 ms post-stimulus.

As scalp EEG is a non-invasive brain imaging technique with high temporal content, it is applied in several areas of research of human behaviour and cognition [NS06], often in the form of EP/ERPs. The spontaneous EEG data is widely utilized in diagnostics; e.g. in the area of epilepsy, head injury and sleep disorders, to name a few [NS06]. Alzheimer's disease is one example where research is performed to expand the knowledge of its pathophysiology and to enable earlier detection of the condition [NS06].

A newer advance in the application of EEG is neurofeedback. The goal is for the user/patient to train specific activation of a brain area by receiving information about the state of their brain [Tep02]. The information can consist of the frequency content of the potentials received at electrode level or of source reconstruction images. The former was done in a study of ADHD patients who trained their brain wave activity through EEG biofeedback [LHR96]. The results were an increase in intelligence functioning and attention ability [LHR96]. Symptom reduction in a patient with Parkinson's disease has also been described [TT08]. The focus was on increasing the activity of the sensor motor rhythm, which is seen decreased in Parkinson's patients [TT08].

Using 3D images as biofeedback, so far, poses a trade off between quality of the applied model and computation complexity. Achieving source reconstructed images of the brain online, thus restricts the method by computation time, as too big a delay between activation and reproduction of activation will confuse the user. Therefore often the minimum norm estimate (MNE) is applied, which has a closed form solution, but not optimal performance. Promising results was seen using a Bayesian MNE approach to perform source reconstruction in a study differentiating emotional responses [PSS11]. Here a delay of 150 ms was achieved. Making this study even more relevant to the focus of the current study, is the use of a wireless EEG headset synchronized with a smartphone, thus making the interface between user and system simple. This interface was introduced in [SLS11]. Initial research on real-time imaging in clinical applications has been performed [IHCL07]. It was found that distinctions in the cortical alpha rhythm between healthy subjects and persons with dementia were indeed detectable. A latency of 200 ms was reported using an MNE approach, and where the cortex was split into 1000 vertices and applied to 128 time samples.

Another real-time application of EEG source reconstruction is Brain Computer Interface (BCI) [BMG11]. In this context source reconstruction has been shown to outperform electrode level information in decoding mental imagery tasks [BMG11]. A combination of source and sensor level information has also shown

to improve results [AHJ12]. It is argued that the improvement is caused by a denoising of the sensor data by projecting it on to source space, thus making previously invisible information visible.

1.2.4 Example of EEG data

The multimodal face-evoked data set is a recording of subjects being presented with stimuli: either a face or a scrambled face [ACM⁺12]. The data is created as described by Henson et al. (2003), see phase 1 in [HGGG⁺03] for paradigm description. The intention is to reveal the difference between the human perception of a face versus an undefinable object. Also human recognition of faces was investigated by using familiar faces versus unfamiliar faces. The unfamiliar and familiar faces have however been collapsed in the present study. Furthermore the subjects in the study indicated by finger tapping the symmetry of the images.

The data set is interesting to work on as it has been analyzed multiple times and contains not only EEG data, but also magnetoencephalography (MEG) and fMRI data. This supplies a certain knowledge about what to expect, when applying a novel algorithm. As several EEG source localization experiments have been conducted on this data set, a qualitative comparison with the current experiment is thus possible. The data set is available through the website of Statistical Parametric Mapping (SPM): <http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/>, created by the Functional Imaging Laboratory (Fil), Wellcome department of Imaging Neuroscience, Institute of Neurology at University College London, UK and is described in the SPM8 manual [ACM⁺12].

Some of the relevant results reported in the SPM8 manual [ACM⁺12] are shown in figure 1.3. For the differential ERP (the average face response minus the average scrambled face response) the strongest source at time = 180 ms is found to be at $[-37, -80, -16]$ mm, and can be seen in figure 1.3a as the red trace. The N170 component is seen in this source. The inverse solution is found using a multiple sparse priors model (MSP) [FHD⁺08]. Activation was recovered in the occipito-temporal areas as well as in the fusiform gyri. Other studies have also indicated face-related activation in occipito-temporal areas [HGGG⁺03].

Findings of the fMRI experiment on the face-evoked response paradigm have shown increased hemodynamic activity in bilateral fusiform as well as in the right superior temporal cortices [HGGG⁺03]. MEG source reconstructions have found activity to be increased in the occipito-temporal cortex [SHH⁺97, LHH⁺91]. Applying MSP to MEG has shown activity in the fusiform gyri [HMPF09].

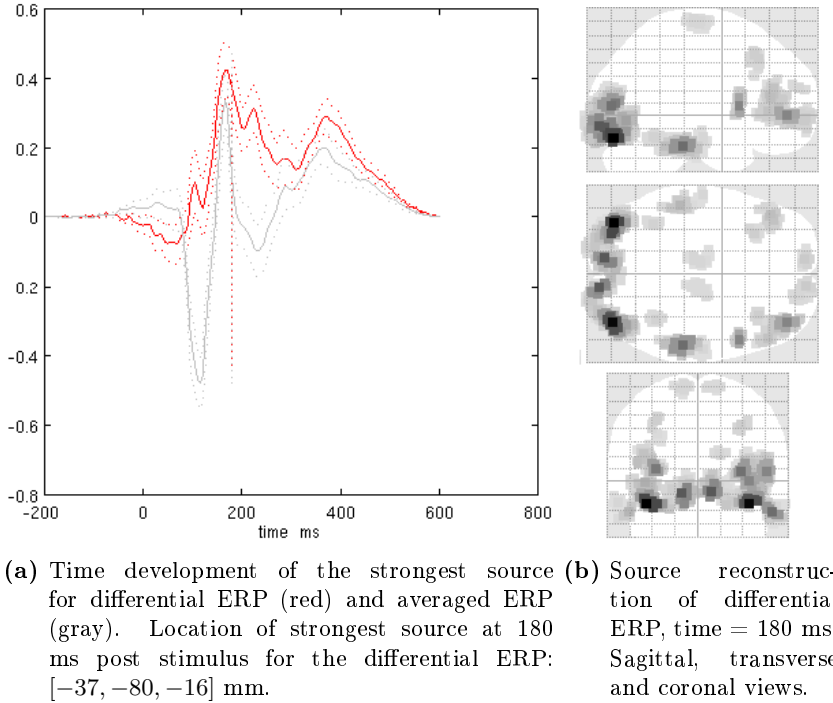


Figure 1.3: Results obtained on the multimodal face-evoked response data set with SPM8 using multiple sparse priors. The paradigm applied to reveal the face-evoked response is described by Henson et al. (2003). Modified from [ACM⁺12].

MSP has been seen to outperform the MNE and the coherence (COH) models [FHD⁺08]. MNE assumes that the sources are independent and identically distributed and imposes solutions with minimum total energy [HMPF09]. MNE has a closed form solution and therefore has low computational complexity, however it is prone to produce very diffuse and superficially located sources [HMPF09, MMHM12]. COH enforces smoothness in the solution, like low resolution brain electromagnetic tomography (LORETA), thus it assumes that if a specific source is active the neighbors probably are too. MSP is a method that combines assumptions of sparsity and smoothness, regularized by pruning priors; i.e. a few well-defined areas are presumed capable of explaining the data [FHD⁺08, HMPF09].

1.3 Motivation of Thesis

As already indicated the objective of this project is to obtain an efficient and precise method that solves the EEG inverse problem. The intention is that the source reconstruction is done in a way that makes it applicable in biofeedback settings, thus a fast algorithm is required. The VG algorithm is suggested to be the answer and is consequently the main focus. VG exploits Bayesian inference to obtain a sparse solution to a linear problem.

In this section some of the motives behind applying the algorithm, described by Kappen et al., on EEG data are presented. Emphasis is on clarifying that the forward problem is linear and that assuming sparsity in the number of EEG generators is reasonable. It also includes reviewing some of the difficulties faced when doing EEG source reconstruction and how VG can be a method in a direction of mending these obstacles.

1.3.1 General considerations

A huge number of neurons (10^{10}) are believed to be EEG generators and with only e.g. 128 electrodes, and often even less, EEG source reconstruction is a highly underdetermined problem. It is important to note that one neuron is not sufficient to generate a measurable electric field at the scalp. Bundles of synchronized neurons are therefore more accurately termed EEG generators. These have a size of 40-200 mm² [PM09].

Programs like SPM (Functional Imaging Laboratory, Wellcome department of Imaging Neuroscience, Institute of Neurology at University College London, UK) offer working with 5124, 8196 or 20484 EEG generators [ACM⁺12]. Assuming significantly fewer EEG generators than cortical neurons makes source reconstruction simpler and less computational heavy. However, obtaining the correct source representation is still a difficult task and an ideal solution has not yet been found.

Proving that an approach finds the correct source distribution is in itself difficult as the 'truth' is not fully known. Part of the solution to this is to either test algorithms on synthetic data or to work on multimodal data, where informed guesses on the source distribution can be made. The multimodal face-evoked data set presented in section 1.2.4 is an example of the latter. Here EEG, MEG and fMRI have been recorded under the same settings. It is expected that on a larger time scale some correlations are visible, as also demonstrated in [HGGG⁺03].

1.3.2 The forward model

The forward model relates the sources in the brain to the measured potentials at the scalp [HVG⁺07]. The problem is most often modeled as a volume conductor. The electric fields generated in the brain are instantaneous, as charge is not build up extracellularly; at least compared to the sampling frequency used in EEG [HVG⁺07]. The lack of time dependency in the electric fields is an important element in describing the relation in the forward model, as magnetic fields can be disregarded and thus facilitates application of Maxwell's quasi-static equations [HVG⁺07][NS06]. Poisson's equations can be derived via Maxwell's equations, but can also be obtained through the divergence operator as done by Hallez et al. (2007). The latter is recapped here, thus notation and equations are from [HVG⁺07] and in part from [KG12].

The divergence of the current density, which is described as the flux or current entering/leaving a small volume making the current negative/positive, respectively, is

$$\nabla \mathbf{J} = I_m, \quad (1.1)$$

where I_m is termed the current density source and can be divided into three cases. First, the case where the volume encases a small extracellular space. Here the flux leaving and entering the volume cancel out each other, thus making $\nabla \mathbf{J} = 0$. Secondly, the case where the volume surrounds a current sink, with the position \mathbf{r}_1 . This will cause current to leave the extracellular space, and is thus described as a negative current, $\nabla \mathbf{J} = -I\delta(\mathbf{r} - \mathbf{r}_1)$. The singularity is added to indicate that the sink is infinitesimally small. The opposite case describes the third case, where the sink is replaced by a source in \mathbf{r}_2 , thus making the current positive; $\nabla \mathbf{J} = I\delta(\mathbf{r} - \mathbf{r}_2)$. Combining these three examples the current source density is

$$\nabla \mathbf{J} = I\delta(\mathbf{r} - \mathbf{r}_2) - I\delta(\mathbf{r} - \mathbf{r}_1). \quad (1.2)$$

The goal is to relate a potential field measured at the scalp to the current sources in the brain. To achieve this Ohm's law is used

$$\mathbf{J} = \kappa \mathbf{E}, \quad (1.3)$$

where κ is conductivity, which can be modeled as being isotropic or more realistically, anisotropic. \mathbf{E} is the electric field and is related to the potential field Y by the gradient operator

$$\mathbf{E} = -\nabla Y. \quad (1.4)$$

Poisson's differential equation can now be presented combining equations (1.2), (1.3) and (1.4)

$$\nabla(\kappa\nabla Y) = I\delta(\mathbf{r} - \mathbf{r}_1) - I\delta(\mathbf{r} - \mathbf{r}_2). \quad (1.5)$$

The head can be modeled as having several layers. At each boundary of these layers, boundary conditions must be met. The Neumann boundary condition dictates that the current exiting one layer must enter another

$$\mathbf{J}_1\mathbf{e}_n = \mathbf{J}_2\mathbf{e}_n. \quad (1.6)$$

Here \mathbf{e}_n is the normal component on the boundary between the two compartments. Additionally, current can not leave the head through the air, as it has very low conductivity

$$\mathbf{J}_3\mathbf{e}_n = 0. \quad (1.7)$$

This restriction is called the homogeneous Neumann boundary condition. Furthermore the Dirichlet boundary condition is considered for potentials crossing an interface inside the head, and is

$$Y_1 = Y_2. \quad (1.8)$$

Each dipole \mathbf{d} at position \mathbf{r}_{dip} in the brain affects the potential measured by an electrode at the scalp in position \mathbf{r} by $g(\mathbf{r}, \mathbf{r}_{dip}, \mathbf{d})$. The potentials caused by the dipoles are summable

$$Y(\mathbf{r}) = \sum_{i=1}^p g(\mathbf{r}, \mathbf{r}_{dip}, \mathbf{d}) = \sum_{i=1}^p g(\mathbf{r}, \mathbf{r}_{dip}, \mathbf{e}_d)w_i, \quad (1.9)$$

where \mathbf{e}_d is the orientation of \mathbf{d} and w its magnitude. Equation (1.9) is true for all p potentials measured at the scalp and can be written as a set of equations or on matrix form. Assuming the neurons are oriented perpendicular to the surface of the cortex the matrix relation is

$$\mathbf{Y}_{p \times T} = (\mathbf{X}_{n \times p})^T \mathbf{w}_{n \times T} + \boldsymbol{\xi}_{p \times T}, \quad (1.10)$$

where the dimension of time has been added. Equation (1.10) represents the forward problem with dimensions noted in subscript. Note that noise $\boldsymbol{\xi}$ has been added. The matrix \mathbf{X}^T is termed the *gain matrix*. A column of this matrix describes a source's contribution to each electrode and is called the forward field. A row relates the potential of one electrode to all of the sources and is called the lead field. Thus *lead field matrix* and *forward field matrix* are used interchangeable with gain matrix. Solving the forward problem is equivalent to solving Poisson's equations, where finding the potentials is the objective. The solution to the forward problem depends on the chosen head model.

1.3.3 Head models

One problem of locating the EEG sources is the geometry and structure of the head. The brain is protected by the five layers of the scalp as well as by the skull, cerebrospinal fluid and the three meningeal layers [NHW03]. Each layer has different conductive characteristics, as does the brain itself, this makes the solution to the forward problem difficult. Especially sources on the interfaces between layers, thus including the sources at the outer rim of the cortex, are prone to be estimated with error [GPOC11].

The three-shell spherical model is a simplification of the electrical properties of the head. Here the head is modeled as three nested spheres acting as the scalp, skull and brain [HVG⁺07]. Each sphere is isotropic and homogeneous, which of course are very crude assumptions [BML01]. The white matter is highly anisotropic as the conductivity on a current flux in the direction of the axons is much higher than it is in an angle [NS06]. One way to reveal the organization is to use diffusion tensor magnetic resonance imaging (MRI), as it is expected that the measured water diffusion tensor is strongly related to electric conductivity tensor [WAT⁺06].

More realistic head models use MRI scans, which also solve the problem of variations between individuals; that is if individual head scans are performed [BML01]. If structural information from MRI scans are not available Akalin Acar et al. (2013) argue that it is crucial to describe the head geometry and conductivity, and the electrode placement as accurately as possible [AM13]. The boundary element method (BEM) is an approach to solve Poisson's equation that can use the interfaces of the head found by MRI scans [BML01]. The method however still assumes isotropic and homogeneous conductivity in the implemented head compartments. The symmetric BEM is an improved version of BEM. It enhances the capability of locating dipoles near interfaces between compartments with high conductivity ratios [HVG⁺07]. BEM calculates the potentials only on each interface as opposed to e.g. the finite element method (FEM). BEM thus has fewer unknowns to find, and thereby computational cost is reduced [KCA⁺05]. FEM however has the ability to model the conductivities of each compartment as being anisotropic [BML01]. As computational complexity is reduced by advances in mathematical modeling of FEM, the method becomes more desirable, as it facilitates more accurate models of the organization of the brain [WGH04, SSJ⁺10].

The VG algorithm can basically be giving any head model, however its solution is expected to be dependent on the choice. Therefore not to introduce errors in the input, a head model which approaches the actual electrical properties of the head is preferred.

1.3.4 Linearity of the forward problem

As mentioned the space between the EEG source and the EEG electrode contains many different layers, many of which are inhomogeneous and have varying conductivities. Experiments have however shown that the tissue of the brain can be modeled as a linear conductor, meaning that superposition of sources is possible [NS06]. This means that the potential difference measured by an electrode set can be assumed to originate from a sum of sources inside the brain, as also seen in equations (1.9) and (1.10). Note that specific weights are placed on each of these sources, determined by their location relative to the scalp electrode.

The VG algorithm assumes the problem at hand is linear by having the linear regression problem at its core. In the aspect of linearity, VG is therefore an appropriate candidate to solve the EEG inverse problem.

1.3.5 Sparsity of the EEG sources

Including the attribute of sparsity in a solution has been done in many machine learning tasks, see [MJOB10] for references. It has the clear advantage of making the solution easier to interpret especially when there are many variables, as in EEG. Additionally using a solution which is capable of describing the data and has high sparsity is often found to be the correct solution to overcomplete systems [DET06].

Single dipole fitting [SB91] assumes one dipole can explain the main part of the measured signal [PM09]. Of course this is very simplistic but does perform well in certain settings. This includes when one area of the brain is responsible for a strong dipole moment, as in locating epileptic foci [PM09]. The problem with dipole fitting is that the number of dipoles must be known beforehand and of course that the maximum allowable number of dipoles could be too low.

Several other EEG source reconstruction techniques attempt to solve the problem by assuming some degree of sparsity. MSP is one example [HMPF09]. As mentioned MSP has been found to be better at explaining the data compared to more diffuse methods such as MNE [FHD⁺08]. In the application of e.g. independent component analysis (ICA) only the strongest dipoles are investigated further [DPO⁺12]. It is in [DPO⁺12] assumed that a few strong sources can explain the measured signal and that they are independent in time. It has however been shown that functional connectivity between brain regions exist in e.g. processing of stimuli [VSSBLC⁺05, Fri94, HT10]. It is thus more cautiously to model the time series of sources simultaneously.

Sparsity can also be employed via regularization with the L_1 -norm as done in minimum current estimate (MCE) [UHS99]. However, this approach is liable to produce a scattered distribution of dipoles around the true source [HTD⁺11]. Adding the L_2 -norm in the regularization (in combination with L_1 -norm), enforces smoothness and has shown promising results [HNZ⁺08][MMHM12].

In the BCI setting, reduction of the number of sources used to describe the measured EEG signal is also sometimes enforced [BMG08]. The goal is also here to make the solutions more interpretable without losing accuracy of the model. Univariate and multivariate variable selection are examples of methods that reduce the solution space [BMG08].

Whether a sparse prior is prudent or not depend on whether the actual source distribution is indeed sparse or not. The performance of the VG algorithm, which enforces sparsity by turning sources on or off, is thus dependent on the physiology of the neuronal interactions. Even though many distributed sources should be present, imposing sparsity is still reasonable if the algorithm finds the most relevant and/or dominating sources.

CHAPTER 2

Linear Regression Theory

The linear regression problem is relevant in many applications. It is used because of its fundamental simplicity and flexibility, i.e. by using different basis functions variable complexity can be added [Bis06]. The description of the linear regression problem and methods of solving it is described below. The symbol notation follows mostly that of [KG12].

The linear regression problem is in its simplest form

$$y_\mu = \sum_{i=1}^n w_i X_{i\mu} + w_0 + \xi_\mu, \quad (2.1)$$

where y_μ is a one of the p responses, $X_{i\mu}$ indexes the n input variables for each sample μ , and ξ_μ is zero-mean noise with inverse variance (precision) β . Finally w_i is one of the n weights and w_0 is a bias.

Relations to the EEG problem:

- \mathbf{y} contains electrodes as samples, i.e. μ indexes over p electrodes.
- \mathbf{X} is the transpose of the forward field matrix, seen in section 1.3.2, equation (1.10).

- \mathbf{w} contains the magnitudes of the n sources (EEG generators).

Presented below are selected solutions to the linear regression problem.

2.1 Ordinary Least Square

The linear regression problem can be solved by minimizing an error function consisting of the sum of squares difference between the target and the predictions made by the model

$$\frac{1}{2} \sum_{\mu=1}^p \left(\sum_{i=1}^n w_i X_{i\mu} - y_{\mu} \right)^2. \quad (2.2)$$

The solution to the above can be solved analytically

$$\begin{aligned} \mathbf{w} &= \boldsymbol{\chi}^{-1} \mathbf{b}, \quad \text{where} \\ w_0 &= \bar{y} - \sum_{i=1}^n w_i \bar{\mathbf{X}}_{i:}, \end{aligned} \quad (2.3)$$

where $\chi_{ij} = \frac{1}{p} \sum_{\mu=1}^p X_{i\mu} X_{j\mu}$ and $b_i = \frac{1}{p} \sum_{\mu=1}^p X_{i\mu} y_{\mu}$. Additionally $\bar{\mathbf{X}}_{i:}$ and \bar{y} are the mean values of $\mathbf{X}_{i:}$ and \mathbf{y} , respectively. By centralization of the data, these are equal to zero. This is assumed done from now on.

The solution in equation (2.3) is called the *ordinary least squares* (OLS) solution. Its solutions are prone to overfitting, especially if the number of samples is smaller than the number of input dimensions [Bis06]. In this situation the model can describe the training set exactly but will not perform well on a test set as noise will probably be falsely modeled as signal. Non-sparse solutions are furthermore difficult to interpret [Tib96]. Remedies are e.g. subset selection, where weights are discarded or kept, this however gives an unstable model. Additionally ridge regression can be applied, which shrinks the coefficients using L_2 -norm regularization. This does not make the solution sparse and overfitting is still a problem. [Tib96].

OLS assumes that the error terms are uncorrelated, which might not be the case, and thus induce further errors [DW50]. Also note that when the dimension n is larger than the number of samples p , $\boldsymbol{\chi}$ is singular and can therefore not be inverted, instead the pseudo-inverse must be applied.

2.2 Least Absolute Shrinkage and Selection Operator

The least absolute shrinkage and selection operator (LASSO) technique [Tib96] revises the solution to the linear regression problem by setting some weights to 0 and shrinking others. This is done using the following linear restriction to the OLS

$$\sum_{i=1}^n |w_i| \leq t, \quad (2.4)$$

i.e. the L_1 -norm is applied to the weights. The size of t determines the degree of sparseness introduced in the model [Tib96]. A scaling of t by the sum of the absolute values of the weights in the OLS solution is termed s ; $s = t/t_0$, where $t_0 = \sum_{i=1}^n |w_i^{OLS}|$ and \mathbf{w}^{OLS} is found using OLS. Setting $t = t_0/2$ broadly corresponds to creating a feature subset of size $n/2$ by the shrinkage and removal of weights [Tib96].

The constraint region created by the L_1 -norm is a rotated square for two weights, and a (hyper-)cube for higher dimensions, centered at origin [Bis06]. The probability of the least squares solution hitting one of this quantity's corners, i.e. 0, is higher than hitting 0 using the circular L_2 -norm constraint region. This explains why the LASSO algorithm finds more weights equal to 0 compared to ridge regression, that only shrinks the parameters. Note that in EEG source reconstruction settings using LASSO corresponds to MCE and ridge regression to MNE.

The following is defined to be the LASSO problem [Tib96]

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{\mu=1}^p (y_{\mu} - \mathbf{w}^T \mathbf{X}_{\cdot\mu})^2 \quad \text{subject to} \quad \sum_{i=1}^n |w_i| \leq t. \quad (2.5)$$

In addition to centering of the data, the rows of \mathbf{X} are also further scaled; $\sum_{\mu=1}^p X_{i\mu}^2/p = 1$. The same solution as found above for a value of t can be found for a value of λ in the following

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{\mu=1}^p (y_{\mu} - \mathbf{w}^T \mathbf{X}_{\cdot\mu})^2 + \lambda \sum_{i=1}^n |w_i|. \quad (2.6)$$

The above can be solved for a range of values of λ , and e.g. optimized by cross-validation. Using least angle regression (LARS) [EHJT04] the problem is solved computationally efficient for $\lambda \in [0, \infty[$ [TT11]. LARS compares to

forward selection, where the variable with highest correlation to the response, \mathbf{y} , is chosen to describe the response entirely, this leaves a residual [EHJT04, HTTW07]. The remaining variables are projected orthogonally to the found variable and the correlation between them and the residual, determines the next variable to be added, and so on [EHJT04]. This method is fast but will be overly greedy for applications with highly correlated variables. The forward stagewise linear regression is an alternative which uses many small steps to build a model with increasing involvement of predictor variables [EHJT04]. The algorithm also finds the variable with highest correlation to the residual (=response in step 1), however this chosen variable's involvement is only incremented to a small degree at each step it obtains highest correlation [HTTW07].

In between these two algorithms LARS is found. At each step the variable with highest correlation to the residual (=response in step 1) is incremented towards its least squares solution until a second variable becomes more correlated with the residual. This is done in one step as opposed to several in forward stagewise regression [EHJT04]. LARS then move in an equiangular direction between the chosen variables until another variable enters the active set, and so on [EHJT04]. This explain the name least angle regression, as the algorithm moves in the direction which has smallest angle between the residual and the variables. LARS has a closed form solution to find the step size needed and is therefore very efficient. Note that one non-zero weight is added at each step, yielding n -steps in the algorithm. A modification must be added in order to make it a 'real' LASSO solution, i.e. the option of dropping a variable in an iteration is necessary, thus yielding more iterations than the pure LARS algorithm [EHJT04, HTTW07]. LARS has similarities with the also piece-wise linear path homotopy approach suggested in [OPT00a].

The MATLAB (The MathWorks Inc.) toolbox *SpaSM* created by Sjöstrand [Sjö05] implements the LASSO algorithm using the adjusted LARS technique. More specifically the function `lasso` calculates the parameter values in a window of regularization values corresponding to all weights being set to 0 (high λ) to applying no regularization ($\lambda = 0$). The latter results in the same solution as the OLS. The regularization terms outputted include λ and s . The latter is calculated straightforward as previously described, while λ is approximated to the median of

$$\lambda = 2|\mathbf{X}_{\mathbf{a}}\mathbf{r}|, \quad (2.7)$$

where \mathbf{a} points to the active set; i.e. contains the indices where \mathbf{w} have values different from 0 [SCLE]. The residual between the output \mathbf{y} and the estimated output $\mathbf{X}^T\mathbf{w}$ is denoted \mathbf{r} . Expression (2.7) is found by decomposing the weights into a positive and negative part and then applying the Karush-Kuhn-Tucker

conditions, after which it is realized that

$$\lambda = \left| \frac{\partial}{\partial \mathbf{w}} (\mathbf{y} - \mathbf{X}_{\mathbf{a}}^T \mathbf{w}_{\mathbf{a}})^2 \right| = 2 |\mathbf{X}_{\mathbf{a}}: (\mathbf{X}_{\mathbf{a}}^T \mathbf{w}_{\mathbf{a}} - \mathbf{y})| = 2 |\mathbf{X}_{\mathbf{a}}: \mathbf{r}|. \quad (2.8)$$

Sjöstrand et al. (preprint) explain that any of the λ s in the above could be chosen, i.e. any of the variables in the active set could be used to calculate the sought regularization λ , however they choose the median to avoid numerical problems.

By setting some weights to zero and shrinking others, LASSO combines the assumptions warranting subset selection and ridge regression [OPT00b]. However LASSO is criticized for low prediction power when the inputs are highly correlated [FHT10, KG12]. More specifically it has been shown by [ZY07] that under certain conditions if a predictor, not included in the true model, is highly correlated with the true predictors, LASSO will include the non-descriptive variable in the set, and this no matter how many samples are added.

Extensions of the LASSO algorithm includes group LASSO [YL05], which uses a penalizing function intermediate to regularizing the linear regression problem through the L_1 - and L_2 -norm.

2.2.1 Performance of LASSO

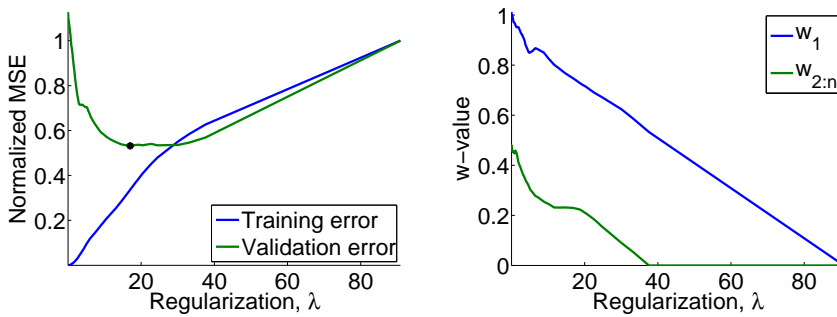
The LASSO algorithm is tested on the data described in [KG12] example 1, also outlined in appendix B. In brief the data is generated by having all weights set to 0 except for the first which is given the value 1. The before mentioned MATLAB toolbox SpaSM [Sjö05] created by Sjöstrand is utilized to find the LASSO solutions. 83 steps of regularization was found to create solutions from no active variables to all variables active.

In figure 2.1a the normalized mean squared training and validation errors are shown as function of applied regularization. The normalized mean squared error (nMSE) is calculated by

$$\text{nMSE} = \frac{E [(\mathbf{X}^T \mathbf{w} - \mathbf{y})^2]}{\sigma_y^2}, \quad (2.9)$$

where E denotes the expectation.

The OLS solution corresponds, as explained, to a regularization λ of 0 and, as expected, the validation error is high here, while the training error is minimal. This is clearly the result of overfitting. In the neighboring figure, figure 2.1b,



(a) Normalized mean squared error for training and validation set. Black dot indicates position of minimum validation MSE. The solution to the corresponding regularization is seen in figure 2.2.

(b) Feature values; w_1 should optimally be 1, $w_{2:n}$ represents the magnitude of the variable from the non-active set with highest absolute value. This value should be 0.

Figure 2.1: LASSO solutions for increasing regularization. The adjusted LARS algorithm is applied, yielding in total 83 levels of regularization. Data set inspired by example 1 in [KG12].

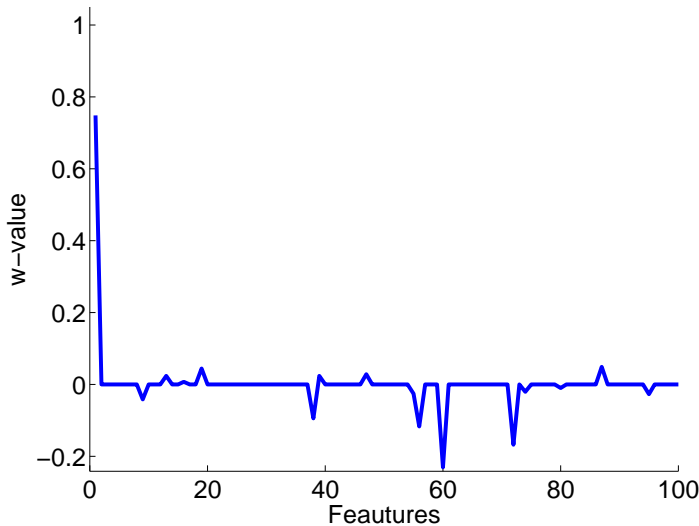


Figure 2.2: Optimum LASSO solution; weight distribution that give lowest validation error, see figure 2.1a. Optimum level of regularization: $\lambda = 16.7$. Data set inspired by example 1 in [KG12].

it can be seen that w_1 starts out by being 1, but it is affected by the regularization and shrunk. The shrinkage of the other weights is however greater and their values fall to 0 at a smaller regularization. The values of the weights obtained by using the solution with smallest validation error is seen in figure 2.2. The algorithm is successful in finding the first weight as being the most dominant. However the weight is given smaller magnitude than 1 and non-predicting variables are given value.

2.3 Non-negative Garrote

The idea for the LASSO algorithm arose from Breiman's non-negative Garrote [Bre95]. The two methods appeared about the same time, but with the non-negative Garrote slightly earlier. The algorithm got its name from an execution device inflicting strangulation, perhaps to emphasize its shrinkage and elimination properties.

Non-negative Garrote achieves sparseness and shrinkage by introducing the new non-negative variable \mathbf{s} . The problem to be minimized is defined as

$$\sum_{\mu=1}^p \left(y_{\mu} - \sum_{i=1}^n X_{i\mu} w_i^{OLS} s_i \right)^2 \quad \text{subject to } s_i \geq 0 \text{ and } \sum_{i=1}^n s_i \leq t. \quad (2.10)$$

The parameter \mathbf{s} thus controls the weights by enforcing shrinkage as t decreases [Bre95, Tib96].

Breiman showed that the non-negative Garrote is more stable to perturbations in applied data than subset regression, and that in settings where the ratio of actual predicting variables to total number of variables is not too big its accuracy is at the level of ridge regression.

As seen from equation (2.10) the solution to the non-negative Garrote is dependent on the OLS solution. This can cause problems if overfitting is present in the initial OLS solution and in cases where the inverse of the input covariance matrix is not computable (when $p < n$) [Tib96]. The latter problem can be solved using the pseudo-inverse, as also mentioned earlier. However, large overfitting can potentially harm the non-negative Garrote solution. Errors in the OLS from correlated error terms on the inputs will also affect the non-negative Garrote solution. However, where LASSO can give inconsistent results if non-predicting variables are correlated with the predicting variables, non-negative Garrote is less sensitive [Zou06]. The adaptive LASSO [Zou06], presented by Zou et al. (2006), is a LASSO variant that includes a weighting of the variables

in the penalizing function. This method can under specific settings be shown to be closely related to the non-negative Garrote [Zou06]. Extensions to the non-negative Garrote have, like for LASSO, been suggested, see e.g. [YL05] and [CFR11].

2.4 Automatic Relevance Determination and Sparse Bayesian Models

The automatic relevance determination (ARD) model [HR94, MN94, Nea95] also includes sparseness in its solution. In ARD each input variable is coupled with a specific regularization, controlled by a hyperparameter. This forces the irrelevant input variables to zero and retains the relevant.

It was originally used in neural network models, where one input variable is associated with several weights, each of these regulated by the same hyperparameter [Nea95]. The variables have a Gaussian prior with zero-mean and individual standard deviations, directly defined by the hyperparameter. A small standard deviation indicates little relevance in the model while large standard deviation indicates large relevance [Nea95].

The use of the ARD prior has been expanded to applications of linear regression problems [Nea95, Tip09a]. In general a linear model has the form

$$y_\mu = \sum_{i=1}^n w_i \phi_i(\mathbf{X}_{:\mu}), \quad (2.11)$$

where $\phi_i(\cdot)$ is a basis function [Bis06]. In the current employment, linearity is assumed between the input and output and the basis functions are therefore of the form $\phi_i(\mathbf{X}_{:\mu}) = X_{i\mu}$. Thus translating equation (2.11) into the linear model described in equation (2.1).

The weight prior is

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^n \mathcal{N}(w_i|0, \alpha_i^{-1}), \quad (2.12)$$

thus explaining that if α_i is large the variance (and standard deviation) of weight w_i is small thus making the probability of $w_i = 0$ high. Models which include this kind of prior and using Bayesian inference are termed *sparse Bayesian models* (SBMs) [Tip09b], also *sparse Bayesian learning*, SBL for short, is often used [WRP⁺07]. The relevance vector machine (RVM) [Tip00] is an example of

an SBM, here the basis function in equation (2.11) is a kernel thus approaching the support vector machine (SVM) technique [Tip01].

The response measured is assumed to be affected by noise with zero-mean and variance σ^2 . So in addition to the $p + 1$ hyperparameters, the variance, often expressed as the precision $\beta = \sigma^{-2}$, of the data also needs to be estimated. Gamma distributions can be used as priors for these [Tip01].

In SBMs the goal is to predict a response given an input vector and at the same time say something about the confidence of the predicting model. The latter is what separates e.g. SVM from RVM. This is however no simple task as the posterior of the unknowns \mathbf{w} , $\boldsymbol{\alpha}$ and β

$$p(\mathbf{w}, \boldsymbol{\alpha}, \beta^{-1} | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{w}, \boldsymbol{\alpha}, \beta^{-1}) p(\mathbf{w}, \boldsymbol{\alpha}, \beta^{-1})}{p(\mathbf{y})} \quad (2.13)$$

is not computable. Therefore SBM employs tricks such as using type-2 maximum likelihood, also known as evidence maximization, to estimate the values of $\boldsymbol{\alpha}$ and β and reiterates to find the optimum solution of these parameters [Tip01, Bis06].

The computation cost of SBM is very high in its most simple implementation, especially for large data sets. A more efficient implementation is suggested by Tipping et al. (2003). The proposed sequential method starts by having all weights pruned and then add (or delete) one at time until convergence occurs [TF03]. A 'refined' edition of this approach is implemented in the MATLAB toolbox *SparseBayes* Version 2 created by Tipping [Tip09b].

Wipf et al. (2007) reviewed the ARD framework in [WRP⁺07]. They concluded, among other things, that ARD is robust to the normalization procedure of the input matrix. Additionally it is explained that the model shows best results when the sources are uncorrelated. Here the algorithm will converge towards the global minimum when increasing the sample size. Trujillo-Barreto et al. (2004) employ many models with different priors and exploits their posterior probability to construct weights that dictate their influence in the final model. This is termed *Bayesian model averaging* and is presented in [TBAVVS04]. The technique is found to increase the ability of finding deep sources, such as activity in the thalamus. Additionally fewer ghost sources was found, i.e. non-predicting variables given activity.

Additions to the SBM framework have been presented by e.g. Zhang et al. (2011) in [ZR11]. Assuming temporal correlation a model is build which enforces block sparsity (identical to row sparsity) but at the same time exploits that an input variable at an instance in time is coupled with the samples obtained within

a certain time frame. This is often applicable to e.g. EEG source distribution. In this type of data Zhang et al. (2011) achieved superior performance to algorithms not including temporal correlation.

2.4.1 Performance of a sparse Bayesian model

The MATLAB toolbox SparseBayes Version 2.0 [Tip09b] created by Tipping is modified to evaluate an SBM model with a Gaussian likelihood model and linear basis. The same data set as used to explore the LASSO model is applied to the SBM model. To make the comparison between the algorithms discussed in this chapter fair, only one parameter is optimized. For SBM, α is estimated while β , the precision of the noise in the data, is chosen through a validation set.

The normalized mean squared training and validation errors are visualized in figure 2.3a for the range of β values: 0.01 to 1, with a total of 50 steps and maximum 100 iterations for each β value are applied.

It is apparent from figure 2.3b that SBM has a range of values until approximately $\beta = 0.1$ where all the non-predicting variables are set to zero. The

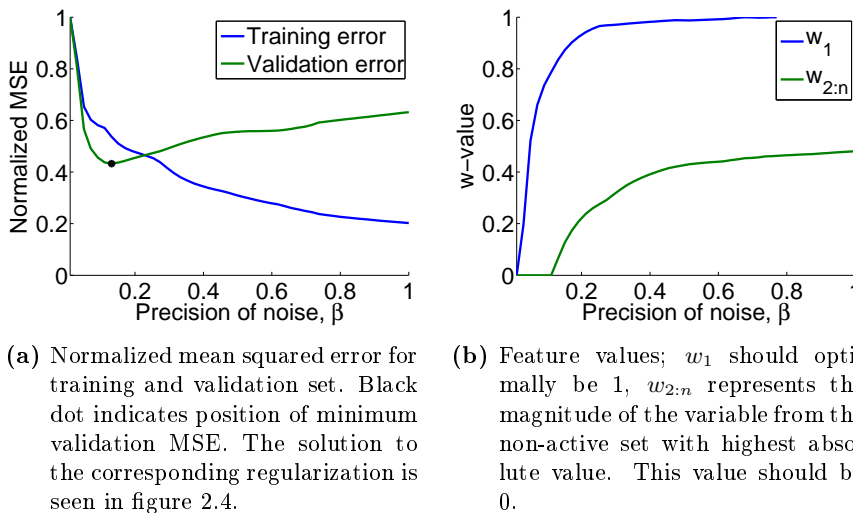


Figure 2.3: SBM solutions for increasing precision of noise β . The β values are investigated in the range from 0.01 to 1, with 50 incrementing steps each with maximum 100 iterations. Data set inspired by example 1 in [KG12].

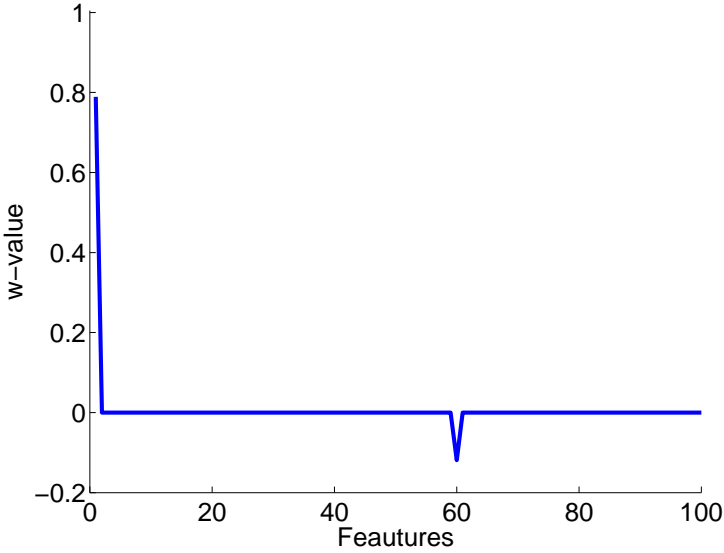


Figure 2.4: Optimum SBM solution; weight distribution that give lowest validation error, see figure 2.3a. Optimum level of precision: $\beta = 0.15$. Data set inspired by example 1 in [KG12].

predicting variable w_1 however does not entirely reach the desired value of 1 in this region.

The SBM solution chosen by the validation set, indicated by a black dot in figure 2.3a, is seen in figure 2.4. In this figure it is clarified that, like LASSO, SBM estimates the predicting variable's value to be smaller than the 'truth'. SBM does however only give one non-predicting variable relevance. Interestingly this variable is the same as found as the strongest non-predicting variable using LASSO. This non-predicting variable could therefore be speculated to have a noise component, both LASSO and SBM are partial to.

2.5 Variational Garrote

Another model which is close in appearance to the non-negative Garrote is the variational Garrote (VG) [KG12] suggested by Kappen et al. in a yet unpublished article. In this approach sparseness is introduced into the regression

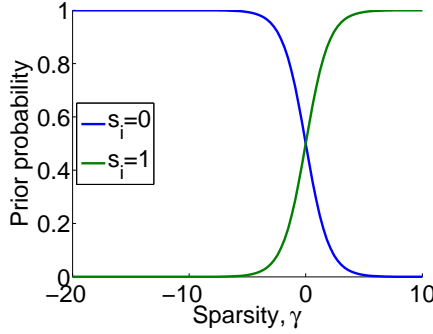


Figure 2.5: Prior probability of the binary switch s_i in VG.

problem by adding the variable \mathbf{s} . The problem is now defined as

$$y_\mu = \sum_{i=1}^n w_i s_i X_{i\mu} + \xi_\mu, \quad (2.14)$$

where s_i is either 0 or 1 and its prior is

$$p(\mathbf{s}|\gamma) = \prod_{i=1}^n p(s_i|\gamma), \quad (2.15)$$

where

$$p(s_i|\gamma) = \frac{\exp(\gamma s_i)}{1 + \exp(\gamma)}. \quad (2.16)$$

When γ is very negative, s_i is very likely to be 0. At $\gamma = 0$ the probabilities of s_i being 0 or 1 are of equal sizes, see figure 2.5. By introducing sparseness in to the likelihood, the problem is no longer convex, as opposed to the LASSO problem where the L_1 -norm guarantees convexity [KG12]. A local optimum might therefore be the result of the VG algorithm. However, Kappen et al. showed that VG performs better than LASSO, and than ridge regression on highly correlated inputs.

As also done in SBM, VG is solved using Bayesian inference. Kappen et al. suggest finding the optimum solution to the problem described in equation (2.14) by variational approximation. First the posterior probability of the model given the data is defined

$$p(\mathbf{s}, \mathbf{w}, \beta | \mathbf{D}, \gamma) = \frac{p(\mathbf{w}, \beta) p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta)}{p(\mathbf{D}|\gamma)}, \quad (2.17)$$

with D being the full data set. Instead of maximizing the posterior probability in equation (2.17)¹, the discrete variable \mathbf{s} is marginalized out, giving rise to the marginal posterior, $p(\mathbf{w}, \beta | \mathbf{D}, \gamma)$. This expression is to be optimized with respect to the parameters \mathbf{w} and β . The denominator in equation (2.17) does not depend on the two latter variables and is therefore not relevant in the maximization. Furthermore defining the joint prior likelihood of \mathbf{w} and β to be uniform, simplifies the problem additionally. The resulting expression to maximize is now

$$\log p(\mathbf{w}, \beta | \mathbf{D}, \gamma) \propto \log \sum_{\mathbf{s}} p(\mathbf{s} | \gamma) p(D | \mathbf{s}, \mathbf{w}, \beta), \quad (2.18)$$

where the logarithm operation has been added in order to make further derivations simpler. Equation (2.18) is difficult to maximize, by setting the differential coefficient equal to 0, due to the sum inside the logarithm expression. Therefore Jensen's inequality is applied. This approach can be used because the logarithmic function is a concave function [Bis06]. Concavity implies that on a chord to the concave function every point's value is smaller than that of the function. A point on a chord which has contact points with the concave function $f(x)$ in $(x_1, f(x_1))$ and $(x_2, f(x_2))$ can be described by $(\theta x_1 + (1 - \theta)x_2, \theta f(x_1) + (1 - \theta)f(x_2))$, where $\theta \in [0, 1]$. Due to concavity the following is thus true

$$f(\theta x_1 + (1 - \theta)x_2) \geq \theta f(x_1) + (1 - \theta)f(x_2). \quad (2.19)$$

The above can be rewritten to

$$f(\theta_1 x_1 + \theta_2 x_2) \geq \theta_1 f(x_1) + \theta_2 f(x_2), \quad (2.20)$$

where $\theta_1 + \theta_2 = 1$. By induction, the above can be extended to

$$f\left(\sum_h \theta_h x_h\right) \geq \sum_h \theta_h f(x_h), \quad (2.21)$$

where $\theta_h \geq 0$ and $\sum_h \theta_h = 1$, corresponding to a probability distribution. Now defining $q(\mathbf{s})$ to have the same properties as $\boldsymbol{\theta}$ and multiplying and dividing with it in equation (2.18) Jensen's inequality can be applied

$$\log \sum_{\mathbf{s}} \frac{q(\mathbf{s})}{q(\mathbf{s})} p(\mathbf{s} | \gamma) p(D | \mathbf{s}, \mathbf{w}, \beta) \geq - \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s} | \gamma) p(D | \mathbf{s}, \mathbf{w}, \beta)}. \quad (2.22)$$

The variational approximation $q(\mathbf{s})$ is defined in [KG12] to be a fully factorized distribution and satisfies $q(\mathbf{s}) = \prod_{i=1}^n q_i(s_i)$, where $q_i(s_i) = m_i s_i + (1 - m_i)(1 - s_i)$. This implies that m_i is the probability that s_i is equal to 1.

¹It would be very complex to find the MAP solution to the 'complete' posterior probability as \mathbf{s} has been defined to be binary.

Now defining the variational free energy

$$F(q, \mathbf{w}, \beta) = \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma)p(D|\mathbf{s}, \mathbf{w}, \beta)}. \quad (2.23)$$

Minimizing $F(q, \mathbf{w}, \beta)$ then corresponds to maximizing the log likelihood in equation (2.18). It is noted that $-F(q, \mathbf{w}, \beta)$ is the lower bound on the log-likelihood and should therefore be maximized, i.e. the same as minimizing $F(q, \mathbf{w}, \beta)$. The latter is expanded

$$\begin{aligned} F(q, \mathbf{w}, \beta) &= \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma)p(D|\mathbf{s}, \mathbf{w}, \beta)} = -F_1 - F_2 + F_3, \text{ with} \\ F_1 &= \sum_{\mathbf{s}} q(\mathbf{s}) \log p(D|\mathbf{s}, \mathbf{w}, \beta), \quad F_2 = \sum_{\mathbf{s}} q(\mathbf{s}) \log p(\mathbf{s}|\gamma) \text{ and } F_3 = \sum_{\mathbf{s}} q(\mathbf{s}) \log q(\mathbf{s}). \end{aligned} \quad (2.24)$$

The derivation of these can be seen in appendix A, the results are presented here

$$\begin{aligned} F_1 &= \frac{p}{2} \log \frac{\beta}{2\pi} \\ &\quad - \frac{p\beta}{2} \left(\sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \right) \end{aligned} \quad (2.25)$$

$$F_2 = \gamma \sum_{i=1}^n m_i - n \log(1 + \exp(\gamma)) \quad (2.26)$$

$$F_3 = \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)), \quad (2.27)$$

where $\sigma_y^2 = \frac{1}{p} \sum_{\mu=1}^p y_{\mu}^2$.

The variational free energy can now be presented

$$\begin{aligned} F(\mathbf{m}, \mathbf{w}, \beta) &= -\frac{p}{2} \log \frac{\beta}{2\pi} + \frac{p\beta}{2} \sigma_y^2 \\ &\quad + \frac{p\beta}{2} \left(\sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \right) \\ &\quad - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\ &\quad + \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)) \end{aligned} \quad (2.28)$$

The expression $F(\mathbf{m}, \mathbf{w}, \beta)$ is minimized by finding its derivatives with respect to \mathbf{w} , \mathbf{m} and β and setting them equal to 0. These derivations can be seen in appendix A. The parameters are found to be

$$\mathbf{w} = (\boldsymbol{\chi}')^{-1} \mathbf{b} \quad (2.29)$$

$$m_i = \left(1 + \exp \left(-\frac{\beta p}{2} w_i^2 \chi_{ii} - \gamma \right) \right)^{-1} = \sigma \left(\frac{\beta p}{2} w_i^2 \chi_{ii} + \gamma \right) \quad (2.30)$$

$$\frac{1}{\beta} = \sigma_y^2 - \sum_{i=1}^n m_i w_i b_i \quad (2.31)$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ and \mathbf{b} and $\boldsymbol{\chi}$ are as defined in section 2.1. Additionally defining: $\chi'_{ij} = m_j \chi_{ij} + (1 - m_j) \chi_{jj} \delta_{ij}$ and noting that the inverse of $\boldsymbol{\chi}'$ must exist, i.e. $\boldsymbol{\chi}'$ should be non-singular, alternatively the pseudo-inverse should be applied.

The implementation of the algorithm is suggested to consist of cross-validation on γ [KG12]. The details can be seen in appendix B. Of applied tricks the incorporation of the smoothing parameter η is worth mentioning. The parameter more precisely smooths the activation vector \mathbf{m} by

$$\mathbf{m}_{\text{new}} = (1 - \eta) \mathbf{m}_{\text{old}} + \eta \mathbf{m}_{\text{current}}, \quad (2.32)$$

where $\mathbf{m}_{\text{current}}$ is calculated by equation (2.30). The value of η is initialized to 1 for each γ , and is halved every time the maximum absolute value of the difference between the new and old \mathbf{m} is bigger than 0.1. This means that if the difference between old and new \mathbf{m} is 'big', then in the next iteration \mathbf{m}_{new} is forced to become closer to the previous value and thereby the activation vector is smoothed across iterations.

2.5.1 Performance of VG

The performance of the VG-algorithm is also verified using the data setup suggested in [KG12], example 1. The prediction of a response can be obtained from the VG solution by

$$y_\mu = \sum_{i=1}^n m_i w_i X_{i\mu} = \sum_{i=1}^n v_i X_{i\mu}, \quad (2.33)$$

where $v_i = m_i w_i$.

Each of the 50 implemented levels of sparsity γ has 100 updating iterations. The activation vector \mathbf{m} is initialized to zeros for each γ . As the number of

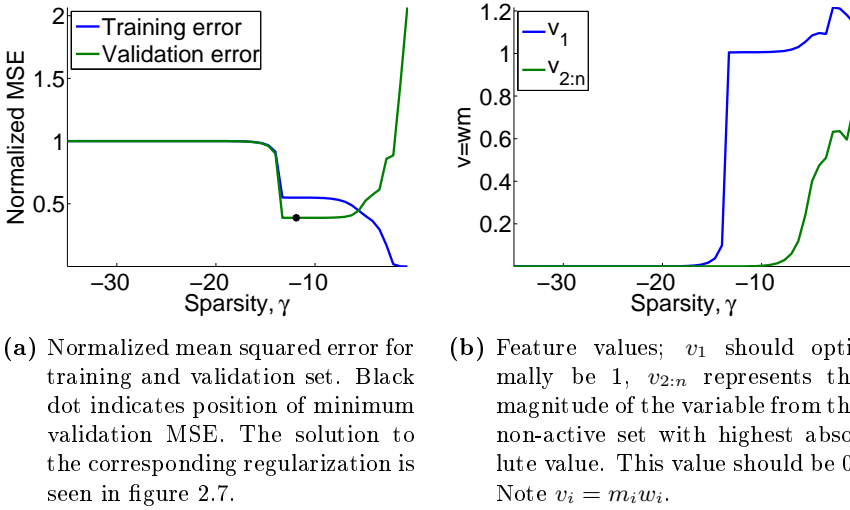


Figure 2.6: VG solutions for decreasing sparsity, from -35 to -0.7 with 50 steps, each with 100 iterations. Data set inspired by example 1 in [KG12].

samples is smaller than the number of input dimensions, the pseudo-inverse is used to calculate \mathbf{w} .

From figure 2.6b it is seen that in the region of sparsity $\gamma = -15$ to -10 , the first weight is approximately equal to 1 and the others are 0. Also in this region, the lowest validation nMSE is found, indicated in figure 2.6a by a black dot. This found optimum solution of feature values is depicted in figure 2.7. Note that the feature value vector, \mathbf{v} , corresponds to the element-wise multiplication of the vectors \mathbf{m} and \mathbf{w} . Visible from figure 2.7 is that VG finds the correct active weight, sets it to 1, and finds the remaining to be 0, thus performing better than LASSO and SBM.

Now 100 data sets are generated with the specifications suggested by Kappen et al. in example 1 in [KG12] through which the algorithms VG, SBM and LASSO are compared. The results are expressed as the mean nMSE \pm the standard deviation around this mean, see table 2.1. Note that the test set contains eight times more samples than the validation and training sets, thus explaining the lower standard deviations seen for the test set. The table clarifies that VG is best at approximating the weights, followed by SBM.

VG, LASSO and SBM are also compared through their learning curves, see figure 2.8. The curves are created by generating 100 data sets and extracting

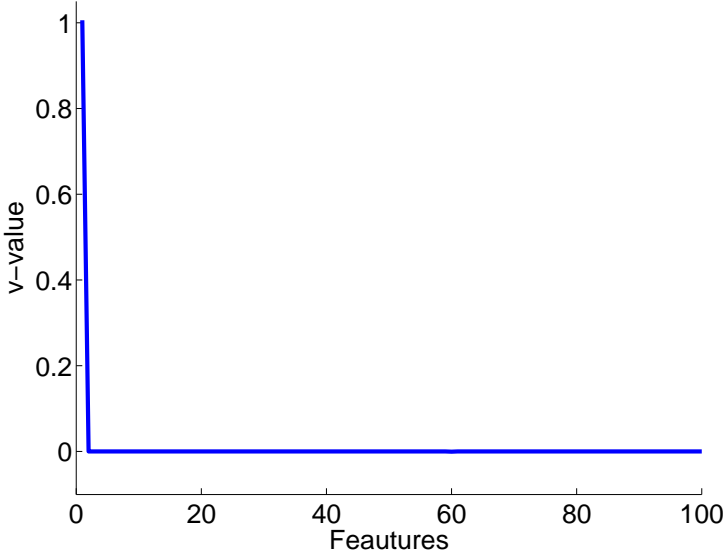


Figure 2.7: Optimum VG solution; weights that give the lowest validation error. Optimum level of sparsity: $\gamma = -11.9$. Data set inspired by example 1 in [KG12].

	Training error	Validation error	Test error
VG	0.44 ± 0.011	0.52 ± 0.011	0.52 ± 0.0042
LASSO	0.44 ± 0.012	0.56 ± 0.012	0.58 ± 0.0069
SBM	0.46 ± 0.011	0.53 ± 0.010	0.55 ± 0.0061
True	0.51 ± 0.011	0.51 ± 0.010	0.50 ± 0.0040

Table 2.1: The normalized MSE of VG, LASSO and SBM compared to application of the 'true' weights. Generated by 100 repetitions of the data set described in example 1 in [KG12]. The mean values \pm the standard deviations around the means are reported.

an increasing number of samples for training and for choosing the best regularization parameter, i.e. number of samples used for validation. A test error is calculated on the same test set with 400 samples for all training sizes in the 100 repetitions. The data again has the same characteristics as the data of example 1 in [KG12]. The training and test error for each repetition are calculated as mean squared errors, for VG; $\text{MSE} = E[(\mathbf{X}^T \mathbf{v} - \mathbf{y})^2]$, and for LASSO and SBM; $\text{MSE} = E[(\mathbf{X}^T \mathbf{w} - \mathbf{y})^2]$.

In figure 2.8 it is demonstrated that VG, LASSO and SBM converge towards the

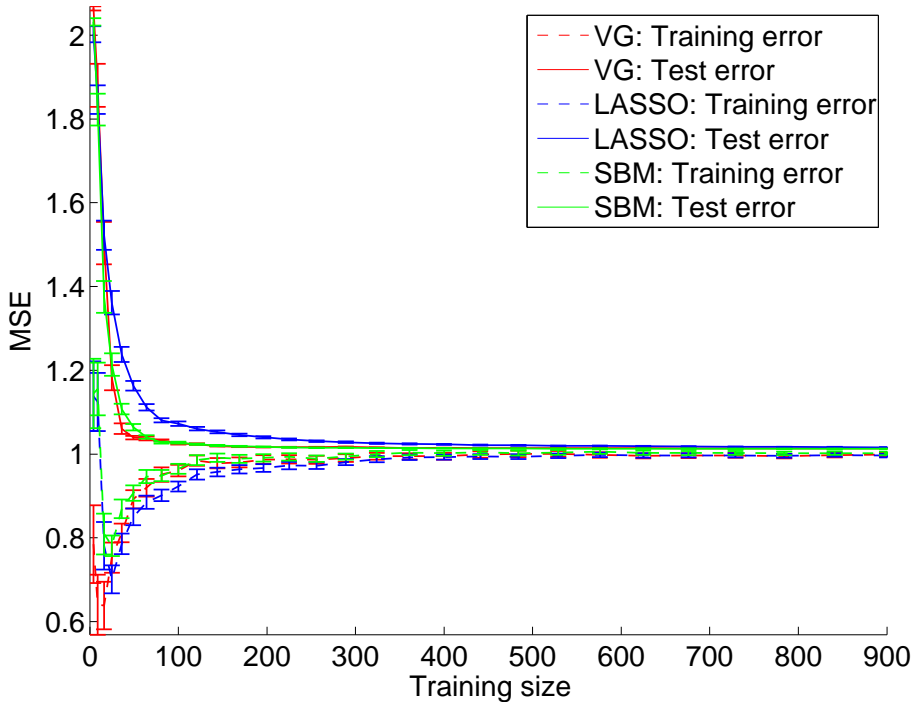


Figure 2.8: Learning curves for VG, LASSO and SBM. The sizes of the training and validation sets are increased and an MSE on a test set is reported. 100 repetitions are performed. VG and SBM are run with 20 iterations for each γ and β , respectively. The distinction between the stipled and non-stipled is difficult, however the training error is always the lower of the two equally colored graphs. Data set is inspired by example 1 in [KG12].

variance of the data, however VG and SBM converge after 40 training examples and LASSO not until 400. This implies that VG and SBM demand a smaller training set compared to LASSO to find a good solution. The figure also shows that as expected, higher variability in error is seen between smaller training sets.

2.5.2 Reformulation using Kailath Variant

With the purpose of reducing computation time, an alternative calculation of \mathbf{w} is presented. Calculating the inverse of a large matrix is high in computational

cost and the pseudo-inverse even higher. The latter is necessary when the matrix at hand is singular. Therefore χ' is rewritten using Kailath Variant, which is expressed as $(\mathbf{A} + \mathbf{BC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}$ [PBT⁺08]. Making this recasting of the problem is of course only relevant if calculating the inverses of \mathbf{A} and $\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B}$ is low in computational cost, i.e. if they are low-dimensional, and if they are in fact invertible.

Breaking χ' into \mathbf{A} , \mathbf{B} and \mathbf{C}

$$\mathbf{A} = \text{diag}((1 - m_j)\chi_{jj})_{j=1:n} \iff A_{ij} = \begin{cases} (1 - m_j)\chi_{ij} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \quad (2.34)$$

$$\mathbf{B} = \frac{1}{p}\mathbf{X} \iff B_{i\mu} = \frac{X_{i\mu}}{p} \quad (2.35)$$

$$\mathbf{C} = \mathbf{X}^T \text{diag}(\mathbf{m}) \iff C_{\mu j} = X_{j\mu} m_j. \quad (2.36)$$

$$(2.37)$$

Verification of the decomposition

$$\mathbf{A} + \mathbf{BC} = \text{diag}((1 - m_j)\chi_{jj})_{j=1:n} + \frac{1}{p}\mathbf{XX}^T \text{diag}(\mathbf{m}) = \chi'. \quad (2.38)$$

The operation $\text{diag}(\mathbf{d})$ refers to inserting vector \mathbf{d} in a diagonal matrix. The first expression to invert is \mathbf{A} . It can be done efficient, as it is a diagonal matrix and the elements in the inverted matrix is just the inverse of the values in the original matrix: $A_{ii}^{-1} = 1/A_{ii}$, or expressed in matrix form: $\mathbf{A}^{-1} = \text{diag}(1 \oslash ((1 - \mathbf{m}) \odot \chi_{diag})) = \text{diag}(\mathbf{a}_{inv})$, where χ_{diag} is an n -vector with elements from the diagonal in the covariance matrix χ . The notations \oslash and \odot indicate an element-wise division and multiplication, respectively. Note that if an element in \mathbf{m} is 1, \mathbf{A}^{-1} is not computable. This can however be fixed by replacing such instances by $1 - \epsilon$, where ϵ is a small number. In the MATLAB implementation of VG in Donders Machine Learning Toolbox (DMLT) [Dis12] created by Donders Institute for Brain, Cognition and Behavior ϵ is set to 10^{-10} .

The next to invert is $\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B}$, which becomes a $p \times p$ -matrix. This means that instead of the computational cost of inversion is dependent on the number of dimensions, it is now dependent on the number of samples (corresponding to the number of electrodes in the EEG problem, which is much smaller than the number of sources in the brain).

In order to increase computation efficiency additionally, \mathbf{b} is included in the derivation of χ' , thus yielding \mathbf{w} directly. The expression to compute is then: $(\mathbf{A} + \mathbf{BC})^{-1}\mathbf{b} = \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}\mathbf{b}$. The first expression

can be calculated element-wise as

$$A_{jj}^{-1}b_j = \frac{b_j}{A_{jj}} = \frac{b_j}{(1 - m_j)\chi_{jj}}, \quad (2.39)$$

or in matrix form

$$\mathbf{A}^{-1}\mathbf{b} = (1 \oslash ((1 - \mathbf{m}) \odot \chi_{diag})) \odot \mathbf{b} = \mathbf{a}_{inv} \odot \mathbf{b}. \quad (2.40)$$

The second expression that demands inversion

$$\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B} = \mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m} \odot \mathbf{a}_{inv}) \frac{1}{p} \mathbf{X}. \quad (2.41)$$

Calculation of \mathbf{w} in vector form can now be completed

$$\begin{aligned} \mathbf{w} = & \mathbf{a}_{inv} \odot \mathbf{b} - \\ & \frac{1}{p} \mathbf{a}_{inv} \odot \left(\mathbf{X} \left(\left(\mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m} \odot \mathbf{a}_{inv}) \frac{1}{p} \mathbf{X} \right)^{-1} (\mathbf{X}^T (\mathbf{m} \odot \mathbf{a}_{inv}) \odot \mathbf{b}) \right) \right). \end{aligned} \quad (2.42)$$

Note that the multiple parentheses ensure that an $n \times n$ -matrix is not created, and thereby avoiding multiplications with a 'big' matrix, hence reducing computation time. The above is calculated more comprehensively and using element-wise notation in appendix C.

2.5.3 Dual formulation

Kappen et al. also suggest a technique that improves computation efficiency. The problem is reformulated to a dual representation. The variables $z_\mu = \sum_{i=1}^n m_i w_i X_{i\mu}$ are defined and Lagrange multipliers λ are added. Making the variational free energy

$$\begin{aligned} F(\mathbf{m}, \mathbf{w}, \beta, \mathbf{z}, \lambda) = & -\frac{p}{2} \log \frac{\beta}{2\pi} + \frac{\beta}{2} \sum_{\mu=1}^p (z_\mu - y_\mu)^2 + \frac{p\beta}{2} \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} \\ & - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\ & + \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)) \\ & + \sum_{\mu=1}^p \lambda_\mu \left(z_\mu - \sum_{i=1}^n m_i w_i X_{i\mu} \right). \end{aligned} \quad (2.43)$$

Setting the partial derivative of the above equation, with respect to the variables, equal to 0 yields the following equations to be iterated

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i X_{i\mu} X_{i\nu}}{(1 - m_i) \chi_{ii}}, \quad (2.44)$$

$$y_\mu = \sum_{\nu=1}^p A_{\mu\nu} \hat{y}_\nu, \quad (2.45)$$

$$\frac{1}{\beta} = \frac{1}{p} \sum_{\mu=1}^p \hat{y}_\mu y_\mu, \quad (2.46)$$

$$\lambda_\mu = \beta \hat{y}_\mu, \quad (2.47)$$

$$w_i = \frac{1}{\beta p \chi_{ii} (1 - m_i)} \sum_{\mu=1}^p \lambda_\mu X_{i\mu}, \quad (2.48)$$

$$m_i = \left(1 + \exp \left(-\frac{\beta p}{2} w_i^2 \chi_{ii} - \gamma \right) \right)^{-1}. \quad (2.49)$$

The derivations can be seen in appendix D.

2.5.4 Time-expanded dual formulation

In this thesis the dual VG formulation is expanded to the application of time windows. It is assumed that an EEG source in the brain has a certain time period of activation. The strength of the activation might vary, e.g. oscillate with some frequency. The binary variable (\mathbf{s}) is therefore held constant in the time window while \mathbf{w} is allowed fluctuations. Using more time samples to calculate the parameters should make the model stronger and thereby improve the performance.

Dual representation of F with time dependent \mathbf{w} , \mathbf{y} , \mathbf{z} and λ

$$\begin{aligned}
F(\mathbf{m}, \mathbf{w}, \beta, \mathbf{z}, \lambda) = & -\frac{Tp}{2} \log \frac{\beta}{2\pi} + \frac{\beta}{2} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{p\beta}{2} \sum_{t=1}^T \sum_{i=1}^n m_i(1 - m_i) w_{it}^2 \chi_{ii} \\
& - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\
& + \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)) \\
& + \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} \left(z_{\mu t} - \sum_{i=1}^n m_i w_{it} X_{i\mu} \right). \tag{2.50}
\end{aligned}$$

Notice that only the parts in the above equation stemming from the likelihood term in the variational free energy, i.e. equation (2.25), are affected by the summation over time samples.

The procedure of finding the parameters follows that of the VG primal and dual formulation. The partial derivatives of F are found and subsequently set to 0.

$$\frac{\partial F}{\partial w_{it}} = \beta p m_i (1 - m_i) \chi_{ii} w_{it} - \sum_{\mu=1}^p \lambda_{\mu t} m_i X_{i\mu}, \tag{2.51}$$

$$\frac{\partial F}{\partial z_{\mu t}} = \beta (z_{\mu t} - y_{\mu t}) + \lambda_{\mu t}, \tag{2.52}$$

$$\frac{\partial F}{\partial \beta} = -\frac{Tp}{2\beta} + \frac{1}{2} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{p}{2} \sum_{t=1}^T \sum_{i=1}^n m_i(1 - m_i) w_{it}^2 \chi_{ii}, \tag{2.53}$$

$$\frac{\partial F}{\partial m_i} = \frac{\beta p}{2} \sum_{t=1}^T (1 - 2m_i) w_{it}^2 \chi_{ii} - \gamma + \log \left(\frac{m_i}{1 - m_i} \right) - \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} w_{it} X_{i\mu}, \tag{2.54}$$

$$\frac{\partial F}{\partial \lambda_{\mu t}} = z_{\mu t} - \sum_{i=1}^n m_i w_{it} X_{i\mu}. \tag{2.55}$$

$$\tag{2.56}$$

Solving $\frac{\partial F}{\partial w_{it}} = 0$ yields

$$w_{it} = \frac{1}{p\beta(1 - m_i)\chi_{ii}} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu}, \tag{2.57}$$

and $\frac{\partial F}{\partial z_{\mu t}} = 0$

$$z_{\mu t} = y_{\mu t} - \frac{1}{\beta} \lambda_{\mu t}. \quad (2.58)$$

These equations are used in the following. Starting with $\frac{\partial F}{\partial \beta} = 0$

$$\beta = \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} \lambda_{\nu t} A_{\mu\nu}, \quad (2.59)$$

when defining

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu}. \quad (2.60)$$

Next $\frac{\partial F}{\partial \lambda_{\mu t}} = 0$

$$\beta y_{\mu t} = \sum_{\nu=1}^p \lambda_{\nu t} \left(\delta_{\mu\nu} + \sum_{i=1}^n \frac{m_i}{p(1 - m_i) \chi_{ii}} X_{i\nu} X_{i\mu} \right) = \sum_{\nu=1}^p \lambda_{\nu t} A_{\mu\nu}. \quad (2.61)$$

Introducing

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu t} = y_{\mu t}, \quad (2.62)$$

and inserting this in equation (2.61) yields

$$\lambda_{\nu t} = \beta \hat{y}_{\nu t}. \quad (2.63)$$

Inserting (2.63) and (2.62) in (2.59) yields a simplification of β

$$\frac{1}{\beta} = \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \hat{y}_{\mu t} y_{\mu t}. \quad (2.64)$$

Using equation (2.57), \mathbf{m} is derived

$$m_i = \sigma \left(\frac{\beta p}{2} \chi_{ii} \sum_{t=1}^T w_{it}^2 + \gamma \right). \quad (2.65)$$

The final equation set is then

$$w_{it} = \frac{1}{p\beta(1-m_i)\chi_{ii}} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu}, \quad (2.66)$$

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1-m_i)\chi_{ii}} X_{i\mu} X_{i\nu}, \quad (2.67)$$

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu t} = y_{\mu t}, \quad (2.68)$$

$$\frac{1}{\beta} = \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \hat{y}_{\mu t} y_{\mu t}, \quad (2.69)$$

$$\lambda_{\nu t} = \beta \hat{y}_{\nu t}, \quad (2.70)$$

$$m_i = \sigma \left(\frac{\beta p}{2} \chi_{ii} \sum_{t=1}^T w_{it}^2 + \gamma \right). \quad (2.71)$$

Appendix E presents more detailed calculations, along with a vector/matrix representation of the above equation set.

A five-fold cross-validation experiment using synthetic data is performed to illustrate the performance of the algorithm. The training and test data each consist of 50 samples, the input dimension is 100 and \mathbf{X} is a random matrix. A time frame of 25 samples is chosen corresponding to 100 ms if using a sampling frequency of 250 Hz. Within this time frame ten parameters are set to be active with the temporal development of a sine wave with amplitude 1 and a period of 100 ms. The temporal source distribution thus corresponds to an alpha frequency wave in ten sources. One 'true' source is depicted in figure 2.9 as the green curve.

Re-initialization of the activation vector, $\mathbf{m} = \mathbf{0}$, is done for each γ . The smoothing parameter η is set heuristically to 0.5. To facilitate a direct comparison between the dual formulation in single and combined time, 100 iterations are used for each γ , thus disregarding possible convergence. For each fold in the cross-validation an optimum sparsity is found as the γ with lowest error on the validation set. The mean value of these five γ s is defined to be the optimum γ . The optimum γ is applied to the combined training and validation set yielding one solution.

Figure 2.9 shows a solution for the two dual formulations with a level of signal to noise ratio (SNR) of 10. The SNR is calculated as the mean value of the pure signal divided by the added noise, across time samples. It is revealed in

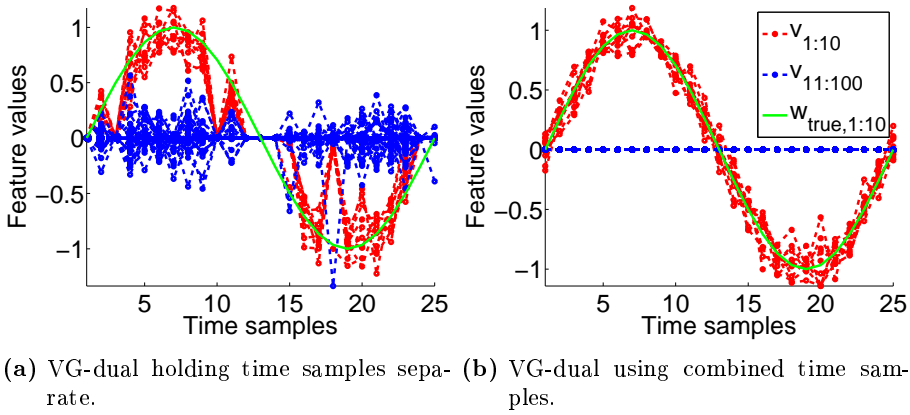


Figure 2.9: The feature values as function of time samples. SNR=10. 'True' appearance of the ten equal strength active sources is shown in bright green. For each level of sparsity 100 iterations are applied. Five-fold cross-validation is used to find optimum level of sparsity.

figure 2.9a that the single time solution only locates the sources in some of the time samples. Especially the activation in time samples with low magnitudes is not recovered. It is clear from figure 2.9b that using multiple time samples greatly increases the proximity to the 'true' feature values. Additionally the non-active sources $\mathbf{v}_{11:100}$ are more accurately represented, having the value 0. It is thus concluded that applying multiple time samples increases the ability of the algorithm to obtain the correct source distribution.

The two dual formulations are further examined as function of the SNR of the applied data. Ten repetitions of five-fold cross-validations for 25 levels of SNR are performed. The test errors obtained is compared to applying the 'true' weight distribution. The comparison of the two algorithms' performances with respect to the test error is seen in figure 2.10. Both solutions start of by having a large test error and both follow the curve of the best possible test errors (blue curve). However the dual time combined model obtains better results and approaches the optimum MSE closer.

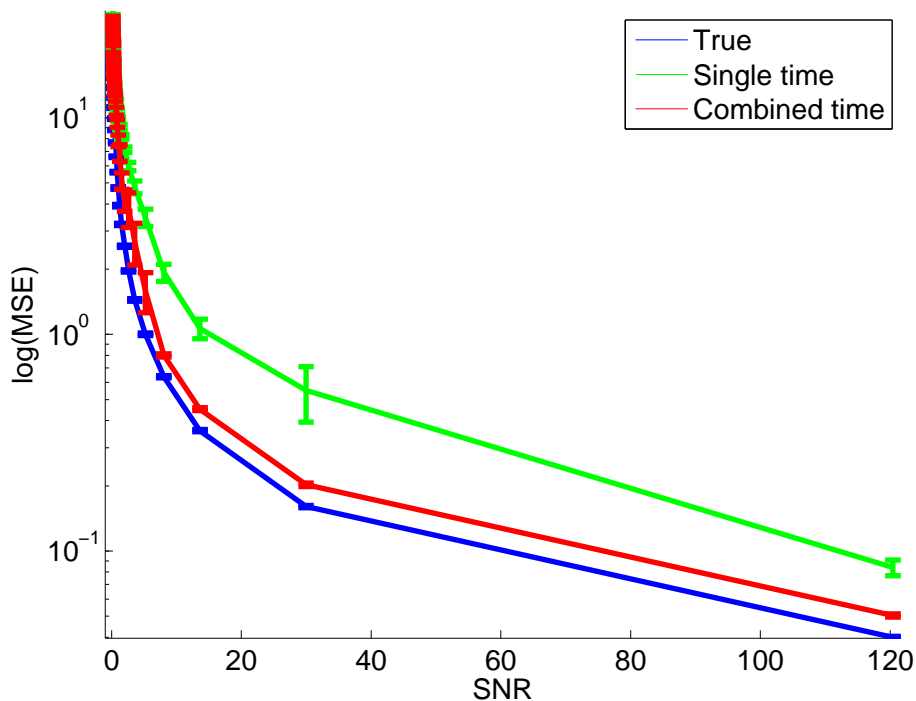


Figure 2.10: Test error as function of 25 levels of SNR. The sources' temporal development are estimated one time sample at a time and combined, respectively. These are compared to the 'true' source distribution. For each level of sparsity 100 iterations are applied in ten five-fold cross-validations.

CHAPTER 3

Experimental Design

As described in the previous chapter, VG outperformed SBM and LASSO on the simple setup presented. VG is thus chosen as the main focus in the following experiments. To ensure superiority in the EEG framework, the algorithms are now compared using a forward field matrix as input.

The VG algorithm should be refined to fit EEG settings, thus several experiments to obtain this are required. Synthetic sources are initially applied as the weight vector \mathbf{w} . This is done to make the solutions acquired easy to evaluate. Later, well-known EEG data will be used to verify the VG algorithm.

First experiments on the instantaneous VGs; the dual formulation of VG (VG-dual) and the Kailath Variant VG (VG-KV), and LASSO and SBM are presented. Then follows a description of the experiments on the time-expanded VG-dual, where a constant mode of activation of each source within a given time window is assumed.

All experiments are performed in MATLAB 2011b (The MathWorks Inc).

3.1 Sparse Algorithms in Single Time

The experiments examining VG, where the input \mathbf{X} is a transposed forward field matrix and a synthetic weight vector is applied, are:

1. Stability in number of cross-validation folds. The number of folds to create in K -fold cross-validation is investigated, i.e. the optimal ratio between training and validation sizes is found.
 - (a) Comparison of performances of VG-dual, VG-KV, LASSO and SBM.
2. Initialization of γ and \mathbf{m} in VG-dual. The optimum solution path of VG-dual is investigated. Re-initialization of the activation vector, $\mathbf{m} = \mathbf{0}$, for each level of sparsity, γ , is compared to using a backward, forward and combined path.

3.1.1 Synthetic data

The synthetic sources are set to 1 in ten of the 8196 positions of the weight vector. The remaining sources are set to 0. The active sources are placed in the back of the left hemisphere, i.e. the left occipital lobe, corresponding to position one through ten in the weight vector.

The same forward field matrix is applied to all experiments in this chapter. This matrix relates 8196 sources to 128 channels. The forward field matrix is the result of solving the forward problem using a symmetric BEM three-layered head model with structural MRI information from a subject enrolled in the multimodal face-evoked response study [HG^{GG}+03]. The forward field matrix is created using SPM8 [ACM⁺12] and applying the open source software *OpenMEEG* [GPOC11].

Data creation in summary

- The transposed forward field matrix is used as input. i.e. \mathbf{X} is a 8196×128 -matrix.
- \mathbf{X} is scaled so $\sum_{\mu=1}^p X_{i\mu}/p = 0$ and $\sum_{\mu=1}^p X_{i\mu}^2/p = 1$.
- The weight matrix contains zeros at all elements except for the first ten, which have ones.

- The response \mathbf{y} is created by $\mathbf{y} = \mathbf{X}^T \mathbf{w}_{\text{true}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, 1)$. Thus making \mathbf{y} a column vector with 128 elements, corresponding to potential differences measured by 128 electrodes/channels.
- Finally the order of the channels is randomized.

3.1.2 Experiment 1.1: Stability in number of cross-validation folds

The performances of VG, LASSO and SBM are investigated across number of folds used in cross-validation. Two different formulations of the solution to VG are evaluated, VG-KV and VG-dual, both presented in section 2.5. The solutions of the models are compared to the weight vector that generated the output; i.e. \mathbf{w}_{true} .

Before describing the implementation of each of the four algorithms the cross-validation steps are outlined in the following.

Two-level K -fold cross-validation

Ten samples (channels) are extracted 50 times from the created input and output, these are used as test sets, thus 50 repetitions are executed. The means are subtracted from \mathbf{X}_{test} and \mathbf{y}_{test} .

1. K is chosen between 2 and 15. For each K :
 - (a) The data remaining after extraction of test set is split into K folds.
 - (b) For $k = 1 : K$ the following is performed:
 - i. The k th data set is used for validation. The rest is used for training. The means are subtracted from the response and input in the two data sets.
 - A. The model is applied to each relevant level of regularization.
 - B. A validation error is calculated for the solutions \mathbf{w} (\mathbf{v} for the VGs) found for each level of regularization

$$\text{MSE}_{\text{val}} = E [(\mathbf{X}_{\text{val}}^T \mathbf{w} - \mathbf{y}_{\text{val}})^2]. \quad (3.1)$$

- C. The minimum validation error is found across regularizations, the corresponding optimum regularization is reported.

- (c) A mean optimal regularization-level is found across the K -folds. This level of regularization is used to calculate the solution \mathbf{w}_K (\mathbf{v}_K for the VGs), where both training and validation data is applied to train the model.
- (d) A test error is calculated for each K as the normalized mean squared error

$$\text{nMSE}_{\text{test},K} = \frac{E[(\mathbf{X}_{\text{test}}^T \mathbf{w}_K - \mathbf{y}_{\text{test}})^2]}{\sigma_{y_{\text{test}}}^2}. \quad (3.2)$$

The 50 splits of training and test data are used to calculate means and standard deviations of the test errors for each K (see figure 4.1). The setup is implemented in the MATLAB function `twolevel_crossval`, seen in appendix F.1.

Application of VG-dual

DMLT [Dis12] created by Donders Institute for Brain, Cognition and Behaviour implements the dual VG-algorithm. The toolbox is available through the open source network github <https://github.com/distrep/DMLT>. The relevant equation set used, is equivalent to equations (2.44) through (2.49). The implementation approach is very similar to that described by Kappen et al. which is also reproduced in appendix B. DMLT however has the following adjustments:

- Convergence/stopping criteria:
 - Maximum absolute difference in current and previous \mathbf{m} , default: 10^{-12}
 - Maximum number of iterations, set to 50 in the current application.
 - Upper boundary of inverse variance β , set to default: 1000.
- The smoothing parameter η is halved when the difference between current and previous variational free energy is bigger than 10^{-10} , instead of when the maximum absolute difference in current and previous \mathbf{m} is bigger than 0.1.
- To avoid numerical problems the values of \mathbf{m} are fixed between 10^{-10} and $1 - 10^{-10}$.

The approach used in the toolbox is adapted to fit the current two-level cross-validation experiment, thus modifications and additions are made. As described

in the cross-validation step 1.(c) an optimal level of sparsity, γ_{opt} , is determined for each K . This value is fed into a new function, which finds the optimal solution \mathbf{v}_{opt} by running VG from γ_{min} to γ_{opt} and then backwards from γ_{max} to γ_{opt} . The direction giving lowest variational free energy at γ_{opt} is chosen to define \mathbf{v}_{opt} . The range of sparsity is heuristically defined as being from $\gamma_{\text{min}} = -50$ to $\gamma_{\text{max}} = -1$ with 50 steps.

Two final alterations are performed to ensure at least three updating iterations are performed for each level of sparsity: 1) not letting the code stop before three iterations have been executed and 2) removing the `break` which is set into action when `'eta<1e10'`. As the smoothing parameter η is always smaller than 1 this will always break the iterations, this implementation must therefore be a mistake.

Application of VG Kailath Variant

The Kailath Variant formulation of VG implements the equations (2.29), (2.30) and (2.31). Equation (2.29), describing \mathbf{w} , is implemented in MATLAB using equation (2.42), repeated here

$$\mathbf{w} = \mathbf{a}_{\text{inv}} \odot \mathbf{b} - \frac{1}{p} \mathbf{a}_{\text{inv}} \odot \left(\mathbf{X} \left(\left(\mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m} \odot \mathbf{a}_{\text{inv}}) \frac{1}{p} \mathbf{X} \right)^{-1} (\mathbf{X}^T (\mathbf{m} \odot \mathbf{a}_{\text{inv}}) \odot \mathbf{b}) \right) \right).$$

One noteworthy difference from VG-dual is that the activation vector, \mathbf{m} , is re-initialized for each level of sparsity. Thus only one pathway search is necessary. The same range of sparsity, as used for VG-dual, is applied. 50 iterations are used if not stopped by an update in \mathbf{m} that is smaller than 10^{-8} . The values of \mathbf{m} are fixed to be below $1 - 10^{-10}$. The Kailath Variant formulation is implemented in the MATLAB function `vgKV`, seen in appendix F.2.

Application of LASSO

The MATLAB toolbox SpaSM [Sjö05], created by Sjöstrand and described in section 2.2, is applied for creating the LASSO solutions. The same preprocessing as done for VG-dual and VG-KV is used; the mean values are subtracted respectively in training, validation and test sets in both \mathbf{X} and \mathbf{y} . The effect of scaling the row variances in \mathbf{X} , in the before mentioned data sets, was investigated beforehand. As the same results were obtained with and without this

scaling, it was found unnecessary to perform the scaling, and thus facilitating equal preprocessing procedure for the four models.

The MATLAB function `lasso` is extended to output the penalty, i.e. t in equation (2.5). When the mean optimum penalty has been found, this value can then be applied and the optimum solution found.

Application of the sparse Bayesian model

The MATLAB toolbox SparseBayes Version 2.0 [Tip09b] created by Tipping is utilized to create an SBM model with a Gaussian likelihood model and a linear basis. As done in the previous chapter, the hyperparameters α controlling the weights are estimated, while β , the precision of the noise in the data, is chosen through cross-validation. Across folds the optimum value of β is found and applied to the SBM model together with both the training and validation set.

3.1.3 Experiment 1.2: Initialization of γ and \mathbf{m} in VG-dual

The dual formulation of the VG problem is tested with respect to the initial level of sparsity, γ_{\min} . The purpose is to examine the stability of the algorithm with respect to this parameter. The forward (starting at γ_{\min}) and backward (starting at γ_{\max}) pathway searches are compared to the solution obtained when combining the two. The combination of the two is for each γ defined by the direction of pathway search with lowest variational free energy. Additionally the effect of initializing the activation vector, $\mathbf{m} = \mathbf{0}$, for each γ is explored.

From the transposed forward field matrix, with the dimensions 8196×128 , 118 electrodes are extracted in 50 different ways and used as \mathbf{X} . This data is used to train the model, thus 50 repetitions are performed. The data is created as in the two-level cross-validation experiment with ten active sources with a value of 1.

The source retrieval index F_s [MKS99]

$$F_s = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{\text{TP} + \text{FP} + \text{TP} + \text{FN}} = \frac{2\text{TP}}{\text{TP} + \text{TP} + 10}, \quad (3.3)$$

is used to evaluate the performance, where TP, FP and FN are the number of true positives, false positives and false negatives, respectively. Note that since ten sources are active is $\text{TP} + \text{FN} = 10$.

An interval of γ_{\min} from -55 to -45 is applied and all solutions on the path to $\gamma_{\max} = 0$ are stored. The best solution from each initialization is defined to be the solution with highest source retrieval index. Note that when re-initializing \mathbf{m} for each γ , this is unnecessary, therefore the direct solutions obtained in the same interval are reported, i.e. $\gamma_{\text{opt}} = \gamma_{\min}$. The results can be seen in figure 4.6.

3.2 Time-expanded VG-dual

The following section details the setup of experiments on the time-expanded dual formulation of VG. The algorithm assumes that each source has a constant mode of activation (on or off) for all of the time samples applied to the algorithm. The equation set, equations (2.66) to (2.71), describing this algorithm, is implemented in MATLAB using the following vector/matrix equations

$$\mathbf{W}_{n \times T} = (1 \odot (p\beta(1 - \mathbf{m}) \odot \chi_{\text{diag}}) \cdot \mathbf{1}'_T) \odot (\mathbf{X} \cdot \boldsymbol{\lambda}), \quad (3.4)$$

$$\mathbf{A}_{p \times p} = \mathbf{I}_{p \times p} + \frac{1}{p} \mathbf{X}^T \cdot \text{diag}(\mathbf{m} \odot ((1 - \mathbf{m}) \odot \chi_{\text{diag}})) \cdot \mathbf{X}, \quad (3.5)$$

$$\hat{\mathbf{Y}}_{p \times T} = \mathbf{A} \setminus \mathbf{Y}, \quad (3.6)$$

$$\beta_{1 \times 1} = Tp / \text{sum}(\hat{\mathbf{Y}} \odot \mathbf{Y}), \quad (3.7)$$

$$\boldsymbol{\lambda}_{p \times T} = \beta \hat{\mathbf{Y}} \quad \text{and} \quad (3.8)$$

$$\mathbf{m}_{n \times 1} = \sigma \left(\frac{\beta p}{2} \text{sum}(\mathbf{W}.^2, 2) + \gamma \right). \quad (3.9)$$

$\mathbf{1}_T$ denotes a column vector of ones of length T . In equation (3.7) 'sum' indicates the sum over all elements in the matrix generated by $\hat{\mathbf{Y}} \odot \mathbf{Y}$. In equation (3.9) the sum is along the rows of the squared elements in \mathbf{W} .

Common for the following experiments is the use of VG-dual applied to a time window, and that the transposed forward field matrix, described in section 3.1.1, functions as the input matrix, \mathbf{X} . The first experiment is designed to validate the approach on a synthetic data set. The second experiment is created to compare the found solution to that obtained by SPM8 [ACM⁺12] with the MSP model. Finally a time window in a single epoch from the multimodal face-evoked data set is fed to the time-expanded VG-dual. In summary the experiments are:

1. Performance on synthetic data. The synthetic temporal source distribution consists of sine waves applied as ten sources.

- Search for an optimum value of the smoothing parameter η .
- 2. Performance on differential ERP. The differential ERP from the multimodal face-evoked response data set is used as samples of the response.
- 3. Performance on single face epoch. A time window from one epoch, where an image of a face is presented to the subject in the multimodal face-evoked response data set, is used as samples of the response.

3.2.1 Multimodal face-evoked response data set

The data set is available through the SPM website: <http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/>. The stimuli setup is detailed in [HG^{GG}+03] and is briefly outlined in section 1.2.4.

The multimodal face-evoked response data was recorded on a 128 channel ActiveTwo system (additionally two earlobes and two bipolar, HEOG and VEOG, channels), sampled at 2048 Hz. The raw data from two runs on one subject is preprocessed in SPM8 as described in the SPM manual [ACM⁺12], including the following steps:

- Converting: loads the data into a .mat and .dat file.
- Downsampling: the sampling rate is decreased to 200 Hz.
- Montage: removes channels without relevant EEG data.
- Epoching: sets the window on the recorded EEG data to the relevant area: -200 ms to 600 ms, with respect to stimulus presentation. Includes baseline correction; baseline from -200 to 0 ms.
- Artifact rejection: marks trials as containing artifacts if data magnitude exceeds 200 μ V. Out of 344 trials, 305 remain.
- Robust averaging: calculates a weighted mean response for each of the two conditions. As there approximately is an equal number of trials in each condition, both conditions are used for calculating the means.
- Contrasting: creates difference (differential ERP) and mean (average ERP) of averaged responses from face and scrambled stimuli.

3.2.2 Experiment 2.1: Performance on synthetic data

Synthetic data is used to verify that the VG-dual time formulation is applicable to the forward field matrix. The synthetic source distribution is similar to that used in section 2.5.4, where ten sources are active with the appearance of a sine wave through 25 time samples, see figure 2.9. The output is generated by $\mathbf{Y} = \mathbf{X}^T \mathbf{W}_{\text{true}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, 1)$.

A search for a suitable level of the smoothing parameter (η) of the activation vector (\mathbf{m}) see equation (2.32), is performed in the range from 0.3 to 1. Kappen et al. describe that if the maximum absolute difference between the old and new \mathbf{m} is bigger than 0.1, η should be halved. This value is reduced to 0.05, as it was seen to improve the solution.

50 combinations of ten channels are used as 50 test sets, leaving 118 channels for each combination which are applied to a five-fold cross-validation. The 118 channels are thus split into a training and a validation set. The activation parameters \mathbf{m} are re-initialized for each level of sparsity, γ . The cross-validation investigates γ from -150 to 0, with 60 steps and 100 iterations for each γ . An SNR of approximately 40 is used.

The performance is evaluated by the mean squared test error, and by the number of true and false positives. The number of positives is defined to be the number of sources with activation m_i higher than 0.5.

The general implementation of the dual VG on time windows can be seen in the MATLAB script presented in appendix F.3

3.2.3 Experiment 2.2: Performance on differential ERP

Real EEG data is now fed to the VG-dual time model, however in this experiment in the form of the differential averaged data. This data set is the difference in EEG between faces and scrambled faces that have been averaged over 305 trials/epochs. This is the final step before applying the model to one single epoch. The experiment is performed to enable a comparison with the results produced by SPM8 using MSP, see section 1.2.4.

As in the above experiment, using synthetic data, a five-fold cross-validation experiment is conducted, the data now being the differential ERP. First the entire frame of time, from -200 to 600 ms is used as the time window. Then only 20 samples, corresponding to 100 ms, are extracted; more specifically from

100 to 200 ms. 100 iterations are used for each γ , which is in the range from -80 to 0 with 100 steps. Finally η is set to the found value in experiment 2.1; i.e. 0.55 .

Additionally, 100 ms of the peristimulus area are extracted, from -100 to 0 ms, and applied to the same settings as the above. This is done to check if background brain activity is effectively removed by the performed baseline-correction.

3.2.4 Experiment 2.3: Performance on single face epoch

The same approach as in experiment 2.2 is executed with a time frame from 100 to 200 ms. The data set now consists of a single epoch, where the stimulus is an image of a face. The experiment is repeated for different epochs and results from two representing epochs are shown.

Results

The outcome of the experiments described in the previous chapter is presented here following the same section structure. Initial comments on the results are included while discussions are left for the next chapter.

4.1 Sparse Algorithms in Single Time

4.1.1 Experiment 1.1: Stability in number of cross-validation folds

The results of applying K -fold cross-validation with K from 2 to 15 using the two formulations of the VG algorithm; VG-dual and VG-KV, and the L_1 -inducing LASSO, as well as the sparse Bayesian learning method SBM are seen in figures 4.1-4.5. Selected findings of the figures have been submitted to International Conference on Acoustics, Speech, and Signal Processing 2013, preprint is seen appendix G.

As seen from figure 4.1, the test errors obtained by the four methods are very similar. VG-dual performs slightly better than SBM, which is followed by LASSO and finally by VG-KV. Especially VG-dual, LASSO and SBM are very

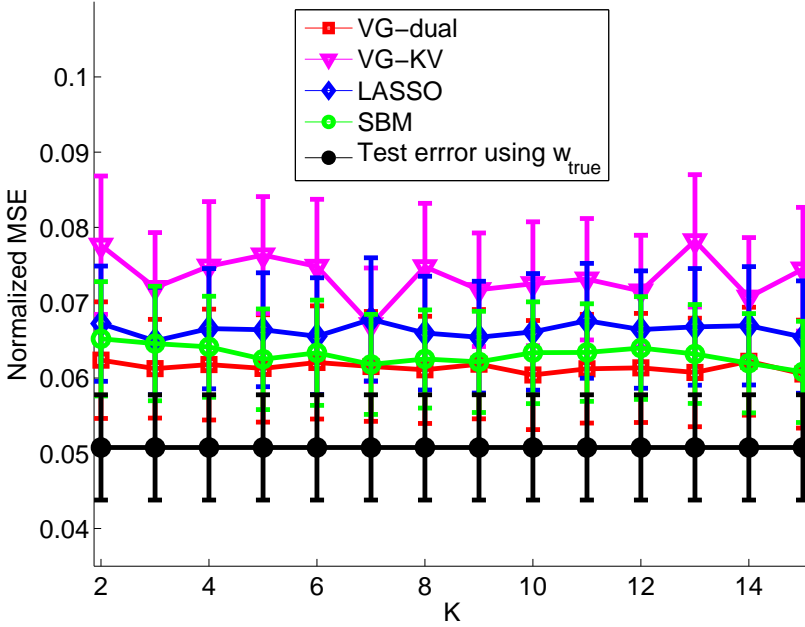


Figure 4.1: Normalized mean squared test error after performing 50 two-level K -fold cross-validations. K is investigated from 2 to 15. The algorithms are optimized with respect to one parameter; for the VGs the sparsity level γ , for LASSO the regularization parameter λ and for SBM the precision of the noise β . The solution of the VGs are in the form of \mathbf{v} , while LASSO and SBM are in the form of \mathbf{w} . Ten sources out of 8196 are defined to be active in the 'true' weight distribution.

stable across number of folds in validation and training sets. Note that the mean squared test errors have been normalized by the variance on \mathbf{y}_{test} which have an average value of 55, thus the test error is presented in the form of the normalized MSE in figure 4.1.

Comparing the two VG methods the difference in test error is perhaps caused by a potential error introduced when/if $m_i = 1$. Of course such an error might also be introduced to VG-dual, since this model also has to 'hard-code' its way out of numerical problems. The solutions obtained by the two VG formulations were compared to the original formulation (not shown) and it was seen that increasing the number of times \mathbf{m} had to be manipulated, worsened the result of VG-KV. VG-dual was less affected.

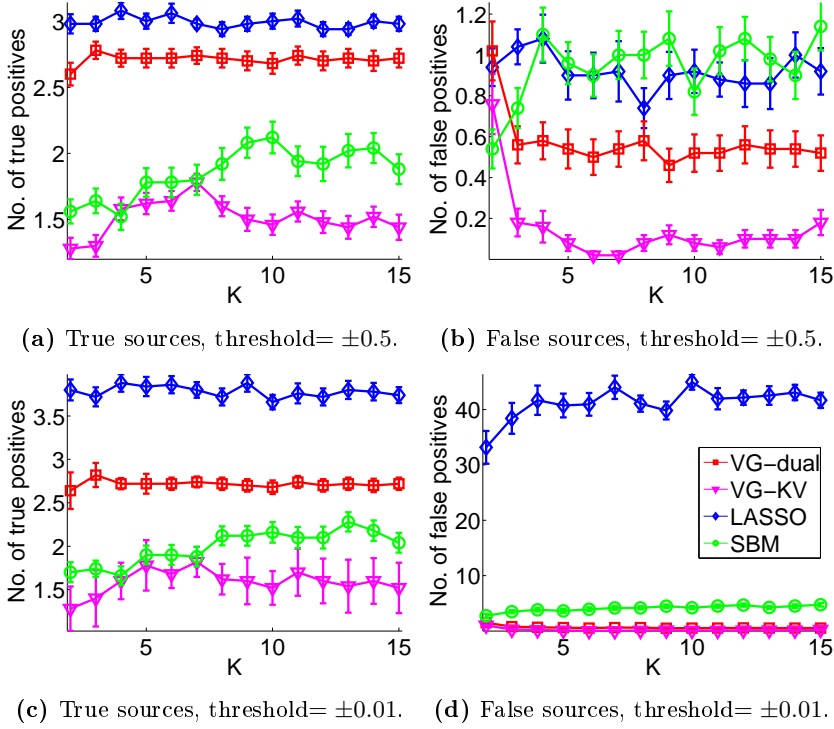


Figure 4.2: Mean number of true/false sources across 50 two-level K -fold cross-validation repetitions with indicated threshold applied to the solutions. The algorithms' performances are evaluated on $K = 2 : 15$. The algorithms are optimized with respect to one parameter; for the VGs the sparsity level γ , for LASSO the regularization parameter λ , and for SBM the precision of the noise β . The solutions of the VGs are in the form of \mathbf{v} , while LASSO's and SBM's are in the form of \mathbf{w} . The actual number of active sources is ten, in total 8196 sources are applied.

The performances of the models are further investigated in figure 4.2. In this figure the number of true sources (active sources in $i = 1 : 10$) and false sources (active sources outside $i = 1 : 10$) is found on two levels of threshold. Note that the thresholds are not scaled to the maximum value in the data. The goal is of course to have ten true sources and zero false sources.

From figure 4.2a and 4.2c it is seen that LASSO obtains most true sources, followed by VG-dual, then SBM and finally VG-KV. LASSO is however also the method which results in most false sources. This is especially true when decreas-

ing the threshold on the weights to 0.01. The two VG methods remain rather stable to the threshold, with the VG-dual solution having slightly fewer false sources on the lower threshold and more true sources on both thresholds. SBM is more stable to the threshold than LASSO but worse than the VG methods.

It should be noted that LASSO and SBM should not need a threshold at all to find sparse solutions, while for VG a 0.5-threshold on the activation \mathbf{m} is natural since $m_i > 0.5$ implies $p(s_i = 1|D) > 0.5$. Additionally the values of \mathbf{m} are seen typically to be either very close to 1 or 0, thus often making the thresholding of \mathbf{m} redundant.

In figure 4.3 the methods are thresholded more fair to their describing algorithms. The VG methods have a threshold applied to their activation parameter \mathbf{m} of 0.5, thus only keeping the sources with a probability greater than 0.5 of

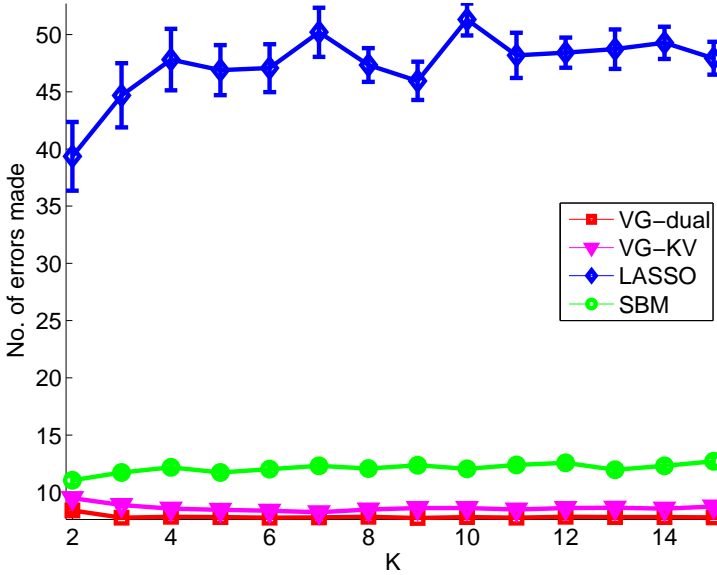


Figure 4.3: Average number of wrong predictions for 50 repetitions of the two-level K -fold cross-validation. K is investigated from 2 to 15. The algorithms are optimized with respect to one parameter; for the VGs the sparsity level γ , for LASSO the regularization parameter λ , and for SBM the precision of the noise β . The solution of the VGs are in the form of \mathbf{v} where a threshold of 0.5 is set on \mathbf{m} . The weights \mathbf{w} found using SBM and LASSO are thresholded at 0.01. In total 8196 sources are applied of which ten are set active.

being active. The weights of LASSO and SBM are thresholded at 0.01. Note that this threshold could have been set even lower. From figure 4.3 it is seen that counting the number of wrongly classified sources, i.e false negatives and false positives a clear distinction between LASSO and the other algorithms is apparent. LASSO makes approximately 47 wrong predictions, SBM 12 and the VGs around eight.

In figure 4.4 the solutions obtained through one ten-fold cross-validation are depicted. The most obvious difference between the solutions of the four methods, is that LASSO has many (small) sources outside the activated area, SBM has a few while the VGs have none. All algorithms have problems estimating the values of the sources they do find, perhaps caused by compensation for their missing sources.

In figure 4.5 the same sources as found in figure 4.4 are visualized in 3D together with the 'true' distribution of the sources. A threshold of 0.5 on \mathbf{m} is set to the VGs and 10^{-10} on LASSO and SBM, corresponding to 'no threshold'. As

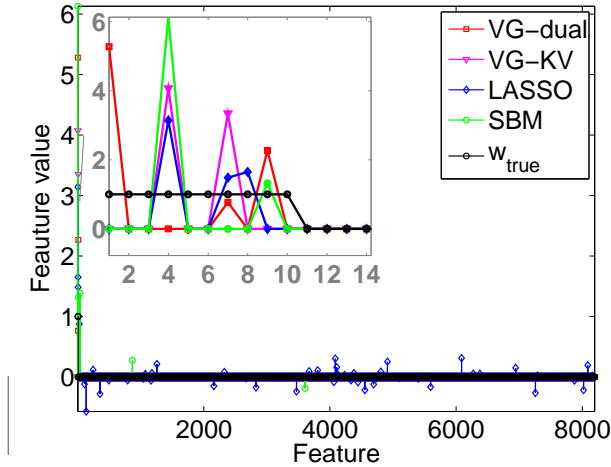


Figure 4.4: Optimum solutions for one example of a ten-fold cross-validation. The algorithms are optimized with respect to one parameter; for the VGs the sparsity level γ , for LASSO the regularization parameter λ , and for SBM the precision of the noise β . The solutions for the VGs correspond to \mathbf{v} , while for LASSO and SBM the solution presented is \mathbf{w} . As seen from the 'true' distribution of sources \mathbf{w}_{true} (black trace) ten sources are active with the value 1, in total 8196 sources are applied. Inset is zoom of the first 14 features.

already seen VG is better at restricting the sources to the active region, with VG-dual finding one 'true' source more than VG-KV.

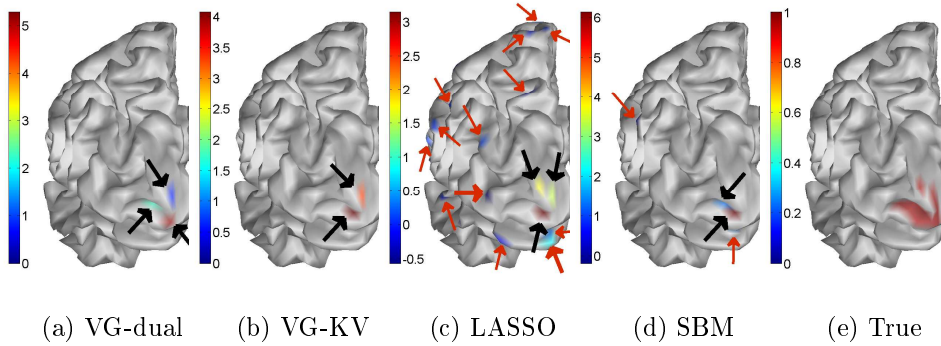


Figure 4.5: Sources estimated in the context of a 3D cortex structure are compared with the 'true' distribution. The algorithms are optimized through ten-fold cross-validation with respect to one parameter; for the VGs the sparsity level γ , for LASSO the regularization parameter λ , and for SBM the precision of the noise β . The solutions for the VGs correspond to \mathbf{v} including a threshold on the activation \mathbf{m} set to 0.5. For LASSO and SBM the solution presented are \mathbf{w} with a threshold on the weights set to 10^{-10} . Heavy or thin arrows indicate sources with magnitudes larger or less than 0.5, respectively. Black arrows indicate true sources and red false sources. View is from the back of the left hemisphere. No sources are found in the right hemisphere for the VGs, only low-strength sources for LASSO and one low strength for SBM. Note individual color maps are used.

4.1.2 Experiment 1.2: Initialization of γ and \mathbf{m} in VG-dual

In this experiment the stability of VG-dual, with respect to the initial level of sparsity applied, is investigated and thereby how the forward and backward solutions are affected by γ_{\min} . These pathway searches are compared to the combined solution, which chooses for each γ the solution from the forward or backward search that gives lowest variational free energy. Additionally the consequence of re-initializing the activation parameters, \mathbf{m} for each γ is examined. The latter makes a forward/backward search redundant, as information obtained from the previous γ is not used.

The source retrieval index F_s is used as a measure of how good the solutions are at reproducing the correct source distribution, see equation (3.3). The results can be seen in figure 4.6.

In figure 4.6a the maximum source retrieval index is found across sparsity in solutions for each γ_{\min} and each repetition. This is done for the forward, backward and combined pathways. For the solutions obtained with re-initialization

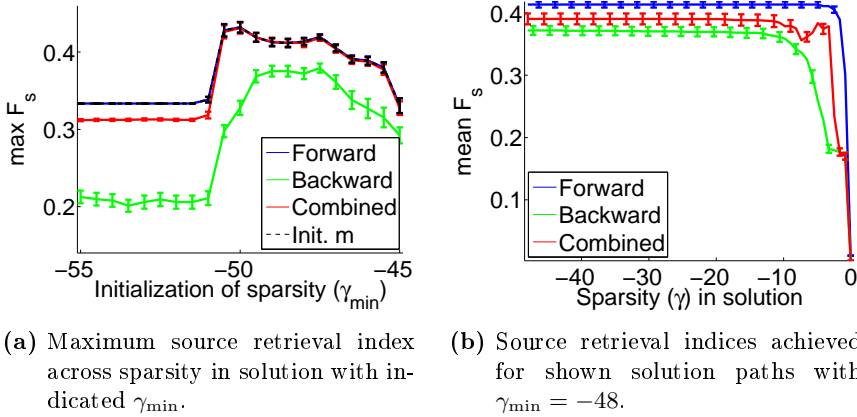


Figure 4.6: Source retrieval index F_s for VG-dual as function of γ_{\min} and applied regularization in solution, respectively. The results are averaged over 50 repetitions of five-fold cross-validations, searching for optimum level of sparsity. Three pathway searches are applied. The forward solution starts at γ_{\min} and ends at γ_{\max} , the backward does the opposite. The combined consists of both the forward and backward solution, where the variational free energy determines the involvement. Finally the solutions where \mathbf{m} is initialized to all zeros for each γ are shown, only included in (a).

of \mathbf{m} for each γ , the maximum source retrieval index is defined to be the solution obtained directly at γ_{\min} , see black dashed line in figure 4.6a. For this reason this solution does not make sense to include in figure 4.6b.

From figure 4.6a it is clear that the dual VG is sensitive to the initialization of γ_{\min} . Additionally it is noticeable that the forward and combined solutions are very similar after $\gamma_{\min} = -52$. After this value of initialization of sparsity, the combined pathway is thus successful in choosing the pathway with highest source retrieval index.

The forward solutions and solutions from re-initializing \mathbf{m} attain the same level of performance. This of course implies that using the connected paths to search for the optimal solution is unnecessary. The reason for this is evident in figure 4.6b, where the mean value of F_s across repetitions is seen for $\gamma_{\min} = -48$. The performances of the methods are seen to be very constant as the sparsity in the solution is reduced.

As the weight distribution found by VG-dual is very much dependent on the initialization of γ , it is no surprise that the backward search performs worse than the others. Remember that the backward search is initialized at $\gamma_{\max} = 0$. In the current application with only ten active sources out of 8196 possible, the solution will thus probably not achieve the sparsity inherent in the data.

4.2 Time-expanded VG-dual

In the following experiments it is investigated whether the time formulation of VG-dual, as described in section 2.5.4, is applicable to EEG settings. In this version of the dual VG the activation parameters are fixed through the time window applied, while the weights are allowed to fluctuate. Again the transposed forward field matrix is utilized as \mathbf{X} . First the performance of the algorithm is shown on synthetic data and finally on the multimodal face-evoked response data.

4.2.1 Experiment 2.1: Performance on synthetic data

The current experiment is made to act as a pre-stage to implementation of actual EEG data. The purpose is to validate the VG-dual time formulation and to find a suitable value of the smoothing parameter η . The sine in figure 2.9, also presented as the green trace in figure 4.8, is applied as the temporal

development of ten sources in the 'true' weight matrix. The mean squared test error, and count of true and false positives as function of η are seen in figure 4.7.

The mean squared test error in figure 4.7a shows several local minima, and it seems as though the relationship between MSE and η is not straightforward. Looking at the true and false positives in figure 4.7b more of a tendency is visible; the number of active sources seems to decrease with η until $\eta = 0.9$. Hereafter, especially for $\eta = 1$, many sources are modeled as being active. Based on 4.7b $\eta = 0.55$ seems a sensible choice, as the number of true positives is relatively high and the number of false positives relatively low. This seems as a reasonable choice looking at the test error too, as a local minimum is present here.

The sources found in one cross-validation run with the chosen level of smoothness of 0.55 are plotted in figure 4.8. The results from figure 4.8 show that although the algorithm is not capable of finding all the active sources it does give a good approximation. Another interesting finding is that taking the mean of the ten first estimated sources returns the appearance of one 'true' source. It can thus be speculated that the algorithm compensates the excess sparsity in the solution by increasing the magnitudes of the found sources.

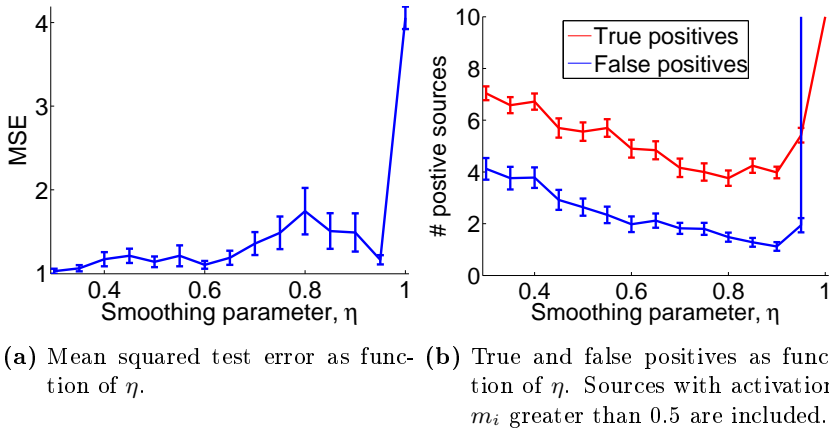


Figure 4.7: A search on η , the smoothing parameter of \mathbf{m} , is conducted with the purpose of optimizing the results of the VG-dual time algorithm. A synthetically generated weight distribution is used. It contains ten active sources out of 8196 possible. The graphs are the result of the average of 50 repetitions of two-level five-fold cross-validations.

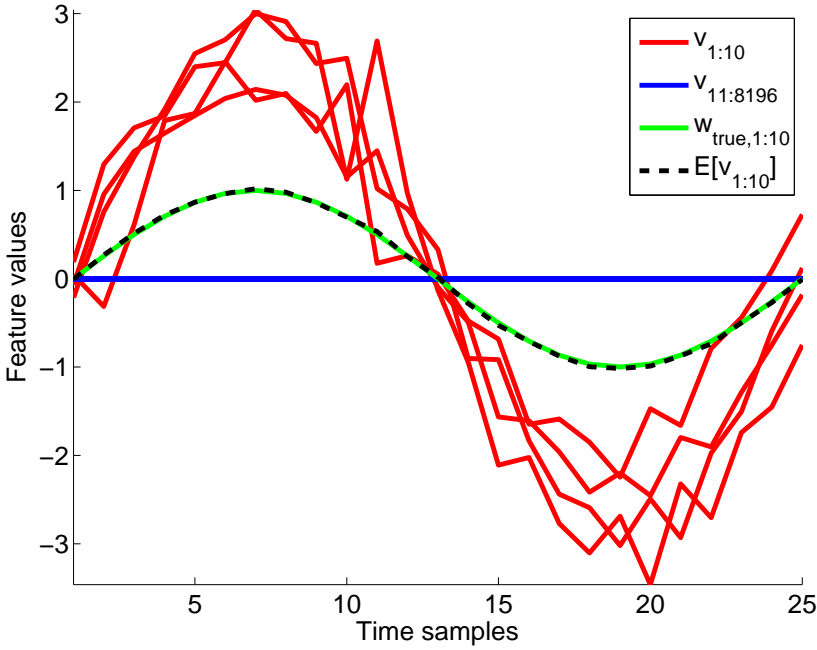
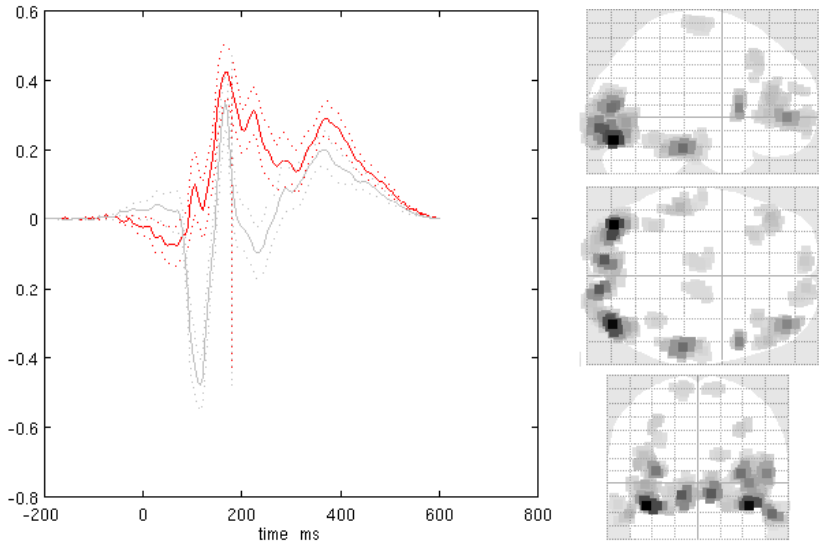


Figure 4.8: One example of the solution to a five-fold cross-validation. The feature values are shown as function of time samples. $\text{SNR}=40$. Synthetic data is applied with 25 time samples. v_i is equal to $m_i w_i$ and denotes the solution obtained for the source in location i in one point in time using time-expanded VG-dual. 'True' appearance of one of the ten equal strength active sources is shown in bright green. Source 1 through 10 have this temporal development, the remaining are constantly 0. The mean value of the first ten estimated sources, $E[\mathbf{v}_{1:10}]$, is also shown.

4.2.2 Experiment 2.2: Performance on differential ERP

The following experiment investigates if the VG-dual time algorithm finds the expected sources on real EEG. The results from running source localization on the differential (faces-scrambled faces) ERP using SPM8 [ACM⁺12] with MSP are seen in section 1.2.4, figure 1.3 and are repeated in figure 4.9 to facilitate comparison with VG-dual. Especially note that the strongest source at 180 ms post-stimulus, found for the differential ERP using SPM8 with MSP, is at location = $[-37, -80, -16 \text{ mm}]$.

Applying VG-dual on the complete ERP, from 200 ms before stimulus to 600 ms



(a) Time development of the strongest source for differential ERP (red) and averaged ERP (gray). Location of strongest source at 180 ms for the differential ERP: $[-37, -80, -16]$ mm. (b) Source reconstruction of differential ERP, time = 180 ms. Sagittal, transverse and coronal views.

Figure 4.9: Results obtained on the multimodal face-evoked data set with SPM8 using MSP. The paradigm applied to reveal the face-evoked response is described by Henson et al. (2003). Modified from [ACM⁺12].

after stimulus, yields at 180 ms the two strongest sources seen in figure 4.10a. The time courses of these two sources have many of the same characteristics as the strongest source obtained by SPM MSP. Especially the peak at 180 ms, which is the N170 component, is visible with both methods. Additionally the position of the strongest source; $[-34.9, -89.9, -22.0]$ mm in the VG-dual solution, approximately matches the position of the strongest source obtained by SPM. Note that the second strongest source obtained by VG-dual is located bilaterally to the strongest source, which seems reasonable since brain functions of the two hemispheres are often located symmetrically.

In figure 4.10b the time frame of the differential ERP is reduced to 100 ms equivalent to 20 time samples; the range is from 100 to 200 ms post stimulus. The same two sources as shown in figure 4.10a are seen to be dominant and to contain similar characteristics in this smaller time frame. The positions of the strongest sources from the two time windows are visualized in the context of

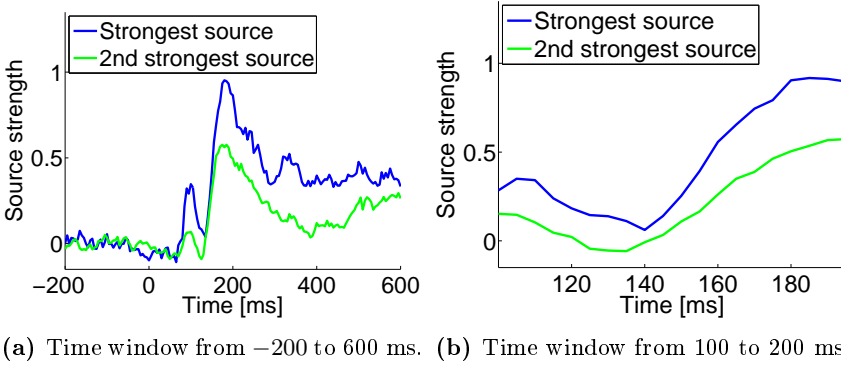


Figure 4.10: The two sources with highest activation at 180 ms found in differential ERP by VG-dual in two time windows. The two implementations give the same two strongest sources which have locations $[-34.9, -89.9, -22.0 \text{ mm}]$ and $[41.1, -78.4, -25.4 \text{ mm}]$. These are found in the left and right occipital lobes, respectively. Five-fold cross-validation is used to find optimum level of sparsity.

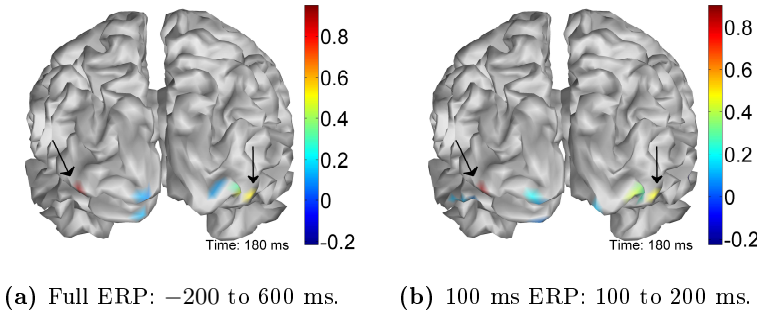


Figure 4.11: Source reconstruction on differential ERP at time instance 180 ms. Arrows point to the two strongest sources. The strongest of the two is found in the left hemisphere. A threshold of 0.5 is imposed on the activation vector \mathbf{m} . The source distribution is found using five-fold cross-validation. Posterior view.

the cerebrum in figure 4.11, where they are marked with arrows.

The results from the differential ERP obtained by the full and 100 ms time window are very similar in the number of sources found, as well as in their locations. This is seen in the glass brain representation of the sources found at

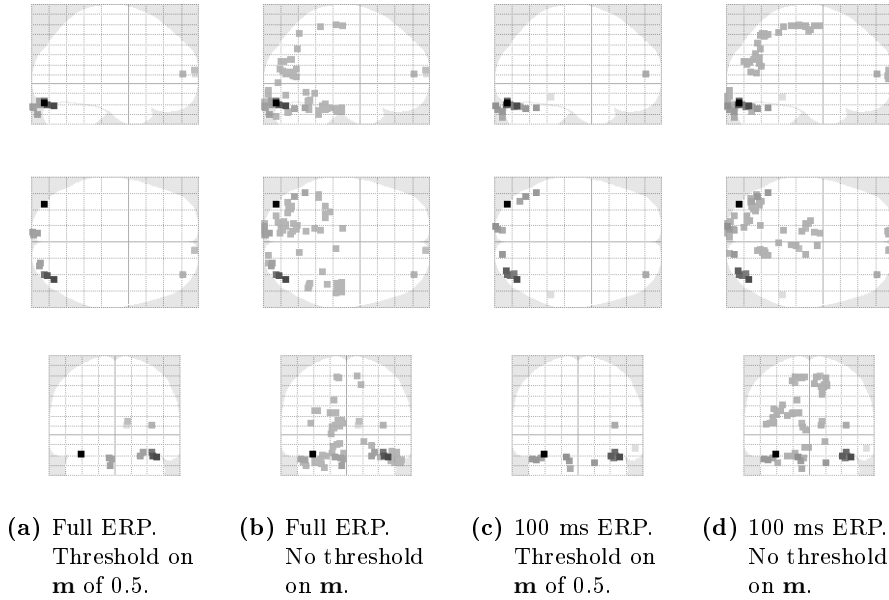


Figure 4.12: Glass brain representation of sources found for differential ERP at time instance 180 ms. The two windows are full ERP: from -200 to 600 ms, and 100 ms ERP: from 100 to 200 ms. The presented source distribution is found using five-fold cross-validation. Sagittal, transverse and coronal views are presented.

180 ms post-stimulus in figure 4.12. In figures 4.12a and 4.12c a threshold of 0.5 is imposed on \mathbf{m} , thus only the sources with probability higher than 0.5 of being active are shown. Both figures show that sources are found mainly in the occipital and temporal lobes, bilaterally. And most non-occipital sources are located in the right hemisphere. However the 100 ms time window has more sources radiating from the occipital lobes to the temporal lobes, and one source clearly in the temporal lobe. Additionally the full time window has two sources in the right frontal lobe where the 100 ms time window only has one. SPM MSP also finds sources in the temporal and frontal lobes, however more scattered and in both hemispheres, though also with the strongest sources on the right.

In figures 4.12b and 4.12d the same results are shown as in figures 4.12a and 4.12c but with no threshold on \mathbf{m} . These of course show more sources active and are more similar to the result obtained by SPM MSP, by additionally showing activity in the fusiform area.

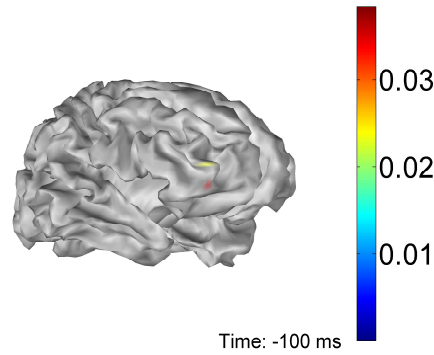


Figure 4.13: 100 ms peristimulus time window. Source reconstruction on differential ERP. Shown is time instance -100 ms. The source distribution is found using five-fold cross-validation. View is such that all sources are visible, here two.

The comparison with the found sources using SPM MSP indicates that the VG-dual time algorithm finds more sparse solutions, but is capable of finding the most dominating sources.

To ensure that the above results are not obtained by chance, a 100 ms window in the peristimulus area is extracted. The sources found at time -100 ms are seen in figure 4.13. The source reconstruction of the peristimulus window reveals only two sources with low magnitude in the right frontal lobe, note the range on the color bar compared to those in figures 4.11a and 4.11b.

4.2.3 Experiment 2.3: Performance on single face epoch

A single face-stimulus epoch is extracted from the face-evoked response data set and a time window from 100 to 200 ms is used to form the employed EEG response. This procedure is repeated several times and the results from two representing epochs are shown here.

A very sparse solution is obtained for epoch 35, visualized in figures 4.14 and 4.15. One source is found in each occipital lobe and two sources are seen in the right temporal lobe. The sources of the occipital lobes are located in the visual cortex and are seen to peak around 160 ms, i.e. similar to the occipito-temporal sources found in the differential ERP. The same applies for the posterior temporal source. The anterior temporal source appears to have a mirrored time

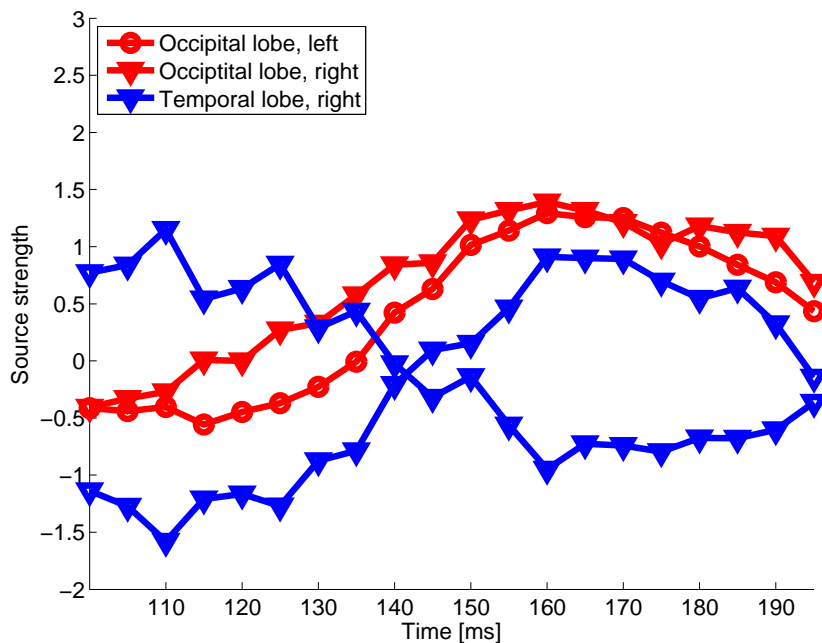


Figure 4.14: Face 35. The sources found in single face-evoked epoch by VG-dual using a time window from 100 to 200 ms. The temporal source distribution is found using five-fold cross-validation.

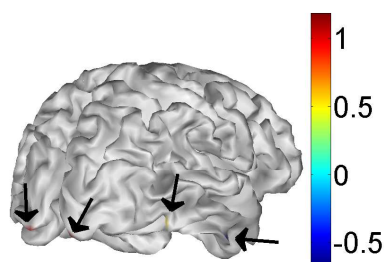


Figure 4.15: Face 35. The sources found in single face-evoked epoch by VG-dual using a time window from 100 to 200 ms, visualized at time instance 180 ms. The source distribution is found using five-fold cross-validation. View is such that all four sources are visible.

course of the three others.

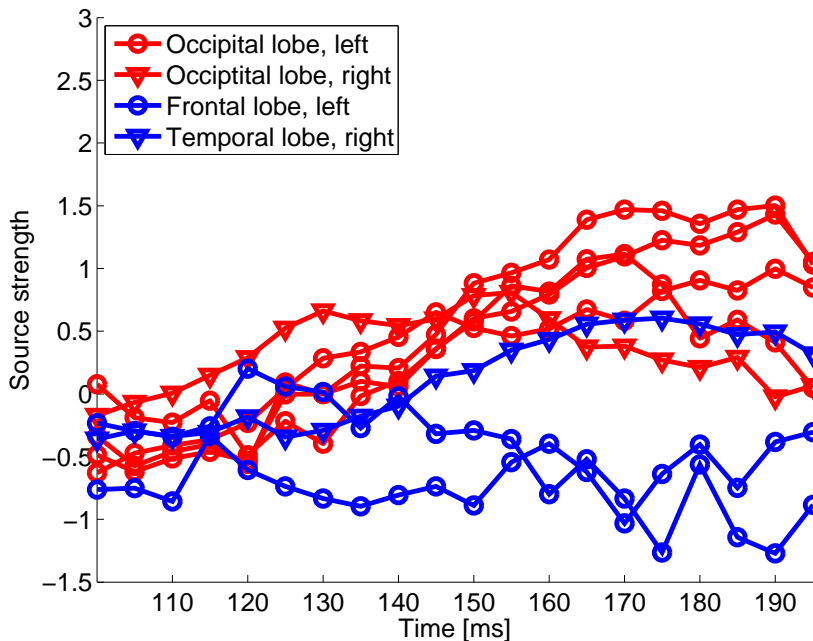


Figure 4.16: Face 173. The sources found in single face-evoked epoch by VG-dual using a time window from 100 ms to 200 ms. The temporal source distribution is found using five-fold cross-validation.

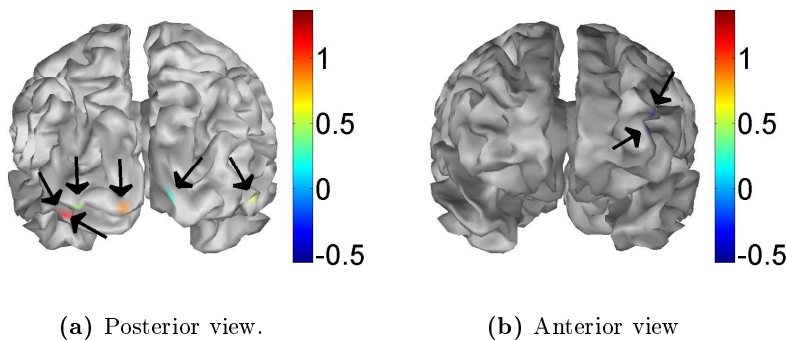


Figure 4.17: Face 173. The sources found in single face-evoked epoch by VG-dual using a time window from 100 to 200 ms, visualized at time instance 180 ms. The source distribution is found using five-fold cross-validation. Views are such that all eight sources are visible.

The source reconstruction from face epoch 173, seen in figures 4.16 and 4.17, shows more activated sources, with most located in the left occipital lobe and in the occipito-temporal area. The found sources, excluding the two sources in the frontal lobe, have peak activity in 160-190 ms. The N170 complex is thus also seen for this epoch.

Most of the examined epochs show activity in the visual cortex and in the occipito-temporal areas. Several also show activity in the frontal lobes, as also seen in the differential ERP and the baseline study. Also unilateral activity of the motor cortex is seen in some epochs, possibly caused by the instructed finger tapping.

Discussion

Solving the highly underdetermined EEG inverse problem often includes either assuming smoothness, e.g. with MNE [HI84, IHCL07, BMG11, PSS11] or LORETA [PMML94, FGPM⁺01, Con06], or sparseness, e.g. with MCE/LASSO [UHS99], dipole fitting [SB91] or Bayesian inference [TBAVVS04, ZR11, PSS11], of the EEG generators. This thesis has focused on applying the sparsity assumption. Not only does it make the results more interpretable it has also been shown to be well suited for EEG applications [FHD⁺08, DPO⁺12, UHS99, TBAVVS04, PM09].

This chapter analyzes the results presented in the previous chapter. The conclusions drawn from the experiments are related to prior studies. The chapter is divided in to three main sections covering the instantaneous algorithms and the dual formulation of VG applied to time windows. The chapter is concluded with reflections that apply to both types of implementation.

5.1 Sparse Algorithms in Single Time

The algorithms discussed in this section; the two VG formulations as well as LASSO and the sparse Bayesian Learning method SBM, all have the following in common; they solve the linear problem under a sparsity assumption.

Additionally their solutions were in this thesis optimized with respect to one parameter through cross-validation.

The two-level cross-validation experiment in section 4.1.1 not only compared the two VG formulations (VG-dual and VG-KV) and LASSO and SBM, but also showed these algorithms' stability with respect to number of folds, K . These results are important for further investigations, as using the optimum value of K will ensure that the algorithms are represented under their most favorable settings. As seen in figure 4.1 the mean squared test error is very stable across number of folds for generally all algorithms. This of course indicates that any of the explored values of K , from 2 to 15, can be applied. That the algorithms are stable to the number of folds is in itself a significant result as it adds to the algorithms' merits.

Comparing the performances of the algorithms with respect to their test errors, see figure 4.1, the dual formulation of VG, SBM and LASSO are found within one standard deviation of each other. The lowest obtainable test error, created by using the 'true' source distribution, had highest proximity to VG-dual, SBM followed very closely, and so did LASSO. These result are very similar to those presented in section 2, where VG was shown to be slightly superior to SBM and even more so to LASSO.

The solutions of the four algorithms were further explored in figure 4.2. Here the number of true and false positives were compared across number of folds in the cross-validation. LASSO was found to obtain most true positives, but also gave many non-predicting variables non-zero weights. The dual formulation of VG showed on average a little fewer true sources than LASSO, but its ability to set the non-predicting variables to zero was much better. For LASSO it thus seems that a higher number of true positives is found with the trade off of lower prediction power in the non-active regions. From figure 4.3 it was seen that the total number of errors made, clearly separates VG-dual from LASSO, with VG-dual being much superior. Another interesting way to evaluate the algorithms would be in the form of a receiver operating curve, however this would not be entirely natural for either of the two algorithms. The solution of LASSO and VG, respectively, would only give one point in the curve, as a threshold is already inherent in the algorithms; for LASSO a threshold of 0, or of a very small value, and for VG a threshold of 0.5 on the activation parameters, are obvious choices.

Judging whether LASSO or VG-dual is better in general, entirely depends on the application. If it is most important to find as many true sources as possible LASSO is slightly better than VG-dual. However if avoiding ghost sources is desired, VG-dual is the most obvious choice between the two. In EEG applications the weighing of the trade offs might also differ. Generally the existence of ghost

sources will confuse the interpretation of the acquired source distribution. They are however often present in linear inverse solutions [TBAVVS04, HNZ⁺08]. Trujillo-Barreto et al. (2004) showed in [TBAVVS04] that using Bayesian inference via Bayesian model averaging, ghost sources could be avoided. The three models using Bayesian inference; VG-dual, VG-KV and SBM, did indeed have fewer false positives than the non-Bayesian LASSO, as seen in figure 4.2d. Among the Bayesian solutions VG-dual found most true sources and was intermediate in the number of false sources (figures 4.2a to 4.2d).

The positions of the wrongly classified sources should also be considered. Many scattered spurious sources far from actual sources might be more confusing than ghost sources located near the 'true' sources. The L_1 -norm has been found to produce sources around the actual source [HTD⁺11]. In figures 4.4 and 4.5 this was investigated for one example. In this presented example the VG formulations did not show any ghost sources, LASSO however had many small. The two strongest were seen to be close to the true region of activation, while the smaller were scattered across the whole cortex, thus partly supporting the claim made about the L_1 -norm by Haufe et al. (2011). SBM had one ghost source close to the active area and one far from it. Supplementary investigations on the location of the ghost sources should be done for it to be used in evaluating the algorithms' performances more generally.

As LASSO only obtained a somewhat higher average of true positives than VG-dual, but had many more false positives, when the threshold was set low, VG-dual was hypothesized to be more applicable to the current problem. For these reasons VG-dual was chosen to expand upon.

In figure 4.1 it was shown that the Kailath Variant formulation had a bit higher test error and some small fluctuations across the number of folds, compared to the three other algorithms. This might be a result of the introduction of numerical errors. These could potentially occur since the possibility of a division with 0 was present in the algorithm, which necessitated an requirement on the activation parameters \mathbf{m} ; if $m_i = 1$ a small value would be subtracted from it. A similar problem is encountered in the dual formulation of VG, and it could therefore be expected that its solution would be affected in the same manner. The formulations of VG were compared (not shown) to the original formulation, seen in equations (2.29) to (2.31). It was found that increasing the number of times a value in \mathbf{m} had to be manipulated the more deviated the Kailath Variant formulation from the results of the original formulation. The deviation was less pronounced for the dual formulation. This could explain the less stable appearance of VG-KV and the higher test error seen in figure 4.1, compared to especially VG-dual.

The two implementations of VG also differed in their search for the optimum

level of sparsity. In the code (from the DMLT toolbox) implementing VG-dual, the solution obtained from the previous level of sparsity was used as initialization of the new level. For VG-KV the parameters were reset for each level of sparsity. This could be expected to influence their respective solutions. However, it does not directly explain why VG-KV performed worse as seen from the experiment on the pathway searches in figure 4.6. From this figure it was found that re-initializing the activation parameter vector \mathbf{m} for each level of sparsity was as good as carrying out a connected pathway search. Most often it will even be better to re-initialize \mathbf{m} , as not choosing the correct initialization of sparsity will cause the algorithm to get stuck in a local optimum. The latter is suspected to be likely to occur, as it was found that the solutions obtained by VG-dual are very much dependent on the initial γ applied. The search for the optimal applied sparsity is however still essential, but only one direction is necessary. Computation time is therefore reduced, as the solution using the found optimum sparsity level does not have to be calculated starting from γ_{\min} and/or γ_{\max} , but can be calculated directly. The main finding was therefore that the levels of sparsity should be evaluated separately.

Additional experiments on the performances of the algorithms could have further revealed their strengths and weaknesses. For example it would have been interesting to use a less or more sparse weight vector as the source distribution, or to include correlations between the input variables. The latter has been examined in [FHT10, KG12, ZY07] for LASSO where it was found that LASSO's performance decrease under specific types of correlation. Also SBMs have been found to be affected by correlated input variables [WRP⁺07]. As the VG is a very new technique, results of its performance by reviewers have not yet been published - note that even the article describing the VG algorithm [KG12] only exists as a preprint. However in this preprint Kappen et al. presented good results for VG on correlated inputs.

In summary the discussed experiments gave insight into the pronounced stability of the algorithms with respect to number of folds to include in cross-validation setups. It showed that under the analyzed conditions VG-dual was superior but also that the algorithm does sacrifice finding some of the actual sources for the benefit of reducing the number of ghost sources. Furthermore it was found that the activation parameter should be initialized for each level of investigated sparsity in order to avoid getting caught in a local optimum. These considerations were taken into account during evaluation of the VG-dual formulation under time-constant activation settings.

5.2 Time-expanded VG-dual

The results obtained with the dual formulation of VG in instantaneous time were applied to the time expanded version, where the activation modes of the sources are assumed constant in the applied time window. It is expected that the nature of the algorithm is mostly unchanged, although it would be interesting to investigate the differences in the two formulations. Convergence could perhaps be found on a lower iteration count for the time-expanded formulation as it exploits knowledge of sources' activity within a range of time. For now it is however assumed that e.g. re-initializing the activation parameter vector \mathbf{m} for each level of sparsity is also the best choice for the time-expanded VG-dual.

Initial investigations on the characteristics of VG-dual using time windows were however performed on the applied smoothing parameter, as indications from prestudies showed relevance of this parameter. Again a synthetic source distribution was applied. The chosen source distribution was a sine wave over a time frame of 100 ms applied to ten neighboring sources. The temporal appearance of each of the ten sources mimicked an alpha frequency wave. EEG signal shows activity in the alpha frequency band in the occipital lobes in a relaxed person with eyes closed [NS06]. Indeed the positions of the ten sources were in one of the occipital lobes, more specifically the left. The smoothing parameter η , which is enforced on the activation parameter vector \mathbf{m} , was by Kappen et al. suggested to initially have the value 1. Only if an update in \mathbf{m} is 'large', is η forced smaller, thus increasing smoothness of the solutions between iterations. Initializing η to be less than 1 will make the initial guess on \mathbf{m} increasingly important, at least for the first couple of iterations. Since \mathbf{m} is initialized to all zeros for each new sparsity level, it could be presumed that an initial low value of η would create more sparse solutions, than e.g. $\eta = 1$. In figure 4.7b it was indicated that the sparsity level of the solution seems to decrease until $\eta = 0.85$, thus in this region the opposite of the expected was found. This can possibly partly be explained by the relatively high number of iterations; i.e. 100.

The number of true and false positives depicted in figure 4.7b showed the same characteristic (by both decreasing with increasing smoothness). Although at $\eta = 0.55$ a small increase in the number of true positives was present, while the number of false positives continued to decrease. Therefore this value was chosen for implementation in the next experiments. The test error in figure 4.7a was difficult to use as guidance for determining η . The smallest value of η gave minimum test error but it was found (not shown) that the active weights did not have much resemblance with an alpha wave when applying small values of η . A local minimum was seen around $\eta = 0.6$, this fits well with the value of η chosen from figure 4.7b.

The value determining whether η should be reduced or not, i.e. the permitted maximum absolute difference between the new and previous \mathbf{m} , was heuristically set. A grid search on this value and η -level would have been to prefer. The number of iterations is another issue that could affect the solution significantly. If few iterations are applied it might be necessary to relax the smoothing as the solution might not otherwise have time to converge. However, since the VG solution seems to be very dependent on the initial parameters, there might not be much difference between iteration numbers.

As explained, the synthetic source distribution imitated an alpha wave, which belongs to the spontaneous type of EEG signals. Applying a synthetic source distribution of a known ERP, could also be interesting. This would indicate if the algorithm's use is appropriate on higher states of brain processing as well.

The above was indirectly and more interestingly done using real EEG data. More specifically the differential ERP from the multimodal face-evoked response data set [HGGG⁺03] was applied. This data set was, as previously mentioned, obtained by presenting visual stimuli consisting of faces or scrambled faces to the enrolled participants. The differential ERP was then obtained by subtracting the averaged scrambled face-evoked response from the averaged face-evoked response. The resulting differential ERP is generally linked with a face-evoked response showing increased activity in areas of the occipito-temporal cortices and fusiform gyri around 150-190 ms [HGGG⁺03]. The peak response is called the N170 complex.

The two strongest sources at 180 ms, found by VG-dual using time windows, shown in figure 4.10a, had the characteristics of the N170 complex. This figure was created using the full length of the ERP as the response. The neighboring figure, where a smaller time window had been extracted around N170, also showed peak activity around 180 ms. The two implementations of the differential ERP resulted in exactly the same locations of the two strongest sources. However all of the sources did not overlap. This is not unexpected as the smaller time window focused on activity in fewer time samples. While the full time window is assumed to find the sources that are dominating over a longer period of time, and/or give relevance to activity occurring before or after the 100 ms time window.

Analyzed fMRI data obtained simultaneous with the described EEG data in [HGGG⁺03] showed that face stimuli triggers activation in bilateral fusiform and lateral ventral occipital regions. While scrambled faces were related to medial and posterior bilateral occipital regions [HGGG⁺03]. Sams et al. (1997) showed in [SHH⁺97] that face specific responses are predominately found in the inferior occipito-temporal cortex using single dipole fitting of MEG data. The greatest difference between faces and non-faces was found around 160 ms poststimulus.

In another MEG study [LHH⁺91] three sources outside the visual cortex was found dominant in face-evoked responses. These were near the occipito-temporal junction, in the inferior parietal lobe and in the middle temporal lobe, listed in chronological order in the time span from 105 to 560 ms from stimuli. Studies on persons with lesions in the occipito-temporal cortices show deficiency in face processing and therefore support the claim of face-processing activity in this region [SHH⁺97, HGGG⁺03].

The VG-dual time formulation showed peak activity in the occipito-temporal areas on the same time scale as the mentioned studies. Additionally the 100 ms time window showed a low magnitude source in the central part of the right temporal lobe. Only when removing the threshold on \mathbf{m} , were sources found in the parietal lobes. Lu et al. (1991) found that the activity in the parietal lobes was also seen doing other forms of visual stimuli [LHH⁺91]. It could therefore be speculated that the parietal response has to some degree been subtracted out of the differential ERP, assuming that the same parietal activity is present for scrambled face stimuli. The fusiform area has, also using the EEG modality, been found to be specific for face-stimulated processes via the MSP method [FHD⁺08]. In VG, only when removing the threshold on the activation parameters, were sources visible in this area. Even though the found activation was low it is noteworthy that the algorithm is capable of finding these sources without a prior assumptions of spatial smoothness.

A study of the baseline activity of the differential ERP was conducted to ensure that baseline correction had been successful in removing spontaneous EEG signal. The results of this study, presented in figure 4.13, showed two very low magnitude sources found in the right frontal lobe, anteriorly. Only this kind of activity is therefore expected to appear as ghost sources in the ERP. Two frontal sources were found applying the full differential ERP window, including the baseline. One was found in the 100 ms time window, from 100 to 200 ms. This phenomenon could either be the result of an algorithm prone to produce low magnitude ghost sources in the frontal lobe or that the baseline correction was not entirely successful. Either way, the frontal sources are of such low magnitude that they do not seem to disturb the interpretation notably.

The application of a single epoch in source reconstruction is very scarce in the literature. Generally an averaging is performed across multiple epochs, thus reducing noise and artifacts. The differential ERP produced in the multimodal face-evoked response is such an example. Additionally in the ERP, two types of responses have been compared, thus further un-clouding information from the EEG data and focusing on the stimuli-evoked processing. Obtaining information about the brain activity on the same time scale as it is produced is however needed, e.g. in EEG biofeedback [PSS11] and BCI [BMG11] applications. The intention with the VG algorithm is to facilitate online tracking of brain activity

from a single epoch, or more precisely from small time windows of a single epoch. Short EEG recordings are thus the only information available for the source reconstruction. Even though this information is possibly more clouded than the averaged ERP, meaningful results have been obtained. Im et al. (2007) were able to show differences of the cortical distribution of alpha activity between patients with dementia and healthy subjects [IHCL07]. They chose to apply MNE for source reconstruction as it has a closed form solution, thus keeping computation time low.

As studies using only a single epoch as the response could not be found in the literature for the face-evoked paradigm, the results of figures 4.14 to 4.17 are compared to general knowledge of brain processing of faces and visual stimuli in general. Dominating sources after visual stimuli should include areas of the visual cortex [SST08]. In many of the inspected face epochs, sources in the visual cortices were reproduced with VG. The more face-specific responses includes activation in the occipito-temporal and fusiform areas, as mentioned. This activation was reproduced in a varying degree by VG. However in none of the investigated face epochs were sources with activation above 0.5 obtained in the fusiform areas. Some epochs additionally showed activation in the motor cortex. The latter can be explained by the finger tapping required to judge the symmetry of the presented images.

A more statistical comparison of the source reconstructions of the single epochs would have been useful to make more quantitative statements about the single epoch results. Perhaps a comparison with the averaged face-evoked response could have given further insight into the applicability of the single epoch. The source reconstruction on the single epoch did however show brain activity in many of the expected areas, and the VG algorithm thus seems to perform well on non-averaged EEG signal as well.

5.3 General Reflections

Two main assumptions have to be fulfilled for the VG algorithm to be applicable to EEG source reconstruction. First of all linearity between the currents generated in the gray matter and the potentials measured at the scalp must be present. This demand is well backed by scientific studies describing the physiology of the brain which models the head as a linear volume conductor [HVG⁺07]. Poisson's equation which is linear, is thus used to relate the currents generated to the potentials measured.

Secondly sparsity is assumed. The presence of sparsity in the number of EEG

generators is difficult to study on a physiological level [SMFK09]. The connections between neurons of the brain have thus only been studied to some degree. It is estimated that approximately 10^{14} connections are found in the brain [NS06]. A comprehensive study showing how many of these are linking close-by neurons and how many transfer information to far away neurons is to be desired. This task is very difficult and even if possible the activation of these connections during stimuli processing must also be known in order to biologically verify functional sparsity [SMFK09]. Studies have however been performed that indicate the existence of localized activity in the brain, e.g. in connections with brain trauma to specific areas, where e.g. face perception is deteriorated after lesions to the occipito-temporal area [SHH⁺97]. Additionally, as also explained in the introduction of this thesis, sparsity has been applied to the EEG inverse problem with success [SHH⁺97, UHS99, DET06, SSL06, FHD⁺08, HNZ⁺08, HTD⁺11, DPO⁺12], and the assumption can therefore be considered reasonable. Especially so if the most relevant sources are found in the sparse representation of the actual source distribution, as seems to be the case with VG.

Inferring spatial smoothness is also a standard technique used in locating EEG sources [TBAVVS04], e.g. LORETA and MSP. MSP obtains smoothness by including smoothing priors where the data supports this. MSP has obtained good results and has facilitated reproduction of the face-evoked response in the fusiform areas, as also found with fMRI and subdural EEG recordings [HGGG⁺03]. So perhaps a next step for the VG algorithm is an expansion where similar activity in neighboring vertices is assumed, thus possibly making the found response in the fusiform area greater.

The current application of VG only gives one degree of freedom to each source. Also in SPM8 [ACM⁺12], source reconstruction is limited to the use of the sources' magnitude component perpendicular to the cortex. It has been shown that dipole orientation does hold important information [PLD⁺05]. Henson et al. (2009) however showed that MSP actually performs best when constraining the dipole direction to the normal of the mesh [HMPF09].

A symmetric BEM head model, generated in SPM8, was used to construct the applied forward field matrix. This limits the reconstruction space to the surface of the cortex. Thus deep sources are projected on to this surface. Using, e.g. an FEM head model, which includes voxels in the entire cerebrum volume, might have improved the solution, including enhancement of the estimation of deep sources. FEM furthermore has the attractive ability to model the conductivity of the head anisotropically [BML01]. As FEM is becoming more competitive with BEM with respect to computation complexity its use is likely to be much increased [WAT⁺06, SSJ⁺10].

Conclusion and Perspectives

The current thesis has described the sparsity enforcing methods; least absolute shrinkage and selection operator (LASSO), sparse Bayesian model (SBM) and variational Garrote (VG), as ways of solving the underdetermined linear problem that is EEG source reconstruction. The dual formulation of VG resulted in a solution with few ghost sources and was a competitor to LASSO with respect to finding the actual predicting variables. This was proven with the application of a synthetic source distribution and an actual forward field matrix, thus imitating EEG settings under controlled conditions.

The dual formulation of VG assuming time-constant activation modes of the sources also showed satisfactory results under these conditions and more importantly, meaningful results on the face-evoked response data set were obtained. As an exact description of the events occurring in processing of face-stimulus does not exist on EEG level, the results obtained were largely verified through similar studies applying EEG, MEG or fMRI. Activation in the fusiform area was only reproduced when removing the threshold on the activation parameters. It is hypothesized that incorporation of spatial smoothness is needed to obtain these sources more clearly, as also done in multiple sparse priors (MSP).

The time window from a single epoch, also from the face-evoked response study, applied to VG-dual returned for many examples of epochs, activation in the visual cortex as well as in the more face-specific areas, including the occipito-

temporal cortices. The VG algorithm thus showed great promise of usability in real-time settings.

Suggestions for improvement of the dual VG, were indicated in the previous chapter. They consist of performing various experiments to tweak the parameters in the model, e.g. the smoothing parameter η and the size of the time window applied. A further description of the performance of the algorithm, e.g. in the presence of correlated input variables and varying the number of active variables, is also desirable. For the time-expanded version of VG-dual it could be interesting to mimic sources as being correlated across time, as higher order brain processing of stimuli includes activation of different centers of the brain on interlocked time scales.

The convergence rates will also be an important next step to investigate, with the goal being application to real-time EEG based imaging. Initial studies revealed that the algorithm converged rather quickly. Additionally computational complexity of the dual and Kailath Variant formulations of VG both scaled with the number of samples used. An expectation of a fast run time is therefore reasonable. Often in real-time EEG imaging minimum norm estimates (MNE) are applied, mostly because of its low computational cost, thus sacrificing model accuracy. Applying the VG algorithm instead, could potentially lead to more precise descriptions of the instantaneous activation pattern of the brain.

An intermediate step between the time indexed VG-dual and actual application to real-time imaging is missing, i.e. a moving time window which allows continually updates of activity. One way would be to create overlapping time windows, with temporal resolution of the representation depending on the degree of overlap, and the time delay reliant on the number of iterations and samples/electrodes. Information from previous time windows could furthermore be used to obtain enhanced solutions. This could be incorporated in the model by modulating the prior on the binary switches to enforce a bias towards activation in the sources found active in the previous time window(s).

This thesis adds to the already extensive field of EEG source localization by employing an algorithm that both assume Ockham's razor to be valid and holds potential for applications to EEG biofeedback. The qualities of VG can thus be exploited in clinical settings, such as training patients with Parkinson's disease to control specific activation of their brain with the objective to reduce symptoms. It is thus the hope that the VG algorithm can be a further step towards understanding the formation of the measured EEG signal, and to use this information to improve the quality of life in persons with neurological disorders.

APPENDIX A

Derivation of VG in Primal Space

The problem of VG is defined as

$$y_\mu = \sum_{i=1}^n w_i s_i X_{i\mu} + \xi_\mu, \quad (\text{A.1})$$

where s_i is either 0 or 1 and its prior is

$$p(\mathbf{s}|\gamma) = \prod_{i=1}^n p(s_i|\gamma), \quad (\text{A.2})$$

where

$$p(s_i|\gamma) = \frac{\exp(\gamma s_i)}{1 + \exp(\gamma)}. \quad (\text{A.3})$$

Variational approximation is used to solve (A.1). First the posterior probability of the model given the data is defined

$$p(\mathbf{s}, \mathbf{w}, \beta | \mathbf{D}, \gamma) = \frac{p(\mathbf{w}, \beta) p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta)}{p(\mathbf{D}|\gamma)}, \quad (\text{A.4})$$

with D being the full data set. Instead of maximizing the posterior probability in equation (A.4)¹, the discrete variable \mathbf{s} has been marginalized out, giving rise to the marginal posterior, $p(\mathbf{w}, \beta | \mathbf{D}, \gamma)$. This expression is to be optimized with respect to \mathbf{w} and β . The denominator in equation (A.4) does not depend on the two latter variables and is therefore not relevant in the maximization. Furthermore defining the joint prior likelihood of \mathbf{w} and β to be uniform, simplifies the problem. The resulting expression to maximize is now

$$\log p(\mathbf{w}, \beta | \mathbf{D}, \gamma) \propto \log \sum_{\mathbf{s}} p(\mathbf{s} | \gamma) p(D | \mathbf{s}, \mathbf{w}, \beta), \quad (\text{A.5})$$

where the logarithm operation has been added in order to make the further derivations simpler. Equation (A.5) is difficult to maximize, by setting the differential coefficient equal to 0, due to the sum inside the logarithm expression. Therefore Jensen's inequality is applied. This approach can be used because the logarithmic function is a concave function. Concavity implies that every point's value on a chord is smaller than that of the function. A point on a chord which has contact points with the concave function $f(x)$ in $(x_1, f(x_1))$ and $(x_2, f(x_2))$ can be described by $(\theta x_1 + (1 - \theta)x_2, \theta f(x_1) + (1 - \theta)f(x_2))$, where $\theta \in [0, 1]$. Due to concavity the following is thus true

$$f(\theta x_1 + (1 - \theta)x_2) \geq \theta f(x_1) + (1 - \theta)f(x_2). \quad (\text{A.6})$$

The above can be rewritten to

$$f(\theta_1 x_1 + \theta_2 x_2) \geq \theta_1 f(x_1) + \theta_2 f(x_2), \quad (\text{A.7})$$

where $\theta_1 + \theta_2 = 1$. By induction the above can then be extended to

$$\begin{aligned} f(\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_i x_i + \dots + \theta_H x_H) \geq \\ \theta_1 f(x_1) + \theta_2 f(x_2) + \dots + \theta_h f(x_h) + \dots + \theta_H f(x_H), \end{aligned} \quad (\text{A.8})$$

hence

$$f\left(\sum_h \theta_h x_h\right) \geq \sum_h \theta_h f(x_h), \quad (\text{A.9})$$

where $\theta_h \geq 0$ and $\sum_h \theta_h = 1$, corresponding to a probability distribution. Now defining $q(\mathbf{s})$ as having the same properties as θ and multiplying and dividing with it in equation (A.5) Jensens's inequality can be applied

¹It would be very complex to find the MAP solution to the 'complete' posterior probability as \mathbf{s} has been defined to be binary.

$$\begin{aligned}
\log \sum_{\mathbf{s}} p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta) &= \log \sum_{\mathbf{s}} \frac{q(\mathbf{s})}{q(\mathbf{s})} p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta) \implies \\
\log \sum_{\mathbf{s}} p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta) &\geq \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta)}{q(\mathbf{s})} \iff \\
\log \sum_{\mathbf{s}} p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta) &\geq - \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta)}. \quad (\text{A.10})
\end{aligned}$$

The variational approximation $q(\mathbf{s})$ is defined by [KG12] to be a fully factorized distribution and satisfies $q(\mathbf{s}) = \prod_{i=1}^n q_i(s_i)$, where $q_i(s_i) = m_i s_i + (1 - m_i)(1 - s_i)$. This implies that m_i is the probability that s_i is equal to 1.

Now defining the variational free energy

$$F(q, \mathbf{w}, \beta) = \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta)}. \quad (\text{A.11})$$

Minimizing $F(q, \mathbf{w}, \beta)$ then corresponds to maximizing the log likelihood in equation (A.5). It is noted that $-F(q, \mathbf{w}, \beta)$ is the lower bound on the log-likelihood and should therefore be maximized, i.e. the same as minimizing $F(q, \mathbf{w}, \beta)$. The latter is expanded

$$\begin{aligned}
F(q, \mathbf{w}, \beta) &= \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma) p(D|\mathbf{s}, \mathbf{w}, \beta)} = -F_1 - F_2 + F_3, \text{ with} \\
F_1 &= \sum_{\mathbf{s}} q(\mathbf{s}) \log p(D|\mathbf{s}, \mathbf{w}, \beta), \quad F_2 = \sum_{\mathbf{s}} q(\mathbf{s}) \log p(\mathbf{s}|\gamma) \text{ and } F_3 = \sum_{\mathbf{s}} q(\mathbf{s}) \log q(\mathbf{s}). \quad (\text{A.12})
\end{aligned}$$

Calculating F_1

Before F_1 is found the likelihood $p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta)$ is defined

$$p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta) = \prod_{\mu=1}^p p(y_{\mu}|\mathbf{X}_{\mu}, \mathbf{s}, \mathbf{w}, \beta) \quad (\text{A.13})$$

The conditional likelihood of one example of the output, y_μ , is assumed to follow a Gaussian distribution centered at $\sum_{i=1}^n w_i s_i X_{i\mu}$ and with variance β^{-1}

$$\begin{aligned} p(y_\mu | \mathbf{X}_\mu, \mathbf{s}, \mathbf{w}, \beta) &= \sqrt{\frac{\beta}{2\pi}} \exp \left(-\frac{\beta}{2} \left(y_\mu - \sum_{i=1}^n w_i s_i X_{i\mu} \right)^2 \right) \\ &= \sqrt{\frac{\beta}{2\pi}} \exp \left(-\frac{\beta}{2} \left(y_\mu^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j s_i s_j X_{i\mu} X_{j\mu} - 2y_\mu \sum_{i=1}^n w_i s_i X_{i\mu} \right) \right). \end{aligned} \quad (\text{A.14})$$

The conditional likelihood for the whole data set using equation (A.13)

$$\begin{aligned} p(\mathbf{D} | \mathbf{s}, \mathbf{w}, \beta) &= \left(\frac{\beta}{2\pi} \right)^{p/2} \exp \left(-\frac{\beta}{2} \left(\sum_{\mu=1}^p \left(y_\mu^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j s_i s_j X_{i\mu} X_{j\mu} - 2y_\mu \sum_{i=1}^n w_i s_i X_{i\mu} \right) \right) \right) \\ &= \left(\frac{\beta}{2\pi} \right)^{p/2} \exp \left(-\frac{p\beta}{2} \left(\sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} - 2 \sum_{i=1}^n w_i s_i b_i \right) \right) \end{aligned} \quad (\text{A.15})$$

Here χ and \mathbf{b} are as defined in section 2.1 and $\sigma_y^2 = \frac{1}{p} \sum_{\mu=1}^p y_\mu^2$. The found expression is plugged into F_1

$$\begin{aligned} \sum_{\mathbf{s}} q(\mathbf{s}) \log p(\mathbf{D} | \mathbf{s}, \mathbf{w}, \beta) &= \\ \sum_{\mathbf{s}} q(\mathbf{s}) \left(\frac{p}{2} \log \left(\frac{\beta}{2\pi} \right) - \left(\frac{p\beta}{2} \left(\sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} - 2 \sum_{i=1}^n w_i s_i b_i \right) \right) \right) \\ \sum_{\mathbf{s}} q(\mathbf{s}) \frac{p}{2} \log \left(\frac{\beta}{2\pi} \right) - \sum_{\mathbf{s}} q(\mathbf{s}) \frac{p\beta}{2} \sigma_y^2 - \sum_{\mathbf{s}} q(\mathbf{s}) \frac{p\beta}{2} \left(\sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} - 2 \sum_{i=1}^n w_i s_i b_i \right). \end{aligned} \quad (\text{A.16})$$

The first two parts only depend on \mathbf{s} in $q(\mathbf{s})$ and since $\sum_{\mathbf{s}} q(\mathbf{s}) = 1$, they can be simplified

$$\sum_{\mathbf{s}} q(\mathbf{s}) \frac{p}{2} \log \left(\frac{\beta}{2\pi} \right) - \sum_{\mathbf{s}} q(\mathbf{s}) \frac{p\beta}{2} \sigma_y^2 = \frac{p}{2} \log \left(\frac{\beta}{2\pi} \right) - \frac{p\beta}{2} \sigma_y^2. \quad (\text{A.17})$$

The third, and final, expression in equation (A.16) is reformulated

$$\begin{aligned} & \frac{p\beta}{2} \sum_{\mathbf{s}} q(\mathbf{s}) \left(\sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} - 2 \sum_{i=1}^n w_i s_i b_i \right) = \\ & \frac{p\beta}{2} \left(\sum_{\mathbf{s}} q(\mathbf{s}) \sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} - \sum_{\mathbf{s}} q(\mathbf{s}) 2 \sum_{i=1}^n w_i s_i b_i \right). \end{aligned} \quad (\text{A.18})$$

The first expression inside the parenthesis

$$\sum_{\mathbf{s}} q(\mathbf{s}) \sum_{i=1}^n \sum_{j=1}^n s_i s_j w_i w_j \chi_{ij} = \sum_{\mathbf{s}} \sum_{i=1}^n \sum_{j=1}^n q(\mathbf{s}) s_i s_j w_i w_j \chi_{ij}. \quad (\text{A.19})$$

Since s_i is binary the following applies

$$\begin{aligned} & \sum_{\mathbf{s}} \sum_{i=1}^n \sum_{j=1}^n q(\mathbf{s}) s_i s_j = \\ & \begin{cases} \sum_{i=1}^n q(s_i) s_i \sum_{j=1}^n q(s_j) s_j = \sum_{i=1}^n \sum_{j=1}^n m_i m_j & \text{for } i \neq j \\ \sum_{i=1}^n q(s_i) s_i^2 = \sum_{i=1}^n q(s_i) s_i = \sum_{i=1}^n m_i & \text{for } i = j \end{cases}, \end{aligned} \quad (\text{A.20})$$

where also using that $\sum_{\mathbf{s}} q(\mathbf{s}) = 1$ and $q(s_i) s_i = m_i$ for $s_i = 1$ and $q(s_i) s_i = 0$ for $s_i = 0$. Finishing equation (A.19)

$$\sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii}, \quad (\text{A.21})$$

where the last sum substitutes the addition of a product with the factor m_i^2 with m_i , thus taking the case where $i = j$ into consideration.

The second expression in equation (A.18)

$$\sum_{\mathbf{s}} q(\mathbf{s}) 2 \sum_{i=1}^n w_i s_i b_i = 2 \sum_{i=1}^n m_i w_i b_i, \quad (\text{A.22})$$

where the result in equation (A.20) is applied.

Combining equations (A.16)-(A.22), F_1 is found

$$F_1 = \frac{p}{2} \log \frac{\beta}{2\pi} - \frac{p\beta}{2} \left(\sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \right). \quad (\text{A.23})$$

Calculating F_2

$$\begin{aligned}
\sum_{\mathbf{s}} q(\mathbf{s}) \log p(\mathbf{s}|\gamma) &= \sum_{\mathbf{s}} q(\mathbf{s}) \sum_{i=1}^n (\gamma s_i - \log(1 + \exp(\gamma))) \\
&= \gamma \sum_{i=1}^n q(s_i) s_i - n \log(1 + \exp(\gamma)) \\
&= \gamma \sum_{i=1}^n m_i - n \log(1 + \exp(\gamma)). \tag{A.24}
\end{aligned}$$

Calculating F_3

$$\begin{aligned}
\sum_{\mathbf{s}} q(\mathbf{s}) \log q(\mathbf{s}) &= \sum_{\mathbf{s}} \left(\prod_{i=1}^n (m_i s_i + (1 - m_i)(1 - s_i)) \sum_{i=1}^n \log(m_i s_i + (1 - m_i)(1 - s_i)) \right) \\
&= \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)) \tag{A.25}
\end{aligned}$$

The total variational free energy

The variational free energy can now be presented

$$\begin{aligned}
F(\mathbf{m}, \mathbf{w}, \beta) &= -\frac{p}{2} \log \frac{\beta}{2\pi} + \frac{p\beta}{2} \sigma_y^2 \\
&\quad + \frac{p\beta}{2} \left(\sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \right) \\
&\quad - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\
&\quad + \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)). \tag{A.26}
\end{aligned}$$

The expression $F(\mathbf{m}, \mathbf{w}, \beta)$ is now minimized by finding its derivatives with respect to \mathbf{w} , \mathbf{m} and β and setting them equal to 0.

Calculating $\frac{\partial F}{\partial w_k} = 0$

$$\begin{aligned}
0 &= \frac{\beta p}{2} \left(\frac{\partial}{\partial w_k} \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \frac{\partial}{\partial w_k} \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2m_k b_k \right) \Longleftrightarrow \\
2m_k b_k &= \frac{\partial}{\partial w_k} \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \frac{\partial}{\partial w_k} \sum_{i=1}^n \sum_{j=1}^n m_i (1 - m_j) w_i w_j \chi_{ij} \delta_{ij} \Longleftrightarrow \\
2m_k b_k &= \frac{\partial}{\partial w_k} \sum_{i=1}^n \sum_{j=1}^n m_i (m_j \chi_{ij} + (1 - m_j) \chi_{ij} \delta_{ij}) w_i w_j \Longleftrightarrow \\
2m_k b_k &= \frac{\partial}{\partial w_k} \sum_{i=1}^n \sum_{j=1}^n m_i \chi'_{ij} w_i w_j, \tag{A.27}
\end{aligned}$$

when defining: $\chi'_{ij} = m_j \chi_{ij} + (1 - m_j) \chi_{jj} \delta_{ij}$, and noting that $(1 - m_j) \chi_{ij} \delta_{ij} = (1 - m_j) \chi_{jj} \delta_{ij}$.

Continuing the derivation

$$\begin{aligned}
2m_k b_k &= \sum_{i=1}^n \sum_{j=1}^n m_i \chi'_{ij} (w_j \delta_{ik} + w_i \delta_{kj}) \Longleftrightarrow \\
2m_k b_k &= \sum_{i=1}^n \sum_{j=1}^n (m_k \chi'_{kj} w_j + m_i w_i \chi'_{ik}) \Longleftrightarrow \\
2m_k b_k &= \sum_{j=1}^n m_k \chi'_{kj} w_j + \sum_{i=1}^n m_i \chi'_{ik} w_i. \tag{A.28}
\end{aligned}$$

The above is expanded to make way for a later simplification

$$\begin{aligned}
2m_k b_k &= \sum_{j=1}^n (m_k \chi_{kj} m_j w_j + m_k (1 - m_j) \chi_{jj} \delta_{kj} w_j) \\
&\quad + \sum_{i=1}^n (m_i \chi_{ik} m_k w_i + m_i (1 - m_k) \chi_{kk} \delta_{ik} w_i) \Longleftrightarrow \\
2m_k b_k &= \sum_{j=1}^n m_k \chi_{kj} m_j w_j + m_k (1 - m_k) \chi_{kk} w_k \\
&\quad + \sum_{i=1}^n m_i \chi_{ik} m_k w_i + m_k (1 - m_k) \chi_{kk} w_k. \tag{A.29}
\end{aligned}$$

Due to symmetry in χ the following is true

$$\begin{aligned}
 2m_k b_k &= 2m_k \sum_{j=1}^n \chi'_{kj} w_j \iff \\
 b_k &= \sum_{j=1}^n \chi'_{kj} w_j \iff \\
 \mathbf{b} &= \chi' \mathbf{w} \implies \\
 \mathbf{w} &= (\chi')^{-1} \mathbf{b}.
 \end{aligned} \tag{A.30}$$

For the final expression to be true the inverse of χ' of course has to exist, i.e. χ' should be non-singular, otherwise the pseudo-inverse must be applied.

Calculating $\frac{\partial F}{\partial m_k}$

Line two in equation (A.26) is considered first, starting with differentiating the first and second expression inside the parenthesis

$$\begin{aligned}
 &\frac{\partial}{\partial m_k} \left(\sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} \right) \\
 &= \sum_{i=1}^n \sum_{j=1}^n (m_j \delta_{ik} + m_i \delta_{jk}) w_i w_j \chi_{ij} + w_k^2 \chi_{kk} - 2m_k w_k^2 \chi_{kk} \\
 &= \sum_{j=1}^n m_j w_k w_j \chi_{kj} + \sum_{i=1}^n m_i w_k w_i \chi_{ik} + w_k^2 \chi_{kk} - 2m_k w_k^2 \chi_{kk} \\
 &= 2 \sum_{j=1}^n m_j w_k w_j \chi_{kj} + w_k^2 \chi_{kk} - 2m_k w_k^2 \chi_{kk}.
 \end{aligned} \tag{A.31}$$

The third expression differentiated, using equation (A.30)

$$\begin{aligned}
 &\frac{\partial}{\partial m_k} 2 \sum_{i=1}^n m_i w_i b_i = 2w_k b_k \iff \\
 &= 2w_k \sum_{j=1}^n (\chi_{kj} m_j + (1 - m_j) \chi_{jj} \delta_{kj}) w_j \iff \\
 &= 2 \sum_{j=1}^n \chi_{kj} m_j w_j w_k + 2(1 - m_k) \chi_{kk} w_k^2.
 \end{aligned} \tag{A.32}$$

Combining equation (A.31) and (A.32)

$$\begin{aligned}
 & 2 \sum_{j=1}^n m_j w_k w_j \chi_{kj} + w_k^2 \chi_{kk} - 2m_k w_k^2 \chi_{kk} - 2 \sum_{j=1}^n \chi_{kj} m_j w_j w_k - 2(1 - m_k) \chi_{kk} w_k^2 \\
 & = -w_k^2 \chi_{kk}.
 \end{aligned} \tag{A.33}$$

Finishing the partial derivation of F with respect to m_k and setting it equal to 0

$$\begin{aligned}
 0 &= -\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma + \log(m_k) + 1 - \log(1 - m_k) - 1 \iff \\
 \log(1 - m_k) - \log(m_k) &= -\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma \iff \\
 \frac{1 - m_k}{m_k} &= \exp\left(-\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma\right) \iff \\
 m_k &= \left(1 + \exp\left(-\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma\right)\right)^{-1} \\
 m_k &= \sigma\left(\frac{\beta p}{2} w_k^2 \chi_{kk} + \gamma\right),
 \end{aligned} \tag{A.34}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$.

Calculating $0 = \frac{\partial F}{\partial \beta}$

$$\begin{aligned}
 0 &= -\frac{p}{2\beta} + \frac{p}{2} \left(\sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \right) \iff \\
 \frac{1}{\beta} &= \sigma_y^2 + \sum_{i=1}^n \sum_{j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} - 2 \sum_{i=1}^n m_i w_i b_i \tag{A.35}
 \end{aligned}$$

Looking only at the last expression in the above equation and inserting the found expression for b_i

$$\begin{aligned}
 2 \sum_{i=1}^n m_i w_i b_i &= 2 \sum_{i=1}^n m_i w_i \sum_{j=1}^n (\chi_{ij} m_j w_j + (1 - m_j) \chi_{jj} w_j \delta_{ij}) \\
 &= 2 \sum_{i=1}^n \sum_{j=1}^n m_i w_i (\chi_{ij} m_j w_j + (1 - m_j) \chi_{jj} w_j \delta_{ij}) \\
 &= 2 \sum_{i=1}^n \sum_{j=1}^n m_i w_i \chi_{ij} m_j w_j + 2 \sum_{i=1}^n m_i (1 - m_i) \chi_{ii} w_i^2.
 \end{aligned} \tag{A.36}$$

Thus the inverse precision can be simplified to

$$\frac{1}{\beta} = \sigma_y^2 - \sum_{i=1}^n m_i w_i b_i. \quad (\text{A.37})$$

APPENDIX B

Details of VG-code

The following is a description of how Kappen et al. suggest implementing VG. In the current thesis, not all of the steps are carried out as proposed.

First, specifications of the data set in example 1 in [KG12] is given:

- Define dimensions on input, $n = 100$, and number of samples, $p = 50$.
- Define input data: $X_{i\mu} \in \mathcal{N}(0, 1)$.
- Construct output data by $y_\mu = \sum_{i=1}^n \hat{w}_i X_{i\mu} + d\xi_\mu$, with:
 - $d\xi_\mu \sim \mathcal{N}(0, \hat{\sigma})$, where $\hat{\sigma} = 1$,
 - $\hat{w}_1 = 1$ and $\hat{w}_i = 0$ for $i \neq 1$,

The implementation of VG:

- Preprocess data so input and output have zero mean; $\frac{1}{p} \sum_{\mu=1}^p X_{i\mu} = 0$
and $\frac{1}{p} \sum_{\mu=1}^p y_\mu = 0$.
- Compute $\sigma_y^2 = \frac{1}{p} \sum_{\mu=1}^p y_\mu^2$.

- Construct the input-output covariance vector \mathbf{b} by: $b_i = \frac{1}{p} \sum_{\mu=1}^p X_{i\mu} y_\mu$.
- Compute the input covariance matrix if $n < p$ by: $\chi_{ij} = \frac{1}{p} \sum_{\mu=1}^p X_{i\mu} X_{j\mu}$
- Compute minimum sparsity input by equation (20) in [KG12] (slightly modified to remove suspected error)

$$\gamma_{\min} = -\frac{pb_i^2}{2\sigma_y^2 \chi_{ii}} + \sigma^{-1}(\epsilon) + \mathcal{O}(\epsilon), \quad (\text{B.1})$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ and $\epsilon = 0.001$ (for example 1 in [KG12]).

- Define $\gamma_{\max} = 0.02\gamma_{\min}$ and $\Delta\gamma = -0.02\gamma_{\min}$.
- Initialize \mathbf{m} by all zeros
- For $\gamma = \gamma_{\min} : \Delta\gamma : \gamma_{\max}$
 - Define $\eta = 1$
 - Iterate until parameters have converged.
 - * If $n < p$ use equations 9-10 in [KG12], corresponding to equations (2.29) and (2.31) in this thesis.

$$\mathbf{w} = (\boldsymbol{\chi}')^{-1} \mathbf{b}, \quad \text{where } \chi'_{ij} = \chi_{ij} m_j + (1 - m_j) \chi_{jj} \delta_{ij} \quad (\text{B.2})$$

$$\frac{1}{\beta} = \sigma_y^2 - \sum_{i=1}^n m_i w_i b_i. \quad (\text{B.3})$$

- * If $n > p$ use equations 15-19 in [KG12], corresponding to equations (2.44) to equation (2.48).

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{1 - m_i} \frac{X_{i\mu} X_{i\nu}}{\chi_{ii}} \quad (\text{B.4})$$

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_\nu = y_\mu \quad (\text{B.5})$$

$$\frac{1}{\beta} = \frac{1}{p} \sum_{\mu=1}^p \hat{y}_\mu y_\mu \quad (\text{B.6})$$

$$\lambda_\mu = \beta \hat{y}_\mu \quad (\text{B.7})$$

$$w_i = \frac{1}{\beta p \chi_{ii}} \frac{1}{1 - m_i} \sum_{\mu=1}^p \lambda_\mu X_{i\mu}. \quad (\text{B.8})$$

* Compute smoothed version of \mathbf{m}

$$m'_i = (1 - \eta)m_i + \eta\sigma(\gamma + \frac{\beta p}{2}w_i^2\chi_{ii}). \quad (\text{B.9})$$

* If $\max_i |m'_i - m_i| > 0.1$ then $\eta = \frac{1}{2}\eta$.

* $\mathbf{m} = \mathbf{m}'$

– Store the found \mathbf{w} , \mathbf{m} , β for the specific value of γ .

– Compute the variational free energy:

$$\begin{aligned} F(\gamma) = & -\frac{p}{2} \log \frac{\beta}{2\pi} + \frac{\beta p}{2} \left(\sum_{i,j=1}^n m_i m_j w_i w_j \chi_{ij} + \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} \right. \\ & \left. - 2 \sum_{i=1}^n m_i w_i b_i + \sigma_y^2 \right) - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\ & + \sum_{i=1}^n (m_i \log m_i + (1 - m_i) \log(1 - m_i)). \end{aligned}$$

- Repeat above for $\gamma = \gamma_{\max} : -\Delta\gamma : \gamma_{\min}$.
- For each γ use the forward or backward algorithm's solution, determined by the one with lowest variational free energy.
- Use the validation error to find the optimum solution.

APPENDIX C

Extensions to VG Kailath Variant Formulation

In this appendix the Kailath Variant formulation of VG is presented more thoroughly than in section 2.5.2. Especially the element-wise calculations are expanded on.

The parameter χ' is rewritten using Kailath Variant, which is expressed as $(\mathbf{A} + \mathbf{B}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$

Breaking χ' into \mathbf{A} , \mathbf{B} and \mathbf{C}

$$\mathbf{A} = \text{diag}((1 - m_j)\chi_{jj})_{j=1:n} \iff A_{ij} = \begin{cases} (1 - m_j)\chi_{ij} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \quad (\text{C.1})$$

$$\mathbf{B} = \frac{1}{p}\mathbf{X} \iff B_{i\mu} = \frac{X_{i\mu}}{p} \quad (\text{C.2})$$

$$\mathbf{C} = \mathbf{X}^T \text{diag}(\mathbf{m}) \iff C_{\mu j} = X_{j\mu}m_j \quad (\text{C.3})$$

$$\mathbf{A} + \mathbf{B}\mathbf{C} = \text{diag}((1 - m_j)\chi_{jj})_{j=1:n} + \frac{1}{p}\mathbf{X}\mathbf{X}^T \text{diag}(\mathbf{m}) = \chi'. \quad (\text{C.4})$$

Here the operation $\text{diag}(\mathbf{d})$ refers to inserting vector \mathbf{d} in a diagonal matrix. The first expression to invert is \mathbf{A} , giving a diagonal matrix with $A_{ii}^{-1} = 1/A_{ii}$. Expressed in matrix form: $\mathbf{A}^{-1} = \text{diag}(1 \oslash ((1 - \mathbf{m}) \odot \chi_{diag})) = \text{diag}(\mathbf{a}_{inv})$, where χ_{diag} is a n -vector with elements from the diagonal in the covariance

matrix χ . The notations \oslash and \odot indicate an element-wise division and multiplication, respectively. Note that if an element in \mathbf{m} is 1, \mathbf{A}^{-1} is not computable. This can however be fixed by replacing such instances by $1 - \epsilon$, where ϵ is a small number.

In order to increase computation efficiency additionally, \mathbf{b} is included in the derivation of χ' and thus yielding \mathbf{w} directly. The expression to compute is then: $(\mathbf{A} + \mathbf{B}\mathbf{C})^{-1}\mathbf{b} = \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}\mathbf{b}$. The first expression can be calculated element-wise as

$$A_{jj}^{-1}b_j = \frac{b_j}{A_{jj}} = \frac{b_j}{(1 - m_j)\chi_{jj}}, \quad (\text{C.5})$$

or in matrix form

$$\mathbf{A}^{-1}\mathbf{b} = (1 \oslash ((1 - \mathbf{m}) \odot \chi_{diag})) \odot \mathbf{b} = \mathbf{a}_{inv} \odot \mathbf{b}. \quad (\text{C.6})$$

Looking at the second expression that demands inversion

$$\begin{aligned} \mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B} &= \mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m}) \text{diag}(1 \oslash ((1 - \mathbf{m}) \odot \chi_{diag})) \frac{1}{p} \mathbf{X} \\ &= \mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m} \odot \mathbf{a}_{inv}) \frac{1}{p} \mathbf{X} = \mathbf{D} \quad \text{or} \end{aligned} \quad (\text{C.7})$$

$$(\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})_{\mu\nu} = \delta_{\mu\nu} + \sum_{j=1}^n C_{\mu j} A_{jj}^{-1} B_{j\nu} = D_{\mu\nu}. \quad (\text{C.8})$$

Finishing the calculation of \mathbf{w} in vector form

$$\mathbf{w} = \mathbf{a}_{inv} \odot \mathbf{b} - \frac{1}{p} \mathbf{a}_{inv} \odot \left(\mathbf{X} \left(\left(\mathbf{I} + \mathbf{X}^T \text{diag}(\mathbf{m} \odot \mathbf{a}_{inv}) \frac{1}{p} \mathbf{X} \right)^{-1} (\mathbf{X}^T (\mathbf{m} \odot \mathbf{a}_{inv}) \odot \mathbf{b}) \right) \right). \quad (\text{C.9})$$

Note that the multiple parentheses ensures that an $n \times n$ -matrix is not created, thereby a big matrix is avoided and computation time is reduced.

Now \mathbf{w} element-wise, when defining the pseudo-inverse of \mathbf{D} as \mathbf{F}

$$\begin{aligned} w_i &= \frac{b_i}{A_{ii}} - \sum_{j=1}^n \left(A_{ii}^{-1} \sum_{\nu=1}^p \sum_{\mu=1}^p B_{i\mu} F_{\mu\nu} \sum_{k=1}^n \frac{C_{\nu k}}{A_{jj}} \delta_{kj} \right) b_j \\ &= \frac{b_i}{A_{ii}} - \sum_{j=1}^n \sum_{\nu=1}^p \sum_{\mu=1}^p A_{ii}^{-1} B_{i\mu} F_{\mu\nu} \frac{C_{\nu j}}{A_{jj}} b_j. \end{aligned} \quad (\text{C.10})$$

Inserting **A**, **B** and **C** finally returns w_i

$$w_i = \frac{b_i}{(1 - m_i)\chi_{ii}} - \sum_{j=1}^n \sum_{\nu=1}^p \sum_{\mu=1}^p \frac{1}{p(1 - m_i)\chi_{ii}} X_{i\mu} F_{\mu\nu} \frac{(X^T)_{\nu j}}{(1 - m_j)\chi_{jj}} b_j. \quad (\text{C.11})$$

APPENDIX D

Extensions to Dual Formulation of VG

Derivation of the fixed point equations, which solve the dual formulation of VG, are presented in this appendix. The partial derivatives of the variational free

energy F , see equation (2.43), are

$$\begin{aligned}\frac{\partial F}{\partial w_k} &= \frac{\beta p}{2} 2m_k(1 - m_k)w_k\chi_{kk} - \sum_{\mu=1}^p \lambda_\mu m_k X_{k\mu} \\ &= m_k \left(\beta p(1 - m_k)w_k\chi_{kk} - \sum_{\mu=1}^p \lambda_\mu X_{k\mu} \right)\end{aligned}\quad (\text{D.1})$$

$$\begin{aligned}\frac{\partial F}{\partial z_\mu} &= \frac{\beta}{2}(2z_\mu - 2y_\mu) + \lambda_\mu \\ &= \beta(z_\mu - y_\mu) + \lambda_\mu\end{aligned}\quad (\text{D.2})$$

$$\frac{\partial F}{\partial \beta} = -\frac{p}{2\beta} + \frac{1}{2} \sum_{\mu=1}^p (z_\mu - y_\mu)^2 + \frac{p}{2} \sum_{i=1}^n m_i(1 - m_i)w_i^2\chi_{ii} \quad (\text{D.3})$$

$$\begin{aligned}\frac{\partial F}{\partial m_k} &= \frac{\beta p}{2}(1 - 2m_k)w_k^2\chi_{kk} - \gamma + 1 + \log(m_k) - \log(1 - m_k) - 1 - \sum_{\mu=1}^p \lambda_\mu w_k X_{k\mu} \\ &= \frac{\beta p}{2}(1 - 2m_k)w_k^2\chi_{kk} - \gamma + \log\left(\frac{m_k}{1 - m_k}\right) - \sum_{\mu=1}^p \lambda_\mu w_k X_{k\mu}\end{aligned}\quad (\text{D.4})$$

$$\frac{\partial F}{\partial \lambda_\mu} = z_\mu - \sum_{i=1}^n m_i w_i X_{i\mu}. \quad (\text{D.5})$$

First the partial derivative of F with respect to w_k is set equal to zero

$$\begin{aligned}0 &= m_k \left(\beta p(1 - m_k)w_k\chi_{kk} - \sum_{\mu=1}^p \lambda_\mu X_{k\mu} \right) \Longleftrightarrow \\ w_k &= \frac{1}{\beta p(1 - m_k)\chi_{kk}} \sum_{\mu=1}^p \lambda_\mu X_{k\mu}.\end{aligned}\quad (\text{D.6})$$

The same is done to find z_μ

$$\begin{aligned}0 &= \beta(z_\mu - y_\mu) + \lambda_\mu \Longleftrightarrow \\ z_\mu &= y_\mu - \frac{1}{\beta}\lambda_\mu.\end{aligned}\quad (\text{D.7})$$

The above two results are used to find the remaining variables, which partial

derivatives also are set to 0. First revision of equation (D.3)

$$\begin{aligned}
\frac{\partial F}{\partial \beta} = 0 &= -\frac{p}{2\beta} + \frac{1}{2} \sum_{\mu=1}^p (z_\mu - y_\mu)^2 + \frac{p}{2} \sum_{i=1}^n m_i(1 - m_i)w_i^2 \chi_{ii} \iff \\
\frac{1}{\beta} &= \frac{1}{p} \sum_{\mu=1}^p (z_\mu - y_\mu)^2 + \sum_{i=1}^n m_i(1 - m_i)w_i^2 \chi_{ii} \iff \\
\frac{1}{\beta} &= \frac{1}{p} \sum_{\mu=1}^p (y_\mu - \frac{1}{\beta} \lambda_\mu - y_\mu)^2 + \\
&\quad \sum_{i=1}^n m_i(1 - m_i) \chi_{ii} \frac{1}{\beta p(1 - m_i) \chi_{ii}} \sum_{\mu=1}^p \lambda_\mu X_{i\mu} \frac{1}{\beta p(1 - m_i) \chi_{ii}} \sum_{\nu=1}^p \lambda_\nu X_{i\nu} \iff \\
\frac{1}{\beta} &= \frac{1}{p} \sum_{\mu=1}^p (-\frac{1}{\beta} \lambda_\mu)^2 + \frac{1}{\beta^2 p^2} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_\mu X_{i\mu} \lambda_\nu X_{i\nu} \iff \\
\beta &= \frac{1}{p} \sum_{\mu=1}^p \lambda_\mu^2 + \frac{1}{p^2} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_\mu X_{i\mu} \lambda_\nu X_{i\nu} \iff \\
\beta &= \frac{1}{p} \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_\mu \lambda_\nu \left(\delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu} \right) \iff \\
\beta &= \frac{1}{p} \sum_{\mu=1}^p \sum_{\nu=1}^p A_{\mu\nu} \lambda_\mu \lambda_\nu \tag{D.8}
\end{aligned}$$

with

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu}. \tag{D.9}$$

Continuing with equation (D.5)

$$\begin{aligned}
\frac{\partial F}{\partial \lambda_\mu} = 0 &= z_\mu - \sum_{i=1}^n m_i w_i X_{i\mu} \iff \\
0 &= y_\mu - \frac{1}{\beta} \lambda_\mu - \sum_{i=1}^n \frac{m_i X_{i\mu}}{\beta p(1 - m_i) \chi_{ii}} \sum_{\nu=1}^p \lambda_\nu X_{i\nu} \iff \\
\beta y_\mu &= \sum_{\nu=1}^p \lambda_\nu \left(\delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\nu} X_{i\mu} \right) \iff \\
\beta y_\mu &= \sum_{\nu=1}^p \lambda_\nu A_{\mu\nu} \tag{D.10}
\end{aligned}$$

Additionally introducing \hat{y}

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_\nu = y_\mu \quad (\text{D.11})$$

and making the following derivations

$$\begin{aligned} \sum_{\nu=1}^p A_{\mu\nu} \hat{y}_\nu &= \frac{1}{\beta} \sum_{\nu=1}^p A_{\mu\nu} \lambda_\nu \iff \\ 0 &= \sum_{\nu=1}^p A_{\mu\nu} \left(\frac{1}{\beta} \lambda_\nu - \hat{y}_\nu \right) \implies \\ \lambda_\nu &= \beta \hat{y}_\nu \end{aligned} \quad (\text{D.12})$$

and using equations (D.8) and (D.10)

$$\begin{aligned} \beta &= \frac{1}{p} \sum_{\mu=1}^p \lambda_\mu \sum_{\nu=1}^p A_{\mu\nu} \lambda_\nu \iff \\ \beta &= \frac{1}{p} \sum_{\mu=1}^p \lambda_\mu \beta y_\mu \iff \\ \beta &= \frac{1}{p} \sum_{\mu=1}^p \beta \hat{y}_\mu \beta y_\mu \iff \\ \frac{1}{\beta} &= \frac{1}{p} \sum_{\mu=1}^p \hat{y}_\mu y_\mu \end{aligned} \quad (\text{D.13})$$

equations (2.46) and (2.47) are obtained. Finally equation (2.30) is derived

$$\begin{aligned} \frac{\partial F}{\partial m_k} = 0 &= \frac{\beta p}{2} (1 - 2m_k) w_k^2 \chi_{kk} - \gamma + \log \left(\frac{m_k}{1 - m_k} \right) - \sum_{\mu=1}^p \lambda_\mu w_k X_{k\mu} \iff \\ 0 &= \frac{\beta p}{2} (1 - 2m_k) w_k^2 \chi_{kk} - \gamma + \log \left(\frac{m_k}{1 - m_k} \right) - w_k^2 \beta p (1 - m_k) \chi_{kk} \iff \\ 0 &= \beta p w_k^2 \chi_{kk} \left(\frac{1}{2} - m_k - 1 + m_k \right) - \gamma + \log \left(\frac{m_k}{1 - m_k} \right) \iff \\ \log \left(\frac{1 - m_k}{m_k} \right) &= -\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma \iff \\ m_k &= \left(1 + \exp \left(-\frac{\beta p}{2} w_k^2 \chi_{kk} - \gamma \right) \right)^{-1}. \end{aligned} \quad (\text{D.14})$$

APPENDIX E

Extensions to Time-expanded VG-dual

The following expands on the calculations given in section 2.5.4.

Original dual representation of F

$$\begin{aligned} F(\mathbf{m}, \mathbf{w}, \beta, \mathbf{z}, \lambda) = & -\frac{p}{2} \log \frac{\beta}{2\pi} + \frac{\beta}{2} \sum_{\mu=1}^p (z_{\mu} - y_{\mu})^2 + \frac{p\beta}{2} \sum_{i=1}^n m_i (1 - m_i) w_i^2 \chi_{ii} \\ & - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\ & + \sum_{i=1}^n (m_i \log(m_i) + (1 - m_i) \log(1 - m_i)) \\ & + \sum_{\mu=1}^p \lambda_{\mu} \left(z_{\mu} - \sum_{i=1}^n m_i w_i X_{i\mu} \right) \end{aligned} \tag{E.1}$$

Dual representation of F with time dependent \mathbf{w} , \mathbf{y} , \mathbf{z} and λ

$$\begin{aligned}
F(\mathbf{m}, \mathbf{w}, \beta, \mathbf{z}, \lambda) = & -\frac{Tp}{2} \log \frac{\beta}{2\pi} + \frac{\beta}{2} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{p\beta}{2} \sum_{t=1}^T \sum_{i=1}^n m_i(1-m_i) w_{it}^2 \chi_{ii} \\
& - \gamma \sum_{i=1}^n m_i + n \log(1 + \exp(\gamma)) \\
& + \sum_{i=1}^n (m_i \log(m_i) + (1-m_i) \log(1-m_i)) \\
& + \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} \left(z_{\mu t} - \sum_{i=1}^n m_i w_{it} X_{i\mu} \right). \tag{E.2}
\end{aligned}$$

Notice that only the parts in the above equation stemming from the likelihood term in the variational free energy, i.e. equation (2.25), are affected by the summation over time samples.

The procedure of finding the parameters follows that of the VG primal and dual formulation. The partial derivatives of F are found and subsequently set to 0.

The partial derivatives

$$\frac{\partial F}{\partial w_{it}} = \beta p m_i (1 - m_i) \chi_{ii} w_{it} - \sum_{\mu=1}^p \lambda_{\mu t} m_i X_{i\mu} \tag{E.3}$$

$$\frac{\partial F}{\partial z_{\mu t}} = \beta (z_{\mu t} - y_{\mu t}) + \lambda_{\mu t} \tag{E.4}$$

$$\frac{\partial F}{\partial \beta} = -\frac{Tp}{2\beta} + \frac{1}{2} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{p}{2} \sum_{t=1}^T \sum_{i=1}^n m_i (1 - m_i) w_{it}^2 \chi_{ii} \tag{E.5}$$

$$\frac{\partial F}{\partial m_i} = \frac{\beta p}{2} \sum_{t=1}^T (1 - 2m_i) w_{it}^2 \chi_{ii} - \gamma + \log \left(\frac{m_i}{1 - m_i} \right) - \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} w_{it} X_{i\mu} \tag{E.6}$$

$$\frac{\partial F}{\partial \lambda_{\mu t}} = z_{\mu t} - \sum_{i=1}^n m_i w_{it} X_{i\mu} \tag{E.7}$$

$$\cdot \tag{E.8}$$

Solving for $\frac{\partial F}{\partial w_{it}} = 0$

$$w_{it} = \frac{1}{p\beta(1 - m_i)\chi_{ii}} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu}, \tag{E.9}$$

and for $\frac{\partial F}{\partial z_{\mu t}}$

$$z_{\mu t} = y_{\mu t} - \frac{1}{\beta} \lambda_{\mu t}. \quad (\text{E.10})$$

These equations are used in the following. Starting with $\frac{\partial F}{\partial \beta} = 0$

$$\begin{aligned} \frac{Tp}{2\beta} &= \frac{1}{2} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{p}{2} \sum_{t=1}^T \sum_{i=1}^n m_i (1 - m_i) w_{it}^2 \chi_{ii} \iff \\ \frac{1}{\beta} &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p (z_{\mu t} - y_{\mu t})^2 + \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^n m_i (1 - m_i) w_{it}^2 \chi_{ii} \iff \\ \frac{1}{\beta} &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \left(y_{\mu t} - \frac{1}{\beta} \lambda_{\mu t} - y_{\mu t} \right)^2 + \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^n \frac{\chi_{ii} m_i (1 - m_i)}{p^2 \beta^2 (1 - m_i)^2 \chi_{ii}^2} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu} \sum_{\nu=1}^p \lambda_{\nu t} X_{i\nu} \iff \\ \beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t}^2 + \frac{1}{Tp^2} \sum_{t=1}^T \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} X_{i\mu} \lambda_{\nu t} X_{i\nu} \iff \\ \beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} \lambda_{\nu t} \delta_{\mu\nu} + \frac{1}{Tp^2} \sum_{t=1}^T \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} \lambda_{\nu t} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu} \iff \\ \beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} \lambda_{\nu t} \left(\delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu} \right) \iff \\ \beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \sum_{\nu=1}^p \lambda_{\mu t} \lambda_{\nu t} A_{\mu\nu}, \end{aligned} \quad (\text{E.11})$$

when defining

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i) \chi_{ii}} X_{i\mu} X_{i\nu}. \quad (\text{E.12})$$

Next $\frac{\partial F}{\partial \lambda_{\mu t}} = 0$

$$\begin{aligned}
0 &= z_{\mu t} - \sum_{i=1}^n m_i w_{it} X_{i\mu} \iff \\
y_{\mu t} - \frac{1}{\beta} \lambda_{\mu t} &= \sum_{i=1}^n m_i \frac{1}{p\beta(1-m_i)\chi_{ii}} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu} X_{i\mu} \iff \\
\beta y_{\mu t} &= \lambda_{\mu t} + \sum_{\nu=1}^p \sum_{i=1}^n \frac{m_i}{p(1-m_i)\chi_{ii}} \lambda_{\nu t} X_{i\nu} X_{i\mu} \iff \\
\beta y_{\mu t} &= \sum_{\nu=1}^p \lambda_{\nu t} \left(\delta_{\mu\nu} + \sum_{i=1}^n \frac{m_i}{p(1-m_i)\chi_{ii}} X_{i\nu} X_{i\mu} \right) = \sum_{\nu=1}^p \lambda_{\nu t} A_{\mu\nu}. \quad (\text{E.13})
\end{aligned}$$

Introducing

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu t} = y_{\mu t}, \quad (\text{E.14})$$

and inserting this in equation (E.13) yields

$$\begin{aligned}
\beta \sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu t} &= \sum_{\nu=1}^p \lambda_{\nu t} A_{\mu\nu} \iff \\
0 &= \sum_{\nu=1}^p A_{\mu\nu} \left(\frac{1}{\beta} \lambda_{\nu t} - \hat{y}_{\nu t} \right) \implies \\
\lambda_{\nu t} &= \beta \hat{y}_{\nu t}. \quad (\text{E.15})
\end{aligned}$$

Inserting (E.15) and (E.14) in (E.11) yields a simplification of β

$$\begin{aligned}
\beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} \sum_{\nu=1}^p \lambda_{\nu t} A_{\mu\nu} \iff \\
\beta &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \beta \hat{y}_{\mu t} \beta y_{\mu t} \iff \\
\frac{1}{\beta} &= \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \hat{y}_{\mu t} y_{\mu t}. \quad (\text{E.16})
\end{aligned}$$

Using from equation (E.9) $w_{it}p\beta(1 - m_i)\chi_{ii} = \sum_{\mu=1}^p \lambda_{\mu t}X_{i\mu}$, \mathbf{m} is derived

$$\begin{aligned}
\log\left(\frac{1 - m_i}{m_i}\right) &= \frac{\beta p}{2} \sum_{t=1}^T (1 - 2m_i) w_{it}^2 \chi_{ii} - \gamma - \sum_{t=1}^T \sum_{\mu=1}^p \lambda_{\mu t} w_{it} X_{i\mu} \iff \\
\log\left(\frac{1 - m_i}{m_i}\right) &= \beta p \sum_{t=1}^T \left(\frac{1}{2} - m_i\right) w_{it}^2 \chi_{ii} - \gamma - \sum_{t=1}^T w_{it} w_{it} p \beta (1 - m_i) \chi_{ii} \iff \\
\log\left(\frac{1 - m_i}{m_i}\right) &= \beta p \sum_{t=1}^T w_{it}^2 \chi_{ii} \left(\frac{1}{2} - m_i - 1 + m_i\right) - \gamma \iff \\
\log\left(\frac{1 - m_i}{m_i}\right) &= -\frac{\beta p}{2} \sum_{t=1}^T w_{it}^2 \chi_{ii} - \gamma \iff \\
m_i &= \left(1 + \exp\left(-\frac{\beta p}{2} \sum_{t=1}^T w_{it}^2 \chi_{ii} - \gamma\right)\right)^{-1} \\
&= \sigma\left(\frac{\beta p}{2} \chi_{ii} \sum_{t=1}^T w_{it}^2 + \gamma\right). \tag{E.17}
\end{aligned}$$

The final equation set is then

$$w_{it} = \frac{1}{p\beta(1 - m_i)\chi_{ii}} \sum_{\mu=1}^p \lambda_{\mu t} X_{i\mu}, \tag{E.18}$$

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i}{(1 - m_i)\chi_{ii}} X_{i\mu} X_{i\nu}, \tag{E.19}$$

$$\sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu t} = y_{\mu t}, \tag{E.20}$$

$$\frac{1}{\beta} = \frac{1}{Tp} \sum_{t=1}^T \sum_{\mu=1}^p \hat{y}_{\mu t} y_{\mu t}, \tag{E.21}$$

$$\lambda_{\nu t} = \beta \hat{y}_{\nu t} \quad \text{and} \tag{E.22}$$

$$m_i = \sigma\left(\frac{\beta p}{2} \chi_{ii} \sum_{t=1}^T w_{it}^2 + \gamma\right). \tag{E.23}$$

Vector/matrix implementation of the above

$$\mathbf{W}_{n \times T} = (1 \oslash (p\beta(1 - \mathbf{m}) \odot \chi_{\text{diag}}) \cdot \mathbf{1}'_T) \odot (\mathbf{X} \cdot \boldsymbol{\lambda}), \quad (\text{E.24})$$

$$\mathbf{A}_{p \times p} = \mathbf{I}_{p \times p} + \frac{1}{p} \mathbf{X}^T \cdot \text{diag}(\mathbf{m} \oslash ((1 - \mathbf{m}) \odot \chi_{\text{diag}})) \cdot \mathbf{X}, \quad (\text{E.25})$$

$$\hat{\mathbf{Y}}_{p \times T} = \mathbf{A} \setminus \mathbf{Y}, \quad (\text{E.26})$$

$$\beta_{1 \times 1} = Tp / \text{sum}(\hat{\mathbf{Y}} \odot \mathbf{Y}), \quad (\text{E.27})$$

$$\boldsymbol{\lambda}_{p \times T} = \beta \hat{\mathbf{Y}} \quad \text{and} \quad (\text{E.28})$$

$$\mathbf{m}_{n \times 1} = \sigma \left(\frac{\beta p}{2} \text{sum}(\mathbf{W}^2, 2) + \gamma \right). \quad (\text{E.29})$$

$\mathbf{1}_T$ denotes a column vector of ones of size T .

APPENDIX F

Selected Matlab Implementations

F.1 Two-level Cross-validation

```
1 function [testMSE trueMSE y_test_var REG_opt] = ...
2     twolevel_crossval(Maxit,Minreg,Maxreg,Nreg,Convalue,Maxrep)
3 %
4 % Two-level K-fold cross-validation setup
5 %
6 % twolevel_crossval – describes the steps involved in a two level
7 % K-fold cross-validation experiment. The script can be used
8 % to test the stability of an algorithm with respect to K, and
9 % to find the optimal number of folds.
10 %
11 % Inputs: Maxit:    Number of maximum iterations performed
12 %           Minreg:  Minimum regularization applied
13 %           Maxreg:  Maximum regularization applied
14 %           Nreg:    Number of regularization levels
15 %           Convalue: Convergence is obtained at this value
16 %           Maxrep:  Number of repetitions, max 50 is allowed
17 %
18 % Outputs: testMSE:    Test MSE found by algorithm
19 %           trueMSE:    MSE found with w_true
20 %           y_test_var: Variance on test set outputs
21 %           REG_opt:    Found optimum regularizations
```

```

22 %
23 % The setup of this script has been used to compare the performances
24 % of the algorithms; variational Garrote, in the Kailath Variant and
25 % in the dual formulation, as well as LASSO and a sparse Bayesian
26 % model with linear basis
27 %
28 % For each algorithm one parameter is optimized through
29 % cross-validation.
30 % The data is split into test and training sets, the latter is
31 % further divided into a smaller training set and a validation set.
32 %
33 % This script shows the implementation of the Kailath Variant
34 % formulation, thus it inputs the function 'vgKV'.
35 %
36 % Created by Sofie Therese Hansen, s072331.
37 %
38
39 % Load data file where data has been randomized across samples:
40 load data_EEG_split_comb_rand
41     % contains x_all (input), y_all (output)
42     % and w_true (true weight distribution)
43
44 [nd,ns] = size(x_all); % No. of dimensions on x and no. of samples
45
46 load samps % Matrix dictating which ten samples to use as test set
47     % for 50 repetitions
48 pt = size(samps,2); % Number of test samples, here 10
49
50 % Define range of regularization/parameter to optimize on:
51 REG = linspace(Minreg,Maxreg,Nreg);
52
53 Kfolds = 2:15; % Range of K values to investigate in K-fold
54     % cross-validation
55 nK = length(Kfolds);
56
57 % Preallocations for speed:
58 WW = zeros(nK,Maxrep,nd);
59 testMSE = zeros(nK,Maxrep);
60 trueMSE = zeros(Maxrep,1);
61 y_test_var = zeros(Maxrep,1);
62 REG_opt = zeros(nK,Maxrep);
63 % Main loop:
64 for rep =1:Maxrep
65     tic
66     % Allocating samples and preprocessing test data:
67     indtt = samps(rep,:); % Choose indices for test set samples
68     x_data = x_all; y_data = y_all;
69     x_test = x_data(:,indtt)- repmat(mean(x_data(:,indtt),2),1,pt);
70     y_test = y_data(indtt);
71     y_test = y_test-mean(y_test);
72     % Variance on test sample outputs:
73     y_test_var(rep) = 1/pt*y_test*(y_test)';
74     % Training and validation data:
75     x_data(:,indtt) = [];y_data(indtt)=[];
76     ppv = ns - pt; % Number of samples in training and validation set

```

```

77 % Investigate number of folds, K, in K-fold cross-validation:
78 for K = 2:15;
79     % Find indices for splits:
80     [indices] = crossvalind('Kfold',ppv,K);
81
82     % Preallocate optimum regularizations:
83     reg_min = zeros(K,1);
84     for i = 1:K; % Each split of K folds
85         % Allocate training and validation data:
86         x_val = x_data(:,indices==i);
87         y_val = y_data(indices==i);
88         x = x_data(:,indices~=i);
89         y = y_data(indices~=i);
90         p = size(x,2); % Size of training set
91
92         % Preprocess training and validation data:
93         x = x-repmat(mean(x,2),1,p);
94         x_val = x_val - repmat(mean(x_val,2),1,ppv-p);
95         y = y - mean(y);
96         y_val = y_val - mean(y_val);
97         % Reset validation error for each split in fold:
98         error_val = zeros(Nreg,1);
99
100        % Test each level of regularization/hyperparameter:
101        for nreg = 1:Nreg
102            reg = REG(nreg);
103            % Run algorithm and find solution:
104            v = vgKV(x,y,reg,Maxit,Convalue);
105            % Calculate mean squared validation error:
106            error_val(nreg) = mean((v'*x_val-y_val).^2);
107        end
108        % Find optimum value of regularization
109        % by the validation error:
110        [valmin i_reg] = min(error_val);
111        reg_min(i) = REG(i_reg);
112    end
113    % Find mean optimum level of regularization across splits
114    reg_opt = mean(reg_min);
115    REG_opt(K-1,rep) = reg_opt; % Save optimal parameters
116    % Run algorithm and find solution
117    ptv = size(x_data,2);
118    x_trainval = x_data-repmat(mean(x_data,2),1,ptv);
119    y_trainval = y_data-mean(y_data);
120    vopt = vgKV(x_trainval,y_trainval,reg_opt,Maxit,Convalue);
121    % Save found solution for each K and repetition
122    WW(K-1,rep,:) = vopt;
123    testMSE(K-1,rep) = mean((vopt'*x_test-y_test).^2);
124    % Mean squared test error
125 end
126 % Mean squared test error using true weights
127 trueMSE(rep) = mean((w_true*x_test-y_test).^2);
128 toc
129 end

```

F.2 Kailath Variant Formulation of VG

```

1 function [v m] = vgKV(x,y,gamma,Maxit,Maxdiff)
2 %
3 % vgKV - Kailath Variant formulation of variational Garrote
4 %
5 % This functions calculates the solution to the linear problem
6 % using the variational Garrote, suggested by Kappen et al. (2012)
7 % The Kailath Variant relation is used to reduce computational
8 % complexity.
9 %
10 % Inputs: x:      Input data with samples in columns and input
11 %              dimensions in rows
12 %           y:      Samples of the one dimensional response
13 %           gamma:   Level of sparsity
14 %           Maxit:   Maximum number of iterations
15 %           Maxdiff: Stop when maximum absolute difference between
16 %                   current and old m is smaller than Maxdiff
17 %
18 % Created by Sofie Therese Hansen, s072331.
19 %
20
21 [n p] = size(x);
22 b = 1/p .* x*y'; % Compute input-output covariance vector
23 x_cov_diag = mean(x.*x,2); % Define input covariance matrix
24 y_var = 1/p*y*(y)'; % Variance of outputs
25 eta = 1; % Smoothing parameter
26 m = zeros(n,1); % Initialize m
27 mdiff = 1;
28 k = 1; % First iteration
29 while k < Maxit && (mdiff > Maxdiff);
30     k = k+1;
31     m = min(m,1-1e-10); % Avoid numerical problems
32     % Calculate weights:
33     dA_inv = 1./((1-m).*x_cov_diag);
34     w = dA_inv.*b-dA_inv.*(x./p*(eye(p)+x'.*...
35         repmat(m.*dA_inv,1,p)'*x./p)^(-1)*(x'*(m.*(dA_inv.*b)))));
36     % Variance and precision:
37     beta_inv = y_var-sum(m.*w.*b);
38     beta = 1/beta_inv;
39     % Updated m is computed:
40     tmp2 = gamma+beta*p/2*w.^2.*x_cov_diag;
41     m_mark = (1-eta).*m+eta*(1+exp(-tmp2)).^(-1);
42
43     % If maximum absolute difference between current and previous m
44     % is bigger than 0.1 decrease eta to increase smoothing:
45     mdiff = max(abs(m_mark-m));
46     if mdiff > 0.1
47         eta = eta/2;
48     end
49     m = m_mark;
50
51 end

```

```

52 V = m.*w;
53 end

```

F.3 Time-expanded VG-dual

```

1  %-----
2  % Time-expanded version of dual formulation of variational Garrote
3  %-----
4  % Locations of activation are held active for the specified time
5  % window.
6  % This script generates synthetic weights corresponding to ten
7  % sines in ten locations. A response is created using this weight
8  % distribution together with a forward field matrix. This can easily
9  % be substituted with real EEG, by replacing y with the measured EEG.
10 %
11 % A five-fold cross-validation setup is applied to the 128 channels.
12 %
13 % Created by Sofie Therese Hansen, s072331.
14 %-----
15 sigmoid1 = @(x) 1./(1+exp(-x));
16
17 % Load forward field matrix
18 load SPMgainmatrix_aceMdsmp8_faces_run1_2_c
19 [p n] = size(G);
20 x_data = G';
21 x_data = x_data-mean(x_data,2)*ones(1,p);%
22 dx = sqrt(1/p*sum(x_data.^2,2));
23 x_data = x_data./(dx*ones(1,p)); % Scale inputs
24
25 % Create weight distribution of sines:
26 a = 10; % No. of active sources
27 T = 25;
28 sinus = sin(linspace(0,2*pi,T));
29 w_true = zeros(n,T);
30 w_true(1:a,:)=repmat(sinus,a,1);
31
32 % Create outputs:
33 stdev_noise = 1;
34 noise = stdev_noise*randn(size(x_data,2),T);
35 y0 = x_data'*w_true;
36 y_data = y0 + noise;
37 SNR = mean(sum(y_data.^2)./sum(noise.^2));
38
39 % Randomize data in samples
40 randind = randperm(size(x_data,2));
41 x_data = x_data(:,randind);y_data = y_data(randind,:);
42
43 % Store training and validation set:
44 x_tv = x_data;

```

```

45 y_tv = y_data;
46 % Subtract means:
47 x_tvc = x_tv-mean(x_tv,2)*ones(1,p);
48 y_tvc = y_tv-ones(p,1)*mean(y_tv,1);
49
50 % Define sparsity range:
51 gamma_min = -150;
52 gamma_max = 0;
53 n_gamma = 60;
54 gamma_all = linspace(gamma_min,gamma_max,n_gamma);
55
56 kmax = 100; % Max. iterations on search for regularization
57 kmaxopt = 100; % Max. iterations on found optimal regularization
58 minit = zeros(n,1); % Initialize m
59 K = 5; % Number of folds in cross-validation
60 [indices] = crossvalind('Kfold',p,K); % Find split
61 % Preallocations:
62 error_val = zeros(1,n_gamma);
63 Gamma = zeros(1,K);
64 %% Main cross-validation loop
65 for kf = 1:K
66     pv = sum(indices==kf);
67     x_val = x_tv(:,indices==kf);
68     x_val = x_val-mean(x_val,2)*ones(1,pv);
69     y_val = y_tv(indices==kf,:);
70     y_val = y_val-ones(pv,1)*mean(y_val,1);
71     y_val_var = std(y_val).^2;
72     x_train = x_tv(:,indices~=kf);
73     x_train = x_train-mean(x_train,2)*ones(1,p-pv);
74     y_train = y_tv(indices~=kf,:);
75     y_train = y_train-ones(p-pv,1)*mean(y_train,1);
76     pt = p-pv;
77     chi_ii = 1/pt*sum(x_train.^2,2); % Diagonal of input covariance
78     % Regularization loop:
79     for i = 1:n_gamma;
80         m=minit % Initialize m
81         eta = 0.55; % Initialize eta
82         gamma = gamma_all(i);
83         % Iterate equation set
84         for k = 1:kmax
85             m=min(m,1-1e-8);
86             A = eye(pt)+1/pt*x_train'*spdiags(m./(1-m)./...
87                 chi_ii,0,n,n)*x_train;
88             yhat = A\y_train;
89             yhaty = yhat.*y_train;
90             beta = T*pt/sum(yhaty(:));
91             lambda = beta*yhat;
92             w = (1./(pt*beta*(1-m).*chi_ii)*ones(1,T)).*...
93                 (x_train*lambda);
94             w2 = w.^2;
95             mold = m;
96             m = (1-eta)*mold + eta*sigmoid1(beta*pt/2*sum(w2,2)).*...
97                 chi_ii+gamma);
98             % If maximum absolute difference between current and
99             % previous m is bigger than 0.1 decrease eta to

```

```

100         % increase smoothing:
101         if max(abs(m-mold)) > 0.05
102             eta = eta/2;
103         end
104
105     end
106     % Calculate mean validation MSE across time samples
107     v = w.*repmat(m,1,T); % Solution
108     error_val(i) = mean(mean((x_val'*v-y_val).^2));
109 end
110 [temp imin] = min(error_val);
111 Gamma(kf) = gamma_all(imin);
112 end
113 gamma_mean = mean(Gamma); % Optimum sparsity level
114 %% Input found optimum sparsity in both training and val. set
115 chi_ii = 1/p*sum(x_tvc.^2,2); % Diagonal of chi
116 m = minit; % Initialize m
117 eta = 0.55; % Initialize eta
118 for k = 1:kmaxopt
119     m = min(m,1-1e-8);
120     A = eye(p)+1/p*x_tvc'*spdiags(m./(1-m)./chi_ii,0,n,n)*x_tvc;
121     yhat = A\y_tvc;
122     yhaty = yhat.*y_tvc;
123     beta = T*p/sum(yhaty(:));
124     lambda = beta*yhat;
125     w = (1./(p*beta*(1-m).*chi_ii)*ones(1,T)).*(x_tvc*lambda);
126     w2 = w.^2;
127     mold = m;
128     m = (1-eta)*mold + eta*sigmoid1(beta*p/2*sum(w2,2).*...
129         chi_ii+gamma_mean);
130     % If maximum absolute difference between current and previous m
131     % is bigger than 0.1 decrease eta to increase smoothing:
132     if max(abs(m-mold)) > 0.05
133         eta = eta/2;
134     end
135 end
136 % Calculate optimum solution:
137 v_opt = w.*repmat(m,1,T);

```


APPENDIX G

Submission to ICASSP2013

SPARSE SOURCE EEG RECONSTRUCTION WITH THE VARIATIONAL GARROTE

Sofie Therese Hansen, Carsten Stahlhut, and Lars Kai Hansen

DTU Informatics, Technical University of Denmark,
Building 324, Kgs. Lyngby, DENMARK

ABSTRACT

EEG imaging is an extremely ill-posed inverse problem. Based on recent work (Delorme et al., 2011) we hypothesize that solutions of interest are sparse. We show that direct search for sparse solutions as implemented by the Variational Garrote (VG, Kappen 2011) can outperform solutions based on convex relaxations (Lasso) both in terms of cross-validation error on test data and in terms of sparsity of the solution.

Index Terms— EEG, Imaging, Variational Garrote, Lasso, Sparsity

1. INTRODUCTION

We are interested in real-time imaging of human brain function by electroencephalography (EEG). The EEG imaging problem is of significant theoretical interest and real-time EEG imaging has many potential applications including quality control, in-line experimental design, brain state decoding, and neuro-feedback. In mobile applications these possibilities are attractive as elements in systems for personal state monitoring and well-being, and indeed in clinical settings were proper care requires imaging under quasi-natural conditions [1]. The first real-time mobile systems are based on reconstruction methods using basic Tikhonov regularization [1]. However, the computational challenges induced by the highly ill-posed nature of the EEG imaging problem escalate in mobile real-time systems and new algorithms may be necessary [2].

In recent work by Delorme et al. [3] it is argued that independent components of EEG signals are dipolar in nature. In particular it was shown that a direct dipolar fit can explain much of the spatially distributed signal measured in scalp electrodes. This suggests localized sparse sources and motivates reconstruction algorithms that emphasize sparsity in contrast to the distributed spatial source patterns promoted in classical alternatives [4].

Unfortunately, the quest for sparse solutions to the EEG imaging problem is entirely non-trivial. We show that the most widely used scheme based on convex relaxation is based

on conditions on the forward model that may not be met. Therefore we here investigate a recent alternative for sparse recovery proposed by Kappen [5]. It is aimed at solving the sparse recovery problem without resorting to convex relaxation, enables separation of the location and magnitude estimation aspects of the reconstruction task, and leads to a relative low-complexity set of non-linear equations that are iterated towards the solution. All known approaches for sparse imaging are based on trade-offs between data fit and sparsity measures. The trade-off is here carried out using two-level cross-validation which allows us both to infer the optimal level of sparsity and provide an un-biased measure of performance.

2. THE EEG INVERSE PROBLEM

In the quasi static approximation the relation between dipolar sources placed at the cortical surface w_i and the measured potentials at multiple scalp locations y_μ is instantaneous and linear,

$$y_\mu = \sum_{i=1}^n w_i X_{i\mu} + \xi_\mu. \quad (1)$$

We have denoted the forward model by $X_{i\mu}$ and allowed for measurement noise ξ_μ , which is further assumed to be independent of the source signal. In a typical laboratory setting the number of measured scalp signals p can be 32–128, while the source distribution can be represented by $n = 1000 - 10.000$ locations. Thus we face a severely underdetermined problem and regularization is necessary to ensure a well-defined solution, see e.g., [6] for an early review. As we have noted key processes appear to be rather dipolar, thus searching for sparse localized solutions seems well-motivated.

Searching for the minimal cardinality source distribution within a given level of misfit represents a non-convex combinatorial optimization problem [7]. Under certain conditions convex relaxations like the least absolute shrinkage and selection operator (Lasso) [8] can be shown to solve the linear regression problem with sparsity constraints. Lasso uses a L_1 penalty on the weights, which produces a sparse solution by forcing some weights to zero and shrinking others using the

This research is supported in part by the Danish Research Council for Technical and Production Sciences and the Lundbeck Foundation.

objective

$$\sum_{\mu=1}^p (y_{\mu} - \mathbf{w}^T \mathbf{X}_{\mu})^2 \quad \text{subject to} \quad \sum_{i=1}^n |w_i| \leq t. \quad (2)$$

The problem can be solved for a range of values of t , if t is small we enforce sparsity. The least angle regression solver (LARS) is a computationally efficient constructive path method that adds one non-zero weight at each step along a path from the zero solution to a dense solution [9]. We will investigate the utility of the LARS approach for EEG imaging using the tools developed by Sjöstrand [10].

3. THE VARIATIONAL GARROTE

It is complex to check whether the conditions for the validity of the convex relaxation are full-filled for a given EEG problem, and therefore we are interested in alternative approaches that aim to solve the sparse approximation problem without these assumptions. The so-called Variational Garrote (VG) introduces sparseness into the regression problem by adding the binary 'location' variable $s_i \in \{0, 1\}$ for absent/present parameters [5]. Thus, the modified linear problem reads

$$y_{\mu} = \sum_{i=1}^n w_i s_i X_{i\mu} + \xi_{\mu}. \quad (3)$$

The location variable is a latent binary variable with a prior

$$p(\mathbf{s}|\gamma) = \prod_{i=1}^n p(s_i|\gamma), \quad \text{where} \quad (4)$$

$$p(s_i|\gamma) = \frac{\exp(\gamma s_i)}{1 + \exp(\gamma)}. \quad (5)$$

Parameter γ will in general be assumed negative $\gamma < 0$, reflecting a bias towards sparsity.

The optimal solution to Eq. ((3)) can be obtained with a variational approximation proposed in [5]. First the posterior probability of the model given the data is established based on a Gaussian noise assumption, $\xi \sim N(0, \beta^{-1})$,

$$p(\mathbf{s}, \mathbf{w}, \beta | \mathbf{D}, \gamma) = \frac{p(\mathbf{w}, \beta) p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta)}{p(\mathbf{D}|\gamma)}, \quad (6)$$

with D being the full data set, while the prior over sources and noise variance is assumed to be uniform $p(\mathbf{w}, \beta) \propto 1$. The discrete variable \mathbf{s} is marginalized out, giving rise to the marginal posterior, $p(\mathbf{w}, \beta | \mathbf{D}, \gamma)$. The denominator does not depend on \mathbf{w} and β and is therefore not relevant in the maximization of these. The resulting expression to maximize is now

$$\log p(\mathbf{w}, \beta | \mathbf{D}, \gamma) \propto \log \sum_{\mathbf{s}} p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta) \quad (7)$$

Expression ((7)) is bounded using Jensen's inequality and introducing a variational posterior over source locations, $q(\mathbf{s})$, a fully factorized distribution with $q(\mathbf{s}) = \prod_{i=1}^n q_i(s_i)$, and factors $q_i(s_i) = m_i s_i + (1 - m_i)(1 - s_i)$ [5]

$$\begin{aligned} \log \sum_{\mathbf{s}} p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta) &\geq - \sum_{\mathbf{s}} q(\mathbf{s}) \log \frac{q(\mathbf{s})}{p(\mathbf{s}|\gamma) p(\mathbf{D}|\mathbf{s}, \mathbf{w}, \beta)} \\ &= -F(q, \mathbf{w}, \beta). \end{aligned} \quad (8)$$

The variational free energy $F(q, \mathbf{w}, \beta)$ is minimized, corresponding to maximizing the log likelihood ((7)). As noted the EEG problem is severely underdetermined, therefore we can simplify the model using the so-called Kernel trick with a dual formulation with update rules for p Lagrange multipliers $\lambda_{\mu}, \hat{y}_{\nu}$ [5]

$$A_{\mu\nu} = \delta_{\mu\nu} + \frac{1}{p} \sum_{i=1}^n \frac{m_i X_{i\mu} X_{i\nu}}{(1 - m_i) \chi_{ii}} \quad (9)$$

$$y_{\mu} = \sum_{\nu=1}^p A_{\mu\nu} \hat{y}_{\nu} \quad (10)$$

$$\frac{1}{\beta} = \frac{1}{p} \sum_{\mu=1}^p \hat{y}_{\mu} y_{\mu} \quad (11)$$

$$\lambda_{\mu} = \beta \hat{y}_{\mu} \quad (12)$$

$$w_i = \frac{1}{\beta p \chi_{ii} (1 - m_i)} \sum_{\mu=1}^p \lambda_{\mu} X_{i\mu} \quad (13)$$

$$m_i = \left(1 + \exp \left(-\frac{\beta p}{2} w_i^2 \chi_{ii} - \gamma \right) \right)^{-1}. \quad (14)$$

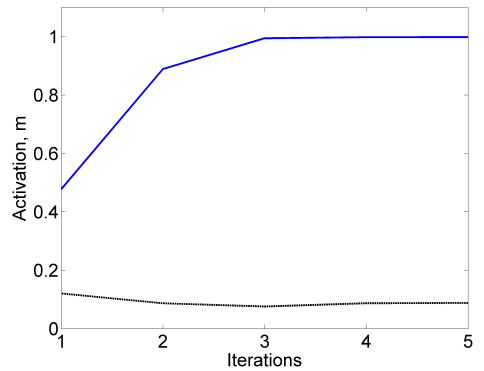


Fig. 1. Simulation with a single active source. Activation of the true source shown in blue. The dotted black line represents the sum of all 'false' sources' activation.

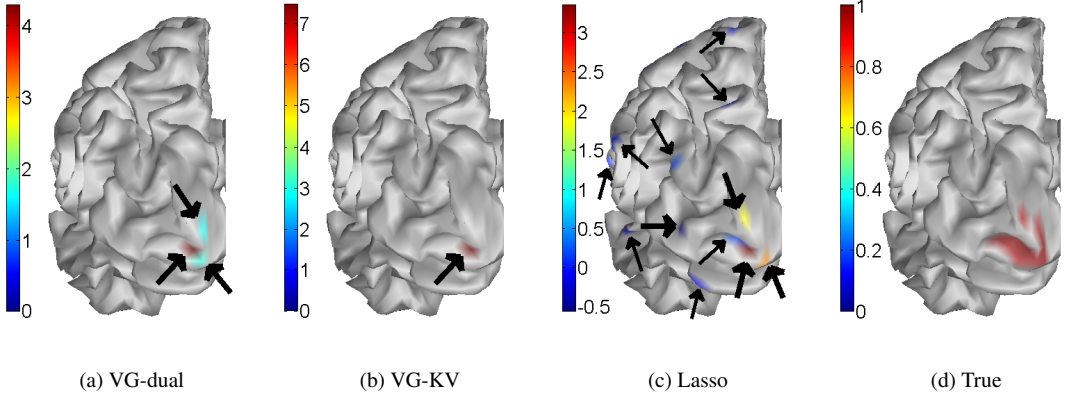


Fig. 4. Sources estimated within a single 10-fold cross-validation run in the context a 3D cortex structure and compared with the true distribution. For VG a threshold on the activation, \mathbf{m} , is set to 0.5. Heavy arrows indicate sources with magnitude larger than 0.5 and thin arrows indicate sources below this value. View is from the back of the left hemisphere. No sources are found in the right hemisphere for VG and only a few low-strength sources for Lasso. Note individual color maps are used.

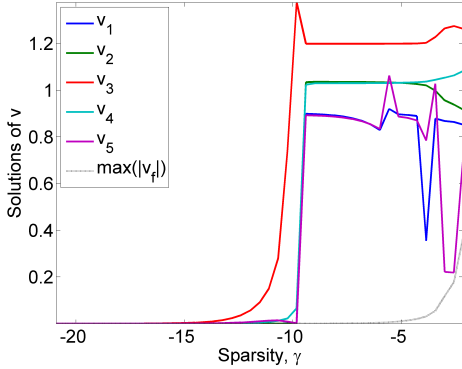


Fig. 2. Simulation of five active sources. True sources (\mathbf{v}) are shown in color together with the false source having the highest absolute value (dotted gray line).

4. SIMULATIONS

We investigate the Variational Garrote (VG) in a series of simulation experiments. The first is based on a random forward model, while the latter are quasi realistic simulations using State-of-the-Art high-dimensional EEG forward models. For the first set of simulations we form $p = 50$ measurements and $n = 100$ unknown sources

$$y_\mu = \sum_{i=1}^n w_i X_{i\mu} + \xi_\mu, \quad (15)$$

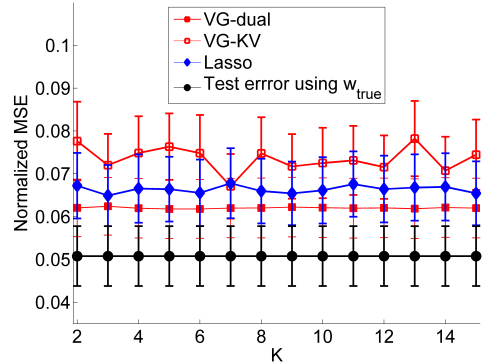


Fig. 3. The normalized mean square error of the solution (nMSE) estimated from 50 splits of test and training set, with K -fold cross-validation run on the latter to find the optimum solution. $K = 2 : 15$.

and the noise precision is $\beta^{-1} = 1$. First, we let only a single source element in the 'true' generating model be set to unity, while the rest are set to 0. VG is run on this data set with γ set to -10 and \mathbf{m} initialized to be all zeros. The convergence of the activation of the first weight, corresponding to m_1 , across iterations is illustrated in Figure 1. Also shown is the sum of the (99) remaining sources. Note the swift convergence of the posterior probabilities of the location variables. Next, to illustrate the role of the sparsity parameter γ a simulation is made with five source locations set to unity in the generating model,

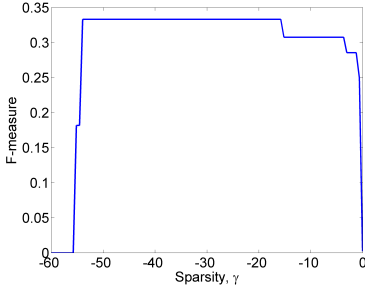


Fig. 5. Ability to retrieve the planted sources (F) as function of sparsity control γ .

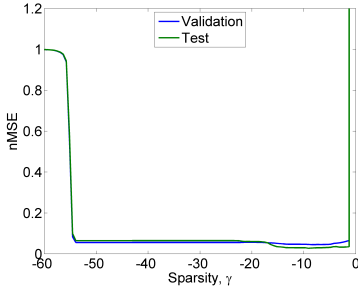


Fig. 6. Cross-validation error as function of γ .

while keeping the rest inactive. VG is run from γ_{\min} to γ_{\max} and reversed. For all values of γ , the solutions with minimum variational free energy across the two paths are stored. The values of the five locations holding the active sources are shown in Figure 2. Further, we show the magnitude of the false positive source with largest absolute value. We note that there is a large window of γ parameters for which the scheme locates the sought five sources and suppresses all other locations.

5. SIMULATIONS WITH A REALISTIC EEG FORWARD MODEL

VG and Lasso's performances are tested in a quasi-realistic EEG setting using synthetic sources. The latter consists of 10 sources set to the value 1, and the rest 0. Again Eq. ((15)) is applied. However, now using a normalized forward field as \mathbf{X} which is created in SPM (Functional Imaging Laboratory, Wellcome department of Imaging Neuroscience, Institute of Neurology at University College London, UK). The forward field \mathbf{X} here maps $n = 8196$ sources to $p = 128$ electrodes. To tune the sparsity parameters in both VG and Lasso, and also have an unbiased test error estimate, the data

is first split into a training and test set with $p_{\text{test}} = 10$ and $p_{\text{train}} = 118$. On the training set we further perform K -fold cross-validation to estimate the optimal sparsity control parameters (γ, t) . We use $K = 2, \dots, 15$, i.e., the training set is subdivided to consist of a training set $((K-1)/K)$ and a validation set $(1/K)$. For each K , performances of the optimal solution, \mathbf{v}_K for VG and \mathbf{w}_K for Lasso, are reported, using the normalized mean squared error nMSE,

$$\text{nMSE}_{\text{test}} = \frac{\text{mean}(\mathbf{v}_K \mathbf{X}_{\text{test}} - \mathbf{y}_{\text{test}})^2}{\sigma_{\mathbf{y}_{\text{test}}}}. \quad (16)$$

The above procedure is repeated 50 times. Figure 3 compares the performances of VG-dual, VG-KV and Lasso with the nMSE for the 'true' source distribution. VG-KV denotes an alternative approach to calculate a solution to the primal VG problem. In this scheme we use Kailath Variant of the matrix inversion lemma to rewrite an inverse and obtain effective scalable update rules (not shown). Figure 3 demonstrates that VG-dual outperforms the Lasso solution, while the primal solution is less accurate. It is interesting that the performance of the optimal sparsity parameters is quite stable with respect to fold size K . Inspection of the optimal solutions reveals that Lasso is less sparse than VG, and in fact has many small 'false' sources. Figure 4 visualizes the spatial structure of the found sources in the context of a 3D 'cortex'; \mathbf{w} for Lasso and \mathbf{v} for the two VG algorithms. As $m_i > 0.5$ implies $P(s_i) > 0.5$, we threshold at this level. It is noted that the values of \mathbf{m} are typically either close to 1 or 0, thus often making the thresholding redundant.

For the simulation we also check how well the VG with optimization of sparsity using the electrode cross-validation procedure is able to identify the actual source locations. For this experiment we plant 10 sources and estimate source distributions for a range of sparsity parameters (γ). In Figure 6 we show that the cross-validation error as function of the sparsity control parameter indeed is minimized in the same range as is source retrieval index $F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ [11], see Figure 5.

6. DISCUSSION AND CONCLUSION

EEG imaging is an extremely underdetermined inverse problem. Based on recent work we hypothesized that solutions of interest are sparse dipole like. We have shown that direct search for sparse solutions as implemented by Kappens Variational Garrote [5] can outperform solutions based on convex relaxations (Lasso) both in terms of cross-validation error on test data, and in terms of sparsity of the solutions. In a quasi-realistic setting with an EEG forward model we found that the VG solution provides an excellent reconstruction of the planted sources.

7. REFERENCES

- [1] A. Stopczynski, J. Larsen, C. Stahlhut, M. Petersen, and L. Hansen, "A smartphone interface for a wireless eeg headset with real-time 3d reconstruction," *Affective Computing and Intelligent Interaction*, pp. 317–318, 2011.
- [2] M. Petersen, C. Stahlhut, A. Stopczynski, J. Larsen, and L. Hansen, "Smartphones get emotional: mind reading images and reconstructing the neural sources," *Affective Computing and Intelligent Interaction*, pp. 578–587, 2011.
- [3] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig, "Independent eeg sources are dipolar," *PloS one*, vol. 7, no. 2, pp. e30135, 2012.
- [4] R.D. Pascual-Marqui, M. Esslen, K. Kochi, D. Lehmann, et al., "Functional imaging with low-resolution brain electromagnetic tomography (loreta): a review," *Methods and findings in experimental and clinical pharmacology*, vol. 24, no. suppl C, pp. 91–95, 2002.
- [5] H. J. Kappen, "The Variational Garrote," *arXiv preprint arXiv:1109.0486*, 2011.
- [6] D.M. Titterton, "Common structure of smoothing techniques in statistics," *International Statistical Review/Revue Internationale de Statistique*, pp. 141–170, 1985.
- [7] D.L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [8] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [9] R.J. Tibshirani and J. Taylor, *The solution path of the generalized lasso*, Number 2. Stanford University, 2011.
- [10] K. Sjöstrand, "Matlab implementation of LASSO, LARS, the elastic net and SPCA," June 2005, Version 2.0.
- [11] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, et al., "Performance measures for information extraction," in *Proceedings of DARPA Broadcast News Workshop*, 1999, pp. 249–252.

APPENDIX H

Submission to IEEE BCI2013

Mobile Real-Time EEG Imaging -*Abstract*

Bayesian inference with sparse, temporally smooth source priors

Lars Kai Hansen, Sofie Therese Hansen, and Carsten Stahlhut
Cognitive Systems Section, DTU Compute,
Technical University of Denmark,
DK-2800 Kongens Lyngby, Denmark
lkh@imm.dtu.dk

Abstract—EEG based real-time imaging of human brain function has many potential applications including quality control, in-line experimental design, brain state decoding, and neuro-feedback. In mobile applications these possibilities are attractive as elements in systems for personal state monitoring and well-being, and in clinical settings where patients may need imaging under quasi-natural conditions. Challenges related to the ill-posed nature of the EEG imaging problem escalate in mobile real-time systems and new algorithms and the use of meta-data may be necessary to succeed. Based on recent work (Delorme et al., 2011) we hypothesize that solutions of interest are sparse. We propose a new Markovian prior for temporally sparse solutions and a direct search for sparse solutions as implemented by the so-called “variational garrote” (Kappen, 2011). We show that the new prior and inference scheme leads to improved solutions over competing sparse Bayesian schemes based on the “multiple measurement vectors” approach.

Keywords—EEG; real-time imaging; ill-posed inverse; temporal sparsity promoting prior.

I. INTRODUCTION

Imaging electro-encephalography (EEG) is possible via solution of the electro-static inverse problem mapping scalp electrode measures to a cortical representation based on an assumed forward propagation model [1]. EEG based *real-time* imaging of human brain function has many potential applications including in-line experimental design, brain state decoding, neuro-feedback, and quality control [2]. Conventional non-imaging, i.e., “scalp based”, real-time EEG analyses has already found use in numerous applications including vigilance monitoring [3,4], human computer interfacing [5] and intensive care units [6]. Real-time imaging EEG will add several new dimensions to such applications including spatial localization of brain activity, improved localization by invoking 3D anchored prior information, and improvement of signal-to-noise by averaging EEG signals from functionally meaningful regions. Such features are highly attractive in mobile applications and systems for personal state monitoring and wellbeing, and indeed in clinical settings whenever proper care requires imaging under quasi-natural conditions [2].

II. IMAGING EEG

A. The inverse problem

Ensembles of coherent dipolar sources can produce measurable electrical potential differences at scalp electrodes. In the quasi-static approximation the relation between such sources placed in the cortical surface and measured scalp potentials is linear and instantaneous. By use of an assumed conductivity distribution, hence, a “head model”, the coefficients of the map can be estimated. In such models the typical number of sources far exceeds the relatively limited number of measurement scalp electrodes, leading to a severely ill-posed inverse problem which most often is regularized using smoothness priors [1,10]. Alternatively, the solution is expanded in spatial basis allowing tunable and spatially variant smoothness [11].

B. Sparsity promoting priors

Recent work by Delorme et al., present evidence that prominent EEG modes have a relatively simple dipolar structure, hence likely stem from well localized regions [12] in contrast to the distributed sources assumed in conventional smooth reconstruction. Finding sparse solutions to ill-posed linear inverse problems, i.e., solutions in which only a few sources are non-zero, is a quite non-trivial combinatorial optimization problem. Sparsity promoting regularization methods have been proposed based on so-called Lasso or $L1$ regularization terms leading to convex relaxations of the search problem that can be solved by efficient procedures [13].

In probabilistic settings sparsity can be realized as sources being drawn from a mixture distribution with two components, a broad component responsible for non-zero sources and another narrow component centered in zero, see e.g. [14]. A closely related mechanism is to introduce a binary $0,1$ -variable for each source indicating presence or absence of that particular source, for a recent example see [16]. Such priors are attractive for mobile real-time systems as they directly allow integration of “real prior information”, e.g., spatial information from relevant neuroimaging studies that can limit the relevant brain structures or networks in a given context.

Another often used mechanism for direct sparse approximation is to assume that sources are drawn from Gaussian priors with individual and tunable variances. If a source’s variance is tuned to zero, the parameter is effectively

pruned [15], this mechanism is in the more recent literature referred to as “sparse Bayesian learning”, see e.g., [17,18].

C. Temporal smoothness

While typical EEG data is sampled at frequencies beyond 100Hz, the typical high-energy modes have slowly varying support or location sparsity pattern [12], e.g., in independent component analysis individual modes are often treated as constant spatial patterns extending for 1000msec or more. To represent such relatively slowly varying sparsity patterns we propose here a prior with binary indicator variables linked with a simple first order Markov process. The 2x2 transition matrix has two free parameters representing the sparsity level and temporal smoothness, respectively.

D. Inference schemes

Probabilistic approaches based on approximate Bayesian inference are attractive as they can typically both find good solutions and furthermore tune prior strengths (e.g., sparsity levels and noise variance) and other control parameters see [19] for a review and references. The so-called “variational garrote” (VG) introduced by Bert Kappen is a new and computationally efficient approximate Bayesian scheme for inference in ill-posed linear inverse systems [16].

III. RESULTS

A. Single time point source reconstruction

We first compare the quality of reconstructed sources obtained with Lasso regularization, sparse Bayesian learning (SBL), and the VG for a simulated sparse localized true source, a semi-realistic head model, and real world levels of measurement noise. We find that the Lasso solutions are somewhat more scattered than the true solutions, while sparse Bayesian learning and the VG produce more localized solutions with a small advantage for VG.

B. Temporal source reconstruction

Next, we simulate spatio-temporal sources. We produce measurement scalp signals from sparse, temporally smooth, but not constant, sources with a semi-realistic head model and additive white noise. We compare here two so-called “multiple measurement vector” schemes SBL [20,21], with solutions produced by the VG, now generalized to approximate spatio-temporal sparsity patterns with the Markov prior. We find that the new spatio-temporal VG provides for an improved source reconstruction relative to the two SBL methods. Our results indicate that the SBL methods find correct locations while both seem to over-estimate the temporal smoothness of the source support.

ACKNOWLEDGMENT

This work is supported by the Lundbeck Foundation through CIMBI - Center for Integrated Molecular Brain Imaging (LKH, CS) and a postdoc grant for CS.

REFERENCES

- [1] Pascual-Marqui, Roberto Domingo. "Review of methods for solving the EEG inverse problem." *International Journal of Bioelectromagnetism* 1, no. 1 (1999): 75-86.
- [2] Stopczynski, Arkadiusz, Jakob Larsen, Carsten Stahlhut, Michael Petersen, and Lars Hansen. "A smartphone interface for a wireless EEG

headset with real-time 3D reconstruction." *Affective Computing and Intelligent Interaction* (2011): 317-318.

- [3] Makeig, Scott, Tzyy-Ping Jung, and Terrence J. Sejnowski. "Using feedforward neural networks to monitor alertness from changes in EEG correlation and coherence." *Advances in neural information processing systems* (1996): 931-937.
- [4] Berka, Chris, Daniel J. Levensowski, Milenko M. Cvetinovic, Miroslav M. Petrovic, Gene Davis, Michelle N. Lumicao, Vladimir T. Zivkovic, Miodrag V. Popovic, and Richard Olmstead. "Real-time analysis of EEG indexes of alertness, cognition, and memory acquired with a wireless EEG headset." *International Journal of Human-Computer Interaction* 17, no. 2 (2004): 151-170.
- [5] Parra, Lucas C., Clay D. Spence, Adam D. Gerson, and Paul Sajda. "Response error correction-a demonstration of improved human-machine performance using real-time EEG monitoring." *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 11, no. 2 (2003): 173-177.
- [6] Jordan, Kenneth G. "Emergency EEG and continuous EEG monitoring in acute ischemic stroke." *Journal of clinical neurophysiology* 21, no. 5 (2004): 341-352.
- [7] Congedo, Marco. "Subspace projection filters for real-time brain electromagnetic imaging." *Biomedical Engineering, IEEE Transactions on* 53, no. 8 (2006): 1624-1634.
- [8] Cannon, Rex, Joel Lubar, Estate Sokhadze, and Debora Baldwin. "LORETA neurofeedback for addiction and the possible neurophysiology of psychological processes influenced: A case study and region of interest analysis of LORETA neurofeedback in right anterior cingulate cortex." *Journal of Neurotherapy* 12, no. 4 (2008): 227-241.
- [9] Im, Chang-Hwan, Han-Jeong Hwang, Huije Che, and Seunghwan Lee. "An EEG-based real-time cortical rhythmic activity monitoring system." *Physiological measurement* 28, no. 9 (2007): 1101.
- [10] Titterton, D. M. "Common structure of smoothing techniques in statistics." *International Statistical Review/Revue Internationale de Statistique* (1985): 141-170.
- [11] Haufe, Stefan, Ryota Tomioka, Thorsten Dickhaus, Claudia Sannelli, Benjamin Blankertz, Guido Nolte, and Klaus-Robert Müller. "Large-scale EEG/MEG source localization with spatial flexibility." *NeuroImage* 54, no. 2 (2011): 851-859.
- [12] Delorme, Arnaud, Jason Palmer, Julie Onton, Robert Oostenveld, and Scott Makeig. "Independent EEG sources are dipolar." *PloS one* 7, no. 2 (2012): e30135.
- [13] Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* (1996): 267-288.
- [14] Kormylo, J., and J. Mendel. "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes." *Information Theory, IEEE Transactions on* 28, no. 3 (1982): 482-488.
- [15] Hansen, Lars Kai, and Carl Edward Rasmussen. "Pruning from adaptive regularization." *Neural Computation* 6, no. 6 (1994): 1223-1232.
- [16] Kappen, H. J. "The variational garrote." *arXiv preprint arXiv:1109.0486* (2011).
- [17] Stahlhut, Carsten, Morten Mørup, Ole Winther, and Lars Kai Hansen. "Simultaneous EEG source and forward model reconstruction (sofomore) using a hierarchical bayesian approach." *Journal of Signal Processing Systems* 65, no. 3 (2011): 431-444.
- [18] Tipping, Michael E. "Sparse Bayesian learning and the relevance vector machine." *The Journal of Machine Learning Research* 1 (2001): 211-244.
- [19] Bishop, Christopher M. *Pattern recognition and machine learning*. Vol. 4, no. 4. New York: springer, 2006.
- [20] Zhang, Zhilin, and Bhaskar D. Rao. "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning." *Selected Topics in Signal Processing, IEEE Journal of* 5, no. 5 (2011): 912-926.
- [21] Wipf, David P., and Bhaskar D. Rao. "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem." *Signal Processing, IEEE Transactions on* 55, no. 7 (2007): 3704-3716.

Identify applicable sponsor/s here. (sponsors)

Bibliography

- [ACM⁺12] J. Ashburner, C.-C. Chen, R. Moran, R. N. Henson, V. Glauche, and C. Phillips. SPM8 manual. Technical report, The FIL Methods Group, 2012.
- [AHJ12] M. Ahn, J. H. Hong, and S. C. Jun. Feasibility of approaches combining sensor and source features in brain-computer interface. *Journal of neuroscience methods*, 204(1):168–178, 2012.
- [AM13] Z. Akalin Acar and S. Makeig. Effect of forward model errors on EEG source localization. *Brain Topography (online)*, 2013.
- [Bis06] C. M. Bishop. *Pattern recognition and Machine Learning*. Springer, New York, 1st edition, 2006.
- [BMG08] M. Besserve, J. Martinerie, and L. Garnero. Non-invasive classification of cortical activities for brain computer interface: A variable selection approach. In *Biomedical Imaging: 5th IEEE International Symposium*, pages 1063–1066. IEEE, 2008.
- [BMG11] M. Besserve, J. Martinerie, and L. Garnero. Improving quantification of functional networks with EEG inverse problem: evidence from a decoding point of view. *NeuroImage*, 55(4):1536–1547, 2011.
- [BML01] S. Baillet, J. C. Mosher, and R. M. Leahy. Electromagnetic brain mapping. *Signal Processing Magazine, IEEE*, 18(6):14–30, 2001.
- [Bre95] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

- [CFR11] E. Cantoni, J. M. Flemming, and E. Ronchetti. Variable selection in additive models by non-negative garrote. *Statistical Modelling*, 11(3):237–252, 2011.
- [Con06] M. Congedo. Subspace projection filters for real-time brain electromagnetic imaging. *IEEE transactions on bio-medical engineering*, 53(8):1624–1634, 2006.
- [DET06] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Transactions on Information Theory, IEEE*, 52(1):6–18, 2006.
- [Dis12] Distrep. Donders Machine Learning Toolbox (DMLT), 2012.
- [DPO⁺12] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig. Independent EEG sources are dipolar. *PloS one*, 7(2):1–14, 2012.
- [DW50] J. Durbin and G. S. Watson. Testing for serial correlation in least squares regression: I. *Biometrika*, 37(3/4):pp. 409–428, 1950.
- [EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.
- [FGPM⁺01] E. Frei, A. Gamma, R. Pascual-Marqui, D. Lehmann, D. Hell, and F. X. Vollenweider. Localization of MDMA-induced brain activity in healthy volunteers using low resolution brain electromagnetic tomography (LORETA). *Human brain mapping*, 14(3):152–165, 2001.
- [FHD⁺08] K. J. Friston, L. Harrison, J. Daunizeau, S. J. Kiebel, C. Phillips, N. Trujillo-Barreto, R. N. Henson, G. Flandin, and J. Mattout. Multiple sparse priors for the M/EEG inverse problem. *NeuroImage*, 39(3):1104–1120, 2008.
- [FHT10] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- [Fri94] K. J. Friston. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, 2(1-2):56–78, 1994.
- [GPOC11] A. Gramfort, T. Papadopoulos, E. Olivi, and M. Clerc. Forward field computation with OpenMEEG. *Computational intelligence and neuroscience*, 2011(1):1–13, 2011.
- [Gra06] GraysAnatomy. Principal fissures and lobes of the cerebrum viewed laterally Figure 728. http://en.wikipedia.org/wiki/File:Lobes_of_the_brain_NL.svg, 2006.

- [HFT00] B. Horwitz, K. J. Friston, and J. Taylor. Neural modeling and functional brain imaging : an overview. *Neural Networks*, 13(8):829–846, 2000.
- [HGGG⁺03] R. N. Henson, Y. Goshen-Gottstein, T. Ganel, L. J. Otten, A. Quayle, and M. D. Rugg. Electrophysiological and haemodynamic correlates of face perception, recognition and priming. *Cerebral cortex*, 13(7):793–805, 2003.
- [HI84] M. S. Hämäläinen and R. J. Ilmoniemi. Interpreting magnetic fields of the brain: minimum norm estimates. Technical Report TKK-F-A559, Helsinki University of Technology, Department of Technical Physics, 1984.
- [HMPF09] R. N. Henson, J. Mattout, C. Phillips, and K. J. Friston. Selecting forward models for MEG source-reconstruction using model-evidence. *Neuroimage*, 46(1):168–176, 2009.
- [HNZ⁺08] S. Haufe, V. V. Nikulin, A. Ziehe, K.-R. Müller, and G. Nolte. Combining sparsity and rotational invariance in EEG/MEG source reconstruction. *NeuroImage*, 42(2):726–738, 2008.
- [HR94] L. K. Hansen and C. E. Rasmussen. Pruning from adaptive regularization. *Neural Computation*, 6(6):1223–1232, 1994.
- [HT10] S. Haufe and R. Tomioka. Modeling sparse connectivity between underlying brain sources for EEG/MEG. *Transactions on Biomedical Engineering, IEEE*, 57(8):1954–1963, 2010.
- [HTD⁺11] S. Haufe, R. Tomioka, T. Dickhaus, C. Sannelli, B. Blankertz, G. Nolte, and K.-R. Müller. Large-scale EEG/MEG source localization with spatial flexibility. *NeuroImage*, 54(2):851–859, 2011.
- [HTTW07] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1(1):1–29, 2007.
- [HVG⁺07] H. Hallez, B. Vanrumste, R. Grech, J. Muscat, W. De Clercq, A. Vergult, Y. D’Asseler, K. P. Camilleri, S. G. Fabri, S. Van Huffel, and I. Lemahieu. Review on solving the forward problem in EEG source analysis. *Journal of neuroengineering and rehabilitation*, 4(46):1–29, 2007.
- [IHCL07] C.-H. Im, H.-J. Hwang, H. Che, and S. Lee. An EEG-based real-time cortical rhythmic activity monitoring system. *Physiological measurement*, 28(9):1101–13, 2007.

- [KCA⁺05] J. Kybic, M. Clerc, T. Abboud, O. Faugeras, R. Keriven, and T. Papadopoulos. A common formalism for the integral formulations of the forward EEG problem. *Transactions on medical imaging, IEEE*, 24(1):12–28, 2005.
- [KG12] H. J. Kappen and V. Gomez. The variational garrote. *arXiv preprint*, 2012.
- [LHH⁺91] S. Lu, M. S. Hämäläinen, R. Hari, R. Ilmoniemi, O. V. Lounasmaa, M. Sams, and V. Vilkmann. Seeing faces activates three separate areas outside the occipital visual cortex in man. *Neuroscience*, 43(2):287–290, 1991.
- [LHR96] M. Linden, T. Habib, and V. Radojevic. A controlled study of the effects of EEG biofeedback on cognition and behavior of children with attention deficit disorder and learning disabilities. *Biofeedback and self-regulation*, 21(1):35–49, 1996.
- [MJOB10] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. *arXiv preprint*, 2010.
- [MKSW99] J. Makhouf, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252. Morgan Kaufmann Pub, 1999.
- [MMHM12] J. Montoya-Martinez, L. K. Hansen, and P. Massimiliano. Structured sparsity regularization approach to the EEG inverse problem. *3rd International Workshop on Cognitive Information Processing*, 1, 2012.
- [MN94] D. J. C. MacKay and R. M. Neal. Automatic relevance determination for neural networks. *Technical Report in preparation, Cambridge University*, Preprint, 1994.
- [MWDD05] D. J. Moore, A. B. West, V. L. Dawson, and T. M. Dawson. Molecular pathophysiology of Parkinson’s disease. *Annual review of neuroscience*, 28(1):57–87, 2005.
- [Nea95] R. M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.
- [NHW03] T. J. Nowak, A. G. Handford, and G. M. Whitelegg. *Pathophysiology: concepts and applications for health care professionals*. McGraw-Hill Higher Education, 3rd edition, 2003.
- [NS06] P. Nunez and R. Srinivasan. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2nd edition, 2006.

- [OPT00a] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA journal of Numerical Analysis*, 20(1):389–403, 2000.
- [OPT00b] M. R. Osborne, B. Presnell, and B. A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.
- [PBT⁺08] M. S. Pedersen, B. Baxter, B. Templeton, C. Rishøj, D. L. Theobald, E. Hoegh-rasmussen, G. Casteel, J. B. Gao, K. Dedecius, K. Strim, L. Christiansen, L. K. Hansen, L. Wilkinson, L. He, M. Bar, O. Winther, P. Sakov, and S. Hattinger. The Matrix Cookbook. Technical report, Technical University of Denmark, 2008.
- [PLD⁺05] E. Patarraia, G. Lindinger, L. Deecke, D. Mayer, and C. Baumgartner. Combined MEG/EEG analysis of the interictal spike complex in mesial temporal lobe epilepsy. *NeuroImage*, 24(3):607–614, 2005.
- [PM09] R. D. Pascual-Marqui. Theory of the EEG inverse problem. In S. Tong and N. Vyomesh Thakor, editors, *Quantitative EEG analysis: methods and clinical applications*, chapter 5, pages 121–137. Artech House, 1st edition, 2009.
- [PMML94] R. D. Pascual-Marqui, C. M. Michel, and D. Lehmann. Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain. *International Journal of Psychology*, 18:49–65, 1994.
- [PSS11] M. Petersen, C. Stahlhut, and A. Stopczynski. Smartphones get emotional: mind reading images and reconstructing the neural sources. *Affective Computing Intelligent Interaction*, pages 578–587, 2011.
- [SB91] M. Scherg and P. Berg. Use of prior knowledge in brain electromagnetic source analysis. *Brain topography*, 4(2):143–50, 1991.
- [SCLEl] K. Sjöstrand, L. H. Clemmensen, R. Larsen, and B. Ersbøll. SpaSM: A Matlab Toolbox for Sparse Statistical Modeling. *Journal of Statistical Software*, Submitted.
- [SHH⁺97] M. Sams, J. K. Hietanen, R. Hari, R. Ilmoniemi, and O. V. Lounasmaa. Face-specific responses from the human inferior occipito-temporal cortex. *Neuroscience*, 77(1):49–55, 1997.
- [Sj05] K. Sjöstrand. Matlab implementation of LASSO, LARS, the elastic net and SPCA, version 2.0, 2005.

- [SLS11] A. Stopczynski, J. Larsen, and C. Stahlhut. A smartphone interface for a wireless EEG headset with real-time 3D reconstruction. *Affective Computing and Intelligent Interaction*, 900(1):317–318, 2011.
- [SMFK09] A. Stepanyants, L. M. Martinez, A. S. Ferecsko, and Z. F. Kisvady. The fractions of short- and long-range connections. *Proceedings of the National Academy of Sciences*, 106(9):3555–3560, 2009.
- [SSJ⁺10] O. Steinsträter, S. Sillekens, M. Junghoefer, M. Burger, and C. H. Wolters. Sensitivity of beamformer source analysis to deficiencies in forward modeling. *Human brain mapping*, 31(12):1907–1927, 2010.
- [SSL06] K. Sjöstrand, M. B. Stegmann, and R. Larsen. Sparse principal component analysis in medical shape modeling. *Symposium on Medical Imaging*, 6144, 2006.
- [SST08] R. R. Seeley, T. D. Stephens, and P. Tate. *Anatomy & Physiology*. Boston: McGraw-Hill, 8th edition, 2008.
- [TBAVVS04] N. Trujillo-Barreto, E. Aubert-Vázquez, and P. A. Valdés-Sosa. Bayesian model averaging in EEG/MEG imaging. *NeuroImage*, 21(4):1300–19, 2004.
- [Tep02] M. Teplan. Fundamentals of EEG measurement. *Measurement science review*, 2(2):1–11, 2002.
- [TF03] M. E. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. *Proceedings of the ninth international workshop on artificial intelligence and statistics*, 1(3), 2003.
- [Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.
- [Tip00] M. E. Tipping. The relevance vector machine. *Advances in Neural Information Processing Systems*, 12(1):652–658, 2000.
- [Tip01] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1(1):211–244, 2001.
- [Tip09a] M. E. Tipping. A baseline Matlab implementation of “ Sparse Bayesian ” model estimation (Version 1.1). Technical report, 2009.

- [Tip09b] M. E. Tipping. An efficient Matlab implementation of the sparse Bayesian modelling algorithm (Version 2.0). Technical report, Vector anomaly, 2009.
- [TT08] M. Thompson and L. Thompson. Biofeedback for movement disorders (dystonia with Parkinson ’ s disease): Theory and preliminary results biofeedback for movement disorders. *Journal of Neurotherapy : Investigations in Neuromodulation , Neuro-feedback and Applied Neuroscience*, 6(4):51–70, 2008.
- [TT11] R. J. Tibshirani and J. Taylor. *The solution path of the generalized lasso*. PhD thesis, Stanford University, 2011.
- [UHS99] K. Uutela, M. S. Hämäläinen, and E. Somersalo. Visualization of magnetoencephalographic data using minimum current estimates. *NeuroImage*, 10(2):173–180, 1999.
- [VSSBLC⁺05] P. A. Valdés-Sosa, J. M. Sánchez-Bornot, A. Lage-Castellanos, M. Vega-Hernández, J. Bosch-Bayard, L. Melie-García, and E. Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 360(1457):969–981, 2005.
- [WAT⁺06] C. H. Wolters, A. Anwander, X. Tricoche, D. Weinstein, M. A. Koch, and R. S. MacLeod. Influence of tissue conductivity anisotropy on EEG/MEG field and return current computation in a realistic head model: a simulation and visualization study using high-resolution finite element modeling. *NeuroImage*, 30(3):813–826, 2006.
- [WGH04] C. H. Wolters, L. Grasedyck, and W. Hackbusch. Efficient computation of lead field bases and influence matrix for the FEM-based EEG and MEG inverse problem. *Inverse Problems*, 20(4):1099–1116, 2004.
- [WRP⁺07] D. Wipf, R. Ramirez, J. Palmer, S. Makeig, and B. D. Rao. Analysis of empirical Bayesian methods for neuroelectromagnetic source localization. *Advances in Neural Information Processing Systems, NIPS*, 19(1):1505–1512, 2007.
- [YL05] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. Technical Report 1095, University of Wisconsin, Department of Statistics, 2005.
- [Zou06] H. Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

- [ZR11] Z. Zhang and B. D. Rao. Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. *Selected Topics in Signal Processing, IEEE*, 5(5):912–926, 2011.
- [ZY07] P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7(1):2541–2563, 2007.