Kensuke Sekihara
Srikantan S. Nagarajan

Electromagnetic Brain Imaging

A Bayesian Perspective

Springer

Electromagnetic Brain Imaging

Kensuke Sekihara · Srikantan S. Nagarajan

Electromagnetic Brain Imaging

A Bayesian Perspective

Springer

Kensuke Sekihara
Department of Systems Design
and Engineering
Tokyo Metropolitan University
Tokyo
Japan

Srikantan S. Nagarajan
Department of Radiology and Biomedical
University of California
San Francisco, CA
USA

ISBN 978-3-319-14946-2 ISBN 978-3-319-14947-9 (eBook)
DOI 10.1007/978-3-319-14947-9

Library of Congress Control Number: 2014959853

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2015
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

Neuronal activities in a human brain generate coherent synaptic and intracellular currents in cortical columns, which generate electric potentials on the scalp surface and magnetic fields outside the head. Electroencephalography (EEG) measures these potentials and magnetoencephalography (MEG) measures these magnetic fields to obtain information on the state of the brain. The class of methodologies that reconstruct and visualize the neuronal activities based on MEG/EEG sensor measurements is referred to as the electromagnetic brain imaging.

In the past two decades there have been significant advances in signal processing and source reconstruction algorithms used in electromagnetic brain imaging. However, electromagnetic brain imaging can still be considered a young field. This is primarily due to the complexity associated with the electrophysiological brain activity underlying the signals. Also, it is true that applying the electromagnetic brain imaging with confidence requires an understanding of the relevant mathematics, physics, biology, and engineering as well as a broad perspective on human brain science. Due to its interdisciplinary nature, such a broad knowledge base takes years to acquire.

This book is intended to provide a coherent introduction to the body of mainstream algorithms used in electromagnetic brain imaging, with specific emphasis on novel Bayesian algorithms that we have developed. It is intended as a graduate level textbook with the goal of helping readers more easily understand the literature in biomedical engineering and in related fields, and be ready to pursue research in either the engineering or the neuroscientific aspects of electromagnetic brain imaging. We hope that this textbook will not only appeal to graduate students but all scientists and engineers engaged in research on electromagnetic brain imaging.

This book begins with an introductory overview of electromagnetic brain imaging in Chap. 1 and then discusses dominant algorithms that are used in electromagnetic brain imaging in Chaps. 2–4. Minimum-norm-based methods, which are classic algorithms and still widely used in this field, are described in Chap. 2, but with a Bayesian perspective in mind. Chapter 3 presents a concise review of adaptive beamformers, which have become a standard tool for analyzing brain

spontaneous activity such as resting-state MEG data, and we also include a Bayesian perspective on adaptive beamformers.

Chapters 4–6 review Bayesian algorithms for electromagnetic brain imaging that we have developed in the past decade. Chapter 4 presents a novel Bayesian algorithm, called Champagne, which has been developed by our group. Since we believe that the Champagne algorithm is a powerful next-generation imaging algorithm, the derivation of the algorithm is presented in detail. Chapter 5 presents Bayesian factor analysis, which is a group of versatile algorithms used for denoising, interference suppression, and source localization. Chapter 6 describes a unified theoretical Bayesian framework, which provides insightful perspective into various source imaging methods and reveals similarities and equivalences between methods that appear to be very different.

Chapters 7–9 deal with newer topics that are currently in vogue in electromagnetic brain imaging—functional connectivity, causality, and cross-frequency coupling analyses. Chapter 7 reviews functional connectivity analysis using imaginary coherence. Chapter 8 provides a review of several directional measures that can detect causal coupling of brain activities. Chapter 9 presents novel empirical results showing that the phase-amplitude coupling can be detected using MEG source-space analysis, and demonstrates that the electromagnetic brain imaging holds great potential in elucidating the mechanisms of brain information processing. This chapter was contributed by Eiichi Okumura and Takashi Asakawa. The first two chapters in the Appendix provide concise explanations of bioelectromagnetic forward modeling and basics of Bayesian inference, and the third chapter provides supplementary mathematical arguments. These chapters are included for the reader's convenience.

Many people have made valuable contributions together with our own efforts in this area. Special mention must be made of Hagai Attias who is an invaluable collaborator. Hagai introduced us to probabilistic graphical models and Bayesian inference methods. He was an integral person in our fruitful collaboration, on which this book is based.

Many students, postdocs, fellows, and UCSF faculty members have collaborated with us over the years. These include Leighton Hinkley, David Wipf, Johanna Zumer, Julia Owen, Sarang Dalal, Isamu Kumihashi, Alex Herman, Naomi Kort, Ethan Brown, Rodney Gabriel, Maneesh Sahani, Adrian Guggisberg, Juan Martino, Kenneth Hild, Matthew Brookes, Carsten Stahlhut, Oleg Portniaguine, Erik Edwards, Ryan Canolty, David McGonigle, Tal Kenet, Theda Heinks-Maldonado, Aliu Sheye, Dameon Harrell, Josiah Ambrose, Sandeep Manyam, William McClay, Virginie van Wassenhove, Ilana Hairston, Maria Ventura, Zhao Zhu, Corby Dale, Tracy Luks, Kelly Westlake, Kamalini Ranasinghe, Karuna Subramanium, Carrie Niziolek, Zarinah Agnew, Carly Demopoulos, Megan Thomson, Peter Lin, Tulaya Limpiti, Lisa Dyson, Pew Puthividya, Jiucang Hao, Phiroz Tarapore, Dario Englot, Heidi Kirsch, Sophia Vinogradov, Michael Merzenich, Christoph Schreiner, Elizabeth Disbrow, Mitchel Berger, Edward Chang, Nick Barbaro, Roland Henry, Sarah Nelson, William Dillon, Jim Barkovich, Nancy Byl, David Blake, Keith Vossel, Elliot Sherr, Elysa Marco, Josh Wooley, Marilu Gorno-Tempini, Robert Knight, and

Pratik Mukherjee. Also, we are deeply indebted to the following fantastic and incredibly dedicated staff at the Biomagnetic Imaging Laboratory: Susanne Honma, Mary Mantle, Anne Findlay-Dowling, Danielle Mizuiri, and Garrett Coleman.

Furthermore, several people have been particularly influential, and have greatly helped us to enhance our understanding of this field. Such people include Thomas Ferree, Mike X. Cohen, Stephen E. Robinson, Jiri Vrba, Guido Nolte, and Barry Van Veen. We would also like to thank Daniel Palomo for his effort in editing the manuscript of this book. Finally, we thank the following collaborators for their friendship and invaluable support to our work over nearly 20 years: John Houde, Steven Cheung, David Poeppel, Alec Marantz, and Tim Roberts.

We kindly ask readers to visit www.electromagneticbrainimaging.info. Supplementary information, as well as error corrections (if necessary), is uploaded to this website.

November 2014

Kensuke Sekihara
Srikantan S. Nagarajan

Contents

Chapter 1
Introduction to Electromagnetic Brain Imaging

1.1 Functional Brain Imaging and Bioelectromagnetic Measurements

Noninvasive functional brain imaging has made a tremendous impact in improving our understanding of the human brain. Functional magnetic resonance imaging (fMRI) has been the predominant modality for imaging the functioning brain since the middle of the 1990s. fMRI measures changes in blood oxygenation-level-dependent (BOLD) signals caused by neuronal activation. It is a noninvasive method that allows for whole-brain measurement and the examination of activity in deep brain structures.

However, fMRI lacks the temporal resolution required to image the dynamic and oscillatory spatiotemporal patterns associated with activities in a brain. This is because the BOLD signal, which is only an indirect measure of neural activity, is fundamentally limited by the rate of oxygen consumption and subsequent blood flow. Furthermore, since the BOLD signal is only an approximate and indirect measure of neuronal activity, it might not accurately reflect true neuronal processes. Hence, to observe neurophysiological processes more directly within relevant timescales, imaging techniques that have both high temporal and adequate spatial resolution are needed.

Neurons in the brain function electrically, as well as chemically. Therefore, their activity generates associated electric and magnetic fields that can be detected outside the head. Electromagnetic brain imaging is a term intended to encompass noninvasive techniques which probe brain electromagnetic activity and properties. Two techniques currently exist for detecting electrical brain activities: electroencephalography (EEG) and magnetoencephalography (MEG). EEG measures the electric potential by means of electrodes placed on the scalp, and MEG measures magnetic fields by means of sensors placed near the head.

In contrast to fMRI, both MEG and EEG directly measure electromagnetic fields emanating from the brain with excellent temporal resolution of less than 1 ms, and allow the study of neural oscillatory processes over a wide frequency range (1–600 Hz). Because of such high temporal resolution, MEG and EEG can provide

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_1

information not obtainable with other functional brain imaging techniques. Most notably, MEG and EEG track neural population activity on millisecond timescales, revealing large-scale dynamics that are crucial for understanding brain function. Furthermore, by applying modern inverse algorithms, it is possible to obtain three-dimensional images, which are reasonable estimates of neural activity. Such images are extremely useful for answering many questions on brain science.

While MEG is mainly sensitive to tangential currents in the brain closer to the surface and relatively insensitive to the conductive properties of the skull, EEG is primarily sensitive to radial sources while being highly sensitive to the conductive properties of the brain, skull, and scalp. Therefore, MEG and EEG can be viewed as being complementary in terms of the sensitivity to underlying neural activity. However, since magnetic fields generated from neurons are not distorted by the heterogeneous electrical properties of a brain, these magnetic fields can be considered an undistorted signature of underlying cortical activity.

In addition, there are several practical or physiological reasons why neuroscientists prefer MEG to EEG. First, MEG setup time is very short, because MEG measurement does not require much preparation in attaching and checking electrodes, as is needed in performing EEG measurement. This simplifies matters both for experimenters and subjects. Second, the anatomical location of primary sensory cortices in sulci makes MEG ideally suited for electrophysiological studies. Furthermore, with whole-head sensor arrays, MEG is also well-suited to investigate hemispheric lateralization effects. Therefore, this chapter is primarily dedicated to giving a review of the methodologies associated with MEG.

1.2 Sensing Magnetic Fields from the Brain

The long apical dendrites of cortical pyramidal cells are arranged perpendicularly to the cortical surface and parallel to each other. This fortuitous anatomical arrangement of these cells allows the magnetic fields to sum up to magnitudes large enough to detect at the scalp. Synchronously fluctuating dendritic currents result in equivalent current dipoles that produce such magnetic fields. However, biomagnetic fields from a brain are extremely small, (in range of tens-to-hundreds of femto-Tesla (fT)) which is about seven orders of magnitude smaller than the earth's magnetic field. As a result, appropriate data collection necessitates a magnetically shielded room and highly sensitive detectors known as superconducting quantum interference devices (SQUIDs). Biomagnetic fields from a brain are typically sensed using detection coils called flux transformers or magnetometers, which are positioned close to the scalp and connected to SQUIDs. A SQUID acts as a magnetic-field-to-voltage converter, and its nonlinear response is linearized by flux-locked loop electronics. SQUIDs have a sensitivity of up to 5 femto-Tesla per square root of Hz, which is adequate for the detection of brain-generated magnetic fields.

MEG sensors are often configured for measuring differential magnetic fields so as to reduce ambient noise in measurements. Such sensors are referred to as

gradiometers, although some MEG systems do not use gradiometers relying on clever noise cancellation methods. The two commonly used gradiometer configurations are axial and planar gradiometers. Axial (first order) gradiometers consist of two coils that share the same axis, whereas planar (first order) gradiometers consist of two coils that share the same plane. The sensitivity profile of a planar gradiometer is somewhat similar to EEG, whereby a sensor is maximally sensitive to a source closest to it. In contrast, the sensitivity profile of an axial gradiometer can be somewhat counterintuitive because it is not maximally sensitive to sources closest to the sensors.

Modern MEG systems consist of simultaneous recordings from many sensors that provide whole head coverage. The total number of sensors varies from 100 to 300. The advent of such large sensor-array systems has significantly advanced MEG studies. Although the maximum sampling rate for many MEG systems reaches more than 10 kHz, most MEG data is usually recorded at a sampling rate of around 1,000 Hz, which still provides excellent temporal resolution for measuring the dynamics of cortical neuronal activity at millisecond order.

1.3 Electromagnetic Brain Imaging

MEG sensor data only provides qualitative information about underlying brain activities. The analysis of the sensor data is typically performed based on the intuitions of experienced users regarding the sensitivity profile of the sensors. To extract more precise information from the observed sensor data, it is essential to apply imaging-type analysis involving the reconstruction of brain activities from the sensor data. Major components for the electromagnetic brain imaging are the forward model and the inverse algorithms.

1.3.1 Forward Model

The forward model consists of three subcomponents: the source model, the volume conductor model, and the measurement model. Typical source models assume that the brain magnetic fields are generated by equivalent current dipoles in the brain. This model is consistent with available measurements of coherent synaptic and intracellular currents in cortical columns that are thought to be major contributors to MEG and EEG signals. Although several more complex source models have been proposed, the equivalent current dipole is the dominant source model. This is because, given the distance between the sources and sensors, the dipole is a reasonable approximation of brain sources.

The volume conductor model refers to the equations that govern the relation between the source model and the sensor measurements, namely the electric potentials or the magnetic fields. These surface integral equations, obtained by solving Maxwell's equations under quasi-static conditions, can be solved analytically for

special geometries of the volume conductor, such as the sphere or an ellipsoid. A concise review on the spherical homogeneous conductor model is found in Appendix A. For realistic volume conductors, various numerical techniques such as finite-element and boundary-element methods may be employed, although these methods are generally time-consuming.

Measurement models refer to the specific measurement systems used in EEG and MEG including the position of the sensors relative to the head. For instance, different MEG systems measure axial versus planar gradients of the magnetic fields with respect to different locations of sensors. The measurement model incorporates such information about the type of measurement and the geometry of the sensors. Measurement of the position of the head relative to the sensor array is accomplished by attaching head-localization coils to fiducial landmarks on the scalp. Modern MEG systems are sometimes equipped with continuous head-localization procedures that enable constant updating of sensor locations relative to the head to compensate for subjects head movements. The source, volume conductor, and measurement models are typically combined into a concept called the lead field that describes a linear relationship between sources and the measurements. When discussing inverse methods, we assume that the lead field matrix is known.

1.3.2 Inverse Algorithms

Inverse algorithms are used for solving the bioelectromagnetic inverse problem, i.e., for estimating the parameters of neural sources from MEG and EEG sensor data. When implementing electromagnetic brain imaging, this estimation of spatial locations and timing of brain sources is a challenging problem because it involves solving for unknown brain activity across thousands of voxels from the recordings of just a few hundred sensors. In general, there are no unique solutions to the inverse problem because there are many source configurations that could produce sensor data equal to the sensor observations, even in the absence of noise and (if given) infinite spatial or temporal sampling. This nonuniqueness is referred to as the ill-posed nature of the inverse problem. Nevertheless, to get around this nonuniqueness, various estimation procedures incorporate prior knowledge and constraints about source characteristics.

Inverse algorithms can be classified into two categories: model-based dipole fitting and (non-model-based) imaging methods. Dipole fitting methods assume that a small set of current dipoles can adequately represent an unknown source distribution. In this case, the dipole locations and its moments form a set of unknown parameters, which are typically estimated using the least-squares fit. The dipole fitting method—particularly the single-dipole fitting method—has clinically been used for localization of early sensory responses in somatosensory and auditory cortices.

However, two major problems exist in a dipole fitting procedure. First, the nonlinear optimization causes a problem of local minima when more than two dipole parameters are estimated. A second, more difficult problem is that the dipole fitting methods require a priori knowledge of the number of dipoles. Often, such information

about the model order is not available a priori, especially for complex brain mapping conditions, and the resulting localization of higher order cortical functions can sometimes be unreliable.

An alternative approach is whole-brain source imaging methods. These methods apply voxel discretization over a whole brain volume, and assume a fixed source at each voxel. These methods estimate the amplitudes (and directions) of the sources by minimizing a cost function. One classic method of this kind is the minimum-norm method [1]. The minimum-norm and related methods are the topic of Chap. 2.

Since the number of voxels is usually much larger than the number of sensors, the cost function should contain a constraint term that is derived from various prior information about the nature of the sources. We have developed a powerful imaging algorithm that incorporates a sparsity constraint that facilitates the sparse source configurations. This algorithm, called the Champagne algorithm, is described in detail in Chap. 4.

The other class of imaging algorithm is the spatial filter. The spatial filter estimates source amplitude at each location independently. It is often called a virtual sensor method, because it forms a virtual sensor, which scans an entire source space to produce a source image. The most popular spatial filter algorithms are adaptive spatial filtering techniques, more commonly referred to as adaptive beamformers. Adaptive beamformers are the topic of Chap. 3 in which a concise review on adaptive beamformers is presented. A comprehensive review of these algorithms is found in [2]. The implementation of adaptive beamformers is quite simple, and they have proven to be powerful algorithms for characterizing cortical oscillations. Therefore, they are popular for the source localization from spontaneous brain activity, particularly resting-state MEG. We have recently proposed novel scanning algorithms [3, 4]. These methods have shown performance improvements over the adaptive beamformers. One of these methods, called the Saketini algorithm, is described in Chap. 5.

In Chap. 6, we discuss a hierarchical Bayesian framework which encompasses various source imaging algorithms in a unified framework and reveals a close relationship between algorithms that are considered quite different [5].

An enduring problem in MEG imaging is that the brain evoked responses to sensory or cognitive events are small compared to the interfering magnetic field. Typical sources of such interference include the background room interference from power lines and electronic equipment, the interference with biological origins such as heartbeat, eye blink, or other muscle artifacts. Ongoing brain activity itself, including the drowsy-state alpha rhythm, is also a major source of interference. All existing methods for brain source imaging are hampered by such interferences present in MEG data. Several signal-processing methods have been developed to reduce interferences by preprocessing the sensor data before submitting it to source localization algorithms. One such algorithm, called partitioned factor analysis, is described in Chap. 5.

1.4 From Source Imaging to Functional Connectivity Imaging

It is now well recognized in neuroscience that it is necessary to examine how the brain integrates information across multiple regions. The term functional connectivity essentially defines the complex functional interaction between separate brain areas. Although functional connectivity analysis using fMRI is common, MEG is better suited for modeling and detecting such interactions, because of its high temporal resolution.

Functional connectivity analysis can be applied either in sensor-space or in source-space. In sensor-space analysis, the field spread across many sensors arriving from a single brain region leads to uncertainties in interpreting the estimation results of brain interactions [6]. Therefore, a number of studies have begun to use source-space analysis, in which voxel time courses are first estimated by an inverse algorithm and brain interactions are then analyzed using those estimated voxel time courses. However, a serious problem arises from the leakage of an inverse algorithm, and such leakages are more or less inevitable in any inverse algorithm. Chapter 7 describes the imaginary coherence analysis in source space. The imaginary coherence analysis is known to be robust to the leakage of an inverse algorithm, and has become a popular method in analyzing functional connectivity using MEG.

Connectivity measures commonly used these days—such as coherence—are bidirectional measures, which cannot detect the directionality of brain interactions. There has been growing interest in analyzing causal networks in the brain, and directional measures are needed to detect such causal networks. Measures for detecting causal networks are the topic of Chap. 8.

Neural mechanisms of brain information processing are the subject of intense investigation. A prevailing hypothesis is that the brain uses temporal encoding based on firing phase, and that there may exist a coupling between oscillators of different frequencies. Chapter 9 presents empirical results that such phase-amplitude coupling can be detected using MEG source-space analysis. The results in Chap. 9 demonstrate that MEG functional connectivity imaging holds great potential in revealing mechanisms of brain information processing.

1.5 Examples of Clinical Applications

1.5.1 Functional Mapping for Preoperative Neurosurgical Planning

The surgical management of brain tumors or epilepsy often requires detailed functional mapping of cortical regions around a tumor or epileptogenic zone. Preoperative mapping techniques aim at accurately estimating the location of functional

areas in relation to a tumor or an epileptic focus to minimize surgical risk. Mass lesions can frequently distort normal neuroanatomy, which makes the identification of eloquent cortices inaccurate with normal neuroanatomical landmarks. MEG is increasingly being used for preoperative functional brain imaging. By mapping relevant somatosensory, auditory, and motor cortices preoperatively, retained areas of function can be delineated. Preoperative localization of functionally intact brain tissue helps guide neurosurgical planning and limits the region of resection, allowing for improved long-term patient morbidity and neurological function.

Examples that show how MEG imaging is used to map the motor cortex in presurgical patients are given in [7]. This study performed localization of β-band desynchronization preceding the index finger flexion for a subject with a frontal tumor, and showed the location of hand motor cortex relative to a single dipole localization of hand somatosensory cortex. These results were confirmed with electrical cortical stimulation. This investigation demonstrated that MEG source images obtained using beta-band event-related desynchronization reliably localize the hand motor cortex.

1.5.2 Functional Connectivity Imaging

Functional connectivity analysis has been shown to have profound clinical significance, because disturbances in networks are manifested as abnormalities in functional connectivity and such abnormalities can be detected using resting state MEG. Recent studies have shown this to be the case, and abnormalities in functional connectivity during resting state are observed in many clinical conditions such as brain tumors [8], strokes [9], traumatic brain injury [10], schizophrenia [11, 12], and Alzheimer disease [13].

Utilizing MEG imaging, changes in the alpha-band functional connectivity in patients with traumatic brain injury were measured and compared to healthy controls in [10]. In this investigation, mean imaginary coherence (MIC) (described in Sect. 7.9) at each brain voxel was calculated as an index for functional connectivity. In one male patient's case, his initial MEG scan was obtained 9 months after his injury and the results of MIC mapping demonstrated several regions of decreased resting state functional connectivity compared to the control group. The follow-up second MEG scan was obtained 23 months after the initial scan and the results of MIC mapping demonstrated a decrease in the volume of cortex with reduced connectivity, although some of the cortical regions with the greatest reductions in functional connectivity seen in the initial MEG remained abnormal even in the second MEG scan.

The investigation described next identifies brain regions that exhibited abnormal resting-state connectivity in patients with schizophrenia [11, 12]. Associations between functional connectivity and clinical symptoms were found in stable outpatient participants. Resting-sate MEG was measured from thirty schizophrenia patients and fifteen healthy control subjects. The MIC mapping was computed in the alpha

frequency band, and the results showed that the functional connectivity of the left inferior parietal cortex was negatively related to positive symptoms and the left prefrontal cortical connectivity was associated with negative symptoms. This study demonstrates direct functional disconnection in schizophrenia patients between specific cortical fields within low-frequency resting state oscillations. Such findings indicate that this level of functional disconnection between cortical regions is an important treatment target in schizophrenia patients.

References

1. M.S. Hämäläinen, R.J. Ilmoniemi, Interpreting measured magnetic fields of the brain: estimates of current distributions. Technical Report TKK-F-A559, Helsinki University of Technology (1984)
2. K. Sekihara, S.S. Nagarajan, *Adaptive Spatial Filters for Electromagnetic Brain Imaging* (Springer, Berlin, 2008)
3. J.M. Zumer, H.T. Attias, K. Sekihara, S.S. Nagarajan, A probabilistic algorithm integrating source localization and noise suppression for MEG and EEG data. NeuroImage **37**, 102–115 (2007)
4. J.M. Zumer, H.T. Attias, K. Sekihara, S.S. Nagarajan, Probabilistic algorithms for MEG/EEG source reconstruction using temporal basis functions learned from data. NeuroImage **41**(3), 924–940 (2008)
5. D. Wipf, S.S. Nagarajan, A unified Bayesian framework for MEG/EEG source imaging. NeuroImage **44**, 947–966 (2009)
6. J.-M. Schoffelen, J. Gross, Source connectivity analysis with MEG and EEG. Hum. Brain Mapp. **30**, 1857–1865 (2009)
7. P.E. Tarapore, M.C. Tate, A.M. Findlay, S.M. Honma, D. Mizuiri, M.S. Berger, S.S. Nagarajan, Preoperative multimodal motor mapping: a comparison of magnetoencephalography imaging, navigated transcranial magnetic stimulation, and direct cortical stimulation: clinical article. J. Neurosurg. **117**(2), 354–362 (2012)
8. A.G. Guggisberg, S.M. Honma, A.M. Findlay, S.S. Dalal, H.E. Kirsch, M.S. Berger, S.S. Nagarajan, Mapping functional connectivity in patients with brain lesions. Ann. Neurol. **63**(2), 193–203 (2008)
9. K.P. Westlake, L.B. Hinkley, M. Bucci, A.G. Guggisberg, A.M. Findlay, R.G. Henry, S.S. Nagarajan, N. Byl, Resting state alpha-band functional connectivity and recovery after stroke. Exp. Neurol. **237**(1), 160–169 (2012)
10. P.E. Tarapore, A.M. Findlay, S.C. LaHue, H. Lee, S.M. Honma, D. Mizuiri, T.L. Luks, G.T. Manley, S.S. Nagarajan, P. Mukherjee, Resting state magnetoencephalography functional connectivity in traumatic brain injury: clinical article. J. Neurosurg. **118**(6), 1306–1316 (2013)
11. L.B. Hinkley, J.P. Owen, M. Fisher, A.M. Findlay, S. Vinogradov, S.S. Nagarajan, Cognitive impairments in schizophrenia as assessed through activation and connectivity measures of magnetoencephalography (MEG) data. Front. Hum. Neurosci. **3**, 73 (2009)
12. L.B. Hinkley, S. Vinogradov, A.G. Guggisberg, M. Fisher, A.M. Findlay, S.S. Nagarajan, Clinical symptoms and alpha band resting-state functional connectivity imaging in patients with schizophrenia: implications for novel approaches to treatment. Biol. Psychiatry **70**(12), 1134–1142 (2011)
13. K.G. Ranasinghe, L.B. Hinkley, A.J. Beagle, D. Mizuiri, A.F. Dowling, S.M. Honma, M.M. Finucane, C. Scherling, B.L. Miller, S.S. Nagarajan et al., Regional functional connectivity predicts distinct cognitive impairments in Alzheimers disease spectrum. NeuroImage: Clin. **5**, 385–395 (2014)

Chapter 2
Minimum-Norm-Based Source Imaging Algorithms

2.1 Introduction

In this chapter, we describe the minimum-norm and related methods, which are classic algorithms for electromagnetic brain imaging [1, 2]. In this chapter, the minimum-norm method is first formulated based on the maximum-likelihood principle, and the properties of the minimum-norm solution are discussed. This discussion leads to the necessity of regularization when implementing the minimum-norm method. We discuss two different representative regularization methods: the L_2-norm regularization and the L_1-norm regularization. The minimum-norm method is, then, formulated based on Bayesian inference—Bayesian formulation providing a form of the minimum-norm method where the regularization is already embedded.

2.2 Definitions

In electromagnetic brain imaging, we use an array of sensors to obtain bioelectromagnetic measurements. We define the output of the mth sensor at time t as $y_m(t)$, and the column vector containing outputs from all sensors, such that

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_M(t) \end{bmatrix}, \tag{2.1}$$

where M is the total number of sensors. This column vector $\boldsymbol{y}(t)$ expresses the outputs of the sensor array, and it may be called the data vector or array measurement.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_2

A spatial location is represented by a three-dimensional vector $\boldsymbol{r}$: $\boldsymbol{r} = (x, y, z)$. A source vector is defined as a three-dimensional column vector $\boldsymbol{s}(\boldsymbol{r}, t)$:

$$\boldsymbol{s}(\boldsymbol{r}, t) = \begin{bmatrix} s_x(\boldsymbol{r}, t) \\ s_y(\boldsymbol{r}, t) \\ s_z(\boldsymbol{r}, t) \end{bmatrix}, \tag{2.2}$$

where $s_x(\boldsymbol{r}, t)$, $s_y(\boldsymbol{r}, t)$, and $s_z(\boldsymbol{r}, t)$ are the x, y, and z components. The physical nature of the source vector is the electromotive force generated by neuronal activities in the brain. Additional discussion regarding the nature of the sources is presented in Sect. A.1 in the Appendix. The magnitude of the source vector is denoted as a scalar $s(\boldsymbol{r}, t)$, and the orientation of the source is denoted as a three-dimensional unit vector $\boldsymbol{\eta}(\boldsymbol{r}) = [\eta_x(\boldsymbol{r}), \eta_y(\boldsymbol{r}), \eta_z(\boldsymbol{r})]^T$, where the superscript T indicates the matrix transpose. Then, the relationship

$$\boldsymbol{s}(\boldsymbol{r}, t) = s(\boldsymbol{r}, t)\boldsymbol{\eta}(\boldsymbol{r}) = s(\boldsymbol{r}, t) \begin{bmatrix} \eta_x(\boldsymbol{r}) \\ \eta_y(\boldsymbol{r}) \\ \eta_z(\boldsymbol{r}) \end{bmatrix} \tag{2.3}$$

holds.

2.3 Sensor Lead Field

We assume that a unit-magnitude source exists at $\boldsymbol{r}$. We denote the output of the mth sensor due to this unit-magnitude source as $l_m^x(\boldsymbol{r})$, $l_m^y(\boldsymbol{r})$, and $l_m^z(\boldsymbol{r})$ when the unit-magnitude source is directed in the x, y, and z directions, respectively. The column vectors $\boldsymbol{l}_x(\boldsymbol{r})$, $\boldsymbol{l}_y(\boldsymbol{r})$, and $\boldsymbol{l}_z(\boldsymbol{r})$ are defined as

$$\begin{aligned} \boldsymbol{l}_x(\boldsymbol{r}) &= [l_1^x(\boldsymbol{r}), l_2^x(\boldsymbol{r}), \ldots, l_M^x(\boldsymbol{r})]^T, \\ \boldsymbol{l}_y(\boldsymbol{r}) &= [l_1^y(\boldsymbol{r}), l_2^y(\boldsymbol{r}), \ldots, l_M^y(\boldsymbol{r})]^T, \\ \boldsymbol{l}_z(\boldsymbol{r}) &= [l_1^z(\boldsymbol{r}), l_2^z(\boldsymbol{r}), \ldots, l_M^z(\boldsymbol{r})]^T. \end{aligned}$$

These vectors express the sensor array sensitivity for a source located at $\boldsymbol{r}$ and directed in the x, y, and z directions. Using these column vectors, the sensitivity of the whole sensor array for a source at $\boldsymbol{r}$ is expressed using an $M \times 3$ matrix:

$$\boldsymbol{L}(\boldsymbol{r}) = [\boldsymbol{l}_x(\boldsymbol{r}), \boldsymbol{l}_y(\boldsymbol{r}), \boldsymbol{l}_z(\boldsymbol{r})]. \tag{2.4}$$

This matrix $\boldsymbol{L}(\boldsymbol{r})$ is called the lead-field matrix. We also define the lead-field vector, $\boldsymbol{l}(\boldsymbol{r})$, that expresses the sensitivity of the sensor array in a particular source direction $\boldsymbol{\eta}(\boldsymbol{r})$, such that

$$\boldsymbol{l}(\boldsymbol{r}) = \boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}(\boldsymbol{r}). \tag{2.5}$$

The problem of estimating the sensor lead field is referred to as the bioelectromagnetic forward problem. Arguments on how to compute the sensor lead field are presented in Appendix A.

2.4 Voxel Source Model and Tomographic Source Reconstruction

Using the lead-field matrix in Eq. (2.4), the relationship between the sensor data, $\boldsymbol{y}(t)$, and the source vector, $\boldsymbol{s}(\boldsymbol{r}, t)$, is expressed as

$$\boldsymbol{y}(t) = \int_{\Omega} \boldsymbol{L}(\boldsymbol{r})\boldsymbol{s}(\boldsymbol{r}, t)\, d\boldsymbol{r}. \tag{2.6}$$

Here, $d\boldsymbol{r}$ indicates the volume element, and the integral is performed over a volume where sources are assumed to exist. This volume is called the source space, which is denoted Ω. Equation (2.6) expresses the relationship between the sensor outputs $\boldsymbol{y}(t)$ and the source distribution $\boldsymbol{s}(\boldsymbol{r}, t)$.

The bioelectromagnetic inverse problem is the problem of estimating the source-vector spatial distribution, $\boldsymbol{s}(\boldsymbol{r}, t)$, from the measurements, $\boldsymbol{y}(t)$. Here, we assume that we know the sensor lead field $\boldsymbol{L}(\boldsymbol{r})$, although our knowledge of the sensor lead field is to some degree imperfect because it must be estimated using an analytical model or numerical computations.

When estimating $\boldsymbol{s}(\boldsymbol{r}, t)$ from $\boldsymbol{y}(t)$, $\boldsymbol{s}(\boldsymbol{r}, t)$ is continuous in space, while $\boldsymbol{y}(t)$ is discrete in space. A common strategy here is to introduce voxel discretization over the source space. Let us define the number of voxels as N, and the locations of the voxels are denoted as $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N$. Then, the discrete form of Eq. (2.6) is expressed as:

$$\boldsymbol{y}(t) = \sum_{j=1}^{N} \boldsymbol{L}(\boldsymbol{r}_j)\boldsymbol{s}(\boldsymbol{r}_j, t) = \sum_{j=1}^{N} \boldsymbol{L}(\boldsymbol{r}_j)\boldsymbol{s}_j(t). \tag{2.7}$$

where the source vector at the jth voxel, $\boldsymbol{s}(\boldsymbol{r}_j, t)$, is denoted $\boldsymbol{s}_j(t)$ for simplicity. We introduce the augmented lead-field matrix over all voxel locations as

$$\boldsymbol{F} = [\boldsymbol{L}(\boldsymbol{r}_1), \boldsymbol{L}(\boldsymbol{r}_2), \ldots, \boldsymbol{L}(\boldsymbol{r}_N)], \tag{2.8}$$

which is an $M \times 3N$ matrix. We define a $3N \times 1$ column vector containing the source vectors at all voxel locations, $\boldsymbol{x}(t)$, such that

$$\boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{s}_1(t) \\ \boldsymbol{s}_2(t) \\ \vdots \\ \boldsymbol{s}_N(t) \end{bmatrix}. \tag{2.9}$$

Equation (2.7) is then rewritten as

$$\boldsymbol{y}(t) = [\boldsymbol{L}(\boldsymbol{r}_1), \boldsymbol{L}(\boldsymbol{r}_2), \ldots, \boldsymbol{L}(\boldsymbol{r}_N)] \begin{bmatrix} \boldsymbol{s}_1(t) \\ \boldsymbol{s}_2(t) \\ \vdots \\ \boldsymbol{s}_N(t) \end{bmatrix} = \boldsymbol{F}\boldsymbol{x}(t). \tag{2.10}$$

Here, since the augmented lead-field matrix $\boldsymbol{F}$ is a known quantity, the only unknown quantity is the $3N \times 1$ column vector, $\boldsymbol{x}(t)$. This vector $\boldsymbol{x}(t)$ is called the voxel source vector.

The spatial distribution of the source orientation, $\boldsymbol{\eta}(\boldsymbol{r})$, may be a known quantity if accurate subject anatomical information (such as high-precision subject MRI) can be obtained with accurate co-registration between the MRI coordinate and the sensor coordinate. In this case, the inverse problem is the problem of estimating the source magnitude, $s(\boldsymbol{r}, t)$, instead of the source vector, $\boldsymbol{s}(\boldsymbol{r}, t)$. Let us consider a situation in which the source orientations at all voxel locations are predetermined. Defining the orientation of a source at the jth voxel as $\boldsymbol{\eta}_j$, the lead field at the jth voxel is expressed as the column vector $\boldsymbol{l}_j$, which is obtained as $\boldsymbol{l}_j = \boldsymbol{L}(\boldsymbol{r}_j)\boldsymbol{\eta}_j$, according to Eq. (2.5). Thus, the augmented lead field is expressed as an $M \times N$ matrix $\boldsymbol{H}$ defined such that

$$\boldsymbol{H} = [\boldsymbol{L}(\boldsymbol{r}_1)\boldsymbol{\eta}_1, \boldsymbol{L}(\boldsymbol{r}_2)\boldsymbol{\eta}_2, \ldots, \boldsymbol{L}(\boldsymbol{r}_N)\boldsymbol{\eta}_N] = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_N], \tag{2.11}$$

whereby Eq. (2.10) can be reduced as follows:

$$\begin{aligned} \boldsymbol{y}(t) &= [\boldsymbol{L}(\boldsymbol{r}_1), \boldsymbol{L}(\boldsymbol{r}_2), \ldots, \boldsymbol{L}(\boldsymbol{r}_N)] \begin{bmatrix} \boldsymbol{s}_1(t) \\ \boldsymbol{s}_2(t) \\ \vdots \\ \boldsymbol{s}_N(t) \end{bmatrix} \\ &= [\boldsymbol{L}(\boldsymbol{r}_1), \boldsymbol{L}(\boldsymbol{r}_2), \ldots, \boldsymbol{L}(\boldsymbol{r}_N)] \begin{bmatrix} \boldsymbol{\eta}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\eta}_2 & \cdot & \vdots \\ \vdots & \cdot & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\eta}_N \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \\ &= [\boldsymbol{L}(\boldsymbol{r}_1)\boldsymbol{\eta}_1, \boldsymbol{L}(\boldsymbol{r}_2)\boldsymbol{\eta}_2, \ldots, \boldsymbol{L}(\boldsymbol{r}_N)\boldsymbol{\eta}_N] \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} = \boldsymbol{H}\boldsymbol{x}(t). \end{aligned} \tag{2.12}$$

Thus, the voxel source vector $\boldsymbol{x}(t)$, in this case, is an $N \times 1$ column vector,

$$\boldsymbol{x}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix}, \tag{2.13}$$

in which the jth component of $\boldsymbol{x}(t)$ is $s_j(t)$, which is the scalar intensity at the jth voxel. In this book, the same notation $\boldsymbol{x}(t)$ is used to indicate either the $3N \times 1$ vector in Eq. (2.9) or the $N \times 1$ vector in Eq. (2.13), unless any confusion arises.

In summary, denoting the additive noise in the sensor data ε, the relationship between the sensor data $\boldsymbol{y}(t)$ and the voxel source vector $\boldsymbol{x}(t)$ is expressed as

$$\boldsymbol{y}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \varepsilon, \tag{2.14}$$

where $\boldsymbol{x}(t)$ is a $3N \times 1$ column vector in Eq. (2.9). When voxels have predetermined orientations, using the augmented lead field matrix $\boldsymbol{H}$ in Eq. (2.11), the relationship between $\boldsymbol{y}(t)$ and $\boldsymbol{x}(t)$ is expressed as

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \varepsilon, \tag{2.15}$$

where $\boldsymbol{x}(t)$ is an $N \times 1$ column vector in Eq. (2.13).

2.5 Maximum Likelihood Principle and the Least-Squares Method

When estimating the unknown quantity $\boldsymbol{x}$ from the sensor data $\boldsymbol{y}$, the basic principle is to interpret the data $\boldsymbol{y}$ as a realization of most probable events. That is, the sensor data $\boldsymbol{y}$ is considered the result of the most likely events. We call this the maximum likelihood principle. In this chapter, we first derive the maximum likelihood solution of the unknown source vector $\boldsymbol{x}$.

We assume that the noise distribution is Gaussian, i.e.,

$$\varepsilon \sim \mathcal{N}(\varepsilon|\boldsymbol{0}, \sigma^2\boldsymbol{I}).$$

Namely, the noise in the sensor data is the identically and independently distributed Gaussian noise with a mean of zero, and the same variance σ^2. According to (C.1) in the Appendix, the explicit form of the noise probability distribution is given by

$$p(\varepsilon) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left[-\frac{1}{2\sigma^2}\|\varepsilon\|^2\right]. \tag{2.16}$$

Since the linear relationship in Eq. (2.14) holds, the probability distribution of the sensor data $\boldsymbol{y}(t)$ is expressed as

$$p(\boldsymbol{y}) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left[-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2\right], \tag{2.17}$$

where the explicit time notation (t) is omitted from the vector notations $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ for simplicity.[1]

This $p(\boldsymbol{y})$ as a function of the unknown parameter $\boldsymbol{x}$ is called the likelihood function, and the maximum likelihood estimate $\widehat{\boldsymbol{x}}$ is obtained such that[2]

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmax}} \log p(\boldsymbol{y}), \tag{2.18}$$

where $\log p(\boldsymbol{y})$ is called the log-likelihood function. Using the probability distribution in Eq. (2.17), the log-likelihood function $\log p(\boldsymbol{y})$ is expressed as

$$\log p(\boldsymbol{y}) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 + \mathcal{C}, \tag{2.19}$$

where $\mathcal{C}$ expresses terms that do not contain $\boldsymbol{x}$. Therefore, the $\boldsymbol{x}$ that maximizes $\log p(\boldsymbol{y})$ is equal to the one that minimizes $\mathcal{F}(\boldsymbol{x})$ defined such that

$$\mathcal{F}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2. \tag{2.20}$$

That is, the maximum likelihood solution $\widehat{\boldsymbol{x}}$ is obtained using

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmin}} \mathcal{F}(\boldsymbol{x}) : \quad \text{where} \quad \mathcal{F} = \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2. \tag{2.21}$$

This $\mathcal{F}(\boldsymbol{x})$ in Eq. (2.20) is referred to as the least-squares cost function, and the method that estimates $\boldsymbol{x}$ through the minimization of the least-squares cost function is the method of least-squares.

2.6 Derivation of the Minimum-Norm Solution

In the bioelectromagnetic inverse problem, the number of voxels N, in general, is much greater than the number of sensors M. Thus, the estimation of the source vector $\boldsymbol{x}$ is an ill-posed problem. When applying the least-squares method to such an ill-posed problem, the problem arises that an infinite number of $\boldsymbol{x}$ could make the cost

[1] For the rest of this chapter, the explicit time notation is omitted from these vector notations, unless otherwise noted.

[2] The notation argmax indicates the value of $\boldsymbol{x}$ that maximizes $\log p(\boldsymbol{y})$ which is an implicit function of $\boldsymbol{x}$.

function equal to zero. Therefore, we cannot obtain an optimum solution of $\boldsymbol{x}$ based only on the least-squares method.

A general strategy for overcoming this problem is to integrate a "desired property" of the unknown parameter $\boldsymbol{x}$ into the estimation problem. That is, we choose $\boldsymbol{x}$ so as to maximize this "desired property," and also satisfy $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$. Quite often, a small norm of the solution vector is used as this "desired property," and in this case, the optimum estimate $\widehat{\boldsymbol{x}}$ is obtained using

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|^2 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}. \tag{2.22}$$

In the optimization above, the notation of "subject to" indicates a constraint, (i.e., the above optimization requires that the estimate $\widehat{\boldsymbol{x}}$ be chosen such that $\boldsymbol{x}$ minimizes $\|\boldsymbol{x}\|^2$ as well as satisfies $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$.) To solve the constraint optimization problem in Eq. (2.22), we use the method of Lagrange multipliers that can convert a constrained optimization problem to an unconstrained optimization problem. In this method, using an $M \times 1$ column vector $\boldsymbol{c}$ as the Lagrange multipliers, we define a function called the Lagrangian $\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})$ such that

$$\mathbb{L}(\boldsymbol{x}, \boldsymbol{c}) = \|\boldsymbol{x}\|^2 + \boldsymbol{c}^T (\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}). \tag{2.23}$$

The solution $\widehat{\boldsymbol{x}}$ is obtained by minimizing $\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})$ above with respect to $\boldsymbol{x}$ and $\boldsymbol{c}$—the solution $\widehat{\boldsymbol{x}}$ being equal to $\widehat{\boldsymbol{x}}$ obtained by solving the constrained optimization in Eq. (2.22).

To derive an $\boldsymbol{x}$ that minimizes Eq. (2.23), we compute the derivatives of $\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})$ with respect to $\boldsymbol{x}$ and $\boldsymbol{c}$, and set them to be zero, giving

$$\frac{\partial \mathbb{L}(\boldsymbol{x}, \boldsymbol{c})}{\partial \boldsymbol{x}} = 2\boldsymbol{x} - \boldsymbol{F}^T \boldsymbol{c} = 0, \tag{2.24}$$

$$\frac{\partial \mathbb{L}(\boldsymbol{x}, \boldsymbol{c})}{\partial \boldsymbol{c}} = \boldsymbol{y} - \boldsymbol{F}\boldsymbol{x} = 0. \tag{2.25}$$

Using the equations above, we can derive

$$\widehat{\boldsymbol{x}} = \boldsymbol{F}^T \left(\boldsymbol{F}\boldsymbol{F}^T\right)^{-1} \boldsymbol{y}. \tag{2.26}$$

The solution in Eq. (2.26) is called the minimum-norm solution, which is well known as a solution for the ill-posed linear inverse problem.

2.7 Properties of the Minimum-Norm Solution

The minimum-norm solution is expressed as

$$\widehat{\boldsymbol{x}} = \boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1}(\boldsymbol{F}\boldsymbol{x} + \boldsymbol{\varepsilon}) = \boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1}\boldsymbol{F}\boldsymbol{x} + \boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1}\boldsymbol{\varepsilon}. \tag{2.27}$$

The first term on the right-hand side is expressed as $E(\widehat{\boldsymbol{x}})$, which indicates the expectation of $\widehat{\boldsymbol{x}}$. This term represents how the solution deviates from its true value even in the noiseless cases . The second term indicates the influence of the noise $\boldsymbol{\varepsilon}$. The first term is rewritten as

$$E(\widehat{\boldsymbol{x}}) = \boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1}\boldsymbol{F}\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{x}, \tag{2.28}$$

where

$$\boldsymbol{Q} = \boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1}\boldsymbol{F}. \tag{2.29}$$

Apparently, the first term is not equal to the true value $\boldsymbol{x}$, and the matrix $\boldsymbol{Q}$ in Eq. (2.29) expresses the relationship between the true value $\boldsymbol{x}$ and the estimated value $E(\widehat{\boldsymbol{x}})$.

Denoting the (i, j)th element of $\boldsymbol{Q}$ as $Q_{i,j}$, the jth element of $E(\widehat{\boldsymbol{x}})$, $E(\widehat{x}_j)$, is expressed as

$$E(\widehat{x}_j) = \sum_{k=1}^{N} Q_{j,k}x_k. \tag{2.30}$$

The above equation shows how each element of the true vector $\boldsymbol{x}$ affects the value of $E(\widehat{x}_j)$. That is, $Q_{j,k}$ expresses the amount of leakage of x_k into $\widehat{x}_j$ when $j \neq k$. If the weight $Q_{j,1}, \ldots, Q_{j,N}$ has a sharp peak at j, $\widehat{x}_j$ may be close to the true value x_j. If the weight has no clear peak or if the weight has a peak at j' that is different from j, $\widehat{x}_j$ may be very different from x_j. Because of such properties, the matrix $\boldsymbol{Q}$ is called the resolution matrix.

We next examine the second term, which expresses the noise influence. The noise influence is related to the singular values of $\boldsymbol{F}$. The singular value decomposition of $\boldsymbol{F}$ is defined as

$$\boldsymbol{F} = \sum_{j=1}^{M} \gamma_j \boldsymbol{u}_j \boldsymbol{v}_j^T, \tag{2.31}$$

where we assume that $M < N$, and the singular values are numbered in decreasing order. Using

$$\boldsymbol{F}^T(\boldsymbol{F}\boldsymbol{F}^T)^{-1} = \sum_{j=1}^{N} \frac{1}{\gamma_j}\boldsymbol{v}_j\boldsymbol{u}_j^T, \tag{2.32}$$

we can express the second term in Eq. (2.27) as

$$\sum_{j=1}^{N} \frac{(\boldsymbol{u}_j^T\boldsymbol{\varepsilon})}{\gamma_j}\boldsymbol{v}_j. \tag{2.33}$$

The equation above shows that the denominator contains the singular values. Thus, if higher order singular values are very small and close to zero, the terms containing such small singular values amplify the noise influence, resulting in a situation where

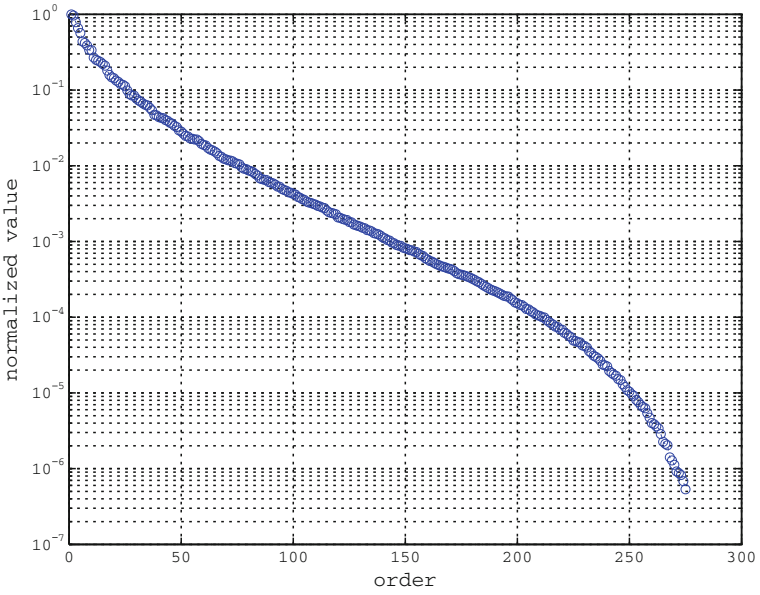

Fig. 2.1 Typical plot of the singular values of the lead field matrix $\boldsymbol{F}$. We assume the 275-channel CTF Omega whole-head MEG sensor system (VMS MedTech Ltd., BC, Canada). A typical location of the subject head relative to the whole head sensor array is assumed. An $8 \times 8 \times 10$ cm region is also assumed as the source space within the subject's head. The spherical homogeneous conductor model is used for computing the sensor lead field. The singular values are normalized with the maximum (i.e., the first) singular value

the second term is dominated in Eq. (2.27), and the minimum norm solution would contain large errors due to the noise.

A plot of a typical singular-value spectrum of the lead field matrix $\boldsymbol{F}$ is shown in Fig. 2.1. To obtain the plot, we used the sensor array of the 275-channel CTF Omega whole-head MEG sensor system (VMS MedTech Ltd., BC, Canada) and spherical homogeneous conductor model to compute the sensor lead field [3].[3] The plot shows that higher order singular values of the lead field matrix are very small. In Fig. 2.1, the ratio of the maximum and minimum singular values reaches the order of 10^{-7}. Therefore, the minimum-norm method in Eq. (2.26) generally produces results highly susceptible to the noise in the sensor data.

2.8 L_2-Regularized Minimum-Norm Solution

When a large amount of noise is overlapped onto the sensor data $\boldsymbol{y}$, if we seek a solution that satisfies $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$, the resultant solution $\boldsymbol{x}$ would be severely affected by the noise. In other words, when noise exists in the sensor data, it is more or less

[3] Computing the lead field using the spherical homogeneous conductor model is explained in Sect. A.2.4 in the Appendix.

meaningless to impose the constraint $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$, so, instead of using the optimization in Eq. (2.22), we should use

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|^2 \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 \leq d, \tag{2.34}$$

where d is a positive constant. In Eq. (2.34), the condition $\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 \leq d$ does not require $\boldsymbol{F}\boldsymbol{x}$ to be exactly equal to $\boldsymbol{y}$, but allow $\boldsymbol{F}\boldsymbol{x}$ to be different from $\boldsymbol{y}$ within a certain range specified by d. Therefore, the solution $\widehat{\boldsymbol{x}}$ is expected to be less affected by the noise in the sensor data $\boldsymbol{y}$.

Unfortunately, there is no closed-form solution for the optimization problem in Eq. (2.34), because of the inequality constraint. Although we can solve Eq. (2.34) numerically, we proceed in solving it by replacing the inequality constraint with the equality constraint. This is possible because the solution of Eq. (2.34) generally exists on the border of the constraint. Thus, we can change the optimization problem in Eq. (2.34) to

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|^2 \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 = d. \tag{2.35}$$

Since this is an equality-constraint problem, we can use the method of Lagrange multipliers. Using the Lagrange multiplier λ, the Lagrangian is defined as

$$\mathbb{L}(\boldsymbol{x}, \boldsymbol{c}) = \|\boldsymbol{x}\|^2 + \lambda \left(\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 - d \right). \tag{2.36}$$

Thus, the solution $\widehat{\boldsymbol{x}}$ is given as

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ \mathbb{L}(\boldsymbol{x}, \boldsymbol{c}) = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[\|\boldsymbol{x}\|^2 + \lambda \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 \right]. \tag{2.37}$$

In the above expression, we disregard the term $-\lambda d$, which does not affect the results of the minimization. Also, we can see that the multiplier λ works as a balancer between the L_2-norm[4] term $\|\boldsymbol{x}\|^2$ and the squared error term $\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2$.

To derive the solution of $\boldsymbol{x}$ that minimizes $\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})$, we compute the derivative of $\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})$ with respect to $\boldsymbol{x}$ and set it to zero, i.e.,

$$\begin{aligned} \frac{\mathbb{L}(\boldsymbol{x}, \boldsymbol{c})}{\partial \boldsymbol{x}} &= \frac{1}{\partial \boldsymbol{x}} \left(\boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{x}^T \boldsymbol{F}^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{F}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{F}^T \boldsymbol{F}\boldsymbol{x} + \xi \boldsymbol{x}^T \boldsymbol{x} \right) \\ &= -2\boldsymbol{F}^T \boldsymbol{y} + 2 \left(\boldsymbol{F}^T \boldsymbol{F} + \xi \boldsymbol{I} \right) \boldsymbol{x} = \boldsymbol{0}, \end{aligned} \tag{2.38}$$

where we use $1/\lambda = \xi$. We can then derive

$$\widehat{\boldsymbol{x}} = \left(\boldsymbol{F}^T \boldsymbol{F} + \xi \boldsymbol{I} \right)^{-1} \boldsymbol{F}^T \boldsymbol{y}. \tag{2.39}$$

[4] A brief summary of the norm of vectors is presented in Sect. C.4 in the Appendix.

Using the matrix inversion lemma in Eq. (C.92), we obtain

$$\widehat{\boldsymbol{x}} = \boldsymbol{F}^T \left(\boldsymbol{F}\boldsymbol{F}^T + \xi \boldsymbol{I} \right)^{-1} \boldsymbol{y}. \tag{2.40}$$

The solution in Eq. (2.40) is called the L_2-norm-regularized minimum-norm solution, or simply L_2-regularized minimum-norm solution.

Let us compute the noise influence term for the L_2-regularized minimum-norm solution. Using Eq. (2.31), we have

$$\boldsymbol{F}^T \left(\boldsymbol{F}\boldsymbol{F}^T + \xi \boldsymbol{I} \right)^{-1} = \sum_{j=1}^{N} \frac{\gamma_j}{\gamma_j^2 + \xi} \boldsymbol{v}_j \boldsymbol{u}_j^T, \tag{2.41}$$

and the L_2-regularized minimum-norm solution is expressed as

$$\begin{aligned} \widehat{\boldsymbol{x}} &= \boldsymbol{F}^T \left(\boldsymbol{F}\boldsymbol{F}^T + \xi \boldsymbol{I} \right)^{-1} (\boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon}) \\ &= \boldsymbol{F}^T \left(\boldsymbol{F}\boldsymbol{F}^T + \xi \boldsymbol{I} \right)^{-1} \boldsymbol{H}\boldsymbol{x} + \sum_{j=1}^{N} \frac{\gamma_j}{\gamma_j^2 + \xi} \boldsymbol{v}_j \boldsymbol{u}_j^T \boldsymbol{\varepsilon}. \end{aligned} \tag{2.42}$$

The second term, expressing the influence of noise, is

$$\sum_{j=1}^{N} \frac{\gamma_j (\boldsymbol{u}_j^T \boldsymbol{\varepsilon})}{\gamma_j^2 + \xi} \boldsymbol{v}_j. \tag{2.43}$$

In the expression above, the denominator contains the positive constant ξ, and it is easy to see that this ξ prevents the terms with smaller singular values from being amplified.

One problem here is how to choose an appropriate value for ξ. Our argument above only suggests that if the noise is large, we need a large ξ, but if small, a smaller ξ can be used. However, the arguments above do not lead to the derivation of an appropriate ξ. We will return to this problem in Sect. 2.10.2 where L_2-regularized minimum-norm solution is re-derived based on a Bayesian formulation, in which deriving the optimum ξ is embedded.

2.9 L_1-Regularized Minimum-Norm Solution

2.9.1 L_1-Norm Constraint

In the preceding section, we derived a solution that minimizes the L_2-norm of the solution vector $\boldsymbol{x}$. In this section, we argue for a solution that minimizes the L_1-norm of $\boldsymbol{x}$, which is defined in Eq. (C.64). The L_1-norm-regularized solution is obtained

using [4–6]

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{j=1}^{N} |x_j| \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 = d. \tag{2.44}$$

The only difference between the equation above and Eq. (2.35) is to minimize either L_2 norm $\|\boldsymbol{x}\|^2$ in Eq. (2.35) or L_1 norm, $\|\boldsymbol{x}\|_1 = \sum_j |x_j|$ in Eq. (2.44). Although it may look as if there is no significant difference between the two methods, the results of source estimation are significantly different. The L_1-norm regularization gives a "so-called" sparse solution, in which only few x_j have nonzero values and a majority of other x_j have values close to zero.

Using the method of Lagrange multipliers and following exactly the same arguments as in Sect. 2.8, the L_1-norm solution can be obtained by minimizing the cost function $\mathcal{F}$, i.e.,

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \mathcal{F}: \quad \mathcal{F} = \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 + \xi \sum_{j=1}^{N} |x_j|, \tag{2.45}$$

where again ξ is a positive constant that controls the balance between the first and the second terms in the cost function above. Unfortunately, the minimization problem in Eq. (2.45) does not have a closed-form solution, so numerical methods are used here to obtain the solution $\widehat{\boldsymbol{x}}$.

2.9.2 Intuitive Explanation for Sparsity

Actually, it is not easy to provide an intuitive explanation regarding why the optimization in Eq. (2.44) or (2.45) causes a sparse solution. The straightforward (and intuitively clear) formulation to obtain a sparse solution should use the L_0-norm minimization, such that

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{j=1}^{N} \mathcal{T}(x_j) \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 = d, \tag{2.46}$$

where the function $\mathcal{T}(x)$ is defined in Eq. (C.65). In the above formulation, since $\sum_{j=1}^{N} \mathcal{T}(x_j)$ indicates the number of nonzero x_j, $\widehat{\boldsymbol{x}}$ is the solution that has the smallest number of nonzero x_j and still satisfies $\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 = d$. The optimization problem in Eq. (2.46) is known to require impractically long computational time. The optimization for the L_1-norm cost function in Eq. (2.44) approximates this L_0-norm optimization in Eq. (2.46) so as to obtain a sparse solution within a reasonable range of computational time [7].

The regularization methods mentioned above can be summarized to have a form of the cost functions expressed as

$$\mathcal{F} = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \xi\phi(\boldsymbol{x}). \tag{2.47}$$

The first term is the data-fitting term, and the second term $\phi(\boldsymbol{x})$ expresses the constraint, which has the following form for general L_p-norm cases ($0 \le p \le 1$):

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{N} \mathcal{T}(x_j) \quad \text{for } L_0\text{-norm}, \tag{2.48}$$

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{N} |x_j| \quad \text{for } L_1\text{-norm}, \tag{2.49}$$

$$\phi(\boldsymbol{x}) = \left[\sum_{j=1}^{N} x_j^p\right]^{1/p} \quad \text{for } L_p\text{-norm}. \tag{2.50}$$

The plots of $\phi(\boldsymbol{x})$ with respect to the x_j axis are shown in Fig. 2.2. In this figure, the four kinds of plots of $\phi(\boldsymbol{x}) = \|\boldsymbol{x}\|_p$ when $p = 0$, $p = 0.3$, $p = 0.7$, and $p = 1$

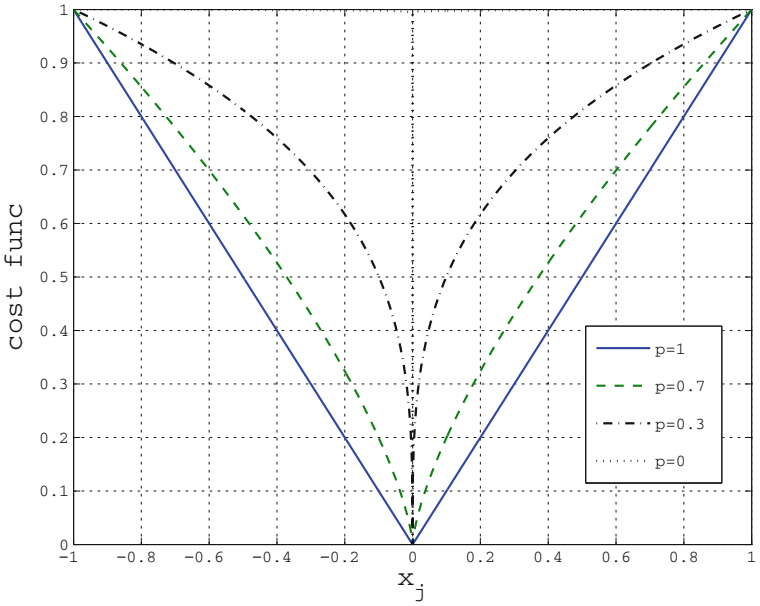

Fig. 2.2 Plots of objective function $\phi(\boldsymbol{x})$ defined in Eqs. (2.48)–(2.50) with respect to the x_j axis. The four cases of $p = 0$, $p = 0.3$, $p = 0.7$, and $p = 1$ are shown. The cases of $p = 0$, and $p = 1$ correspond to the L_0 and L_1 norm constraints (A brief summary of the norm of vectors is presented in Sect. C.4 in the Appendix.)

are shown. It can be seen in this figure that the L_0-norm constraint is approximated by the L_p-norm constraint, and as p becomes closer to 0, the L_p-norm provides a better approximation.

Let us see how the L_p-norm regularization causes sparse solutions when $0 \leq p \leq 1$. To do so, we consider a simplest estimation problem in which only two voxels exist and the voxels have source intensity of x_1 and x_2. We assume a noiseless measurement using a single-sensor; the sensor data being represented by a scalar y. The optimization for the L_1-norm solution is expressed in this case as

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left(|x_1| + |x_2|\right) \quad \text{subject to} \quad y = h_1 x_1 + h_2 x_2, \tag{2.51}$$

where $\widehat{\boldsymbol{x}} = (\widehat{x}_1, \widehat{x}_2)^T$, and h_1 and h_2 are the sensor lead field. For the sake of comparison, we also argue the L_2-norm regularization whose optimization is given as follows:

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left(x_1^2 + x_2^2\right)^{1/2} \quad \text{subject to} \quad y = h_1 x_1 + h_2 x_2. \tag{2.52}$$

The optimization process is depicted in Fig. 2.3. In Fig. 2.3a, the tetragon at the center represents the L_1-norm objective function, $|x_1| + |x_2| =$ constant. The broken line represents the x_1 and x_2 that satisfy the measurement equation $y = h_1 x_1 + h_2 x_2$. Thus, as a result of the optimization in Eq. (2.51), the x_1 and x_2 on the broken line that minimize $|x_1| + |x_2|$ should be chosen as the solution, i.e., the point (x_1, x_2) at which the tetragon touches the broken line is chosen as the solution for the optimization. Such solution is indicated by the small filled circle in Fig. 2.3a. In this solution, x_2 has a nonzero value but x_1 is zero, i.e., a sparse solution is obtained. It can be seen in this figure that in most cases, the point at which the tetragon touches the broken

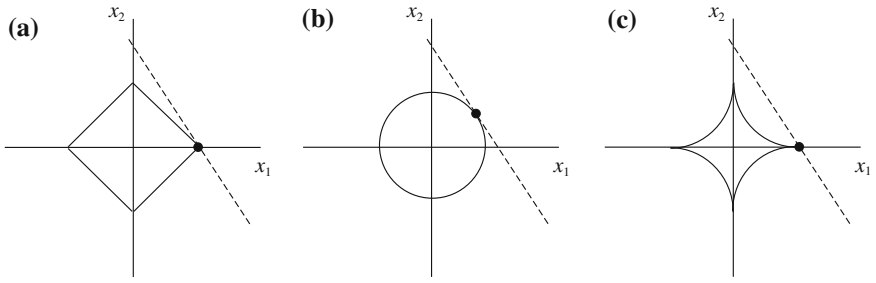

Fig. 2.3 The optimization process is depicted for the simple case in which a single sensor and two voxels exist. Source magnitudes at the voxels are represented by x_1 and x_2. The broken lines represent the x_1 and x_2 that satisfy the measurement equation, $y = h_1 x_1 + h_2 x_2$. The *filled black circles* indicate an example of the solution for each case. **a** L_1-norm regularization in Eq. (2.51). The tetragon at the center represents the L_1-norm objective function $|x_1| + |x_2| =$ constant. **b** L_2-norm regularization in Eq. (2.52). The circle at the center represents the L_2-norm objective function $x_1^2 + x_2^2 =$ constant. **c** L_p-norm regularization where $0 < p < 1$

line is likely to be located at one of its vertices, so a sparse solution is likely to be obtained.

Figure 2.3b shows the case of the L_2-norm minimization in Eq. (2.52). In this figure, the broken line again represents the x_1 and x_2 that satisfy the measurement equation $y = h_1 x_1 + h_2 x_2$, and the circle represents the L_2-norm objective function $x_1^2 + x_2^2 =$ constant. In this case, the x_1 and x_2 on the broken line that minimizes $x_1^2 + x_2^2$ should be chosen, and the resultant solution is (x_1, x_2) at which the circle touches the broken line. An example of such solution is indicated by the small filled circle. In this case, both x_1 and x_2 have nonzero values, and a non-sparse solution is likely to be obtained using L_2-norm regularization.

Finally, Fig. 2.3c shows a case of the general L_p norm minimization ($0 < p < 1$). An example of such solution is indicated by the small, filled circle. Using the general L_p norm regularization, the solution is more likely to be sparse than the case of the L_1-norm minimization. However, the computational burden for the general L_p norm minimization is so high that it is seldom used in practical applications.

2.9.3 Problem with Source Orientation Estimation

When applying the L_1-norm regularization to the bioelectromagnetic source localization, it has been known that the method fails in estimating correct source orientations. The reason for this is described as follows: The components of the solution vector $\boldsymbol{x}$ is denoted explicitly as

$$\boldsymbol{x} = \left[s_1^x, s_1^y, s_1^z, \ldots, s_j^x, s_j^y, s_j^z, \ldots, s_N^x, s_N^y, s_N^z \right]^T,$$

where s_j^x, s_j^y, s_j^z are the x, y, and z components of the source at the jth voxel. When the jth voxel has a source activity, it is generally true that s_j^x, s_j^y, s_j^z have nonzero values. However, when using the L_1 regularization, only one of s_j^x, s_j^y, s_j^z tends to have nonzero value, and others tend to be close to zero because of the nature of a sparse solution. As a result, the source orientation may be erroneously estimated.

To avoid this problem, the source orientation is estimated in advance using some other method [4] such as the L_2-norm minimum-norm method. Then, the L_1-norm method is formulated using the orientation-embedded data model in Eq. (2.15). That is, we use

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \mathcal{F} : \quad \mathcal{F} = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \xi \sum_{j=1}^{N} |x_j|. \tag{2.53}$$

In this case, the sparsity is imposed on the source vector magnitude, $s_1, s_2, \ldots, s_N$, and only a few of $s_1, s_2, \ldots, s_N$ have nonzero values, allowing for the reconstruction of a sparse source distribution.

2.10 Bayesian Derivation of the Minimum-Norm Method

2.10.1 Prior Probability Distribution and Cost Function

In this section, we derive the minimum-norm method based on Bayesian inference. As in Eq. (2.16), we assume that the noise $\boldsymbol{\varepsilon}$ is independently and identically distributed Gaussian, i.e.,

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0}, \beta^{-1}\boldsymbol{I}), \tag{2.54}$$

where the precision β is used, which is the inverse of the noise variance, $\beta^{-1} = \sigma^2$. Thus, using Eq. (2.14), the conditional probability distribution of the sensor data for a given $\boldsymbol{x}$, $p(\boldsymbol{y}|\boldsymbol{x})$ is

$$p(\boldsymbol{y}|\boldsymbol{x}) = \left(\frac{\beta}{2\pi}\right)^{M/2} \exp\left[-\frac{\beta}{2}\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2\right]. \tag{2.55}$$

This conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ is equal to the likelihood $p(\boldsymbol{y})$ in the arguments in Sect. 2.5. Since $\boldsymbol{x}$ is a random variable in the Bayesian arguments, we use the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$, instead of $p(\boldsymbol{y})$.

Let us derive a cost function for estimating $\boldsymbol{x}$. Taking a logarithm of the Bayes's rule in Eq. (B.3) in the Appendix, we have

$$\log p(\boldsymbol{x}|\boldsymbol{y}) = \log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x}) + \mathcal{C}, \tag{2.56}$$

where $\mathcal{C}$ represents the constant terms. Neglecting $\mathcal{C}$, the cost function $\mathcal{F}(\boldsymbol{x})$ in general form is obtained as

$$\mathcal{F}(\boldsymbol{x}) = -2\log p(\boldsymbol{x}|\boldsymbol{y}) = \beta\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 - 2\log p(\boldsymbol{x}). \tag{2.57}$$

The first term on the right-hand side is a squared error term, which expresses how well the solution $\boldsymbol{x}$ fits the sensor data $\boldsymbol{y}$. The second term $-2\log p(\boldsymbol{x})$ is a constraint imposed on the solution. The above equation indicates that the constraint term in the cost function is given from the prior probability distribution in the Bayesian formulation. The optimum estimate of $\boldsymbol{x}$ is obtained by minimizing the cost function $\mathcal{F}(\boldsymbol{x})$.

2.10.2 L_2-Regularized Method

Let us assume the following Gaussian distribution for the prior probability distribution of $\boldsymbol{x}$,

$$p(\boldsymbol{x}) = \left(\frac{\alpha}{2\pi}\right)^{N/2} \exp\left[-\frac{\alpha}{2}\|\boldsymbol{x}\|^2\right]. \tag{2.58}$$

Substituting Eq. (2.58) into (2.57), we get the cost function

$$\mathcal{F}(\boldsymbol{x}) = \beta \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 + \alpha \|\boldsymbol{x}\|^2. \tag{2.59}$$

The cost function in Eq. (2.59) is the same as the cost function in Eq. (2.37), assuming $\lambda = \beta/\alpha$. Thus, the solution obtained by minimizing this cost function is equal to the solution of the L_2-norm regularized minimum-norm method introduced in Sect. 2.8.

To obtain the optimum estimate of $\boldsymbol{x}$, we should compute the posterior distribution. In this case, the posterior is known to have a Gaussian distribution because $p(\boldsymbol{y}|\boldsymbol{x})$ and $p(\boldsymbol{x})$ are both Gaussian, and the mean and the precision matrix of this posterior distribution is derived as in Eqs. (B.24) and (B.25). Substituting $\boldsymbol{\Phi} = \alpha \boldsymbol{I}$ and $\boldsymbol{\Lambda} = \beta \boldsymbol{I}$ into these equations, we have

$$\boldsymbol{\Gamma} = \alpha \boldsymbol{I} + \beta \boldsymbol{F}^T \boldsymbol{F}, \tag{2.60}$$

$$\bar{\boldsymbol{x}}(t) = \left(\boldsymbol{F}^T \boldsymbol{F} + \frac{\alpha}{\beta} \boldsymbol{I} \right)^{-1} \boldsymbol{F}^T \boldsymbol{y}(t). \tag{2.61}$$

The Bayesian solution which minimizes the cost function in Eq. (2.59) is given in Eq. (2.61). This solution is the same as Eq. (2.39). Comparison between Eqs. (2.61) and (2.39) shows that the regularization constant is equal to α/β, which is the inverse of the signal-to-noise ratio of the sensor data. This is in accordance with the arguments in Sect. 2.8 that when the sensor data contains larger amounts of noise, a larger regularization constant must be used.

The optimum values of the hyperparameters α and β can be obtained using the EM algorithm, as described in Sect. B.5.6. The update equations for the hyperparameters are:

$$\widehat{\alpha}^{-1} = \frac{1}{3N} \left[\frac{1}{K} \sum_{k=1}^{K} \bar{\boldsymbol{x}}^T(t_k) \bar{\boldsymbol{x}}(t_k) + \mathrm{tr}\left(\boldsymbol{\Gamma}^{-1} \right) \right], \tag{2.62}$$

$$\widehat{\beta}^{-1} = \frac{1}{M} \left[\frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{y}(t_k) - \boldsymbol{F}\bar{\boldsymbol{x}}(t_k)\|^2 + \mathrm{tr}\left(\boldsymbol{F}^T \boldsymbol{F} \boldsymbol{\Gamma}^{-1} \right) \right]. \tag{2.63}$$

Here, we assume that multiple K time-point data is available to determine α and β.

The Bayesian minimum-norm method is summarized as follows. First, $\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{x}}(t_k)$ are computed using Eqs. (2.60) and (2.61) with initial values set to α and β. Then, the values of α and β are updated using (2.62) and (2.63). Using the updated α and β, the values of $\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{x}}(t_k)$ are updated using Eqs. (2.60) and (2.61). These procedures are repeated and the resultant $\bar{\boldsymbol{x}}(t_k)$ is the optimum estimate of $\boldsymbol{x}(t_k)$.

The EM iteration may be stopped by monitoring the marginal likelihood, which is obtained using Eq. (B.29) as

$$\log p(\boldsymbol{y}(t_1), \ldots, \boldsymbol{y}(t_K)|\alpha, \beta) = -\frac{1}{2} K \log |\boldsymbol{\Sigma}_y| - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}^T(t_k) \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}(t_k), \tag{2.64}$$

where according to Eq. (B.30), $\boldsymbol{\Sigma}_y$ is expressed as

$$\boldsymbol{\Sigma}_y = \beta^{-1}\boldsymbol{I} + \alpha^{-1}\boldsymbol{F}\boldsymbol{F}^T. \tag{2.65}$$

If the increase of the likelihood in Eq. (2.64) with respect to the iteration count becomes very small, the iteration may be stopped.

2.10.3 L_1-Regularized Method

The method of L_1-norm regularization can also be derived based on the Bayesian formulation. To derive the L_1-regularization. we use the Laplace distribution as the prior distribution

$$p(\boldsymbol{x}) = \prod_{j=1}^{N} \frac{1}{2b} \exp\left[-\frac{1}{b}|x_j|\right]. \tag{2.66}$$

Then, using Eq. (2.57), (and replacing $\boldsymbol{F}$ with $\boldsymbol{H}$), the cost function is derived as

$$\mathcal{F}(\boldsymbol{x}) = \beta\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + 2b\sum_{j=1}^{N}|x_j|, \tag{2.67}$$

which is exactly equal to Eq. (2.53), if we set $\xi = 2b/\beta$.

Another formulation for deriving the L_1-regularized method is known. It uses the framework of the sparse Bayesian learning described in Chap. 4. In Chap. 4, assuming the Gaussian prior,

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{j=1}^{N}\mathcal{N}(x_j|0,\alpha_j^{-1}) = \prod_{j=1}^{N}\left(\frac{\alpha_j}{2\pi}\right)^{1/2}\exp\left[-\frac{\alpha_j}{2}x_j^2\right], \tag{2.68}$$

we derive the marginal likelihood for the hyperparameter $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]$, $p(\boldsymbol{y}|\boldsymbol{\alpha})$, using,

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{\alpha})d\boldsymbol{x}, \tag{2.69}$$

and eventually derive the Champagne algorithm. However, instead of implementing Eq. (2.69), there is another option in which we compute the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ using

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\boldsymbol{\alpha} = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \tag{2.70}$$

where

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\boldsymbol{\alpha}. \tag{2.71}$$

The estimate $\widehat{\boldsymbol{x}}$ is, then, obtained by

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmax}}\, p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}).$$

To compute $p(\boldsymbol{x})$ using Eq. (2.71), we need to specify the hyperprior $p(\boldsymbol{\alpha})$. However, we usually have no such information and may use noninformed prior $p(\boldsymbol{\alpha}) = \text{const}$. Substituting this flat prior into Eq. (2.71), we have

$$p(\boldsymbol{x}) \propto \int p(\boldsymbol{x}|\boldsymbol{\alpha})d\boldsymbol{\alpha}.$$

However, the integral in the above equation is difficult to compute. The formal procedure to compute $p(\boldsymbol{x})$ in this case is to first assume the Gamma distribution for the hyperprior $p(\boldsymbol{\alpha})$, such that

$$p(\boldsymbol{\alpha}) = \prod_{j=1}^{N} p(\alpha_j) = \prod_{j=1}^{N} \Gamma(a)^{-1} b^a (\alpha_j)^{a-1} e^{-b\alpha_j}. \tag{2.72}$$

Then, $p(\boldsymbol{x})$ in Eq. (2.71) is known to be obtained as Student t-distribution, such that [8]

$$\begin{aligned} p(x_j) &= \int p(x_j|\alpha_j)p(\alpha_j)d\alpha_j \\ &= \int \left(\frac{\alpha_j}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_j}{2}x_j^2\right) \frac{b^a}{\Gamma(a)} \left(\alpha_j\right)^{a-1} e^{-b\alpha_j} d\alpha_j \\ &= \frac{b^a \Gamma(a+\frac{1}{2})}{\sqrt{2\pi}\Gamma(a)} \left(b + \frac{x_j^2}{2}\right)^{-(a+\frac{1}{2})}. \end{aligned} \tag{2.73}$$

We then assume that $a \to 0$ and $b \to 0$, (which is equivalent to making $p(\boldsymbol{\alpha})$ a noninformed prior,) $p(x_j)$ then becomes

$$p(x_j) \to \frac{1}{|x_j|} \quad \text{i.e.} \quad p(\boldsymbol{x}) \to \prod_{j=1}^{N} \frac{1}{|x_j|}. \tag{2.74}$$

Using Eq. (2.57), the cost function, in this case, is derived as

$$\mathcal{F}(\boldsymbol{x}) = \beta\|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}\|^2 + \sum_{j=1}^{N} \log|x_j|. \tag{2.75}$$

This Eq. (2.75) is not exactly equal to the L_1-norm cost function, since the constraint term is not equal to $\sum_{j=1}^{N} |x_j|$ but has form of $\sum_{j=1}^{N} \log |x_j|$. Since these constraint terms have similar properties, the solution obtained by minimizing this cost function has a property very similar to the L_1-norm-regularized minimum-norm solution. Related arguments are found in Chap. 6.

References

1. M.S. Hämäläinen, R.J. Ilmoniemi, Interpreting measured magnetic fields of the brain: estimates of current distributions. Technical Report TKK-F-A559, Helsinki University of Technology (1984)
2. M.S. Hämäläinen, R.J. Ilmoniemi, Interpreting magnetic fields of the brain: minimum norm estimates. Med. Biol. Eng. Comput. **32**, 35–42 (1994)
3. J. Sarvas, Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. Phys. Med. Biol. **32**, 11–22 (1987)
4. K. Uutela, M. Hämäläinen, E. Somersalo, Visualization of magnetoencephalographic data using minimum current estimate. NeuroImage **10**, 173–180 (1999)
5. B.D. Jeffs, Maximally sparse constrained optimization for signal processing applications. Ph.D. thesis, University of Southern California (1989)
6. K. Matsuura, Y. Okabe, Multiple current-dipole distribution reconstructed by modified selective minimum-norm method, in *Biomag 96*, (Springer, Heidelberg, 2000), pp. 290–293
7. R. Tibshirani, Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B (Methodological) **58**(1), 267–288 (1996)
8. M.E. Tipping, Sparse Bayesian learning and relevance vector machine. J. Mach. Learn. Res. **1**, 211–244 (2001)

Chapter 3
Adaptive Beamformers

3.1 Introduction and Basic Formulation

The beamformer is a location-dependent spatial filter applied to the sensor data. It is used for estimating the strength of brain activity at a particular spatial location, which is referred to as the beamformer pointing location. Thus, the beamformer may be interpreted as a technique that forms a virtual sensor whose sensitivity pattern is localized at its pointing location. By postprocessing, the pointing location can be scanned over the source space to obtain the source three-dimensional reconstruction.

Let us assume momentarily that the source orientation is predetermined and avoid the orientation estimation. In this case, using the data vector, $\boldsymbol{y}(t)$, the beamformer reconstructs the source magnitude, $s(\boldsymbol{r}, t)$, using

$$\widehat{s}(\boldsymbol{r}, t) = \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{y}(t), \tag{3.1}$$

where $\widehat{s}(\boldsymbol{r}, t)$ is the estimated source magnitude at location $\boldsymbol{r}$ and time t. In Eq. (3.1), a column vector $\boldsymbol{w}(\boldsymbol{r})$ expresses the beamformer's weight, which characterizes the properties of the beamformer. There are two types of beamformers. One is the non-adaptive beamformer in which the weight vector depends solely on the lead field of the sensor array, and the other is the adaptive beamformer in which the weight depends on the measured data as well as the lead field of the sensor array.

Adaptive beamformers were originally developed in the field of seismic exploration [1] and introduced later into the field of electromagnetic brain imaging [2–5]. In recent years, brain imaging with adaptive beamformers has increasingly been used for clinical and basic human neuroscience studies, and it is a very popular method for analyzing rhythmic brain activity. You may find detailed arguments on various aspects of adaptive beamformers in [6].

This chapter presents a concise review on the adaptive beamformers. It presents Bayesian-flavored formulations of scalar and vector adaptive beamformers, as well as the conventional derivations. It also describes the narrow-band beamformer and its application to five-dimensional (space–time–frequency) brain imaging [7], which can be used for the source-space connectivity analysis.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_3

3.2 Classical Derivation of Adaptive Beamformers

3.2.1 Minimum-Variance Beamformers with Unit-Gain Constraint

The weight vector of the adaptive beamformer is derived using the optimization:

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}}\, \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{R}\boldsymbol{w}(\boldsymbol{r}), \quad \text{subject to} \quad \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = 1. \tag{3.2}$$

Here, $\boldsymbol{R}$ is the data covariance matrix obtained using $\langle \boldsymbol{y}(t)\boldsymbol{y}^T(t)\rangle$ where $\langle \cdot \rangle$ indicates the ensemble average. In Eq. (3.2), the inner product $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r})$ represents the beamformer output from a unit-magnitude source located at $\boldsymbol{r}$. Therefore, setting $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = 1$ guarantees that the beamformer passes the signal from $\boldsymbol{r}$ with the gain equal to one. The constraint $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = 1$ is called the unit-gain constraint.

The output power of the beamformer $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{R}\boldsymbol{w}(\boldsymbol{r})$ generally contains not only the noise contributions but also unwanted contributions such as the influence of sources at locations other than $\boldsymbol{r}$. Accordingly, by minimizing the output power with this unit-gain constraint, we can derive a weight that minimizes such unwanted influence without affecting the signal coming from $\boldsymbol{r}$, the pointing location of the beamformer.

This constrained minimization problem can be solved using a method of the Lagrange multiplier. We define the Lagrange multiplier as a scalar ζ, and the Lagrangian as $\mathbb{L}(\boldsymbol{w}, \zeta)$, such that

$$\mathbb{L}(\boldsymbol{w}, \zeta) = \boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w} + \zeta(\boldsymbol{w}^T\boldsymbol{l}(\boldsymbol{r}) - 1), \tag{3.3}$$

where the explicit notation of $(\boldsymbol{r})$ is omitted from $\boldsymbol{w}(\boldsymbol{r})$ for simplicity. The weight vector satisfying Eq. (3.2) can be obtained by minimizing the Lagrangian $\mathbb{L}(\boldsymbol{w}, \zeta)$ in Eq. (3.3) with no constraints.

The derivative of $\mathbb{L}(\boldsymbol{w}, \zeta)$ with respect to $\boldsymbol{w}$ is given by:

$$\frac{\partial \mathbb{L}(\boldsymbol{w}, \zeta)}{\partial \boldsymbol{w}} = 2\boldsymbol{R}\boldsymbol{w} + \zeta\boldsymbol{l}(\boldsymbol{r}). \tag{3.4}$$

By setting the right-hand side of the above equation to zero, we obtain

$$\boldsymbol{w} = -\zeta\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})/2. \tag{3.5}$$

Substituting this relationship back into the constraint equation $\boldsymbol{w}^T\boldsymbol{l}(\boldsymbol{r}) = 1$, we get $\zeta = -2/[\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]$. Substituting this ζ into Eq. (3.5), the weight vector satisfying Eq. (3.2) is obtained as

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})}{[\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.6}$$

Using Eq. (3.1), the estimated source magnitude (the beamformer output) is expressed as

$$\widehat{s}(\boldsymbol{r}, t) = \frac{\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{y}(t)}{[\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]}, \tag{3.7}$$

and the output power is given by

$$\langle \widehat{s}(\boldsymbol{r}, t)^2 \rangle = \frac{1}{[\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.8}$$

3.2.2 Minimum-Variance Beamformer with Array-Gain Constraint

We have derived the minimum-variance beamformer with the unit-gain constraint, $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = 1$. However, imposing the unit-gain constraint is somewhat ad hoc, and there may be other possibilities. In electromagnetic brain imaging, the norm of the lead field $\|\boldsymbol{l}(\boldsymbol{r})\|$ represents the gain of the sensor array, and is spatially dependent. Particularly, when the spherical homogeneous conductor model[1] is used, $\|\boldsymbol{l}(\boldsymbol{r})\|$ is zero at the center of the sphere and a false intensity increase around the center of the sphere arises, because the weight vector (in Eq. (3.6)) becomes infinite at the center of the sphere. Such false results occur because the unit-gain constraint forces a nonzero gain at a location where the sensor array has zero gain.

When $\|\boldsymbol{l}(\boldsymbol{r})\|$ has a spatial dependence, it is more reasonable to use the constraint $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|$. We can derive, by using $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|$, a beamformer whose gain exactly matches the gain of the sensor array. The weight, in this case, is obtained using

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}}\, \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{R}\boldsymbol{w}(\boldsymbol{r}), \quad \text{subject to} \quad \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|. \tag{3.9}$$

Exactly, the same derivation used for deriving Eq. (3.6) leads to the weight vector:

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{R}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r})}{[\widetilde{\boldsymbol{l}}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r})]}, \tag{3.10}$$

where $\widetilde{\boldsymbol{l}}(\boldsymbol{r})$ is the normalized lead field vector defined as $\widetilde{\boldsymbol{l}}(\boldsymbol{r}) = \boldsymbol{l}(\boldsymbol{r})/\|\boldsymbol{l}(\boldsymbol{r})\|$. In Eq. (3.10), the weight is independent of the norm of the lead field, and we can avoid the artifacts around the center of the sphere. This type of beamformer is referred to as the array-gain minimum-variance beamformer [6], and the constraint $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|$ is referred to as the array-gain constraint. The estimated source magnitude for the array-gain minimum-variance beamformer is given by

[1] An explanation of the spherical homogeneous conductor model is given in the Appendix A.

$$\widehat{s}(\boldsymbol{r},t)=\frac{\widetilde{\boldsymbol{l}}^{T}(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{y}(t)}{[\widetilde{\boldsymbol{l}}^{T}(\boldsymbol{r})\boldsymbol{R}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r})]}, \tag{3.11}$$

and the output power is expressed as

$$\langle\widehat{s}(\boldsymbol{r},t)^{2}\rangle=\frac{\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r})}{[\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.12}$$

3.2.3 Minimum-Variance Beamformer with Unit-Noise-Gain Constraint

Another possible constraint is the unit-noise-gain constraint, which is expressed as $\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{w}(\boldsymbol{r})=1$. That is, the filter weight is obtained using

$$\begin{aligned}\boldsymbol{w}(\boldsymbol{r})=\underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}}\,\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{R}\boldsymbol{w}(\boldsymbol{r}),\ \text{subject to}\ \boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r})=\tau,\\ \text{and}\ \boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{w}(\boldsymbol{r})=1,\end{aligned} \tag{3.13}$$

where the minimization problem is solved with the first constraint, $\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r})=\tau$. The scalar constant τ is determined by the second constraint, $\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{w}(\boldsymbol{r})=1$. To obtain the weight vector derived from the above minimization, we first calculate the weight using

$$\boldsymbol{w}(\boldsymbol{r})=\underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}}\,\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{R}\boldsymbol{w}(\boldsymbol{r})\quad\text{subject to}\quad\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r})=\tau. \tag{3.14}$$

Following the same steps from Eqs. (3.3) to (3.6), the weight satisfying Eq. (3.14) is obtained as

$$\boldsymbol{w}(\boldsymbol{r})=\tau\frac{\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})}{[\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.15}$$

Substituting this expression to $\boldsymbol{w}^{T}(\boldsymbol{r})\boldsymbol{w}(\boldsymbol{r})=1$ leads to

$$\tau=\frac{\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})}{\sqrt{\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{l}(\boldsymbol{r})}}, \tag{3.16}$$

and the weight is given by:

$$\boldsymbol{w}(\boldsymbol{r})=\frac{\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})}{\sqrt{\boldsymbol{l}^{T}(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{l}(\boldsymbol{r})}}. \tag{3.17}$$

This weight vector again does not depend on the norm of the lead field $\|\boldsymbol{l}(\boldsymbol{r})\|$. The output power of this beamformer is given by

$$\langle \widehat{s}(\boldsymbol{r}, t)^2 \rangle = \frac{\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r})}{[\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.18}$$

This beamformer was first proposed by Borgiotti and Kaplan [8] and it is referred to as the unit-noise-gain (constraint) minimum-variance beamformer,[2] or the weight-normalized minimum-variance beamformer.

3.3 Semi-Bayesian Derivation of Adaptive Beamformers

The adaptive beamformer can be derived based on a Bayesian formulation [9]. Let the relationship between the sensor data $\boldsymbol{y}(t)$ and the voxel source distribution $\boldsymbol{x}(t)$ be expressed in Eq. (2.15), rewritten here as:

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \varepsilon,$$

where $\boldsymbol{H}$ is the lead field matrix defined in Eq. (2.11). We also assume that the noise in the sensor data is assumed such that

$$\varepsilon \sim \mathcal{N}(\varepsilon|\boldsymbol{0}, \sigma^2\boldsymbol{I}). \tag{3.19}$$

We assume that the prior distribution for the source vector $\boldsymbol{x}(t)$ is the zero mean Gaussian with a diagonal precision matrix, i.e.,

$$p(\boldsymbol{x}(t)) = \mathcal{N}(\boldsymbol{x}(t)|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}), \tag{3.20}$$

where the precision matrix is expressed as

$$\boldsymbol{\Phi} = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_N \end{bmatrix}.$$

An entirely rigorous Bayesian treatment using this prior distribution leads to the Champagne algorithm, which is the topic of Chap. 4. In the following, we show that adaptive beamformer algorithm can also be derived using this prior distribution.

[2] This name comes from the fact that spatial filter's noise gain is equal to the squared weight norm $\|\boldsymbol{w}(\boldsymbol{r})\|^2$.

The prior distribution in Eq. (3.20) leads to Bayes' estimate of $\boldsymbol{x}$ expressed in Eq. (B.26) in Appendix. Setting the noise precision $\boldsymbol{\Lambda}$ equal to $\sigma^{-2}\boldsymbol{I}$ in Eq. (B.26), we get

$$\bar{\boldsymbol{x}}(t) = \boldsymbol{\Phi}^{-1}\boldsymbol{H}^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y}(t), \tag{3.21}$$

where

$$\boldsymbol{\Sigma}_y = \boldsymbol{H}\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T + \sigma^2\boldsymbol{I}$$

is the model data covariance matrix, which is the covariance of the marginal distribution $p(\boldsymbol{y}(t))$. Taking a look at the jth component of $\bar{\boldsymbol{x}}(t)$ in Eq. (3.21), we have

$$\bar{x}_j(t) = \widehat{s}(\boldsymbol{r}_j, t) = \alpha_j^{-1}\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y}(t), \tag{3.22}$$

where $\boldsymbol{l}_j$ is the lead field vector: $\boldsymbol{l}_j = \boldsymbol{l}(\boldsymbol{r}_j)$. In the equation above, the voxel precision α_j is an unknown quantity. Thus, computing $\bar{x}_j(t)$ first requires estimating α_j. A rigorous Bayesian treatment for estimating the voxel precision leads to the Champagne algorithm. Here, we perform a non-Bayesian treatment for estimating α_j.

Let us treat $\boldsymbol{\Phi}$ as a deterministic (i.e., nonrandom variable) unknown matrix. Using Eq. (3.22), we compute the power of $\bar{x}_j(t)$ as

$$\langle \bar{x}_j(t)^2 \rangle = \alpha_j^{-2}\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\langle \boldsymbol{y}(t)\boldsymbol{y}^T(t)\rangle\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j. \tag{3.23}$$

If we assume that $\langle \boldsymbol{y}(t)\boldsymbol{y}^T(t)\rangle$ is equal to the model data covariance $\boldsymbol{\Sigma}_y$, Eq. (3.23) may be reexpressed as

$$\langle \widehat{s}(\boldsymbol{r}_j, t)^2 \rangle = \alpha_j^{-2}\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_y\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j = \alpha_j^{-2}\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j. \tag{3.24}$$

Next we assume that the relationship,

$$\langle \widehat{s}(\boldsymbol{r}_j, t)^2 \rangle = \alpha_j^{-1} \tag{3.25}$$

holds. This relationship indicates that the reconstructed power of a brain source is equal to its true power, and this is equal to the unit-gain constraint.

Assuming that the unit-gain constraint holds, we have

$$\alpha_j^{-1} = \alpha_j^{-2}\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j,$$

and thus,

$$\alpha_j^{-1} = \frac{1}{\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j}. \tag{3.26}$$

Substituting this equation into Eq. (3.22), we can derive the beamformer expression,

$$\widehat{s}(\boldsymbol{r}_j, t) = \boldsymbol{w}^T(\boldsymbol{r}_j)\boldsymbol{y}(t), \tag{3.27}$$

where the weight vector is expressed as

$$\boldsymbol{w}(\boldsymbol{r}_j) = \frac{\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j^T}{\boldsymbol{l}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{l}_j}. \tag{3.28}$$

If the model data covariance matrix $\boldsymbol{\Sigma}_y$ is replaced with the sample data covariance matrix $\boldsymbol{R}$, the weight equation (3.28) becomes

$$\boldsymbol{w}(\boldsymbol{r}_j) = \frac{\boldsymbol{R}^{-1}\boldsymbol{l}^T(\boldsymbol{r}_j)}{\boldsymbol{l}^T(\boldsymbol{r}_j)\boldsymbol{R}^{-1}\boldsymbol{l}(\boldsymbol{r}_j)}, \tag{3.29}$$

which is exactly equal to the weight expression of the minimum-variance beamformer. To derive the array-gain constraint beamformer, we use the relationship,

$$\langle \widehat{s}(\boldsymbol{r}_j, t)^2 \rangle = \alpha_j^{-1} \|\boldsymbol{l}(\boldsymbol{r}_j)\|, \tag{3.30}$$

The weight vector in this case is obtained as

$$\boldsymbol{w}(\boldsymbol{r}_j) = \frac{\boldsymbol{R}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r}_j)}{[\widetilde{\boldsymbol{l}}^T(\boldsymbol{r}_j)\boldsymbol{R}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r}_j)]}, \tag{3.31}$$

which is equal to the weight expression of the array-gain constraint minimum-variance beamformer.

3.4 Diagonal-Loading and Bayesian Beamformers

The sample covariance matrix sometimes has a large condition number. This situation happens, for example, when the number of time samples is considerably fewer than the size of the matrix or when the SNR of the sensor data is very high. In such cases, since the direct inversion of the sample covariance $\boldsymbol{R}^{-1}$ may cause numerical instability, the regularized inverse, $(\boldsymbol{R}+\kappa\boldsymbol{I})^{-1}$, may be a better approximation of the inverse of the model data covariance $\boldsymbol{\Sigma}_y^{-1}$, where κ is a small positive real number called the regularization constant. This gives the weight expression,

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{(\boldsymbol{R}+\kappa\boldsymbol{I})^{-1}\boldsymbol{l}(\boldsymbol{r})}{[\boldsymbol{l}^T(\boldsymbol{r})(\boldsymbol{R}+\kappa\boldsymbol{I})^{-1}\boldsymbol{l}(\boldsymbol{r})]}. \tag{3.32}$$

The beamformer that uses the above weight is called the diagonal-loading minimum-variance beamformer [10]. The diagonal loading is a form of regularization used when inverting the sample covariance matrix, and is nearly equivalent to adding noise to the sensor data. Contrary to the L_2-regularization in the minimum-norm method, the diagonal loading is not needed when the SNR of the sensor data is low, because the condition number of the sample covariance is generally low in such cases.

The diagonal loading is needed when the sensor data has high SNR, because, in such cases, a significant SNR degradation is caused due to the problem of array mismatch [6], which indicates a situation where the lead field used for computing the beamformer's weight vector is different from the true lead field. When computing a weight vector, the exact lead field is generally unknown and the lead field is usually estimated from some kind of forward models such as the homogeneous conductor model. Thus, we cannot completely avoid this array mismatch problem. The diagonal loading can reduce the SNR degradation due to the array mismatch. On the other hand, however, it degrades the spatial resolution of reconstructed source distribution, so it provides a trade-off between the SNR degradation and spatial resolution [6].

The sample data covariance $\boldsymbol{R}$, which is the maximum-likelihood estimate of the model data covariance, is not necessarily the best estimate of $\boldsymbol{\Sigma}_y$. We can obtain a better estimate by applying the Bayesian factor analysis described in Chap. 5. Applying the VBFA algorithm, we can get an estimate of the data covariance $\bar{\boldsymbol{R}}$, as shown in Eq. (5.104). The beamformer with a weight expression computed using a Bayesian-inferred data covariance matrix, (such as $\bar{\boldsymbol{R}}$ in Eq. (5.104)), is called the Bayesian beamformer [11]. Note that a Bayesian-inferred data covariance matrix has an intrinsic regularization term, and thus the regularization is embedded in the Bayesian beamformer.

3.5 Scalar Adaptive Beamformer with Unknown Source Orientation

3.5.1 Expressions for the Unit-Gain Constraint Beamformer

So far, we have derived the weight of the adaptive beamformer by assuming that the source orientation is predetermined. The source orientation may be predetermined using an accurate three-dimensional anatomical image of the subject, if available. However, in general, the source orientation $\boldsymbol{\eta}(\boldsymbol{r})$ is an unknown quantity, and should be estimated from the data. There are two types of beamformers that can handle the estimation of the source vector. One is the scalar-type and the other is the vector-type adaptive beamformer. In the following, we first describe the scalar adaptive beamformer [5].

In the scalar-type adaptive beamformer, the weight is first formulated using the unknown source orientation $\boldsymbol{\eta}$, such that [5]

$$\boldsymbol{w}(\boldsymbol{r},\boldsymbol{\eta}) = \frac{\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}}{[\boldsymbol{\eta}^T\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}]}, \tag{3.33}$$

and the output power obtained using this weight is given by

$$\langle \widehat{s}(\boldsymbol{r},\boldsymbol{\eta})^2\rangle = \frac{1}{\boldsymbol{\eta}^T[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}}. \tag{3.34}$$

The source orientation is then determined by maximizing this output power. That is, the optimum orientation $\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})$ is derived as [5]

$$\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r}) = \underset{\boldsymbol{\eta}(\boldsymbol{r})}{\rm argmax}\left[\frac{1}{\boldsymbol{\eta}^T(\boldsymbol{r})\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}(\boldsymbol{r})}\right]. \tag{3.35}$$

According to the Rayleigh–Ritz formula in Sect. C.9, the orientation $\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})$ is obtained as

$$\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r}) = \underset{\boldsymbol{\eta}(\boldsymbol{r})}{\rm argmin}\left[\boldsymbol{\eta}^T(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}(\boldsymbol{r})\right] = \vartheta_{\rm min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\}, \tag{3.36}$$

where $\vartheta_{\rm min}\{\cdot\}$ indicates the eigenvector corresponding to the minimum eigenvalue of the matrix in the curly braces.[3] Namely, the optimum orientation $\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})$ is given by the eigenvector corresponding to the minimum eigenvalue of $\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})$.

Once $\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})$ is obtained, the explicit form of the weight vector for the scalar minimum-variance beamformer is expressed as

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})}{[\boldsymbol{\eta}_{\rm opt}^T(\boldsymbol{r})\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})]}. \tag{3.37}$$

The output power is given by

$$\langle \widehat{s}(\boldsymbol{r},t)^2\rangle = \frac{1}{[\boldsymbol{\eta}_{\rm opt}^T(\boldsymbol{r})\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}_{\rm opt}(\boldsymbol{r})]} = \frac{1}{\mathcal{S}_{\rm min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\}}, \tag{3.38}$$

where $\mathcal{S}_{\rm min}\{\cdot\}$ is the minimum eigenvalue of the matrix in the curly braces.

3.5.2 *Expressions for the Array-Gain and Weight-Normalized Beamformers*

For the scalar-type array-gain minimum-variance beamformer, the optimum orientation is derived using

[3] The notations such as $\vartheta_{\rm min}\{\cdot\}$ and $\mathcal{S}_{\rm min}\{\cdot\}$ are defined in Sect. C.9 in the Appendix.

$$\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r}) = \underset{\boldsymbol{\eta}(\boldsymbol{r})}{\text{argmax}} \left[\frac{\boldsymbol{\eta}^T(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}(\boldsymbol{r})}{\boldsymbol{\eta}^T(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}(\boldsymbol{r})} \right], \tag{3.39}$$

which can be computed using

$$\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r}) = \vartheta_{\min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r}), \boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})\}. \tag{3.40}$$

Once $\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r})$ is obtained, the weight vector is obtained using Eq. (3.10) with $\boldsymbol{l}(\boldsymbol{r}) = \boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r})$ and $\widetilde{\boldsymbol{l}}(\boldsymbol{r}) = \boldsymbol{l}(\boldsymbol{r})/\|\boldsymbol{l}(\boldsymbol{r})\|$. The output power of this scalar beamformer is given by

$$\langle \widehat{s}(\boldsymbol{r}, t)^2 \rangle = \frac{1}{\mathcal{S}_{\min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r}), \boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})\}}. \tag{3.41}$$

For the scalar-type weight-normalized minimum-variance beamformer, the optimum orientation is derived using

$$\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r}) = \underset{\boldsymbol{\eta}(\boldsymbol{r})}{\text{argmax}} \left[\frac{\boldsymbol{\eta}^T(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}(\boldsymbol{r})}{\boldsymbol{\eta}^T(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{L}(\boldsymbol{r})]\boldsymbol{\eta}(\boldsymbol{r})} \right], \tag{3.42}$$

which can be computed using

$$\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r}) = \vartheta_{\min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{L}(\boldsymbol{r}), \boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\}. \tag{3.43}$$

Once $\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r})$ is obtained, the weight vector is obtained using Eq. (3.17) with $\boldsymbol{l}(\boldsymbol{r}) = \boldsymbol{L}(\boldsymbol{r})\boldsymbol{\eta}_{\text{opt}}(\boldsymbol{r})$. The output power of this scalar beamformer is given by

$$\langle \widehat{s}(\boldsymbol{r}, t)^2 \rangle = \frac{1}{\mathcal{S}_{\min}\{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-2}\boldsymbol{L}(\boldsymbol{r}), \boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})\}}. \tag{3.44}$$

3.6 Vector-Type Adaptive Beamformer

3.6.1 Vector Beamformer Formulation

The vector adaptive beamformer is another type of adaptive beamformers that can reconstruct the source orientation as well as the source magnitude. It uses a set of three weight vectors $\boldsymbol{w}_x(\boldsymbol{r})$, $\boldsymbol{w}_y(\boldsymbol{r})$, and $\boldsymbol{w}_z(\boldsymbol{r})$, which, respectively, detect the x, y, and z components of the source vector $\boldsymbol{s}(\boldsymbol{r}, t)$. That is, the weight matrix $\boldsymbol{W}(\boldsymbol{r})$ is defined as

$$\boldsymbol{W}(\boldsymbol{r}) = [\boldsymbol{w}_x(\boldsymbol{r}), \boldsymbol{w}_y(\boldsymbol{r}), \boldsymbol{w}_z(\boldsymbol{r})], \tag{3.45}$$

and the source vector can be estimated using

$$\widehat{\boldsymbol{s}}(\boldsymbol{r},t)=[\widehat{s}_x(\boldsymbol{r},t),\widehat{s}_y(\boldsymbol{r},t),\widehat{s}_z(\boldsymbol{r},t)]^T=\boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{y}(t). \tag{3.46}$$

To derive the weight matrix of a vector-type minimum-variance beamformer, we use the optimization

$$\boldsymbol{W}(\boldsymbol{r})=\underset{\boldsymbol{W}(\boldsymbol{r})}{\operatorname{argmin}}\ \operatorname{tr}[\boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{R}\boldsymbol{W}(\boldsymbol{r})],\ \text{subject to}\quad \boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})=\boldsymbol{I}. \tag{3.47}$$

A derivation similar to that for Eq. (3.6) leads to the solution for the weight matrix $\boldsymbol{W}(\boldsymbol{r})$, which is expressed as [4]

$$\boldsymbol{W}(\boldsymbol{r})=\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1}. \tag{3.48}$$

The beamformer that uses the above weight is called the vector-type adaptive beamformer. Using the weight matrix above, the source-vector covariance matrix is estimated as

$$\langle\widehat{\boldsymbol{s}}(\boldsymbol{r},t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r},t)\rangle=[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1}, \tag{3.49}$$

and the source power estimate is obtained using

$$\langle\|\widehat{\boldsymbol{s}}(\boldsymbol{r},t)\|^2\rangle=\langle\widehat{s}(\boldsymbol{r},t)^2\rangle=\operatorname{tr}\left[[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1}\right]. \tag{3.50}$$

This vector-type beamformer is sometimes referred to as the linearly constrained minimum-variance (LCMV) beamformer [4].

The weight matrix of the array-gain vector beamformer is derived, such that

$$\boldsymbol{W}(\boldsymbol{r})=\underset{\boldsymbol{W}(\boldsymbol{r})}{\operatorname{argmin}}\ \operatorname{tr}\left[\boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{R}\boldsymbol{W}(\boldsymbol{r})\right],\quad \text{subject to}\quad \boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{L}(\boldsymbol{r})=\|\boldsymbol{L}(\boldsymbol{r})\|. \tag{3.51}$$

The vector version of the array-gain minimum-variance beamformer is expressed as

$$\boldsymbol{W}(\boldsymbol{r})=\boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r})[\widetilde{\boldsymbol{L}}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r})]^{-1}, \tag{3.52}$$

where $\widetilde{\boldsymbol{L}}(\boldsymbol{r})$ is the normalized lead field matrix defined as $\widetilde{\boldsymbol{L}}(\boldsymbol{r})=\boldsymbol{L}(\boldsymbol{r})/\|\boldsymbol{L}(\boldsymbol{r})\|$. The source-vector covariance matrix is estimated as

$$\langle\widehat{\boldsymbol{s}}(\boldsymbol{r},t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r},t)\rangle=[\widetilde{\boldsymbol{L}}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r})]^{-1}. \tag{3.53}$$

The source power estimate is obtained by

$$\langle \|\widehat{\boldsymbol{s}}(\boldsymbol{r},t)\|^2 \rangle = \langle \widehat{s}(\boldsymbol{r},t)^2 \rangle = \mathrm{tr}\left[[\widetilde{\boldsymbol{L}}^T(\boldsymbol{r})\boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r})]^{-1} \right]. \tag{3.54}$$

The vector-type beamformer can be formulated with the unit-noise-gain constraint, and the detail of the formulation is found in [6, 12].

3.6.2 Semi-Bayesian Formulation

The weight matrix of a vector-type adaptive beamformer can be derived using the same Bayesian formulation in Sect. 3.3. The prior distribution in Eq. (3.20) and the noise assumption in Eq. (3.19) lead to the Bayesian estimate of $\boldsymbol{x}$ in Eq. (3.21), which is written as

$$\bar{\boldsymbol{x}}(t) = \boldsymbol{\Phi}^{-1}\boldsymbol{F}^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y}(t), \tag{3.55}$$

where, $\bar{\boldsymbol{x}}(t)$ is expressed as

$$\bar{\boldsymbol{x}}(t) = \begin{bmatrix} \widehat{\boldsymbol{s}}(\boldsymbol{r}_1,t) \\ \widehat{\boldsymbol{s}}(\boldsymbol{r}_2,t) \\ \vdots \\ \widehat{\boldsymbol{s}}(\boldsymbol{r}_N,t) \end{bmatrix}. \tag{3.56}$$

To derive the vector beamformer, we use the matrix $\boldsymbol{\Phi}^{-1}$ which is a $3N \times 3N$ Block diagonal matrix such that

$$\boldsymbol{\Phi}^{-1} = \begin{bmatrix} \boldsymbol{\Upsilon}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Upsilon}_2 & \cdot & \vdots \\ \vdots & \cdot & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\Upsilon}_N \end{bmatrix}, \tag{3.57}$$

where the 3×3 matrix $\boldsymbol{\Upsilon}_j$ is the prior covariance matrix of the source vector at the jth voxel. Thus, the estimated source vector for the jth voxel is obtained as

$$\widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t) = \boldsymbol{\Upsilon}_j\boldsymbol{L}_j^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y}(t). \tag{3.58}$$

Computing the estimated source-vector covariance matrix, $\langle \widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r}_j,t) \rangle$, leads to

$$\begin{aligned}\langle \widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r}_j,t)\rangle &= \boldsymbol{\Upsilon}_j \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \langle \boldsymbol{y}(t)\boldsymbol{y}^T(t)\rangle \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j \boldsymbol{\Upsilon}_j \\ &= \boldsymbol{\Upsilon}_j \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j \boldsymbol{\Upsilon}_j, \end{aligned} \tag{3.59}$$

where we assume $\langle \boldsymbol{y}(t)\boldsymbol{y}^T(t)\rangle = \boldsymbol{\Sigma}_y$.

In this vector case, the unit-gain constraint is expressed as

$$\langle \widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r}_j,t)\rangle = \boldsymbol{\Upsilon}_j. \tag{3.60}$$

Imposing this relationship, we get,

$$\boldsymbol{\Upsilon}_j = \boldsymbol{\Upsilon}_j \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j \boldsymbol{\Upsilon}_j,$$

and

$$\boldsymbol{\Upsilon}_j = \left[\boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j\right]^{-1}. \tag{3.61}$$

Substituting the above equation into Eq. (3.58), we obtain

$$\widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t) = \left[\boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j\right]^{-1} \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}(t). \tag{3.62}$$

Assuming that the model data covariance can be replaced by the sample data covariance $\boldsymbol{R}$, Eq. (3.62) is rewritten as

$$\widehat{\boldsymbol{s}}(\boldsymbol{r},t) = \boldsymbol{W}^T(\boldsymbol{r}_j)\boldsymbol{y}(t), \tag{3.63}$$

where the weight matrix is given by

$$\boldsymbol{W}(\boldsymbol{r}_j) = \boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r}_j)\left[\boldsymbol{L}^T(\boldsymbol{r}_j)\boldsymbol{R}^{-1}\boldsymbol{L}(\boldsymbol{r}_j)\right]^{-1}. \tag{3.64}$$

The weight matrix of the vector array-gain minimum-variance beamformer can be derived using the array-gain constraint,

$$\langle \widehat{\boldsymbol{s}}(\boldsymbol{r}_j,t)\widehat{\boldsymbol{s}}^T(\boldsymbol{r}_j,t)\rangle = \boldsymbol{\Upsilon}_j \|\boldsymbol{L}(\boldsymbol{r}_j)\|. \tag{3.65}$$

Substituting this relationship into Eq. (3.59) and using Eq. (3.58), the resultant weight matrix is expressed as

$$\boldsymbol{W}(\boldsymbol{r}_j) = \boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r}_j)\left[\widetilde{\boldsymbol{L}}^T(\boldsymbol{r}_j)\boldsymbol{R}^{-1}\widetilde{\boldsymbol{L}}(\boldsymbol{r}_j)\right]^{-1}. \tag{3.66}$$

3.7 Narrow-Band Beamformer

3.7.1 Background

Stimulus-induced power modulation of spontaneous brain activity has been the subject of intense investigations. Such power modulation is sometimes referred to as the event-related spectral power change. When the power change is negative, it is customarily termed as event-related desynchronization (ERD), and when it is positive, it is termed as event-related synchronization (ERS) [13].

The narrow-band dual-state beamformer [7] is a powerful tool for localization of specific brain activities related to these power changes. This is because the power change is frequency-specific, and the narrow-band beamformer uses a weight tuned to a specific target frequency. In this section, we describe the time-domain and the frequency-domain implementations of the narrow-band beamformer.

3.7.2 Time-Domain Implementation

Since the induced brain activity is not time-locked to the stimulus, the sample covariance matrix should be computed from nonaveraged raw trials. We assume that total N_E trials denoted as $\boldsymbol{b}_1(t), \ldots, \boldsymbol{b}_{N_E}$ are obtained. To compute a frequency-specific weight, these raw trials are band-pass filtered with a specific frequency band of interest. The band-pass filtered sensor data for the nth trial is denoted as $\widetilde{\boldsymbol{b}}_n(t, f)$, where f represents the frequency band of interest.[4]

The sample covariance matrix is computed using the band-pass filtered data, such that

$$\widehat{\boldsymbol{R}}(f) = \frac{1}{N_E} \sum_{n=1}^{N_E} \sum_{k=1}^{K} \widetilde{\boldsymbol{b}}_n(t_k, f) \widetilde{\boldsymbol{b}}_n^T(t_k, f). \tag{3.67}$$

Since we use the nonaveraged trial data, the data naturally contains a significant amount of influence from brain activities that are not related to the activity of interest.

To remove the influence of such unwanted brain activities, we use dual-state datasets: one from the target time period and the other from the baseline period. We try to reconstruct source activities that cause the power change in a specific frequency band. That is, using Eq. (3.67), we compute the frequency-specific covariance matrix for the target period, $\widehat{\boldsymbol{R}}_T(f)$, and for the baseline period, $\widehat{\boldsymbol{R}}_C(f)$.

We then reconstruct frequency-specific power changes between the target and baseline periods by using the F-ratio method. That is, using $\widehat{\boldsymbol{R}}_T(f)$ and $\widehat{\boldsymbol{R}}_C(f)$, the F-ratio method computes the frequency-specific filter weight such that

[4] The notation f may indicate the center frequency of the frequency band of interest.

$$\boldsymbol{w}(\boldsymbol{r}, f) = \frac{\widehat{\boldsymbol{R}}_{\text{total}}^{-1}(f)\boldsymbol{l}(\boldsymbol{r})}{\boldsymbol{l}^T(\boldsymbol{r})\widehat{\boldsymbol{R}}_{\text{total}}^{-1}(f)\boldsymbol{l}(\boldsymbol{r})}, \tag{3.68}$$

where $\widehat{\boldsymbol{R}}_{\text{total}}(f) = \widehat{\boldsymbol{R}}_T(f) + \widehat{\boldsymbol{R}}_C(f)$. Then, the F-ratio image, $F(\boldsymbol{r}, f)$, is obtained such that

$$F(\boldsymbol{r}, f) = \frac{\boldsymbol{w}^T(\boldsymbol{r}, f)\widehat{\boldsymbol{R}}_T(f)\boldsymbol{w}(\boldsymbol{r}, f) - \boldsymbol{w}^T(\boldsymbol{r}, f)\widehat{\boldsymbol{R}}_C(f)\boldsymbol{w}(\boldsymbol{r}, f)}{\boldsymbol{w}^T(\boldsymbol{r}, f)\widehat{\boldsymbol{R}}_C(f)\boldsymbol{w}(\boldsymbol{r}, f)}. \tag{3.69}$$

On the right-hand side of Eq. (3.69), the first term in the numerator represents the reconstruction of the source power in the target period and the second represents that in the control period. Thus, $F(\boldsymbol{r}, f)$ represents the ratio of the reconstructed source power change to the power of baseline activities.

3.7.3 Frequency-Domain Implementation

The narrow-band beamformer can also be implemented in the frequency domain. We first define the Fourier transform of the raw-trial vector $\boldsymbol{b}_n(t)$ as $\boldsymbol{g}_n(f)$. The sample cross-spectrum matrix $\widetilde{\boldsymbol{R}}(f)$ is computed using $\boldsymbol{g}_n(f)$, such that

$$\widetilde{\boldsymbol{R}}(f) = \frac{1}{N_E}\sum_{n=1}^{N_E}\boldsymbol{g}_n(f)\boldsymbol{g}_n(f)^H, \tag{3.70}$$

where the superscript H indicates the Hermitian transpose (complex conjugation plus matrix transpose). Using Eq. (3.70), we compute the frequency-specific cross-spectral matrix for the target period, $\widetilde{\boldsymbol{R}}_T(f)$, and for the control period, $\widetilde{\boldsymbol{R}}_C(f)$. The frequency-selective weight $\boldsymbol{w}(\boldsymbol{r}, f)$ is obtained such that

$$\boldsymbol{w}(\boldsymbol{r}, f) = \frac{\widetilde{\boldsymbol{R}}_{\text{total}}^{-1}(f)\boldsymbol{l}(\boldsymbol{r})}{\boldsymbol{l}^T(\boldsymbol{r})\widetilde{\boldsymbol{R}}_{\text{total}}^{-1}(f)\boldsymbol{l}(\boldsymbol{r})}, \tag{3.71}$$

where $\widetilde{\boldsymbol{R}}_{\text{total}}(f) = \widetilde{\boldsymbol{R}}_T(f) + \widetilde{\boldsymbol{R}}_C(f)$. The pseudo F-ratio image is computed using

$$F(\boldsymbol{r}, f) = \frac{\boldsymbol{w}^H(\boldsymbol{r}, f)\widetilde{\boldsymbol{R}}_T(f)\boldsymbol{w}(\boldsymbol{r}, f) - \boldsymbol{w}^H(\boldsymbol{r}, f)\widetilde{\boldsymbol{R}}_C(f)\boldsymbol{w}(\boldsymbol{r}, f)}{\boldsymbol{w}^H(\boldsymbol{r}, f)\widetilde{\boldsymbol{R}}_C(f)\boldsymbol{w}(\boldsymbol{r}, f)}. \tag{3.72}$$

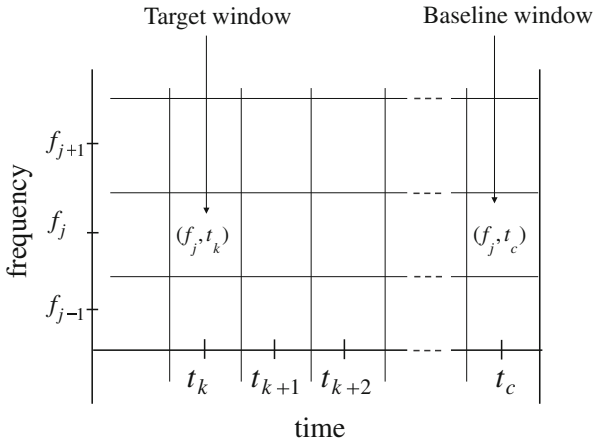

Fig. 3.1 Depiction of time–frequency domain discretization. The target window is set at the one represented by (f_j, t_k), and the baseline window is set at the one represented by (f_j, t_c)

3.7.4 Five-Dimensional Brain Imaging

Using the narrow-band dual-state beamformer, we can implement the five-dimensional (time–frequency–space) imaging of brain activities [7]. In this implementation, we use the sliding window method in which the target window is moved along the time and frequency directions. The discretization of the time–frequency domain is depicted in Fig. 3.1.

We assign the window denoted (f_j, t_k) to the target time–frequency window, and the window denoted (f_j, t_c) to the baseline window. By implementing the narrow-band dual-state beamformer using these two time–frequency windows, we obtain the pseudo F-ratio image, $F(\boldsymbol{r}, f_j, t_k)$, which represents the source power difference at the frequency f_j between the time windows at t_k and t_c. If we move the target window along the time and frequency directions, we can obtain the five-dimensional source power difference map of the induced brain activity, $F(\boldsymbol{r}, f_j, t_k)$, where $j = 1, \ldots, N_f$ and $k = 1, \ldots, K$, where N_f is the number of frequency bins and K is the number of time windows.

3.8 Nonadaptive Spatial Filters

3.8.1 Minimum-Norm Filter

There is a different class of beamformers that uses the nonadaptive weight, the weight computed only using the sensor lead field. These are customarily called as

the nonadaptive spatial filters.[5] A representative and basic nonadaptive spatial filter is the minimum-norm filter, which is the spatial filter version of the minimum-norm method described in Chap. 2.

The minimum-norm source reconstruction method is formulated as a nonadaptive spatial filter in the following manner. In Eq. (2.26), $\boldsymbol{F}\boldsymbol{F}^T$ is expressed such that [14]

$$\boldsymbol{F}\boldsymbol{F}^T = \sum_{n=1}^{N} \boldsymbol{L}(\boldsymbol{r}_n)\boldsymbol{L}^T(\boldsymbol{r}_n) \approx \int_{\Omega} \boldsymbol{L}(\boldsymbol{r})\boldsymbol{L}^T(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \boldsymbol{G}, \tag{3.73}$$

where Ω indicates the source space. That is, $\boldsymbol{F}\boldsymbol{F}^T$ is equal to the gram matrix $\boldsymbol{G}$, if we ignore the voxel discretization error. Thus, let us rewrite Eq. (2.26) as

$$\begin{bmatrix} \widehat{s}(\boldsymbol{r}_1, t) \\ \widehat{s}(\boldsymbol{r}_2, t) \\ \vdots \\ \widehat{s}(\boldsymbol{r}_N, t) \end{bmatrix} = \boldsymbol{F}^T\boldsymbol{G}^{-1}\boldsymbol{y}(t) = \begin{bmatrix} \boldsymbol{L}^T(\boldsymbol{r}_1) \\ \boldsymbol{L}^T(\boldsymbol{r}_2) \\ \vdots \\ \boldsymbol{L}^T(\boldsymbol{r}_N) \end{bmatrix} \boldsymbol{G}^{-1}\boldsymbol{y}(t). \tag{3.74}$$

We can see that, at each voxel location $\boldsymbol{r}_n$, the relationship,

$$\widehat{s}(\boldsymbol{r}_n, t) = \boldsymbol{L}^T(\boldsymbol{r}_n)\boldsymbol{G}^{-1}\boldsymbol{y}(t), \tag{3.75}$$

holds.

The equation above has the same form as the vector-type beamformer in Eq. (3.46), i.e., Eq. (3.75) is rewritten as

$$\widehat{s}(\boldsymbol{r}, t) = \boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{y}(t), \tag{3.76}$$

where the weight matrix $\boldsymbol{W}(\boldsymbol{r})$ is given by:

$$\boldsymbol{W}(\boldsymbol{r}) = \boldsymbol{G}^{-1}\boldsymbol{L}(\boldsymbol{r}). \tag{3.77}$$

The two equations above indicate that the minimum-norm method is formulated as a nonadaptive spatial filter, in which the weight matrix is given by Eq. (3.77). Also, the L_2-regularized version of the weight matrix is given by:

$$\boldsymbol{W}(\boldsymbol{r}) = (\boldsymbol{G} + \xi\boldsymbol{I})^{-1}\boldsymbol{L}(\boldsymbol{r}), \tag{3.78}$$

where a scalar ξ is the regularization constant. The spatial filter whose weight is expressed in either Eq. (3.77) or (3.78) is called the minimum-norm filter.

[5] They might be also called as the nonadaptive beamformers, but this usage is uncommon.

3.8.2 Weight-Normalized Minimum-Norm Filter

The weight-normalized minimum-norm filter has been proposed by Dale et al. and the method is often called as dynamic statistical parametric mapping (dSPM) [15]. The idea is to normalize the minimum-norm weight with its weight's norm to ensure that the spatial distribution of the noise is uniform. The scalar-type weight is thus given by

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})}{\|\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})\|} = \frac{\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})}{\sqrt{\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{G}^{-2}\boldsymbol{l}(\boldsymbol{r})}}. \tag{3.79}$$

The idea of weight normalization can be extended to derive the weight matrix for the vector-type spatial filter, such that

$$\boldsymbol{W}(\boldsymbol{r}) = \frac{\boldsymbol{G}^{-1}\boldsymbol{L}(\boldsymbol{r})}{\sqrt{\mathrm{tr}\left[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-2}\boldsymbol{L}(\boldsymbol{r})\right]}}. \tag{3.80}$$

Using this weight matrix, the source vector is estimated as

$$\widehat{\boldsymbol{s}}(\boldsymbol{r}, t) = \boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{y}(t) = \frac{\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{y}(t)}{\sqrt{\mathrm{tr}\left[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-2}\boldsymbol{L}(\boldsymbol{r})\right]}}. \tag{3.81}$$

The weight matrix for the L_2-regularized version is given by

$$\boldsymbol{W}(\boldsymbol{r}) = \frac{(\boldsymbol{G} + \xi\boldsymbol{I})^{-1}\boldsymbol{L}(\boldsymbol{r})}{\sqrt{\mathrm{tr}\left[\boldsymbol{L}^T(\boldsymbol{r})(\boldsymbol{G} + \xi\boldsymbol{I})^{-2}\boldsymbol{L}(\boldsymbol{r})\right]}}. \tag{3.82}$$

3.8.3 sLORETA Filter

Standardized low resolution electromagnetic tomography (sLORETA) was originally proposed by Pascual-Marqui [16]. The method can be reformulated as a nonadaptive spatial filter. In this method, the minimum-norm filter outputs are normalized by the quantity $\sqrt{\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})}$. This normalization is called as the standardization. The scalar-type weight is given by

$$\boldsymbol{w}(\boldsymbol{r}) = \frac{\boldsymbol{l}\boldsymbol{G}^{-1}}{\sqrt{\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})}}. \tag{3.83}$$

The extension to the vector-type sLORETA filter results in the weight matrix expressed as

$$\boldsymbol{W}(\boldsymbol{r}) = \boldsymbol{G}^{-1}\boldsymbol{L}(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1/2}, \tag{3.84}$$

and the source vector $\widehat{\boldsymbol{s}}(\boldsymbol{r}, t)$ is estimated as

$$\widehat{\boldsymbol{s}}(\boldsymbol{r}, t) = \boldsymbol{W}^T(\boldsymbol{r})\boldsymbol{y}(t) = [\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1/2}\boldsymbol{L}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{y}(t). \tag{3.85}$$

The weight matrix for the L_2-regularized version is given by,

$$\boldsymbol{W}(\boldsymbol{r}) = (\boldsymbol{G} + \xi\boldsymbol{I})^{-1}\boldsymbol{L}(\boldsymbol{r})[\boldsymbol{L}^T(\boldsymbol{r})(\boldsymbol{G} + \xi\boldsymbol{I})^{-1}\boldsymbol{L}(\boldsymbol{r})]^{-1/2}. \tag{3.86}$$

Although the standardization makes the sLORETA filter to have no localization bias [6], the theoretical basis of this standardization is not entirely clear.

3.9 Recursive Null-Steering (RENS) Beamformer

3.9.1 Beamformer Obtained Based on Beam-Response Optimization

In the beamformer formulation, the product, $\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}')$, is called the beam response, which expresses the sensitivity of the beamformer pointing at $\boldsymbol{r}$ to a source located at $\boldsymbol{r}'$. We wish to derive a beamformer that only passes the signal from a source at the pointing location and suppresses the leakage from sources at other locations. Such a beamformer may be derived by imposing a delta-function-like property on the beam response. That is, the weight vector is obtained using

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}} \int_{\Omega} \left[\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}') - \delta(\boldsymbol{r} - \boldsymbol{r}')\right]^2 \mathrm{d}\boldsymbol{r}'. \tag{3.87}$$

The weight obtained from the optimization above is expressed as

$$\boldsymbol{w}(\boldsymbol{r}) = \boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r}). \tag{3.88}$$

The beamformer in Eq. (3.88) is exactly equal to the minimum-norm filter.

Variants of the minimum-norm filter can be obtained by adding various constraints to the optimization in Eq. (3.87) [17]. For example, the optimization

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}} \int_{\Omega} \left[\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}') - \delta(\boldsymbol{r} - \boldsymbol{r}')\right]^2 \mathrm{d}\boldsymbol{r}' \quad \text{subject to} \quad \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = 1, \tag{3.89}$$

leads to the weight expression

$$\boldsymbol{w}(\boldsymbol{r}) = [\boldsymbol{l}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r})]^{-1}\boldsymbol{G}^{-1}\boldsymbol{l}(\boldsymbol{r}). \tag{3.90}$$

The beamformer using the above weight is called the unit-gain (constraint) minimum-norm filter. Using the optimization

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}} \int_{\Omega} \left[\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}') - \delta(\boldsymbol{r}-\boldsymbol{r}')\right]^2 \mathrm{d}\boldsymbol{r}'$$
$$\text{subject to} \quad \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|, \tag{3.91}$$

we obtain the array-gain (constraint) minimum-norm filter, such that

$$\boldsymbol{w}(\boldsymbol{r}) = [\widetilde{\boldsymbol{l}}^T(\boldsymbol{r})\boldsymbol{G}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r})]^{-1}\boldsymbol{G}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r}), \tag{3.92}$$

where $\widetilde{\boldsymbol{l}}(\boldsymbol{r}) = \boldsymbol{l}(\boldsymbol{r})/\|\boldsymbol{l}(\boldsymbol{r})\|$.

3.9.2 Derivation of RENS Beamformer

The recursive null-steering (RENS) beamformer is derived from the following optimization [18]:

$$\boldsymbol{w}(\boldsymbol{r}) = \underset{\boldsymbol{w}(\boldsymbol{r})}{\operatorname{argmin}} \int_{\Omega} \left[\boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}') - \delta(\boldsymbol{r}-\boldsymbol{r}')\right]^2 s(\boldsymbol{r},t)^2 \mathrm{d}\boldsymbol{r}'$$
$$\text{subject to} \quad \boldsymbol{w}^T(\boldsymbol{r})\boldsymbol{l}(\boldsymbol{r}) = \|\boldsymbol{l}(\boldsymbol{r})\|, \tag{3.93}$$

where $s(\boldsymbol{r},t)^2$ is the instantaneous source power. The idea behind the above optimization is that we wish to impose a delta-function-like property on the beam response only over a region where sources exist, instead of imposing that property over the entire source space.

The filter weight is obtained as

$$\boldsymbol{w}(\boldsymbol{r}) = [\widetilde{\boldsymbol{l}}^T(\boldsymbol{r})\bar{\boldsymbol{G}}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r})]^{-1}\bar{\boldsymbol{G}}^{-1}\widetilde{\boldsymbol{l}}(\boldsymbol{r}), \tag{3.94}$$

where

$$\bar{\boldsymbol{G}} = \int_{\Omega} s(\boldsymbol{r},t)^2\boldsymbol{l}(\boldsymbol{r})\boldsymbol{l}^T(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}. \tag{3.95}$$

However, to compute the weight in Eq. (3.94), we need to know the source magnitude distribution $s(\boldsymbol{r},t)^2$, which is unknown. Therefore, we use an estimated source magnitude $\widehat{s}(\boldsymbol{r},t)^2$ when computing $\bar{\boldsymbol{G}}$, i.e.,

$$\bar{\boldsymbol{G}} = \int_{\Omega} \widehat{s}(\boldsymbol{r},t)^2\boldsymbol{l}(\boldsymbol{r})\boldsymbol{l}^T(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}. \tag{3.96}$$

The proposed weight is therefore derived in a recursive manner. That is, by setting an initial $\widehat{s}(\boldsymbol{r}, t)^2$ to have a uniform value, the weight $\boldsymbol{w}(\boldsymbol{r})$ is derived using Eqs. (3.94) and (3.96). The estimated source intensity $\widehat{s}(\boldsymbol{r}, t)$ is obtained using Eq. (3.1). This $\widehat{s}(\boldsymbol{r}, t)$ is then used in Eqs. (3.94) and (3.96) to update $\boldsymbol{w}(\boldsymbol{r})$. These procedures are repeated until some stopping criterion is satisfied. Here we describe a derivation of the scalar-type RENS beamformer. An extension for deriving the vector-type RENS beamformer is straightforward [18].

The RENS beamformer described here provides spatial resolution higher than that of the nonadaptive spatial filters described in Sect. 3.8. Moreover, it is free from the limitations of adaptive beamformers. That is, the RENS beamformer is robust to the source correlation problem. It does not require large time samples and works even with single time point data.

References

1. J. Capon, High-resolution frequency wavenumber spectrum analysis. Proc. IEEE **57**, 1408–1419 (1969)
2. S.E. Robinson, D.F. Rose, Current source image estimation by spatially filtered MEG, in *Biomagnetism Clinical Aspects*, ed. by M. Hoke, et al. (Elsevier Science Publishers, New York, 1992), pp. 761–765
3. M.E. Spencer, R.M. Leahy, J.C. Mosher, P.S. Lewis, Adaptive filters for monitoring localized brain activity from surface potential time series, in *Conference Record for 26th Annual Asilomer Conference on Signals, Systems, and Computers*, pp. 156–161, November 1992
4. B.D. Van Veen, W. Van Drongelen, M. Yuchtman, A. Suzuki, Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. IEEE Trans. Biomed. Eng. **44**, 867–880 (1997)
5. K. Sekihara, B. Scholz, Generalized Wiener estimation of three-dimensional current distribution from biomagnetic measurements, in *Biomag 96: Proceedings of the Tenth International Conference on Biomagnetism*, ed. by C.J. Aine et al. (Springer, New York, 1996), pp. 338–341
6. K. Sekihara, S.S. Nagarajan, *Adaptive Spatial Filters for Electromagnetic Brain Imaging* (Springer, Berlin, 2008)
7. S.S. Dalal, A.G. Guggisberg, E. Edwards, K. Sekihara, A.M. Findlay, R.T. Canolty, M.S. Berger, R.T. Knight, N.M. Barbaro, H.E. Kirsch, S.S. Nagarajan, Five-dimensional neuroimaging: localization of the time-frequency dynamics of cortical activity. NeuroImage **40**, 1686–1700 (2008)
8. G. Borgiotti, L.J. Kaplan, Superresolution of uncorrelated interference sources by using adaptive array technique. IEEE Trans. Antennas Propag. **27**, 842–845 (1979)
9. K. Sekihara, B. Scholz, Generalized Wiener estimation of three-dimensional current distribution from biomagnetic measurements. IEEE Trans. Biomed. Eng. **43**, 281–291 (1996)
10. H. Cox, R.M. Zeskind, M.M. Owen, Robust adaptive beamforming. IEEE Trans. Signal Process. **35**, 1365–1376 (1987)
11. M. Woolrich, L. Hunt, A. Groves, G. Barnes, MEG beamforming using Bayesian PCA for adaptive data covariance matrix regularization. Neuroimage **57**(4), 1466–1479 (2011)
12. K. Sekihara, S.S. Nagarajan, D. Poeppel, A. Marantz, Y. Miyashita, Reconstructing spatio-temporal activities of neural sources using an MEG vector beamformer technique. IEEE Trans. Biomed. Eng. **48**, 760–771 (2001)
13. G. Pfurtscheller, F.H. Lopes da Silva, Event-related EEG/MEG synchronization and desynchronization: basic principles. Clin. Neurophysiol. **110**, 1842–1857 (1999)

14. K. Sekihara, M. Sahani, S.S. Nagarajan, Location bias and spatial resolution of adaptive and non-adaptive spatial filters for MEG source reconstruction. NeuroImage **25**, 1056–1067 (2005)
15. A.M. Dale, A.K. Liu, B.R. Fischl, R.L. Buckner, J.W. Belliveau, J.D. Lewine, E. Halgren, Dynamic statistical parametric mapping: combining fMRI and MEG for high-resolution imaging of cortical activity. Neuron **26**, 55–67 (2000)
16. R.D. Pascual-Marqui, Standardized low resolution brain electromagnetic tomography (sLORETA): technical details. Methods. Find. Exp. Clin. Pharmacol. **24**, 5–12 (2002)
17. R.E. Greenblatt, A. Ossadtchi, M.E. Pflieger, Local linear estimators for the bioelectromagnetic inverse problem. IEEE Trans. Signal Process. **53**, 3403–3412 (2005)
18. I. Kumihashi, K. Sekihara, Array-gain constraint minimum-norm spatial filter with recursively updated gram matrix for biomagnetic source imaging. IEEE Trans. Biomed. Eng. **57**(6), 1358–1365 (2010)

Chapter 4
Sparse Bayesian (Champagne) Algorithm

4.1 Introduction

In this chapter, we provide a detailed description of an algorithm for electromagnetic brain imaging, called the Champagne algorithm [1, 2]. The Champagne algorithm is formulated based on an empirical Bayesian schema, and can provide a sparse solution, since the sparsity constraint is embedded in the algorithm. The algorithm is free from the problems that cannot be avoided in other sparse-solution methods, such as the L_1-regularized minimum-norm method. Such problems include the difficulty in reconstructing voxel time courses or the difficulty in incorporating the source-orientation estimation.

In Sect. 2.10.2, we show that the L_2-regularized minimum-norm method is derived using the Gaussian prior for the jth voxel value,[1]

$$x_j \sim \mathcal{N}(x_j|0, \alpha^{-1}), \tag{4.1}$$

where the precision α is common to all x_j. In this chapter, we use the Gaussian prior whose precision (variance) is specific to each x_j, i.e.,

$$x_j \sim \mathcal{N}(x_j|0, \alpha_j^{-1}). \tag{4.2}$$

We show that this "slightly different" prior distribution gives a solution totally different from the L_2-norm solution. Actually, the prior distribution in Eq. (4.2) leads to a sparse solution. The estimation method based on the prior in Eq. (4.2) is called the sparse Bayesian learning in the field of machine learning [3, 4], and the source reconstruction algorithm derived using Eq. (4.2) is called the Champagne algorithm [1].

In this chapter, we formulate the source reconstruction problem as the spatiotemporal reconstruction, i.e., the voxel time series $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K$ is reconstructed using the sensor time series $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$ where $\boldsymbol{y}(t_k)$ and $\boldsymbol{x}(t_k)$ are denoted $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$. We use collective expressions $\boldsymbol{x}$ and $\boldsymbol{y}$, indicating the whole voxel time series

[1] We use the notational convenience $\mathcal{N}$(variable|mean, covariance matrix) throughout this book.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_4

$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K$ and the whole sensor time series $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$. We first formulate the Champagne algorithm omitting the source orientation estimation. That is, we assume that the source orientation is predetermined at each voxel and use the measurement model in Eq. (2.15). The relationship between $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ is expressed as

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \varepsilon, \tag{4.3}$$

where the lead field matrix $\boldsymbol{H}$ is defined in Eq. (2.11).

4.2 Probabilistic Model and Method Formulation

Defining a column vector $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, the prior distribution is expressed as

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{\alpha}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}), \tag{4.4}$$

where $\boldsymbol{\Phi}$ is equal to $\boldsymbol{\Phi} = \mathrm{diag}(\boldsymbol{\alpha})$, which indicates a diagonal matrix whose diagonal elements are those of a vector in the parenthesis. Thus, we have

$$p(\boldsymbol{x}_k|\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}) = \prod_{j=1}^{N} \mathcal{N}(x_j|0, \alpha_j^{-1}). \tag{4.5}$$

Since we assume that the noise ε is independent across time, the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ is expressed as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{H}\boldsymbol{x}_k, \beta^{-1}\boldsymbol{I}). \tag{4.6}$$

In this chapter, the noise precision matrix is assumed to be $\beta\boldsymbol{I}$ where β is known. Therefore, the parameters we must estimate is the voxel source distribution $\boldsymbol{x}$ and the hyperparameter $\boldsymbol{\alpha}$.

In truly Bayesian formulation, we have to derive the joint posterior distribution of all the unknown parameters, $p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y})$, using

$$p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha})p(\boldsymbol{x}, \boldsymbol{\alpha})}{p(\boldsymbol{y})}. \tag{4.7}$$

However, we cannot compute[2]

$$p(\boldsymbol{y}) = \iint p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha})p(\boldsymbol{x}, \boldsymbol{\alpha})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\alpha}, \tag{4.8}$$

[2] In Eq. (4.8), the notation $\mathrm{d}\boldsymbol{x}$ indicates $\mathrm{d}\boldsymbol{x}_1\mathrm{d}\boldsymbol{x}_2 \ldots \mathrm{d}\boldsymbol{x}_K$.

because the integral on the right-hand side does not have a closed-form solution. Accordingly, the explicit form of $p(\boldsymbol{y})$, and thus the joint distribution $p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y})$ cannot be obtained.

Therefore, instead, using

$$p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})p(\boldsymbol{\alpha}|\boldsymbol{y}), \tag{4.9}$$

we can derive the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$, such that

$$p(\boldsymbol{x}|\boldsymbol{y}) = \int p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y})\mathrm{d}\boldsymbol{\alpha} = \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})p(\boldsymbol{\alpha}|\boldsymbol{y})\mathrm{d}\boldsymbol{\alpha}, \tag{4.10}$$

where we eliminate the unknown hyperparameter $\boldsymbol{\alpha}$ by marginalization. Denoting $\widehat{\boldsymbol{\alpha}}$ such that $\widehat{\boldsymbol{\alpha}} = \mathrm{argmax}\, p(\boldsymbol{\alpha}|\boldsymbol{y})$, and assuming that $p(\boldsymbol{\alpha}|\boldsymbol{y})$ has a (hopefully sharp) peak at $\widehat{\boldsymbol{\alpha}}$, we can use the approximation

$$p(\boldsymbol{x}|\boldsymbol{y}) = \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})p(\boldsymbol{\alpha}|\boldsymbol{y})\mathrm{d}\boldsymbol{\alpha} \approx \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})\delta(\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}})\mathrm{d}\boldsymbol{\alpha} = p(\boldsymbol{x}|\boldsymbol{y}, \widehat{\boldsymbol{\alpha}}), \tag{4.11}$$

where $\delta(\boldsymbol{\alpha})$ indicates the delta function. The above equation shows that $p(\boldsymbol{x}|\boldsymbol{y}, \widehat{\boldsymbol{\alpha}})$, (the posterior distribution obtained with setting $\boldsymbol{\alpha}$ to $\widehat{\boldsymbol{\alpha}}$), approximates the true posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$.

Using the arguments in Sect. B.3 in the Appendix, when $p(\boldsymbol{x}|\boldsymbol{\alpha})$ and $p(\boldsymbol{y}|\boldsymbol{x})$ are expressed in Eqs. (4.4) and (4.6), the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})$ is given by

$$p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}) = \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{\alpha}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{x}}_k, \boldsymbol{\Gamma}^{-1}), \tag{4.12}$$

where the precision and the mean are obtained as

$$\boldsymbol{\Gamma} = \boldsymbol{\Phi} + \beta \boldsymbol{H}^T \boldsymbol{H}, \tag{4.13}$$

$$\bar{\boldsymbol{x}}_k = \beta \boldsymbol{\Gamma}^{-1} \boldsymbol{H}^T \boldsymbol{y}_k. \tag{4.14}$$

Therefore, once $\widehat{\boldsymbol{\alpha}}$ is obtained, we can compute $\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{x}}_k$ by substituting $\boldsymbol{\Phi} = \mathrm{diag}(\widehat{\boldsymbol{\alpha}})$ into Eqs. (4.13) and (4.14), and we obtain $p(\boldsymbol{x}|\boldsymbol{y}, \widehat{\boldsymbol{\alpha}})$ using Eq. (4.12).

The problem here is how to estimate $\widehat{\boldsymbol{\alpha}}$. The posterior distribution of the hyperparameter $\boldsymbol{\alpha}$, $p(\boldsymbol{\alpha}|\boldsymbol{y})$, is expressed using the Bayes' rule,

$$p(\boldsymbol{\alpha}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}). \tag{4.15}$$

When we assume the flat (noninformative) prior for $p(\boldsymbol{\alpha})$, we have

$$p(\boldsymbol{\alpha}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}). \tag{4.16}$$

Thus, since the relationship,

$$\underset{\boldsymbol{\alpha}}{\mathrm{argmax}}\, p(\boldsymbol{\alpha}|\boldsymbol{y}) = \underset{\boldsymbol{\alpha}}{\mathrm{argmax}}\, p(\boldsymbol{y}|\boldsymbol{\alpha})$$

holds, $\widehat{\boldsymbol{\alpha}}$ is obtained as the one that maximizes $p(\boldsymbol{y}|\boldsymbol{\alpha})$. This $p(\boldsymbol{y}|\boldsymbol{\alpha})$ is referred to as the data evidence or the marginal likelihood.

Let us summarize the procedure to estimate the source distribution $\boldsymbol{x}$. First, we estimate the hyperparameter $\boldsymbol{\alpha}$ by maximizing the marginal likelihood function,

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{argmax}}\, p(\boldsymbol{y}|\boldsymbol{\alpha}).$$

Next, this $\widehat{\boldsymbol{\alpha}}$ is substituted into the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})$ to obtain $p(\boldsymbol{x}|\boldsymbol{y}, \widehat{\boldsymbol{\alpha}})$. When this posterior is the Gaussian distribution in Eq. (4.12), the precision and mean are obtained by substituting $\boldsymbol{\Phi} = \text{diag}(\widehat{\boldsymbol{\alpha}})$ into Eqs. (4.13) and (4.14). The voxel time courses are reconstructed by computing $\bar{\boldsymbol{x}}_k$ in Eq. (4.14) for $k = 1, \ldots, K$.

4.3 Cost Function for Marginal Likelihood Maximization

As described in the preceding section, the hyperparameter $\boldsymbol{\alpha}$ is estimated by maximizing the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{\alpha})$. In this section, we describe the maximization of the marginal likelihood, and to do so, let us derive an explicit form of the log marginal likelihood, $\log p(\boldsymbol{y}|\boldsymbol{\alpha})$. Substituting[3]

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{\alpha}) = \left[\frac{|\boldsymbol{\Phi}|^{1/2}}{(2\pi)^{N/2}}\right]^K \exp\left[-\frac{1}{2}\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{\Phi} \boldsymbol{x}_k\right], \tag{4.17}$$

and

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \left[\left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}}\right]^K \exp\left[-\frac{\beta}{2}\sum_{k=1}^{K} \|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2\right], \tag{4.18}$$

into

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \int p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{x} = \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{x},$$

we obtain

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \left[\frac{|\boldsymbol{\Phi}|^{1/2}}{(2\pi)^{N/2}}\left(\frac{\beta}{2\pi}\right)^{M/2}\right]^K \int \exp\left[-D\right]\mathrm{d}\boldsymbol{x}, \tag{4.19}$$

where

$$D = \frac{\beta}{2}\sum_{k=1}^{K} \|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2 + \frac{1}{2}\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{\Phi} \boldsymbol{x}_k, \tag{4.20}$$

[3] Note that M and N are respectively the sizes of $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$.

and this D can be rewritten as

$$D = \frac{1}{2}\sum_{k=1}^{K} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)^T \boldsymbol{\Gamma} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) + \Delta. \tag{4.21}$$

The derivation of the above equation is presented in Sect. 4.10.1, and Δ on the right-hand side of Eq. (4.21) is given in Eq. (4.97).

Using the results in Eq. (4.21), let us compute the integral on the right-hand side of Eq. (4.19). First, we have

$$\int \exp[-D]\,\mathrm{d}\boldsymbol{x} = \exp[-\Delta] \int \exp\left[-\frac{1}{2}\sum_{k=1}^{K} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)^T \boldsymbol{\Gamma} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)\right] \mathrm{d}\boldsymbol{x}. \tag{4.22}$$

Considering the Gaussian with its mean $\bar{\boldsymbol{x}}_k$ and precision $\boldsymbol{\Gamma}$, the integral on the right-hand side is computed such that,

$$\int \exp\left[-\frac{1}{2} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)^T \boldsymbol{\Gamma} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)\right] \mathrm{d}\boldsymbol{x}_k = \frac{(2\pi)^{N/2}}{|\boldsymbol{\Gamma}|^{1/2}}.$$

Substituting the above results into Eqs. (4.22) and (4.19), we have

$$p(\boldsymbol{y}|\boldsymbol{\alpha}) = \left[\frac{|\boldsymbol{\Phi}|^{1/2}}{(2\pi)^{N/2}} \left(\frac{\beta}{2\pi}\right)^{M/2}\right]^K \exp[-\Delta] \left[\frac{(2\pi)^{N/2}}{|\boldsymbol{\Gamma}|^{1/2}}\right]^K. \tag{4.23}$$

Taking the logarithm of both sides, and omitting constant terms, which are unrelated to the arguments, we get

$$\log p(\boldsymbol{y}|\boldsymbol{\alpha}) = K\left(-\frac{1}{2}\log|\boldsymbol{\Gamma}| + \frac{1}{2}\log|\boldsymbol{\Phi}| + \frac{M}{2}\log\beta\right) - \Delta. \tag{4.24}$$

Let us define $\boldsymbol{\Sigma}_y$ such that

$$\boldsymbol{\Sigma}_y = \beta^{-1}\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T. \tag{4.25}$$

Using the formula in Eq. (C.95) in the Appendix, the relationship

$$|\boldsymbol{\Phi}||\beta^{-1}\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T| = |\beta^{-1}\boldsymbol{I}||\boldsymbol{\Phi} + \beta\boldsymbol{H}^T\boldsymbol{H}| \tag{4.26}$$

holds. Thus, substituting Eqs. (4.25) and (4.13) into the above equation, we get

$$|\boldsymbol{\Phi}||\boldsymbol{\Sigma}_y| = |\beta^{-1}\boldsymbol{I}||\boldsymbol{\Gamma}|, \tag{4.27}$$

and

$$\log|\boldsymbol{\Sigma}_y| = \log|\boldsymbol{\Gamma}| - M\log\beta - \log|\boldsymbol{\Phi}|. \tag{4.28}$$

On the other hand, according to Sect. 4.10.2, Δ is expressed as

$$\Delta = \frac{1}{2}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k, \tag{4.29}$$

where $\boldsymbol{\Sigma}_y$ is given in Eq. (4.25). Thus, substituting Eqs. (4.28) and (4.29), into (4.24), we get

$$\log p(\boldsymbol{y}|\boldsymbol{\alpha}) = -\frac{1}{2}K \log |\boldsymbol{\Sigma}_y| - \frac{1}{2}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{4.30}$$

The above equation indicates that $p(\boldsymbol{y}|\boldsymbol{\alpha})$ is Gaussian with the mean equal to zero and covariance matrix equal to $\boldsymbol{\Sigma}_y$. This $\boldsymbol{\Sigma}_y$ is called the model data covariance. Therefore, the estimate of $\boldsymbol{\alpha}$, $\widehat{\boldsymbol{\alpha}}$, is obtained by maximizing $\log p(\boldsymbol{y}|\boldsymbol{\alpha})$ expressed above. Alternatively, defining the cost function such that

$$\mathcal{F}(\boldsymbol{\alpha}) = \log |\boldsymbol{\Sigma}_y| + \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k, \tag{4.31}$$

the estimate $\widehat{\boldsymbol{\alpha}}$ is obtained by minimizing this cost function.

4.4 Update Equations for $\boldsymbol{\alpha}$

In this section, we derive the update equation for $\boldsymbol{\alpha}$. As will be shown, the updated equation contains the parameters of the posterior distribution. Since the value of $\boldsymbol{\alpha}$ is needed to compute the posterior distribution, the algorithm for computing $\boldsymbol{\alpha}$ is a recursive algorithm, as is the case of the EM algorithm presented in Sect. B.5 in the Appendix. That is, first setting an initial value for $\boldsymbol{\alpha}$, the posterior distribution is computed. Then, using the parameters of the posterior distribution, $\boldsymbol{\alpha}$ is updated. These procedures are repeated until a certain stopping condition is met.

Let us derive the update equation for $\boldsymbol{\alpha}$ by minimizing the cost function $\mathcal{F}(\boldsymbol{\alpha})$, i.e.,

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, \mathcal{F}(\boldsymbol{\alpha}).$$

The derivative of $\mathcal{F}(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ is computed,

$$\frac{\partial \mathcal{F}(\boldsymbol{\alpha})}{\partial \alpha_j} = \frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Sigma}_y| + \frac{\partial}{\partial \alpha_j} \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{4.32}$$

The first term in the right-hand side is expressed using Eq. (4.28) as

$$\begin{aligned} \frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Sigma}_y| &= \frac{\partial}{\partial \alpha_j} \left[-M \log \beta - \log |\boldsymbol{\Phi}| + \log |\boldsymbol{\Gamma}| \right] \\ &= -\frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Phi}| + \frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Gamma}|. \end{aligned} \tag{4.33}$$

First, we have

$$\frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Phi}| = \frac{\partial}{\partial \alpha_j} \sum_{j=1}^{N} \log \alpha_j = \alpha_j^{-1}. \tag{4.34}$$

Defining $\boldsymbol{\Pi}_{j,j}$ as an $(N \times N)$ matrix in which the (j, j)th component is equal to 1 and all other components are zero, we can compute the second term of Eq. (4.33), such that

$$\begin{aligned} \frac{\partial}{\partial \alpha_j} \log |\boldsymbol{\Gamma}| &= \mathrm{tr}\left[\boldsymbol{\Gamma}^{-1} \frac{\partial}{\partial \alpha_j} \boldsymbol{\Gamma}\right] = \mathrm{tr}\left[\boldsymbol{\Gamma}^{-1} \frac{\partial}{\partial \alpha_j} \left[\boldsymbol{\Phi} + \beta \boldsymbol{H}^T \boldsymbol{H}\right]\right] \\ &= \mathrm{tr}\left[\boldsymbol{\Gamma}^{-1} \frac{\partial}{\partial \alpha_j} \boldsymbol{\Phi}\right] = \mathrm{tr}\left[\boldsymbol{\Gamma}^{-1} \boldsymbol{\Pi}_{j,j}\right] = [\boldsymbol{\Gamma}^{-1}]_{j,j}, \end{aligned} \tag{4.35}$$

where $[\cdot]_{j,j}$ indicates the (j, j)th element of a matrix in the squared brackets. Note that $[\boldsymbol{\Gamma}^{-1}]_{j,j}$ above is the (j, j)th element of the posterior covariance matrix.

Next, we compute the second term of the right-hand side of Eq. (4.32). Using Eqs. (4.98) and (4.99), we can derive,

$$\begin{aligned} \frac{\partial}{\partial \alpha_j} \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k &= \frac{\partial}{\partial \alpha_j} \frac{1}{K} \sum_{k=1}^{K} \left[\beta \|\boldsymbol{y}_k - \boldsymbol{H} \bar{\boldsymbol{x}}_k\|^2 + \bar{\boldsymbol{x}}_k^T \boldsymbol{\Phi} \bar{\boldsymbol{x}}_k\right] \\ &= \frac{1}{K} \sum_{k=1}^{K} \bar{\boldsymbol{x}}_k^T \left[\frac{\partial}{\partial \alpha_j} \boldsymbol{\Phi}\right] \bar{\boldsymbol{x}}_k = \frac{1}{K} \sum_{k=1}^{K} \bar{\boldsymbol{x}}_k^T \boldsymbol{\Pi}_{j,j} \bar{\boldsymbol{x}}_k \\ &= \frac{1}{K} \sum_{k=1}^{K} \bar{x}_j^2(t_k). \end{aligned} \tag{4.36}$$

Therefore, denoting $[\boldsymbol{\Gamma}^{-1}]_{j,j}$ as $\Sigma_{j,j}$, the relationship

$$\frac{\partial \mathcal{F}(\boldsymbol{\alpha})}{\partial \alpha_j} = \Sigma_{j,j} - \alpha_j^{-1} + \frac{1}{K} \sum_{k=1}^{K} \bar{x}_j^2(t_k) = 0 \tag{4.37}$$

holds, and we get

$$\alpha_j^{-1} = \Sigma_{j,j} + \frac{1}{K} \sum_{k=1}^{K} \bar{x}_j^2(t_k). \tag{4.38}$$

This is equal to the update equation in the EM algorithm derived in Eq. (B.42).

On the other hand, we can derive a different update equation. To do so, we rewrite the expression for the posterior precision matrix in Eq. (4.13) to $\boldsymbol{I} - \boldsymbol{\Gamma}^{-1} \boldsymbol{\Phi} = \beta \boldsymbol{\Gamma}^{-1} \boldsymbol{H}^T \boldsymbol{H}$, where the (j, j)th element of this equation is

$$1 - \alpha_j [\boldsymbol{\Gamma}^{-1}]_{j,j} = \left[\beta \boldsymbol{\Gamma}^{-1} \boldsymbol{H}^T \boldsymbol{H}\right]_{j,j}. \tag{4.39}$$

We then rewrite Eq. (4.37) as

$$1 - \alpha_j [\boldsymbol{\Gamma}^{-1}]_{j,j} = \alpha_j \frac{1}{K} \sum_{k=1}^{K} \bar{x}_j^2(t_k). \tag{4.40}$$

Thus, we can derive the following equation:

$$\alpha_j = \frac{\left[\beta \boldsymbol{\Gamma}^{-1} \boldsymbol{H}^T \boldsymbol{H}\right]_{j,j}}{\frac{1}{K} \sum_{k=1}^{K} \bar{x}_j^2(t_k)}. \tag{4.41}$$

Equation (4.41) is called the MacKay update equation [5]. The MacKay update is known to be faster than the EM update in Eq. (4.38) particularly when the estimation problem is highly ill-posed, i.e., $M < N$, although there is no theoretical proof that guarantees the convergence of the MacKay update.

Let us summarize the algorithm to estimate the source distribution $\boldsymbol{x}_k$. First, $\boldsymbol{\alpha}$ is set to an appropriate initial value and the parameters of the posterior distribution, $\bar{\boldsymbol{x}}_k$ and $\boldsymbol{\Gamma}$, are computed using Eqs. (4.13) and (4.14). Then, the hyperparameter $\boldsymbol{\alpha}$ is updated using Eq. (4.41) with the values of $\bar{\boldsymbol{x}}_k$ and $\boldsymbol{\Gamma}$ obtained in the preceding step. The algorithm is similar to the EM algorithm and the only difference is the update equation for $\boldsymbol{\alpha}$.

4.5 Modified Algorithm Integrating Interference Suppression

A simple modification of the algorithm described in the preceding sections leads to an algorithm robust to the interference overlapped onto the sensor data $\boldsymbol{y}$. The idea is similar to the one for the PFA algorithm described in Sect. 5.4, and the prerequisite is that a control measurement, which contains the interference but not the signal of interest, be available. Using the factor analysis model, the data is expressed such that

$$\boldsymbol{y}_k = \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{\varepsilon} \quad \text{for control data,} \tag{4.42}$$

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{\varepsilon} \quad \text{for target data.} \tag{4.43}$$

In the above equations, $L \times 1$ column vector $\boldsymbol{u}_k$ is the factor activity and $\boldsymbol{B}$ is an $M \times L$ mixing matrix. As in Sect. 5.4, $\boldsymbol{B}\boldsymbol{u}_k$ represents the interference.

Similar to the PFA algorithm, this modified Champagne algorithm has a two-step procedure. The first step applies the VBFA (or BFA) algorithm to the control data, and estimates the interference mixing matrix $\boldsymbol{B}$ and the sensor-noise precision β. The target data is modeled as

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{\varepsilon} = [\boldsymbol{H},\ \boldsymbol{B}] \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} + \boldsymbol{\varepsilon} = \boldsymbol{H}_c \boldsymbol{z}_k + \boldsymbol{\varepsilon}, \tag{4.44}$$

where

$$\boldsymbol{H}_c = [\boldsymbol{H}, \boldsymbol{B}] \quad \text{and} \quad z_k = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix}.$$

Here, $\boldsymbol{H}_c$ and z_k can respectively be called the extended lead field matrix and the extended source vector. The second step applies the Champagne algorithm to the target data using the extended lead field matrix $\boldsymbol{H}_c$ where $\boldsymbol{B}$ is given from the first step.

The prior probability for the extended source vector z_k is:

$$\begin{aligned} p(z_k) &= p(\boldsymbol{x}_k)p(\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{0}, \boldsymbol{\Phi}^{-1})\mathcal{N}(\boldsymbol{u}_\mathbf{k}|\boldsymbol{0}, \boldsymbol{I}) \\ &= \left|\frac{\boldsymbol{\Phi}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}\boldsymbol{x}_k^T \boldsymbol{\Phi} \boldsymbol{x}_k\right] \left|\frac{\boldsymbol{I}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}\boldsymbol{u}_k^T \boldsymbol{u}_k\right] \\ &= \left|\frac{\widetilde{\boldsymbol{\Phi}}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}z_k^T \widetilde{\boldsymbol{\Phi}} z_k\right] = \mathcal{N}(z_k|\boldsymbol{0}, \widetilde{\boldsymbol{\Phi}}), \end{aligned} \tag{4.45}$$

where the $(N+L) \times (N+L)$ matrix $\widetilde{\boldsymbol{\Phi}}$ is the prior precision matrix of the extended source vector, expressed as

$$\widetilde{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \alpha_N & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.46}$$

In this modified version of the Champagne algorithm, we use the same update equation for $\boldsymbol{\alpha}$ with $\boldsymbol{\Phi}$ replaced with $\widetilde{\boldsymbol{\Phi}}$. Thus, when updating the hyperparameter $\boldsymbol{\alpha}$, only components up to the Nth diagonal element of $\widetilde{\boldsymbol{\Phi}}$ are updated, and the rest of the elements are fixed to 1.

4.6 Convexity-Based Algorithm

In this section, we describe an alternative algorithm that minimizes the cost function in Eq. (4.31). The algorithm is called the convexity-based algorithm [1, 2]. It is faster than the EM algorithm. Unlike the MacKay update, this algorithm is guaranteed to converge. The algorithm also provides a theoretical basis for the sparsity analysis described in Sect. 4.7.

4.6.1 Deriving an Alternative Cost Function

The convexity-based algorithm makes use of the fact that $\log|\boldsymbol{\Sigma}_y|$ is a concave function of $1/\alpha_1, \ldots, 1/\alpha_N$, which is the voxel variance of the prior probability

distribution. Because of this, we use the voxel variance instead of the voxel precision in this section. We define prior voxel variance of the jth voxel as ν_j, which is equal to $1/\alpha_j$, and define the column vector $\boldsymbol{\nu}$ such that $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_N]^T$. We rewrite the cost function in Eq. (4.31) using $\boldsymbol{\nu}$,

$$\mathcal{F}(\boldsymbol{\nu}) = \log|\boldsymbol{\Sigma}_y| + \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k, \tag{4.47}$$

where the model data covariance $\boldsymbol{\Sigma}_y$ is expressed as

$$\boldsymbol{\Sigma}_y = \beta^{-1}\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Upsilon}\boldsymbol{H}^T, \tag{4.48}$$

and $\boldsymbol{\Upsilon}$ is the covariance matrix of the voxel prior distribution, defined as $\boldsymbol{\Upsilon} = \text{diag}([\nu_1, \ldots, \nu_N])$.

The first term in Eq. (4.47), $\log|\boldsymbol{\Sigma}_y|$, is a concave function of $\nu_1, \ldots, \nu_N$. Thus, for an arbitrary $\boldsymbol{\nu}$, we can find $\boldsymbol{z}$ that satisfies the relationship

$$\boldsymbol{z}^T\boldsymbol{\nu} - z_o \geq \log|\boldsymbol{\Sigma}_y|, \tag{4.49}$$

where $\boldsymbol{z}$ is the column vector $\boldsymbol{z} = [z_1, \ldots, z_N]$, which contains auxiliary variables z_j $(j = 1, \ldots, N)$, and z_o is a scalar term that depends on $\boldsymbol{z}$.

The second term in Eq. (4.47) can be rewritten using

$$\boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k = \min_{\boldsymbol{x}_k} \left[\beta\|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T\boldsymbol{\Upsilon}^{-1}\boldsymbol{x}_k\right]. \tag{4.50}$$

The proof for the equation above is presented in Sect. 4.10.3. Therefore, we have the relationship

$$\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k = \min_{\boldsymbol{x}} \frac{1}{K}\sum_{k=1}^{K}\left[\beta\|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T\boldsymbol{\Upsilon}^{-1}\boldsymbol{x}_k\right]. \tag{4.51}$$

Using auxiliary variables $\boldsymbol{z}$ and $\boldsymbol{x}$ (where $\boldsymbol{x}$ collectively expresses $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$), we define a new cost function $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z})$, such that

$$\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z}) = \frac{1}{K}\sum_{k=1}^{K}\left[\beta\|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T\boldsymbol{\Upsilon}^{-1}\boldsymbol{x}_k\right] + \boldsymbol{z}^T\boldsymbol{\nu} - z_o. \tag{4.52}$$

Equations (4.49) and (4.51) guarantee that the relationship

$$\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z}) \geq \mathcal{F}(\boldsymbol{\nu}),$$

always holds. That is, the alternative cost function $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z})$ forms an upper bound of the true cost function $\mathcal{F}(\boldsymbol{\nu})$. When we minimize $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z})$ with respect to $\boldsymbol{\nu}$, $\boldsymbol{x}$ and $\boldsymbol{z}$, such $\boldsymbol{\nu}$ also minimizes the true cost function $\mathcal{F}(\boldsymbol{\nu})$.

4.6.2 Update Equation for z

Let us derive the update equation for the auxiliary variable z. The update value of z, $\widehat{z}$, minimizes the cost function $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, z)$ and at the same time satisfies the constraint in Eq. (4.49). Since $z^T\boldsymbol{\nu} - z_o$ are the only terms that depend on z in $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, z)$, this minimization problem is expressed as

$$\widehat{z} = \underset{z}{\operatorname{argmin}} \left(z^T\boldsymbol{\nu} - z_o \right) \quad \text{subject to} \quad z^T\boldsymbol{\nu} - z_o \geq \log|\boldsymbol{\Sigma}_y|. \tag{4.53}$$

That is, the minimization problem is equivalent to finding the hyperplane $z^T\boldsymbol{\nu} - z_o$ that forms a closest upper bound of $\log|\boldsymbol{\Sigma}_y|$. Such a hyperplane is found as the plane that is tangential to $\log|\boldsymbol{\Sigma}_y|$[6]. Therefore, the update value $\widehat{z}$ is given by

$$\widehat{z} = \frac{\partial}{\partial \boldsymbol{\nu}} \log|\boldsymbol{\Sigma}_y|. \tag{4.54}$$

This $\widehat{z}$ forms a bound tightest to the concave function $\log|\boldsymbol{\Sigma}_y|$, and minimizes the cost function $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, z)$ with respect to z. To compute $\widehat{z}$, we use

$$\widehat{z}_j = \frac{\partial}{\partial \nu_j} \log|\boldsymbol{\Sigma}_y| = \operatorname{tr}\left[\boldsymbol{\Sigma}_y^{-1} \frac{\partial}{\partial \nu_j} \boldsymbol{\Sigma}_y \right], \tag{4.55}$$

and

$$\begin{aligned} \frac{\partial}{\partial \nu_j} \boldsymbol{\Sigma}_y &= \frac{\partial}{\partial \nu_j} \left[\beta^{-1}\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Upsilon}\boldsymbol{H}^T \right] \\ &= \boldsymbol{H} \frac{\partial}{\partial \nu_j} \boldsymbol{\Upsilon} \boldsymbol{H}^T = \boldsymbol{H}\boldsymbol{\Pi}_{j,j}\boldsymbol{H}^T = \boldsymbol{l}_j \boldsymbol{l}_j^T, \end{aligned} \tag{4.56}$$

where $\boldsymbol{l}_j$ is the lead field vector at the jth voxel, which is the jth column of $\boldsymbol{H}$. Substituting Eqs. (4.56) into (4.55), we get

$$\frac{\partial}{\partial \nu_j} \log|\boldsymbol{\Sigma}_y| = \operatorname{tr}\left[\boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_j \boldsymbol{l}_j^T \right] = \operatorname{tr}\left[\boldsymbol{l}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_j \right] = \boldsymbol{l}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_j. \tag{4.57}$$

Using the equation above, we arrive at

$$\widehat{z}_j = \underset{z_j}{\operatorname{argmin}}\, \widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, z) = \boldsymbol{l}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_j \tag{4.58}$$

for the update equation of $\widehat{z}_j$.

4.6.3 Update Equation for $\boldsymbol{x}_k$

Now, let us derive the update equation for the auxiliary variable $\boldsymbol{x}$. Since the relationship

$$\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}_k}{\operatorname{argmin}} \left[\beta \|\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k \right] \tag{4.59}$$

holds, according to Eq. (4.101) in Sect. 4.10.3, $\widehat{\boldsymbol{x}}_k$ is obtained as

$$\widehat{\boldsymbol{x}}_k = \beta \left(\boldsymbol{\Upsilon}^{-1} + \beta \boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{y}_k. \tag{4.60}$$

This is the update equation for $\boldsymbol{x}_k$. Using the matrix inversion formula in Eq. (C.92), this equation can be rewritten as

$$\widehat{\boldsymbol{x}}_k = \boldsymbol{\Upsilon} \boldsymbol{H}^T \left(\beta^{-1} \boldsymbol{I} + \boldsymbol{H} \boldsymbol{\Upsilon} \boldsymbol{H}^T \right)^{-1} \boldsymbol{y}_k = \boldsymbol{\Upsilon} \boldsymbol{H}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{4.61}$$

Since this expression uses $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Sigma}_y$, it is conveniently used in the convexity-based algorithm. Note that the auxiliary variable $\boldsymbol{x}_k$ is equal to the posterior mean of the voxel source distribution because the update equation above is exactly equal to Eq. (B.26).

4.6.4 Update Equation for $\boldsymbol{\nu}$

Let us derive $\boldsymbol{\nu}$ that minimizes the cost function $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{z}, \boldsymbol{x})$. Since only the second and the third terms in $\widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{z}, \boldsymbol{x})$ contains $\boldsymbol{\nu}$ (as shown in Eq. (4.52)), we have:

$$\begin{aligned} \widehat{\boldsymbol{\nu}} &= \underset{\boldsymbol{\nu}}{\operatorname{argmin}}\, \widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{x}, \boldsymbol{z}) = \underset{\boldsymbol{\nu}}{\operatorname{argmin}} \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k + \boldsymbol{z}^T \boldsymbol{\nu} \right] \\ &= \underset{\boldsymbol{\nu}}{\operatorname{argmin}} \sum_{j=1}^{N} \left[z_j \nu_j + \frac{\frac{1}{K} \sum_{k=1}^{K} x_j^2(t_k)}{\nu_j} \right]. \end{aligned} \tag{4.62}$$

Thus, considering the relationship

$$\frac{\partial}{\partial \nu_j} \sum_{k=1}^{N} \left[z_j \nu_j + \frac{\frac{1}{K} \sum_{k=1}^{K} x_j^2(t_k)}{\nu_j} \right] = z_j - \frac{\frac{1}{K} \sum_{k=1}^{K} x_j^2(t_k)}{\nu_j^2} = 0,$$

we can derive

$$\widehat{\nu}_j = \underset{\nu_j}{\operatorname{argmin}}\, \widetilde{\mathcal{F}}(\boldsymbol{\nu}, \boldsymbol{z}, \boldsymbol{x}) = \sqrt{\frac{\frac{1}{K} \sum_{k=1}^{K} x_j^2(t_k)}{z_j}}. \tag{4.63}$$

4.6.5 Summary of the Convexity-Based Algorithm

The update for $\boldsymbol{z}$ in Eq. (4.58) and the update for $\boldsymbol{x}$ in Eq. (4.61) require $\boldsymbol{\nu}$ to be known. The update equation for $\boldsymbol{\nu}$ in Eq. (4.63) requires $\boldsymbol{x}$ and $\boldsymbol{z}$ to be known. Therefore, the convexity-based algorithm updates $\boldsymbol{z}$, $\boldsymbol{x}$ and $\boldsymbol{\nu}$ using Eqs. (4.58), (4.61) and (4.63), in

a recursive manner. Since the auxiliary variable $\boldsymbol{x}_k$ is the posterior mean of the voxel source distribution, the value of the auxiliary variable $\widehat{\boldsymbol{x}}_k$ is equal to the Bayesian estimate of the source distribution, $\bar{\boldsymbol{x}}_k$, after the iterative procedure is terminated.

4.7 The Origin of the Sparsity

Why does the Champagne algorithm produce sparse solutions? This section tries to answer this question, and consider the origin of the sparsity by analyzing the cost function of the Champagne algorithm [7]. For simplicity, we set $K = 1$, and by omitting the time index, $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ are denoted $\boldsymbol{x}$ and $\boldsymbol{y}$. Using Eq. (4.47), we have

$$\widehat{\boldsymbol{\nu}} = \underset{\boldsymbol{\nu}}{\operatorname{argmin}} \left(\log |\boldsymbol{\Sigma}_y| + \boldsymbol{y}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y} \right).$$

As shown in Eq. (4.50), we have

$$\boldsymbol{y}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y} = \min_{\boldsymbol{x}} \left[\beta \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \boldsymbol{x}^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x} \right].$$

Combining the two equations above, the cost function for estimating $\boldsymbol{x}$ is given by

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \mathcal{F} : \quad \mathcal{F} = \beta \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \phi(\boldsymbol{x}), \tag{4.64}$$

where the constraint $\phi(\boldsymbol{x})$ is expressed as

$$\phi(\boldsymbol{x}) = \min_{\boldsymbol{\nu}} \left(\boldsymbol{x}^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x} + \log |\boldsymbol{\Sigma}_y| \right) = \min_{\boldsymbol{\nu}} \left(\sum_{j=1}^{N} \frac{x_j^2}{\nu_j} + \log |\boldsymbol{\Sigma}_y| \right). \tag{4.65}$$

However, computing $\phi(\boldsymbol{x})$ in Eq. (4.65) is not so easy because $\log |\boldsymbol{\Sigma}_y|$ contains $\boldsymbol{\nu}$ as in Eq. (4.48). Therefore, we further introduce a simplification by assuming that the columns in the matrix $\boldsymbol{H}$ are orthogonal, i.e., the relationship $\boldsymbol{l}_i^T \boldsymbol{l}_j = I_{i,j}$ is assumed to hold. In this case, using Eq. (4.48), we get

$$\log |\boldsymbol{\Sigma}_y| = \sum_{j=1}^{N} \log(\beta^{-1} + \nu_j),$$

and, accordingly,

$$\phi(\boldsymbol{x}) = \min_{\boldsymbol{\nu}} \sum_{j=1}^{N} \left(\frac{x_j^2}{\nu_j} + \log(\beta^{-1} + \nu_j) \right). \tag{4.66}$$

By computing the minimum on the right-hand side, we finally have

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{N} \varphi(x_j), \tag{4.67}$$

where

$$\varphi(x) = \frac{2|x|}{\sqrt{x^2 + 4\beta^{-1}} + |x|} + \log\left[\beta^{-1} + \frac{x^2}{2} + \frac{1}{2}|x|\sqrt{x^2 + 4\beta^{-1}}\right]. \tag{4.68}$$

The constraint $\varphi(x)$ in Eq. (4.68) is plotted in Fig. 4.1. In Fig. 4.1a, the plot of $\varphi(x)$ is shown by the solid line. For comparison, the constraint for the L_1-norm solution, $|x|$, is also shown by the broken line. The plots in Fig. 4.1a show that the constraint of the Champagne algorithm $\varphi(x)$ is very similar to (but sharper than) the L_1-norm constraint $|x|$, suggesting that the Champagne algorithm produces sparse solutions.

In Fig. 4.1b, the plots of $\varphi(x)$ when $\beta^{-1} = 0.1$, $\beta^{-1} = 1$ and $\beta^{-1} = 10$ are shown by the dot-and-dash, broken, and solid lines, respectively. The vertical broken line at $x = 0$ shows the L_0-norm constraint, $\|x\|_0$, for comparison. It is shown here that the shape of the constraint $\varphi(x)$ depends on β^{-1}—namely the noise variance. When the noise variance is small (i.e., a high SNR), $\varphi(x)$ is much sharper than $|x|$, and becomes closer to the L_0-norm constraint. That is, the Champagne algorithm uses an adaptive constraint. When the SNR of the sensor data is high, it gives solutions with enhanced sparsity. In contrast, when the sensor data is noisy, the shape of $\varphi(x)$ becomes similar to the shape of the L_1-norm constraint and the algorithm gives solutions with mild sparsity.

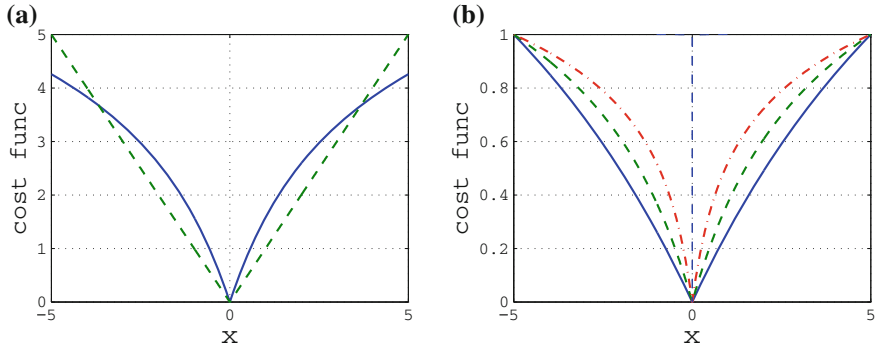

Fig. 4.1 Plots of the cost function $\varphi(x)$ shown in Eq. (4.68). **a** The *solid line* shows the plot of $\varphi(x)$ with $\beta^{-1} = 1$, and the *broken line* the plot of $|x|$, which is the constraint for the L_1-norm minimum-norm solution. **b** Plots of $\varphi(\boldsymbol{x})$ when $\beta^{-1} = 0.1$, $\beta^{-1} = 1$ and $\beta^{-1} = 10$ shown by the *dot*-and-*dash*, *broken*, and *solid lines*, respectively. In this figure, the *vertical broken line* at $x = 0$ shows the constraint for the L_0-norm solution

4.8 Extension to Include Source Vector Estimation

In Sect. 4.6, we derive the convexity-based algorithm when the source orientation at each voxel is known, i.e., when the relationship between the sensor data and the voxel source distribution is expressed as in Eq. (4.3).

In this section, we derive the convexity-based algorithm when the source orientation at each voxel is unknown and the relationship,

$$\boldsymbol{y}_k = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{\varepsilon}, \tag{4.69}$$

holds, where $\boldsymbol{x}_k$ is given by

$$\boldsymbol{x}_k = \left[\boldsymbol{s}_1^T(t_k), \ldots, \boldsymbol{s}_N^T(t_k)\right]^T, \tag{4.70}$$

and $\boldsymbol{s}_j(t_k)$ is the 3×1 source vector at the jth voxel. In other words, we describe an extension that enables estimating the source vector at each voxel. The algorithms in the preceding sections use a diagonal prior covariance (or precision) matrix, and the use of the diagonal matrix is possible because the source orientation is known. Therefore, a naïve application of those algorithms to cases where the source orientation is unknown generally results in the shrinkage of source vector components. Namely, the naïve application possibly leads to a solution where only a single component of a source vector has nonzero value and other components have values close to zero, leading to incorrect estimation of source orientations.

Since the unknown parameter at each voxel is not a scalar quantity but the 3×1 vector quantity in this section, the algorithm being developed here uses the nondiagonal covariance matrix of the prior distribution. That is, the prior distribution is assumed to be [1];

$$p(\boldsymbol{x}_k|\boldsymbol{\Upsilon}) = \prod_{j=1}^{N} \mathcal{N}(\boldsymbol{s}_j(t_k)|\boldsymbol{0}, \boldsymbol{\Upsilon}_j), \tag{4.71}$$

where $\boldsymbol{\Upsilon}_j$ is a 3×3 covariance matrix of the source vector $\boldsymbol{s}_j(t_k)$. Thus, the covariance matrix of the voxel source distribution, $\boldsymbol{\Upsilon}$, is a $3N \times 3N$ block diagonal matrix expressed as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{\Upsilon}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Upsilon}_2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Upsilon}_N \end{bmatrix}. \tag{4.72}$$

The cost function in this case has the same form:

$$\mathcal{F}(\boldsymbol{\Upsilon}) = \log|\boldsymbol{\Sigma}_y| + \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{4.73}$$

However, $\boldsymbol{\Sigma}_y$ is given by

$$\boldsymbol{\Sigma}_y = \beta^{-1}\boldsymbol{I} + \sum_{j=1}^{N} \boldsymbol{L}_j \boldsymbol{\Upsilon}_j \boldsymbol{L}_j^T, \tag{4.74}$$

where the $M \times 3$ lead field matrix at the jth voxel, $\boldsymbol{L}(\boldsymbol{r}_j)$, is denoted $\boldsymbol{L}_j$ for simplicity.

In agreement with Eq. (4.49), since $\log|\boldsymbol{\Sigma}_y|$ is a concave function, we can find 3×3 auxiliary parameter matrices, $\boldsymbol{Z}_j$, $(j = 1, \ldots, N)$ that satisfy

$$\sum_{j=1}^{N} \mathrm{tr}\left(\boldsymbol{Z}_j^T \boldsymbol{\Upsilon}_j\right) - Z_o \geq \log|\boldsymbol{\Sigma}_y|, \tag{4.75}$$

where Z_o is a scalar term that depends on $\boldsymbol{Z}_j$. Regarding the second term in the right-hand side of Eq. (4.73), we have the relationship

$$\begin{aligned} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k &= \min_{\boldsymbol{x}_k} \left[\beta \|\boldsymbol{y}_k - \boldsymbol{F}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k\right] \\ &= \min_{\boldsymbol{x}_k} \left[\beta \|\boldsymbol{y}_k - \boldsymbol{F}\boldsymbol{x}_k\|^2 + \sum_{j=1}^{N} \boldsymbol{s}_j^T(t_k) \boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j(t_k)\right]. \end{aligned} \tag{4.76}$$

Let us use $\boldsymbol{Z}$ to collectively express $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N$, and use $\boldsymbol{s}$ to collectively express $\boldsymbol{s}_j(t_k)$, where $j = 1, \ldots, N$ and $k = 1, \ldots, K$. The alternative cost function, $\widetilde{\mathcal{F}}(\boldsymbol{\Upsilon}, \boldsymbol{s}, \boldsymbol{Z})$, is obtained as

$$\begin{aligned} \widetilde{\mathcal{F}}(\boldsymbol{\Upsilon}, \boldsymbol{s}, \boldsymbol{Z}) = \frac{1}{K} \sum_{k=1}^{K} &\left[\beta \|\boldsymbol{y}_k - \boldsymbol{F}\boldsymbol{x}_k\|^2 + \sum_{j=1}^{N} \boldsymbol{s}_j(t_k) \boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j^T(t_k)\right] \\ &+ \sum_{j=1}^{N} \mathrm{tr}\left(\boldsymbol{Z}_j^T \boldsymbol{\Upsilon}_j\right) - Z_o. \end{aligned} \tag{4.77}$$

This $\widetilde{\mathcal{F}}(\boldsymbol{\Upsilon}, \boldsymbol{s}, \boldsymbol{Z})$ forms an upper bound of the true cost function in Eq. (4.73). Accordingly, when we minimize $\widetilde{\mathcal{F}}(\boldsymbol{\Upsilon}, \boldsymbol{s}, \boldsymbol{Z})$ with respect to $\boldsymbol{\Upsilon}$, $\boldsymbol{s}$, and $\boldsymbol{Z}$, we can minimize the true cost function $\mathcal{F}(\boldsymbol{\Upsilon})$.

4.8.1 Update Equation for $\boldsymbol{Z}_j$

Using the same arguments in Sect. 4.6.2, the update equation for $\boldsymbol{Z}_j$ is derived as

$$\widehat{\boldsymbol{Z}}_j = \frac{\partial}{\partial \boldsymbol{\Upsilon}_j} \log|\boldsymbol{\Sigma}_y|. \tag{4.78}$$

Here, the hyperplane $\sum_{j=1}^{N} \mathrm{tr}\left(\widehat{\boldsymbol{Z}}_j^T \boldsymbol{\Upsilon}_j\right) - Z_o$ forms a tightest upper bound of the concave function $\log|\boldsymbol{\Sigma}_y|$. Let us express the lead field matrix at the jth voxel using its column vectors,

$$\boldsymbol{L}_j = \left[\boldsymbol{l}_x(\boldsymbol{r}_j), \boldsymbol{l}_y(\boldsymbol{r}_j), \boldsymbol{l}_z(\boldsymbol{r}_j)\right] = \left[\boldsymbol{l}_x, \boldsymbol{l}_y, \boldsymbol{l}_z\right],$$

where explicit notation of the voxel location $(\boldsymbol{r}_j)$ is omitted. Using exactly the same derivation from Eqs. (4.55) to (4.58), we can derive the update equation

$$\begin{aligned}
\widehat{\boldsymbol{Z}}_j &= \frac{\partial}{\partial \boldsymbol{\Upsilon}_j} \log|\boldsymbol{\Sigma}_y| \\
&= \begin{bmatrix} \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{1,1} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{1,2} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{1,3} \\ \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{2,1} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{2,2} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{2,3} \\ \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{3,1} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{3,2} & \partial \log|\boldsymbol{\Sigma}_y|/\partial[\boldsymbol{\Upsilon}_j]_{3,3} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{l}_x^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_x & \boldsymbol{l}_x^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_y & \boldsymbol{l}_x^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_z \\ \boldsymbol{l}_y^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_x & \boldsymbol{l}_y^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_y & \boldsymbol{l}_y^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_z \\ \boldsymbol{l}_z^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_x & \boldsymbol{l}_z^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_y & \boldsymbol{l}_z^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{l}_z \end{bmatrix} = \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{L}_j,
\end{aligned} \tag{4.79}$$

where $[\boldsymbol{\Upsilon}_j]_{\ell,m}$ indicates the (ℓ, m)th element of the matrix $\boldsymbol{\Upsilon}_j$.

4.8.2 Update Equation for $s_j(t_k)$

The update equation for $\boldsymbol{x}_k$ can be obtained using

$$\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}_k}{\mathrm{argmin}} \left[\beta \|\boldsymbol{y}_k - \boldsymbol{F}\boldsymbol{x}_k\|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k\right]. \tag{4.80}$$

The solution is given by Eq. (4.61) with replacing $\boldsymbol{H}$ with $\boldsymbol{F}$, which is rewritten as

$$\begin{bmatrix} \widehat{\boldsymbol{s}}_1(t_k) \\ \widehat{\boldsymbol{s}}_2(t_k) \\ \vdots \\ \widehat{\boldsymbol{s}}_N(t_k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Upsilon}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Upsilon}_2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Upsilon}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_1^T \\ \boldsymbol{L}_2^T \\ \vdots \\ \boldsymbol{L}_N^T \end{bmatrix} \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{4.81}$$

Therefore, the source vector at the jth voxel $\widehat{\boldsymbol{s}}_j(t_k)$ is given by

$$\widehat{\boldsymbol{s}}_j(t_k) = \boldsymbol{\Upsilon}_j \boldsymbol{L}_j^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k, \tag{4.82}$$

and Eq. (4.82) is the update equation for $\boldsymbol{s}_j(t_k)$.

4.8.3 Update Equation for $\boldsymbol{\Upsilon}_j$

The update equation for $\boldsymbol{\Upsilon}_j$ is obtained using

$$\widehat{\boldsymbol{\Upsilon}}_j = \underset{\boldsymbol{\Upsilon}_j}{\operatorname{argmin}} \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}_j^T(t_k) \boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j(t_k) + \operatorname{tr}\left(\boldsymbol{Z}_j^T \boldsymbol{\Upsilon}_j \right) \right]. \tag{4.83}$$

Taking

$$\frac{\partial}{\partial \boldsymbol{\Upsilon}_j} \boldsymbol{s}_j^T(t_k) \boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j(t_k) = -\boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j(t_k) \boldsymbol{s}_j^T(t_k) \boldsymbol{\Upsilon}_j^{-1} \tag{4.84}$$

into consideration, we have

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{\Upsilon}_j} \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}_j^T(t_k) \boldsymbol{\Upsilon}_j^{-1} \boldsymbol{s}_j(t_k) + \operatorname{tr}\left(\boldsymbol{Z}_j^T \boldsymbol{\Upsilon}_j \right) \right] \\ &\quad = -\boldsymbol{\Upsilon}_j^{-1} \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}_j(t_k) \boldsymbol{s}_j^T(t_k) \right] \boldsymbol{\Upsilon}_j^{-1} + \boldsymbol{Z}_j. \end{aligned} \tag{4.85}$$

Setting the right-hand side to zero, we get the equation,

$$\boldsymbol{\Upsilon}_j \boldsymbol{Z}_j \boldsymbol{\Upsilon}_j = \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}_j(t_k) \boldsymbol{s}_j^T(t_k) \right]. \tag{4.86}$$

However, there are multiple solutions for $\boldsymbol{\Upsilon}_j$ that satisfy Eq. (4.86). We should find a positive semidefinite matrix that satisfies Eq. (4.86). Defining $\boldsymbol{\Xi} = (1/K) \sum_{k=1}^{K} \boldsymbol{s}_j(t_k) \boldsymbol{s}_j^T(t_k)$, and using

$$\begin{aligned} \boldsymbol{\Xi} &= \boldsymbol{Z}_j^{-1/2} (\boldsymbol{Z}_j^{1/2} \boldsymbol{\Xi} \boldsymbol{Z}_j^{1/2}) \boldsymbol{Z}_j^{-1/2} \\ &= \boldsymbol{Z}_j^{-1/2} (\boldsymbol{Z}_j^{1/2} \boldsymbol{\Xi} \boldsymbol{Z}_j^{1/2})^{1/2} (\boldsymbol{Z}_j^{1/2} \boldsymbol{\Xi} \boldsymbol{Z}_j^{1/2})^{1/2} \boldsymbol{Z}_j^{-1/2} \\ &= \boldsymbol{Z}_j^{-1/2} (\boldsymbol{Z}_j^{1/2} \boldsymbol{\Xi} \boldsymbol{Z}_j^{1/2})^{1/2} \boldsymbol{Z}_j^{-1/2} \boldsymbol{Z}_j \boldsymbol{Z}_j^{-1/2} (\boldsymbol{Z}_j^{1/2} \boldsymbol{\Xi} \boldsymbol{Z}_j^{1/2})^{1/2} \boldsymbol{Z}_j^{-1/2}, \end{aligned} \tag{4.87}$$

the solution for $\boldsymbol{\Upsilon}_j$ that is a positive semidefinite matrix is derived such that

$$\widehat{\boldsymbol{\Upsilon}}_j = \boldsymbol{Z}_j^{-1/2} \left[\boldsymbol{Z}_j^{1/2} \left[\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}_j(t_k) \boldsymbol{s}_j^T(t_k) \right] \boldsymbol{Z}_j^{1/2} \right]^{1/2} \boldsymbol{Z}_j^{-1/2}. \tag{4.88}$$

Equation (4.88) is the update equation for $\boldsymbol{\Upsilon}_j$.

In summary, $\boldsymbol{Z}_j$, $\boldsymbol{s}_j(t_k)$, and $\boldsymbol{\Upsilon}_j$ are updated using Eqs. (4.79), (4.82) and (4.88), respectively. Since the auxiliary variable $\boldsymbol{x}_k$ is equal to the posterior mean of the

voxel source distribution, the Bayes estimate of the voxel source distribution, $\bar{\boldsymbol{x}}_k$, can be obtained as the final updated results of $\widehat{\boldsymbol{s}}_j(t_k)$.

4.9 Source Vector Estimation Using Hyperparameter Tying

In Sect. 4.8, we describe an algorithm that reconstructs the source distribution when the source orientation is unknown. The algorithm uses the block diagonal covariance matrix $\boldsymbol{\Upsilon}$ expressed in Eq. (4.72). However, because the matrix $\boldsymbol{\Upsilon}$ is a nondiagonal matrix, the resultant algorithm is rather complex and computationally expensive, compared to the algorithm that uses a diagonal covariance matrix. In this section, we describe an extension of the Champagne algorithm that still uses a diagonal covariance matrix for the prior distribution but enables us to estimate the voxel source vectors.

The method uses a technique called hyperparameter tying, and simple modification of the algorithm described in Sects. 4.2–4.6 leads to a successful estimation of voxel source vectors. We use the forward relationship given in Eq. (4.69), and a $3N \times 3N$ diagonal covariance matrix for the Gaussian prior:

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \ddots & & \vdots & & \vdots \\ \cdots & \nu_{3n+1} & 0 & 0 & \cdots \\ \cdots & 0 & \nu_{3n+2} & 0 & \cdots \\ \cdots & 0 & 0 & \nu_{3n+3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{4.89}$$

where we show a diagonal block matrix corresponding to the nth voxel. The equation in Eq. (4.63) is applied to update ν_j. In the method proposed here, the three hyperparameters ν_{3n+1}, ν_{3n+2}, and ν_{3n+3} from the same voxel are tied together to have the same new update value. That is, for $\widehat{\nu}_{3n+1}$, $\widehat{\nu}_{3n+2}$, and $\widehat{\nu}_{3n+3}$, we compute a new single update value by using

$$\widehat{\nu}_{3n+m}^{\text{new}} = \frac{1}{3}\sum_{i=1}^{3} \widehat{\nu}_{3n+i}, \quad \text{for} \quad m = 1, 2, 3 \tag{4.90}$$

where $\widehat{\nu}_{3n+i}$ (for $i = 1, 2, 3$) are the update values obtained from Eq. (4.63), and $\widehat{\nu}_{3n+m}^{\text{new}}$ ($m = 1, 2, 3$) is a new update value for these hyperparameters.

The hyperparameter tying method can be implemented in exactly the same manner when using the MacKay update in Eq. (4.41). A new single update value for the voxel precision corresponding to the nth voxel is computed by using

$$\widehat{\alpha}_{3n+m}^{\text{new}} = \frac{1}{3}\sum_{i=1}^{3} \widehat{\alpha}_{3n+i}, \quad \text{for} \quad m = 1, 2, 3 \tag{4.91}$$

where $\widehat{\alpha}_{3n+i}$ (for $i = 1, 2, 3$) is the update value from Eq. (4.41) and $\widehat{\alpha}^{\text{new}}_{3n+m}$ (for $m = 1, 2, 3$) is a new update value for these hyperparameters.

The rationale for this hyperparameter tying can be explained using the cost function analysis described in Sect. 4.7. Let us compute the constraint function in Eq. (4.66) for a two-dimensional case in which the unknown parameters are denoted x_1 and x_2. The constraint is rewritten in this case as,

$$\phi(x_1, x_2) = \min_{\nu_1, \nu_2} \sum_{j=1}^{2} \left(\frac{x_j^2}{\nu_j} + \log(\beta^{-1} + \nu_j) \right). \tag{4.92}$$

When the hyperparameters ν_1 and ν_2 are tied together, i.e., when we set these hyperparameters at the same value ν, the constraint function is changed to

$$\phi(x_1, x_2) = \min_{\nu} \left(\frac{x_1^2 + x_2^2}{\nu} + 2\log(\beta^{-1} + \nu) \right). \tag{4.93}$$

By implementing this minimization, the value of ν that minimizes the right-hand side of the above equation, $\widehat{\nu}$, is derived as

$$\widehat{\nu} = \frac{a + \sqrt{a^2 + 8\beta^{-1}a}}{4},$$

where $a = x_1^2 + x_2^2$. Substituting this $\widehat{\nu}$ into Eq. (4.93), we derive the constraint function,

$$\phi(x_1, x_2) = \frac{4a}{a + \sqrt{a^2 + 8\beta^{-1}a}} + 2\log\left[\beta^{-1} + \frac{a + \sqrt{a^2 + 8\beta^{-1}a}}{4} \right]. \tag{4.94}$$

The plot of the constraint function in Eq. (4.94) is shown in Fig. 4.2a. For comparison, the Champagne constraint function when untying ν_1 and ν_2 (Eq. 4.92) is shown in Fig. 4.2b. The constraint functions for the L_2 and L_1-norm regularizations are also shown in Fig. 4.2c, d for comparison. These plots show that the Champagne constraint function when tying ν_1 and ν_2 has a shape very similar to the constraint function for the L_2 regularization. According to the arguments in Sect. 2.9.2, this type of constraint does not generate a sparse solution. Thus, when tying the hyperparameter update values, the sparsity is lost among the solutions of x_{2n+1}, x_{2n+2}, and x_{2n+3} and there is no shrinkage over the source vector components. However, since the sparsity is maintained across voxels, a sparse source distribution can still be reconstructed.

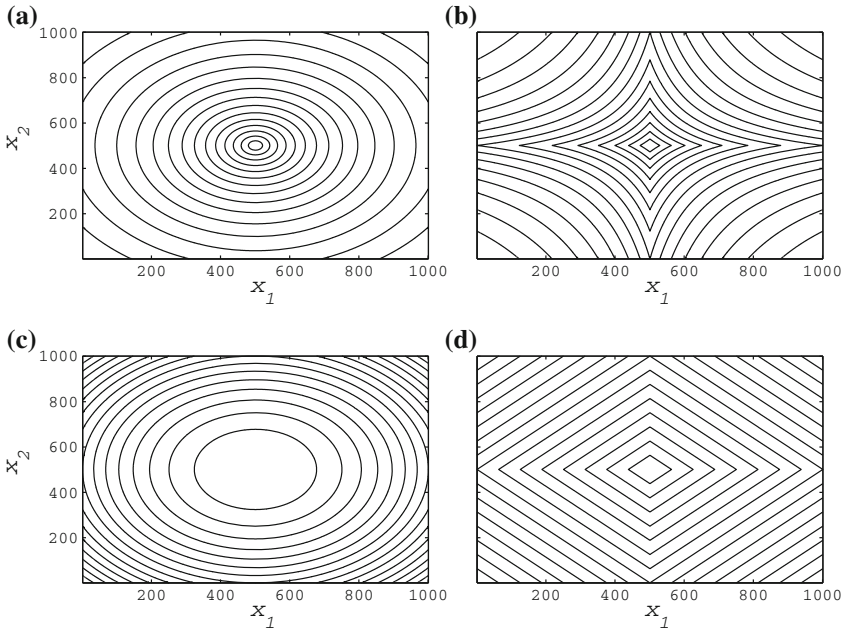

Fig. 4.2 Plots of the constraint functions for the two-dimensional case. **a** The plot of $\phi(x_1, x_2)$ in Eq. (4.94). **b** The plot of $\phi(x_1, x_2)$ in Eq. (4.92). **c** The plot of the constraint function for the L_2-norm constraint: $\phi(x_1, x_2) = x_1^2 + x_2^2$. **d** The plot of the constraint function for the L_1-norm constraint: $\phi(x_1, x_2) = |x_1| + |x_2|$. The parameter β was set to 1 when computing the plots in **a** and **b**

4.10 Appendix to This Chapter

4.10.1 Derivation of Eq. (4.21)

D in Eq. (4.20) is rewritten as

$$D = \sum_{k=1}^{K} \left[\frac{1}{2} \boldsymbol{x}_k^T \left[\boldsymbol{\Phi} + \beta \boldsymbol{H}^T \boldsymbol{H} \right] \boldsymbol{x}_k - \beta \boldsymbol{x}_k^T \boldsymbol{H}^T \boldsymbol{y}_k \right] + \mathcal{C}. \tag{4.95}$$

Here, terms not containing $\boldsymbol{x}_k$ are expressed as $\mathcal{C}$. We apply the completion of the square with respect to $\boldsymbol{x}_k$, i.e., we change Eq. (4.95) to have a form,

$$D = \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{x}_k - \boldsymbol{m}_k)^T \boldsymbol{A} (\boldsymbol{x}_k - \boldsymbol{m}_k) + \Delta,$$

where $\boldsymbol{A}$ is a real symmetric matrix and $\boldsymbol{m}_k$ is a column vector, and Δ represents remaining terms. The first term on the right-hand side is rewritten as

$$\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{x}_k-\boldsymbol{m}_k)^T\,\boldsymbol{A}\,(\boldsymbol{x}_k-\boldsymbol{m}_k)=\sum_{k=1}^{K}\left[\frac{1}{2}\boldsymbol{x}_k^T\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{x}_k^T\boldsymbol{A}\boldsymbol{m}_k+\cdots\right].$$

Comparing the right-hand side of the equation above with that of Eq. (4.95), we get

$$\begin{aligned}\boldsymbol{A}&=\boldsymbol{\Phi}+\beta\boldsymbol{H}^T\boldsymbol{H},\\ \boldsymbol{A}\boldsymbol{m}_k&=\beta\boldsymbol{H}^T\boldsymbol{y}_k\quad\text{namely}\quad\boldsymbol{m}_k=\beta\left[\boldsymbol{\Phi}+\beta\boldsymbol{H}^T\boldsymbol{H}\right]^{-1}\boldsymbol{H}^T\boldsymbol{y}_k.\end{aligned}$$

Comparing the equations above with Eqs. (4.13) and (4.14), we have the relationships $\boldsymbol{A}=\boldsymbol{\Gamma}$ and $\boldsymbol{m}_k=\bar{\boldsymbol{x}}_k$, giving

$$D=\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k)^T\,\boldsymbol{\Gamma}\,(\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k)+\Delta. \tag{4.96}$$

This equation shows that D reaches the minimum at $\boldsymbol{x}_k=\bar{\boldsymbol{x}}_k$, and the minimum value is equal to Δ. The value of Δ is obtained by substituting $\boldsymbol{x}_k=\bar{\boldsymbol{x}}_k$ into Eq. (4.20), such that

$$\Delta=\frac{\beta}{2}\sum_{k=1}^{K}\|\boldsymbol{y}_k-\boldsymbol{H}\bar{\boldsymbol{x}}_k\|^2+\frac{1}{2}\sum_{k=1}^{K}\bar{\boldsymbol{x}}_k^T\boldsymbol{\Phi}\bar{\boldsymbol{x}}_k. \tag{4.97}$$

4.10.2 Derivation of Eq. (4.29)

The derivation starts from Δ in Eq. (4.97). On the right-hand side of this equation, the kth terms containing $\boldsymbol{x}_k$ are

$$\begin{aligned}\frac{\beta}{2}\|\boldsymbol{y}_k-\boldsymbol{H}\bar{\boldsymbol{x}}_k\|^2+\frac{1}{2}\bar{\boldsymbol{x}}_k^T\boldsymbol{\Phi}\bar{\boldsymbol{x}}_k&=\frac{\beta}{2}\left(\boldsymbol{y}_k^T\boldsymbol{y}_k-2\bar{\boldsymbol{x}}_k^T\boldsymbol{H}^T\boldsymbol{y}_k+\bar{\boldsymbol{x}}_k^T\boldsymbol{H}^T\boldsymbol{H}\bar{\boldsymbol{x}}_k\right)+\frac{1}{2}\bar{\boldsymbol{x}}_k^T\boldsymbol{\Phi}\bar{\boldsymbol{x}}_k\\&=\frac{1}{2}\left[\beta\boldsymbol{y}_k^T\boldsymbol{y}_k-2\bar{\boldsymbol{x}}_k^T\beta\boldsymbol{H}^T\boldsymbol{y}_k+\bar{\boldsymbol{x}}_k^T(\boldsymbol{\Phi}+\beta\boldsymbol{H}^T\boldsymbol{H})\bar{\boldsymbol{x}}_k\right]\\&=\frac{1}{2}\left[\beta\boldsymbol{y}_k^T\boldsymbol{y}_k-2\bar{\boldsymbol{x}}_k^T\beta\boldsymbol{H}^T\boldsymbol{y}_k+\bar{\boldsymbol{x}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k\right].\end{aligned} \tag{4.98}$$

Using the relationship $\beta\boldsymbol{H}^T\boldsymbol{y}_k=\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k$, we have

$$\begin{aligned}\frac{1}{2}\left[\beta\boldsymbol{y}_k^T\boldsymbol{y}_k-2\bar{\boldsymbol{x}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k+\bar{\boldsymbol{x}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k\right]&=\frac{1}{2}\left[\beta\boldsymbol{y}_k^T\boldsymbol{y}_k-\bar{\boldsymbol{x}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k\right]\\&=\frac{1}{2}\left[\beta\boldsymbol{y}_k^T\boldsymbol{y}_k-(\beta\boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T\boldsymbol{y}_k)^T\boldsymbol{\Gamma}(\beta\boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T\boldsymbol{y}_k)\right]\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2} \boldsymbol{y}_k^T \left[\beta \boldsymbol{I} - \beta \boldsymbol{H} \boldsymbol{\Gamma}^{-1} \boldsymbol{H}^T \beta \right] \boldsymbol{y}_k \\
&= \frac{1}{2} \boldsymbol{y}_k^T \left[\beta \boldsymbol{I} - \beta \boldsymbol{H} (\boldsymbol{\Phi} + \beta \boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \beta \right] \boldsymbol{y}_k \\
&= \frac{1}{2} \boldsymbol{y}_k^T \left[\beta^{-1} \boldsymbol{I} + \boldsymbol{H} \boldsymbol{\Phi}^{-1} \boldsymbol{H}^T \right]^{-1} \boldsymbol{y}_k. \qquad (4.99)
\end{aligned}
$$

In the equation above, we use the matrix inversion formula in Eq. (C.91). Using the model covariance matrix $\boldsymbol{\Sigma}_y$ defined in Eq. (4.25), we get

$$
\Delta = \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \left[\beta^{-1} \boldsymbol{I} + \boldsymbol{H} \boldsymbol{\Phi}^{-1} \boldsymbol{H}^T \right]^{-1} \boldsymbol{y}_k = \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k.
$$

The above equation is equal to Eq. (4.29).

4.10.3 Proof of Eq. (4.50)

The proof of Eq. (4.50) begins with

$$
\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}_k}{\operatorname{argmin}} \left[\beta \| \boldsymbol{y}_k - \boldsymbol{H} \boldsymbol{x}_k \|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k \right]. \qquad (4.100)
$$

The solution of this minimization, $\widehat{\boldsymbol{x}}_k$, is known as the weighted minimum-norm solution. To derive it, we define the cost function $\mathcal{F}$,

$$
\mathcal{F} = \beta \| \boldsymbol{y}_k - \boldsymbol{H} \boldsymbol{x}_k \|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k.
$$

Let us differentiate $\mathcal{F}$ with respect to $\boldsymbol{x}_k$, and set it to zero,

$$
\frac{\partial}{\partial \boldsymbol{x}_k} \mathcal{F} = -2\beta \boldsymbol{H}^T \left(\boldsymbol{y}_k - \boldsymbol{H} \boldsymbol{x}_k \right) + 2 \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k = 0.
$$

Thus, the weighted minimum-norm solution is given by

$$
\widehat{\boldsymbol{x}}_k = \beta \left(\boldsymbol{\Upsilon}^{-1} + \beta \boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{y}_k. \qquad (4.101)
$$

The above $\widehat{\boldsymbol{x}}_k$ is exactly the same as the posterior mean $\bar{\boldsymbol{x}}_k$ in Eq. (4.14). Therefore, according to the arguments in the preceding subsection, we have the relationship,

$$
\begin{aligned}
\min_{\boldsymbol{x}_k} \left[\beta \| \boldsymbol{y}_k - \boldsymbol{H} \boldsymbol{x}_k \|^2 + \boldsymbol{x}_k^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{x}_k \right] &= \left[\beta \| \boldsymbol{y}_k - \boldsymbol{H} \bar{\boldsymbol{x}}_k \|^2 + \bar{\boldsymbol{x}}_k^T \boldsymbol{\Upsilon}^{-1} \bar{\boldsymbol{x}}_k \right] \\
&= \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \qquad (4.102)
\end{aligned}
$$

References

1. D.P. Wipf, J.P. Owen, H.T. Attias, K. Sekihara, S.S. Nagarajan, Robust Bayesian estimation of the location, orientation, and time course of multiple correlated neural sources using MEG. NeuroImage **49**, 641–655 (2010)
2. D. Wipf, S. Nagarajan, A unified Bayesian framework for MEG/EEG source imaging. NeuroImage **44**(3), 947–966 (2009)
3. M.E. Tipping, Sparse Bayesian learning and relevance vector machine. J. Mach. Learn. Res. **1**, 211–244 (2001)
4. M.E. Tipping, Bayesian inference: an introduction to principles and practice in machine learning, in *Advanced Lectures on Machine Learning* (Springer, New York, 2004), pp. 41–62
5. D.J.C. MacKay, Bayesian interpolation. Neural Comput. **4**, 415–447 (1992)
6. M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models. Mach. Learn. **37**, 183–233 (1999)
7. D.P. Wipf, S.S. Nagarajan, A new view of automatic relevance determination, in *Advances in Neural Information Processing Systems*, pp. 1625–1632 (2008)

Chapter 5
Bayesian Factor Analysis: A Versatile Framework for Denoising, Interference Suppression, and Source Localization

5.1 Introduction

This chapter describes Bayesian factor analysis (BFA), which is a technique that can decompose multiple sensor time courses into time courses of independent factor activities, where the number of factors is much smaller than the number of sensors. The factors are artificially-introduced "abstract" causes that explain the temporal behavior of the sensor data, and do not necessarily correspond to physical sources. Since the factor activities cannot directly be observed, they are considered latent variables.

The Bayesian factor analysis is a versatile framework. It can be used for selectively extracting signal components from noise-contaminated sensor data. It can provide an estimate of the data covariance matrix better than a sample covariance matrix, and such covariance estimate can be used in source reconstruction algorithms such as adaptive beamformers. It is extended to suppress interference components in interference-overlapped sensor data [1]. It is also extended to a virtual-sensor type source localization method, which is called the Saketini algorithm [2].

We start this chapter by explaining the basic form of the Bayesian factor analysis and then extend it to the variational Bayesian factor analysis (VBFA) [3] in which the model order (the number of factors) is determined by the algorithm itself. Following VBFA arguments, we describe the extensions for interference suppression and source localization.

5.2 Bayesian Factor Analysis

5.2.1 Factor Analysis Model

As in the previous chapters, let us define the output of the mth sensor at time t as $y_m(t)$, and the data vector as $\boldsymbol{y}(t)$, such that

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_5

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_M(t) \end{bmatrix}, \tag{5.1}$$

where M is the total number of sensors. This column vector $\boldsymbol{y}(t)$ expresses the outputs of the whole sensor array. We assume that $\boldsymbol{y}(t)$ is measured at discrete time points $t = t_1, \ldots, t_K$ where K is the total number of time points, and $\boldsymbol{y}(t_k)$ is denoted $\boldsymbol{y}_k$ for simplicity. In the factor analysis model, the data vector $\boldsymbol{y}_k$ is expressed as

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{u}_k + \varepsilon, \tag{5.2}$$

where ε is the sensor noise. On the right-hand side of the equation above, the first term, $\boldsymbol{A}\boldsymbol{u}_k$, expresses a signal component which is a product of an unknown matrix $\boldsymbol{A}$ and unknown variables $\boldsymbol{u}_k$. The matrix $\boldsymbol{A}$ is an $M \times L$ matrix, referred to as the mixing matrix, and $\boldsymbol{u}_k$ is an L-dimensional column vector,

$$\boldsymbol{u}_k = \begin{bmatrix} u_1(t_k) \\ \vdots \\ u_L(t_k) \end{bmatrix}, \tag{5.3}$$

where the number of factors L is much smaller than the number of sensors M; i.e., $L \ll M$. In the factor analysis model, the temporal variation of the sensor data $\boldsymbol{y}_k$ is explained by the temporal variation of a small number of L independent factors, $u_1(t_k), \ldots, u_L(t_k)$. Here, since the number of factors L specifies the model complexity, this number is called the model order. The factor is a latent variable, which cannot be directly observed, and each factor, in general, does not correspond to any physical source. The Bayesian factor analysis estimates the mixing matrix $\boldsymbol{A}$ and the factor activity $\boldsymbol{u}_k$ from the data $\boldsymbol{y}_k$ where $k = 1, \ldots, K$.

5.2.2 Probability Model

The prior probability distribution of the factor $\boldsymbol{u}_k$ is assumed to be the zero-mean Gaussian with its precision matrix equal to the identity matrix, i.e.,

$$p(\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{u}_k|\boldsymbol{0}, \boldsymbol{I}). \tag{5.4}$$

The factor activity is assumed to be independent across time. Thus, the joint prior probability distribution for all time points is given by

$$p(\boldsymbol{u}) = p(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) = \prod_{k=1}^{K} p(\boldsymbol{u}_k) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{u}_k|\boldsymbol{0}, \boldsymbol{I}), \tag{5.5}$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ are collectively denoted $\boldsymbol{u}$. The noise ε is assumed to be Gaussian with the mean of zero, i.e.,

$$p(\varepsilon) = \mathcal{N}(\varepsilon|\boldsymbol{0}, \boldsymbol{\Lambda}^{-1}), \tag{5.6}$$

where $\boldsymbol{\Lambda}$ is a diagonal precision matrix. With this assumption, the conditional probability $p(\boldsymbol{y}_k|\boldsymbol{u}_k)$ is expressed as

$$p(\boldsymbol{y}_k|\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{A}\boldsymbol{u}_k, \boldsymbol{\Lambda}^{-1}). \tag{5.7}$$

The noise ε is also assumed to be independent across time. Thus, we have

$$p(\boldsymbol{y}|\boldsymbol{u}) = p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) = \prod_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{u}_k) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{A}\boldsymbol{u}_k, \boldsymbol{\Lambda}^{-1}), \tag{5.8}$$

where $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K$ are collectively denoted $\boldsymbol{y}$. Using the probability distributions defined above, the Bayesian factor analysis can factorize the sensor data into L independent factor activity and additive sensor noise. This factorization is achieved using the EM algorithm [4, 5]. Explanation of the basics of the EM algorithm is provided in Sect. B.5 in the Appendix.

5.2.3 EM Algorithm

The E-step of the EM algorithm derives the posterior distribution $p(\boldsymbol{u}_k|\boldsymbol{y}_k)$. Derivation of the posterior distribution in the Gaussian model is described in Sect. B.3 in the Appendix. Since the posterior distribution is also Gaussian, we define the posterior distribution $p(\boldsymbol{u}_k|\boldsymbol{y}_k)$ such that

$$p(\boldsymbol{u}_k|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{u}_k|\bar{\boldsymbol{u}}_k, \boldsymbol{\Gamma}^{-1}), \tag{5.9}$$

where $\bar{\boldsymbol{u}}_k$ is the mean and $\boldsymbol{\Gamma}$ is the precision matrix. Using Eqs. (B.24) and (B.25) with setting $\boldsymbol{\Phi}$ to $\boldsymbol{I}$ and $\boldsymbol{H}$ to $\boldsymbol{A}$ in these equations, we get

$$\boldsymbol{\Gamma} = (\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A} + \boldsymbol{I}), \tag{5.10}$$

$$\bar{\boldsymbol{u}}_k = (\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A} + \boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{y}_k. \tag{5.11}$$

The equations above are the E-step update equations in the Bayesian factor analysis.

Let us derive the M-step update equations for the mixing matrix $\boldsymbol{A}$ and the noise precision matrix $\boldsymbol{\Lambda}$. To do so, we derive the average log likelihood, $\Theta(\boldsymbol{A}, \boldsymbol{\Lambda})$, and according to Eqs. (B.34), (5.5), and (5.8), it is expressed as

$$\begin{aligned}\Theta(\boldsymbol{A}, \boldsymbol{\Lambda}) &= E_{\boldsymbol{u}}\left[\log p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)\right] \\ &= E_{\boldsymbol{u}}\left[\log p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K | \boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)\right] + E_{\boldsymbol{u}}\left[\log p(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)\right] \\ &= E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} \log p(\boldsymbol{y}_k | \boldsymbol{u}_k)\right] + E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} \log p(\boldsymbol{u}_k)\right], \end{aligned} \tag{5.12}$$

where $E_{\boldsymbol{u}}[\cdot]$ indicates the expectation with respect to the posterior probability $p(\boldsymbol{u}|\boldsymbol{y})$. Taking a look at Eqs. (5.4) and (5.7), only the first term on the right-hand side of Eq. (5.12) contains $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$. Thus, we have

$$\begin{aligned}\Theta(\boldsymbol{A}, \boldsymbol{\Lambda}) &= E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} \left[\log p(\boldsymbol{y}_k | \boldsymbol{u}_k)\right]\right] + \mathcal{C} \\ &= \frac{K}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)\right] + \mathcal{C}, \end{aligned} \tag{5.13}$$

where $\mathcal{C}$ expresses terms not containing $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$.

The derivative of $\Theta(\boldsymbol{A}, \boldsymbol{\Lambda})$ with respect to $\boldsymbol{A}$ is given by

$$\frac{\partial \Theta(\boldsymbol{A}, \boldsymbol{\Lambda})}{\partial \boldsymbol{A}} = \boldsymbol{\Lambda} E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)\boldsymbol{u}_k^T\right] = \boldsymbol{\Lambda}(\boldsymbol{R}_{yu} - \boldsymbol{A}\boldsymbol{R}_{uu}), \tag{5.14}$$

where

$$\boldsymbol{R}_{uu} = E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} \boldsymbol{u}_k \boldsymbol{u}_k^T\right] = \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T + K\boldsymbol{\Gamma}^{-1}, \tag{5.15}$$

$$\boldsymbol{R}_{yu} = E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} \boldsymbol{y}_k \boldsymbol{u}_k^T\right] = \sum_{k=1}^{K} \boldsymbol{y}_k \bar{\boldsymbol{u}}_k^T, \tag{5.16}$$

$$\boldsymbol{R}_{uy} = \boldsymbol{R}_{yu}^T. \tag{5.17}$$

Setting the right-hand side of Eq. (5.14) to zero gives $\boldsymbol{R}_{yu} = \boldsymbol{A}\boldsymbol{R}_{uu}$. That is, the M-step update equation for $\boldsymbol{A}$ is derived as

$$\boldsymbol{A} = \boldsymbol{R}_{yu} \boldsymbol{R}_{uu}^{-1}. \tag{5.18}$$

Next, the update equation for $\boldsymbol{\Lambda}$ is derived. The partial derivative of $\Theta(\boldsymbol{A}, \boldsymbol{\Lambda})$ with respect to $\boldsymbol{\Lambda}$ is expressed as

$$\frac{\partial \Theta(\boldsymbol{A}, \boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} = \frac{K}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} E_{\boldsymbol{u}}\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T\right]. \tag{5.19}$$

Setting the right-hand side to zero gives

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{K} E_{\boldsymbol{u}} \left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \right]. \tag{5.20}$$

The right-hand side is further changed to

$$\begin{aligned}
&E_{\boldsymbol{u}} \left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \right] \\
&= \sum_{k=1}^{K} E_{\boldsymbol{u}}[\boldsymbol{y}_k \boldsymbol{y}_k^T] - \sum_{k=1}^{K} E_{\boldsymbol{u}}[\boldsymbol{y}_k \boldsymbol{u}_k^T] \boldsymbol{A}^T - \sum_{k=1}^{K} \boldsymbol{A} E_{\boldsymbol{u}}[\boldsymbol{u}_k \boldsymbol{y}_k^T] + \sum_{k=1}^{K} \boldsymbol{A} E_{\boldsymbol{u}}[\boldsymbol{u}_k \boldsymbol{u}_k^T] \boldsymbol{A}^T \\
&= \sum_{k=1}^{K} \boldsymbol{y}_k \boldsymbol{y}_k^T - \sum_{k=1}^{K} \boldsymbol{y}_k \bar{\boldsymbol{u}}_k^T \boldsymbol{A}^T - \sum_{k=1}^{K} \boldsymbol{A} \bar{\boldsymbol{u}}_k \boldsymbol{y}_k^T + \sum_{k=1}^{K} \boldsymbol{A} E_{\boldsymbol{u}}[\boldsymbol{u}_k \boldsymbol{u}_k^T] \boldsymbol{A}^T \\
&= \boldsymbol{R}_{yy} - \boldsymbol{R}_{yu} \boldsymbol{A}^T - \boldsymbol{A} \boldsymbol{R}_{uy} + \boldsymbol{A} \boldsymbol{R}_{uu} \boldsymbol{A}^T,
\end{aligned} \tag{5.21}$$

where $\boldsymbol{R}_{yy} = \sum_{k=1}^{K} \boldsymbol{y}_k \boldsymbol{y}_k^T$. Using $\boldsymbol{R}_{yu} = \boldsymbol{A}\boldsymbol{R}_{uu}$ derived from Eq. (5.18), we get

$$\frac{1}{K} E_{\boldsymbol{u}} \left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \right] = \frac{1}{K} \left[\boldsymbol{R}_{yy} - \boldsymbol{A}\boldsymbol{R}_{uy} \right]. \tag{5.22}$$

Considering that $\boldsymbol{\Lambda}^{-1}$ is a diagonal matrix, we finally derive

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{K} \operatorname{diag}(\boldsymbol{R}_{yy} - \boldsymbol{A}\boldsymbol{R}_{uy}), \tag{5.23}$$

where $\operatorname{diag}(\cdot)$ indicates a diagonal matrix obtained using the diagonal entries of a matrix between the brackets.

5.2.4 Computation of Marginal Likelihood

Let us derive an expression to compute the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$, which can be used for monitoring the progress of the EM algorithm. Here, $\boldsymbol{\theta}$ is used to collectively express the hyperparameters $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$. Using the marginalization, we obtain $p(\boldsymbol{y}|\boldsymbol{\theta})$, such that

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{\theta}) p(\boldsymbol{u}) d\boldsymbol{u}. \tag{5.24}$$

Based on the arguments in Sect. B.4 of the Appendix, $p(\boldsymbol{y}|\boldsymbol{\theta})$ is derived such that

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{\theta}) \quad \text{where} \quad p(\boldsymbol{y}_k|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{0}, \boldsymbol{\Sigma}_y), \tag{5.25}$$

where $\boldsymbol{\Sigma}_y$ is the model data covariance. Setting $\boldsymbol{A}$ to $\boldsymbol{H}$ and $\boldsymbol{I}$ to $\boldsymbol{\Phi}^{-1}$ in Eq. (B.30), we derive $\boldsymbol{\Sigma}_y$, such that

$$\boldsymbol{\Sigma}_y = \boldsymbol{\Lambda}^{-1} + \boldsymbol{A}\boldsymbol{A}^T. \tag{5.26}$$

According to Eq. (B.29), the log likelihood function is obtained as

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{\theta}) = \frac{K}{2} \log |\boldsymbol{\Sigma}_y^{-1}| - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k. \tag{5.27}$$

Using the matrix inversion formula in Eq. (C.91), we have

$$\begin{aligned} \boldsymbol{\Sigma}_y^{-1} &= \left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{A}\boldsymbol{A}^T\right)^{-1} \\ &= \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A} + \boldsymbol{I}\right)^{-1} \boldsymbol{A}^T\boldsymbol{\Lambda} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T\boldsymbol{\Lambda}, \end{aligned} \tag{5.28}$$

Using the equation above and Eq. (5.11), we get

$$\boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k = \boldsymbol{y}_k^T \left[\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T\boldsymbol{\Lambda}\right] \boldsymbol{y}_k = \boldsymbol{y}_k^T \boldsymbol{\Lambda} \boldsymbol{y}_k - \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma} \bar{\boldsymbol{u}}_k.$$

Also, since the relationship

$$|\boldsymbol{\Sigma}_y^{-1}| = |\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T\boldsymbol{\Lambda}| = |\boldsymbol{I} - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T||\boldsymbol{\Lambda}| = |\boldsymbol{\Gamma}^{-1}||\boldsymbol{\Lambda}| \tag{5.29}$$

holds,[1] by substituting the above two equations into Eq. (5.27), we finally obtain the expression for the likelihood:

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{\theta}) = \frac{K}{2} \log \frac{|\boldsymbol{\Lambda}|}{|\boldsymbol{\Gamma}|} - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Lambda} \boldsymbol{y}_k + \frac{1}{2} \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma} \bar{\boldsymbol{u}}_k. \tag{5.30}$$

A more informative derivation of Eq. (5.30) uses the free energy expression in Eq. (B.63) from the Appendix, which claims the following relationship holds:

[1] Defining $\boldsymbol{C} = \boldsymbol{I} - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T$, we have

$$\boldsymbol{A}^T\boldsymbol{C} = \boldsymbol{A}^T - \boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T = \boldsymbol{A}^T - (\boldsymbol{\Gamma} - \boldsymbol{I})\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T = \boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T.$$

Taking the determinant of the equation above, we get $|\boldsymbol{C}| = |\boldsymbol{\Gamma}^{-1}|$.

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{\theta}) &= \int d\boldsymbol{u}\, p(\boldsymbol{u}|\boldsymbol{y})[\log p(\boldsymbol{u}, \boldsymbol{y}|\boldsymbol{\theta}) - \log p(\boldsymbol{u}|\boldsymbol{y})] \\ &= E_{\boldsymbol{u}}\left[\log p(\boldsymbol{u}, \boldsymbol{y}|\boldsymbol{\theta})\right] + \mathcal{H}\left[p(\boldsymbol{u}|\boldsymbol{y})\right], \end{aligned} \tag{5.31}$$

where the second term on the right-hand side is the entropy regarding $p(\boldsymbol{u}|\boldsymbol{y})$. Using Eqs. (5.9) and (C.9) in the Appendix, we get

$$\mathcal{H}\left[p(\boldsymbol{u}|\boldsymbol{y})\right] = \sum_{k=1}^{K} \mathcal{H}\left[p(\boldsymbol{u}_k|\boldsymbol{y}_k)\right] = -\frac{K}{2}\log|\boldsymbol{\Gamma}|. \tag{5.32}$$

Using Eqs. (5.5) and (5.8), the first term in Eq. (5.31) can be rewritten as

$$\begin{aligned} E_{\boldsymbol{u}}\left[\log p(\boldsymbol{u}, \boldsymbol{y}|\boldsymbol{\theta})\right] &= E_{\boldsymbol{u}}\left[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{\theta}) + \log p(\boldsymbol{u})\right] \\ &= -\frac{1}{2}\sum_{k=1}^{K} E_{\boldsymbol{u}}\left[(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) + \boldsymbol{u}_k^T \boldsymbol{u}_k\right] \\ &\quad + \frac{K}{2}\log|\boldsymbol{\Lambda}|, \end{aligned} \tag{5.33}$$

where constant terms are omitted. On the right-hand side of the equation above,

$$\begin{aligned} E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T \boldsymbol{A}^T \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{u}_k\right] &= E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T \left(\boldsymbol{A}^T \boldsymbol{\Lambda}\boldsymbol{A} + \boldsymbol{I}\right)\boldsymbol{u}_k\right] = E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T \boldsymbol{\Gamma}\boldsymbol{u}_k\right] \\ &= E_{\boldsymbol{u}}\left[\mathrm{tr}\left(\boldsymbol{u}_k \boldsymbol{u}_k^T \boldsymbol{\Gamma}\right)\right] = \mathrm{tr}\left[E_{\boldsymbol{u}}\left(\boldsymbol{u}_k \boldsymbol{u}_k^T\right)\boldsymbol{\Gamma}\right] \\ &= \mathrm{tr}\left[(\bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T + \boldsymbol{\Gamma}^{-1})\boldsymbol{\Gamma}\right] = \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k, \end{aligned} \tag{5.34}$$

where a constant term is again omitted. Also, we can show that

$$\begin{aligned} E_{\boldsymbol{u}}\left[\boldsymbol{y}_k^T \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{A}^T \boldsymbol{\Lambda}\boldsymbol{y}_k\right] &= \boldsymbol{y}_k^T \boldsymbol{\Lambda}\boldsymbol{A}\bar{\boldsymbol{u}}_k + \bar{\boldsymbol{u}}_k^T \boldsymbol{A}^T \boldsymbol{\Lambda}\boldsymbol{y}_k \\ &= \boldsymbol{y}_k^T \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k + \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T \boldsymbol{\Lambda}\boldsymbol{y}_k = 2\bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k, \end{aligned} \tag{5.35}$$

where Eq. (5.11) is used. Substituting the above four equations into Eq. (5.31), we can get Eq. (5.30).

5.2.5 *Summary of the BFA Algorithm*

Let us summarize the BFA algorithm. The algorithm is based on the factor analysis model presented in Eq. (5.2). The algorithm estimates the factor activity $\boldsymbol{u}_k$, the mixing matrix $\boldsymbol{A}$, and the sensor-noise precision matrix $\boldsymbol{\Lambda}$. In the estimation, first

the model order L is set according to some prior knowledge, and appropriate initial values are given to $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$. In the E-step, $\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{u}}_k$ $(k = 1, \ldots, L)$ are updated according to Eqs. (5.10) and (5.11). In the M-step, parameters $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ are updated using Eqs. (5.18) and (5.23). The marginal likelihood $p(\boldsymbol{y}|\boldsymbol{A}, \boldsymbol{\Lambda})$ in Eq. (5.30) can be used for monitoring the progress of the EM iteration. If the likelihood increase becomes very small with respect to the iteration count, the EM iteration may be stopped.

The BFA can be applied to denoising the sensor data. Using the BFA algorithm, the estimate of the signal component, $\widehat{\boldsymbol{y}}_k^S$, is given by

$$\widehat{\boldsymbol{y}}_k^S = E_{\boldsymbol{u}}[\boldsymbol{A}\boldsymbol{u}_k] = \boldsymbol{A}E_{\boldsymbol{u}}[\boldsymbol{u}_k] = \boldsymbol{A}\bar{\boldsymbol{u}}_k. \tag{5.36}$$

This $\widehat{\boldsymbol{y}}_k^S$ may be used for further analysis such as source localization. We can compute the sample data covariance using only the signal component $\boldsymbol{A}\boldsymbol{u}_k$, which is

$$\begin{aligned}\bar{\boldsymbol{R}} &= \frac{1}{K}\sum_{k=1}^{K} E_{\boldsymbol{u}}\left[(\boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{A}\boldsymbol{u}_k)^T\right] = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{A}E_{\boldsymbol{u}}\left[\boldsymbol{u}_k\boldsymbol{u}_k^T\right]\boldsymbol{A}^T \\ &= \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{A}\left[\bar{\boldsymbol{u}}_k\bar{\boldsymbol{u}}_k^T + \boldsymbol{\Gamma}^{-1}\right]\boldsymbol{A}^T = \boldsymbol{A}\left[\frac{1}{K}\sum_{k=1}^{K}\bar{\boldsymbol{u}}_k\bar{\boldsymbol{u}}_k^T\right]\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T. \end{aligned} \tag{5.37}$$

This $\bar{\boldsymbol{R}}$ can be used in source imaging algorithms such as adaptive beamformers. Note that, on the right-hand side of Eq. (5.37), the second term $\boldsymbol{A}\boldsymbol{\Gamma}^{-1}\boldsymbol{A}^T$ works as a regularization term and $\bar{\boldsymbol{R}}$ has a form in which the regularization is already incorporated.

5.3 Variational Bayes Factor Analysis (VBFA)

5.3.1 Prior Distribution for Mixing Matrix

In the BFA algorithm described in the preceding section, a user must determine the model order L, according to prior knowledge of the measurement. However, determination of the model order is not easy in most practical applications. We describe here an extension of the Bayesian factor analysis based on the variational Bayesian method [6], The method is called the variational Bayesian factor analysis (VBFA), in which the model order determination is embedded in the algorithm. On the basis of the factor analysis model in Eq. (5.2), the VBFA algorithm estimates the posterior probability distributions not only for the factor activity $\boldsymbol{u}_k$ but also for the mixing matrix $\boldsymbol{A}$.

For convenience in the following arguments, we define the jth row of the mixing matrix $\boldsymbol{A}$ as the column vector $\boldsymbol{a}_j$, i.e., the $M \times L$ matrix $\boldsymbol{A}$ is expressed as

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & \dots & A_{1,L} \\ A_{2,1} & \dots & A_{2,L} \\ \vdots & \ddots & \vdots \\ A_{M,1} & \dots & A_{M,L} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_M^T \end{bmatrix}, \tag{5.38}$$

where

$$\boldsymbol{a}_j = \left[A_{j,1}, \dots, A_{j,L}\right]^T. \tag{5.39}$$

The prior distribution of $\boldsymbol{a}_j$ is assumed to be[2]

$$p(\boldsymbol{a}_j) = \mathcal{N}(\boldsymbol{a}_j | \boldsymbol{0}, (\lambda_j \boldsymbol{\alpha})^{-1}), \tag{5.40}$$

where the jth diagonal element of the noise precision matrix $\boldsymbol{\Lambda}$ is denoted λ_j, and $\boldsymbol{\alpha}$ is a diagonal matrix given by

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_L \end{bmatrix}. \tag{5.41}$$

Here, note that we assume that elements of $\boldsymbol{a}_j$ are statistically independent because the precision matrix in Eq. (5.40) is diagonal. We further assume that $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are statistically independent when $i \neq j$. Thus, the prior probability distribution for the whole mixing matrix $\boldsymbol{A}$ is expressed as

$$p(\boldsymbol{A}) = \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{a}_j | \boldsymbol{0}, (\lambda_j \boldsymbol{\alpha})^{-1}). \tag{5.42}$$

This equation is equivalent to

$$p(\boldsymbol{A}) = \prod_{j=1}^{M} \prod_{\ell=1}^{L} \mathcal{N}(A_{j,\ell} | 0, (\lambda_j \alpha_\ell)^{-1}). \tag{5.43}$$

[2] In the prior distribution in Eq. (5.40), the precision matrix has a form of a diagonal matrix $\boldsymbol{\alpha}$ multiplied by a scalar λ_j. The inclusion of this scalar, λ_j, is just for convenience in the mathematical expressions for the update equations of $\boldsymbol{\Psi}$ and $\bar{\boldsymbol{a}}_j$. The inclusion of λ_j actually makes these update equations significantly simpler.

Since $p(\boldsymbol{A})$ is a so-called conjugate prior, the posterior probability distribution has the form of the Gaussian distribution:

$$p(\boldsymbol{a}_j|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{a}_j|\bar{\boldsymbol{a}}_j, (\lambda_j \boldsymbol{\Psi})^{-1}),$$

and

$$p(\boldsymbol{A}|\boldsymbol{y}) = p(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M|\boldsymbol{y}) = \prod_{j=1}^{M} p(\boldsymbol{a}_j|\boldsymbol{y}) = \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{a}_j|\bar{\boldsymbol{a}}_j, (\lambda_j \boldsymbol{\Psi})^{-1}), \quad (5.44)$$

where $\bar{\boldsymbol{a}}_j$ and $\lambda_j \boldsymbol{\Psi}$ are the mean and precision matrix of the posterior distribution. Namely, the posterior distribution $p(\boldsymbol{A}|\boldsymbol{y})$ has a form identical to the prior distribution $p(\boldsymbol{A})$ with the diagonal $\boldsymbol{\alpha}$ replaced by the non-diagonal $\boldsymbol{\Psi}$. We define for later use the matrix $\bar{\boldsymbol{A}}$, such that

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \bar{\boldsymbol{a}}_1^T \\ \bar{\boldsymbol{a}}_2^T \\ \vdots \\ \bar{\boldsymbol{a}}_M^T \end{bmatrix} = \begin{bmatrix} \bar{A}_{1,1} & \ldots & \bar{A}_{1,L} \\ \bar{A}_{2,1} & \ldots & \bar{A}_{2,L} \\ \vdots & \ddots & \vdots \\ \bar{A}_{M,1} & \ldots & \bar{A}_{M,L} \end{bmatrix}. \quad (5.45)$$

In the VBFA algorithm, an overspecified model order L is used, i.e., the value of L is set greater than the true model order L_0. The posterior mean of $\boldsymbol{a}_j$, $\bar{\boldsymbol{a}}_j$:

$$\bar{\boldsymbol{a}}_j = \left[\bar{A}_{j,1}, \ldots, \bar{A}_{j,L_0}, \bar{A}_{j,L_0+1}, \ldots, \bar{A}_{j,L}\right]^T$$

is the Bayes estimate of the jth row of the mixing matrix. In this estimated mixing matrix, the matrix elements $\bar{A}_{j,L_0+1}, \ldots, \bar{A}_{j,L}$ are those corresponding to non-existing factors. In the VBFA algorithm, those elements are estimated to be significantly small, and the influence of the overspecified factors are automatically eliminated in the final estimation results.

5.3.2 Variational Bayes EM Algorithm (VBEM)

5.3.2.1 E-Step

We follow the arguments in Sect. B.6 in the Appendix, and derive the variational Bayes EM algorithm. The posterior distribution $p(\boldsymbol{u}, \boldsymbol{A}|\boldsymbol{y})$ is approximated by

$$p(\boldsymbol{u}, \boldsymbol{A}|\boldsymbol{y}) = p(\boldsymbol{u}|\boldsymbol{y})p(\boldsymbol{A}|\boldsymbol{y}), \quad (5.46)$$

The E-step of the VBEM algorithm is a step that estimates $p(\boldsymbol{u}|\boldsymbol{y})$, and according to the arguments in Sect. B.6, the estimate of the posterior distribution, $\widehat{p}(\boldsymbol{u}|\boldsymbol{y})$, is obtained as

$$\log \widehat{p}(\boldsymbol{u}|\boldsymbol{y}) = E_{\boldsymbol{A}}\left[\log p(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{A})\right], \tag{5.47}$$

where $E_{\boldsymbol{A}}[\cdot]$ indicates the expectation with respect to $p(\boldsymbol{A}|\boldsymbol{y})$. To obtain $\widehat{p}(\boldsymbol{u}|\boldsymbol{y})$, we substitute

$$p(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{A}) = p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A})p(\boldsymbol{u}, \boldsymbol{A}) = p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A})p(\boldsymbol{u})p(\boldsymbol{A}), \tag{5.48}$$

into Eq. (5.47). Neglecting constant terms, we can derive,

$$\begin{aligned}\log \widehat{p}(\boldsymbol{u}|\boldsymbol{y}) &= E_{\boldsymbol{A}}\left[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A}) + \log p(\boldsymbol{u}) + \log p(\boldsymbol{A})\right] \\ &= E_{\boldsymbol{A}}\left[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A}) + \log p(\boldsymbol{u})\right] \\ &= \sum_{k=1}^{K} E_{\boldsymbol{A}}\left[\log p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{A}) + \log p(\boldsymbol{u}_k)\right].\end{aligned} \tag{5.49}$$

In the equation above, the term $\log p(\boldsymbol{A})$ is omitted because it does not contain $\boldsymbol{u}$.

Since we have assumed that the prior and the noise probability distributions are independent across time, we have the independence of the posterior with respect to time, i.e.,

$$\widehat{p}(\boldsymbol{u}|\boldsymbol{y}) = \prod_{k=1}^{K} \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k). \tag{5.50}$$

Using Eqs. (5.49) and (5.50), we obtain

$$\log \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k) = E_{\boldsymbol{A}}\left[\log p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{A}) + \log p(\boldsymbol{u}_k)\right]. \tag{5.51}$$

Substituting Eqs. (5.4) and (5.7) into (5.51), we get

$$\log \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k) = E_{\boldsymbol{A}}\left[-\frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) - \frac{1}{2}\boldsymbol{u}_k^T \boldsymbol{u}_k\right]. \tag{5.52}$$

Since the posterior $\widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k)$ is Gaussian, we assume

$$\widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{u}_k|\bar{\boldsymbol{u}}_k, \boldsymbol{\Gamma}^{-1}), \tag{5.53}$$

where $\bar{\boldsymbol{u}}_k$ and $\boldsymbol{\Gamma}$ are the mean and the precision of this posterior distribution.

The E-step of the VBEM algorithm estimates $\widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k)$, i.e., it estimates $\bar{\boldsymbol{u}}_k$, and $\boldsymbol{\Gamma}$. For this estimation, we compute the derivative $\frac{\partial}{\partial \boldsymbol{u}_k} \log \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k)$. Using Eq. (5.52), the derivative is given by

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{u}_k} \log \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k) &= E_{\boldsymbol{A}}\left[\boldsymbol{A}^T \boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)\right] - \boldsymbol{u}_k \\ &= \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{y}_k - E_{\boldsymbol{A}}\left[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}\right] \boldsymbol{u}_k - \boldsymbol{u}_k, \end{aligned} \tag{5.54}$$

where $E_{\boldsymbol{A}}(\boldsymbol{A}) = \bar{\boldsymbol{A}}$, i.e., $\bar{\boldsymbol{A}}$ (defined in Eq. (5.45)) is the mean of the posterior $p(\boldsymbol{A}|\boldsymbol{y})$. Let us compute $E_{\boldsymbol{A}}\left[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}\right]$ contained in the right-hand side of Eq. (5.54). Noting that the diagonal elements of $\boldsymbol{\Lambda}$ are denoted $\lambda_1, \ldots, \lambda_M$, $\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}$ is rewritten as

$$\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M] \begin{bmatrix} \lambda_1 & \ldots & 0 \\ & \ddots & \\ 0 & \ldots & \lambda_M \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_1^T \\ \vdots \\ \boldsymbol{a}_M^T \end{bmatrix} = \sum_{j=1}^{M} \lambda_j \boldsymbol{a}_j \boldsymbol{a}_j^T,$$

giving

$$E_{\boldsymbol{A}}\left[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}\right] = \sum_{j=1}^{M} \lambda_j E_{\boldsymbol{A}}\left[\boldsymbol{a}_j \boldsymbol{a}_j^T\right]. \tag{5.55}$$

Since the precision matrix of the posterior distribution of $\boldsymbol{a}_j$ is $\lambda_j \boldsymbol{\Psi}$, we have

$$E_{\boldsymbol{A}}\left[\boldsymbol{a}_j \boldsymbol{a}_j^T\right] = \bar{\boldsymbol{a}}_j \bar{\boldsymbol{a}}_j^T + \frac{1}{\lambda_j} \boldsymbol{\Psi}^{-1}.$$

Therefore, the relationship

$$\begin{aligned} E_{\boldsymbol{A}}\left[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}\right] &= \sum_{j=1}^{M} \lambda_j \left(\bar{\boldsymbol{a}}_j \bar{\boldsymbol{a}}_j^T + \frac{1}{\lambda_j} \boldsymbol{\Psi}^{-1}\right) \\ &= \sum_{j=1}^{M} \lambda_j \bar{\boldsymbol{a}}_j \bar{\boldsymbol{a}}_j^T + M \boldsymbol{\Psi}^{-1} = \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M \boldsymbol{\Psi}^{-1} \end{aligned} \tag{5.56}$$

holds.

Substituting Eq. (5.56) into (5.54), we get

$$\frac{\partial}{\partial \boldsymbol{u}_k} \log \widehat{p}(\boldsymbol{u}_k|\boldsymbol{y}_k) = \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{y}_k - \left[\bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M \boldsymbol{\Psi}^{-1} + \boldsymbol{I}\right] \boldsymbol{u}_k. \tag{5.57}$$

The posterior precision matrix $\boldsymbol{\Gamma}$ is obtained as the coefficient of $\boldsymbol{u}_k$ in the right-hand side of Eq. (5.57), i.e., $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} = \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M\boldsymbol{\Psi}^{-1} + \boldsymbol{I}. \tag{5.58}$$

The mean $\bar{\boldsymbol{u}}_k$ is obtained as the value of $\boldsymbol{u}_k$ that makes the right-hand side of Eq. (5.57) equal to zero. That is, we have

$$\bar{\boldsymbol{u}}_k = \left[\bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M\boldsymbol{\Psi}^{-1} + \boldsymbol{I}\right]^{-1} \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{y}_k. \tag{5.59}$$

5.3.2.2 M-Step

The M-step of the VBEM algorithm estimates $p(\boldsymbol{A}|\boldsymbol{y})$. According to the arguments in Sect. B.6 in the Appendix, the estimate of the posterior distribution, $\widehat{p}(\boldsymbol{A}|\boldsymbol{y})$, is obtained as

$$\log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) = E_{\boldsymbol{u}}\left[\log p(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{A})\right], \tag{5.60}$$

where $E_{\boldsymbol{u}}[\cdot]$ indicates the expectation with respect to the posterior distribution, $p(\boldsymbol{u}|\boldsymbol{y})$. To obtain $\widehat{p}(\boldsymbol{A}|\boldsymbol{y})$, we substitute Eq. (5.48) into (5.60). Neglecting terms (such as $\log p(\boldsymbol{u})$) that do not contain $\boldsymbol{A}$, we derive,

$$\log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) = E_{\boldsymbol{u}}\left[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A}) + \log p(\boldsymbol{A})\right]. \tag{5.61}$$

Substituting Eqs. (5.8) and (5.42) into (5.61), omitting terms not containing $\boldsymbol{A}$, we get

$$\log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) = \sum_{k=1}^{K} E_{\boldsymbol{u}}\left[-\frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)\right] - \frac{1}{2}\sum_{j=1}^{M} \lambda_j \boldsymbol{a}_j^T \boldsymbol{\alpha} \boldsymbol{a}_j. \tag{5.62}$$

As in Eq. (5.44), the posterior $p(\boldsymbol{a}_j|\boldsymbol{y})$ is Gaussian with the mean $\bar{\boldsymbol{a}}_j$ and the precision $\lambda_j \boldsymbol{\Psi}$. The precision $\lambda_j \boldsymbol{\Psi}$ can be obtained by the coefficient of $\boldsymbol{a}_j$ in the derivative $\frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})$, and $\bar{\boldsymbol{a}}_j$ can be obtained as $\boldsymbol{a}_j$ that makes this derivative equal to zero. To compute the derivative, we first rewrite:

$$\sum_{j=1}^{M} \lambda_j \boldsymbol{a}_j^T \boldsymbol{\alpha} \boldsymbol{a}_j = \sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2, \tag{5.63}$$

and we can find

$$\frac{\partial}{\partial A_{j,\ell}} \frac{1}{2} \sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2 = \lambda_j \alpha_\ell A_{j,\ell} = [\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\alpha}]_{j,\ell}. \tag{5.64}$$

Therefore, we have

$$\frac{\partial}{\partial \boldsymbol{A}} \frac{1}{2} \sum_{j=1}^{M} \lambda_j \boldsymbol{a}_j^T \boldsymbol{\alpha} \boldsymbol{a}_j = \boldsymbol{\Lambda} \boldsymbol{A} \boldsymbol{\alpha}. \tag{5.65}$$

We also have

$$\frac{\partial}{\partial \boldsymbol{A}} \sum_{k=1}^{K} -\frac{1}{2} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) = \boldsymbol{\Lambda} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) \boldsymbol{u}_k^T. \tag{5.66}$$

Consequently, we can derive

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) &= E_{\boldsymbol{u}} \left[\boldsymbol{\Lambda} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) \boldsymbol{u}_k^T \right] - \boldsymbol{\Lambda} \boldsymbol{A} \boldsymbol{\alpha} \\ &= \boldsymbol{\Lambda} \boldsymbol{R}_{yu} - \boldsymbol{\Lambda} \boldsymbol{A} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}). \end{aligned} \tag{5.67}$$

Setting the right-hand side of the equation above to zero, we get

$$\bar{\boldsymbol{A}} = \boldsymbol{R}_{yu} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha})^{-1}. \tag{5.68}$$

where $\boldsymbol{R}_{uu}$ and $\boldsymbol{R}_{yu}$ are defined in Eqs. (5.15) and (5.16). The precision $\lambda_j \boldsymbol{\Psi}$ is obtained as the coefficient of $\boldsymbol{a}_j$ in the right-hand side of Eq. (5.67). The second term in the right-hand side of this equation can be rewritten as

$$\boldsymbol{\Lambda} \boldsymbol{A} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}) = \begin{bmatrix} \lambda_1 \boldsymbol{a}_1^T \\ \vdots \\ \lambda_M \boldsymbol{a}_M^T \end{bmatrix} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}) = \begin{bmatrix} \lambda_1 \boldsymbol{a}_1^T (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}) \\ \vdots \\ \lambda_M \boldsymbol{a}_M^T (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}) \end{bmatrix}. \tag{5.69}$$

Thus, $\boldsymbol{\Psi}$ is obtained as

$$\boldsymbol{\Psi} = \boldsymbol{R}_{uu} + \boldsymbol{\alpha}. \tag{5.70}$$

Equations (5.68) and (5.70) are the M-step update equations in the VBFA algorithm. However, to compute these equations, we need to know the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$. The next subsection deals with the estimation of $\boldsymbol{\alpha}$.

5.3.2.3 Update Equation for Hyperparameter α

The hyperparameter $\boldsymbol{\alpha}$ is estimated by maximizing the free energy.[3] According to the arguments in Sect. B.6, the free energy is expressed as

[3] An estimate that maximizes the free energy is the MAP estimate under the assumption of the non-informative prior for the hyperparameter.

$$\begin{aligned}\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}] &= E_{(\boldsymbol{A},\boldsymbol{u})}\left[\log p(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{A}) - \log \widehat{p}(\boldsymbol{u}|\boldsymbol{y}) - \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})\right] \\ &= E_{(\boldsymbol{A},\boldsymbol{u})}[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A}) + \log p(\boldsymbol{u}) \\ &\qquad + \log p(\boldsymbol{A}) - \log \widehat{p}(\boldsymbol{u}|\boldsymbol{y}) - \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})], \end{aligned} \tag{5.71}$$

where $E_{(\boldsymbol{A},\boldsymbol{u})}[\,\cdot\,]$ indicates the average regarding both $\widehat{p}(\boldsymbol{A}|\boldsymbol{y})$ and $\widehat{p}(\boldsymbol{u}|\boldsymbol{y})$. On the right-most-side of the equation above, the terms containing $\boldsymbol{\alpha}$ are only $\log p(\boldsymbol{A})$. Thus, omitting the terms not containing $\boldsymbol{\alpha}$, we can rewrite the free energy as

$$\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}] = E_{(\boldsymbol{A},\boldsymbol{u})}\left[\log p(\boldsymbol{A})\right]. \tag{5.72}$$

Considering

$$\begin{aligned}\log p(\boldsymbol{A}) &= \log \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{a}_j|\boldsymbol{0}, \lambda_j \boldsymbol{\alpha}) = \frac{1}{2}\sum_{j=1}^{M} \log|\lambda_j \boldsymbol{\alpha}| - \frac{1}{2}\sum_{j=1}^{M} \boldsymbol{a}_j^T \lambda_j \boldsymbol{\alpha} \boldsymbol{a}_j \\ &= \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \log(\lambda_j \alpha_\ell) - \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2, \end{aligned} \tag{5.73}$$

The free energy in Eq. (5.72) is rewritten as

$$\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}] = E_{(\boldsymbol{A},\boldsymbol{u})}\left[\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \log(\lambda_j \alpha_\ell) - \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2\right]. \tag{5.74}$$

To derive the estimate of $\boldsymbol{\alpha}$, we differentiate the free energy in (5.74) with respect to $\boldsymbol{\alpha}$, resulting in

$$\frac{\partial}{\partial \boldsymbol{\alpha}}\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}] = E_{\boldsymbol{A}}\left[\frac{\partial}{\partial \boldsymbol{\alpha}}\left[\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \log(\lambda_j \alpha_\ell) - \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2\right]\right], \tag{5.75}$$

where the expectation $E_{\boldsymbol{u}}[\,\cdot\,]$ is omitted because the terms on the right-hand side of the equation above do not contain $\boldsymbol{u}$. To compute the equation above, we consider the relationship

$$\frac{\partial}{\partial \alpha_\ell}\left[\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \log(\lambda_j \alpha_\ell) - \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2\right] = \frac{M}{2}\alpha_\ell^{-1} - \frac{1}{2}\sum_{j=1}^{M} \lambda_j A_{j,\ell}^2, \tag{5.76}$$

and obtain

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\alpha}}\left[\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\log(\lambda_j\alpha_\ell)-\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\lambda_j\alpha_\ell A_{j,\ell}^2\right]\\
&\quad=\frac{M}{2}\boldsymbol{\alpha}^{-1}-\frac{1}{2}\begin{bmatrix}\sum_{j=1}^{M}\lambda_j A_{j,1}^2 & 0 & \dots & 0\\ 0 & \sum_{j=1}^{M}\lambda_j A_{j,2}^2 & \dots & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & \dots & \sum_{j=1}^{M}\lambda_j A_{j,L}^2\end{bmatrix}\\
&\quad=\frac{1}{2}(M\boldsymbol{\alpha}^{-1}-\mathrm{diag}[\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{A}^T]),
\end{aligned}\tag{5.77}$$

where diag[·] indicates a diagonal matrix whose diagonal entries are equal to those of a matrix in the brackets.[4] The relationship

$$E_{\boldsymbol{A}}\left[\sum_{j=1}^{M}\lambda_j A_{j,\ell}^2\right]=\sum_{j=1}^{M}\lambda_j\bar{A}_{j,\ell}^2+\sum_{j=1}^{M}\lambda_j\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{\ell,\ell}=\sum_{j=1}^{M}\lambda_j\bar{A}_{j,\ell}^2+M[\boldsymbol{\Psi}^{-1}]_{\ell,\ell}$$

also holds. (Note that Eq. (5.168) shows the general case of computing $E_{\boldsymbol{A}}[A_{i,k}A_{j,\ell}]$.) Thus, we finally obtain

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\alpha}}\mathcal{F}[\boldsymbol{\alpha},\boldsymbol{\Lambda}]&=E_{\boldsymbol{A}}\left[\frac{\partial}{\partial\boldsymbol{\alpha}}\left[\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\log(\lambda_j\alpha_\ell)-\frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\lambda_j\alpha_\ell A_{j,\ell}^2\right]\right]\\
&=\frac{1}{2}M\boldsymbol{\alpha}^{-1}-\frac{1}{2}\begin{bmatrix}\sum_{j=1}^{M}\lambda_j\bar{A}_{j,1}^2 & 0 & \dots & 0\\ 0 & \sum_{j=1}^{M}\lambda_j\bar{A}_{j,2}^2 & \dots & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & \dots & \sum_{j=1}^{M}\lambda_j\bar{A}_{j,L}^2\end{bmatrix}\\
&\quad-\frac{1}{2}M\begin{bmatrix}[\boldsymbol{\Psi}^{-1}]_{1,1} & 0 & \dots & 0\\ 0 & [\boldsymbol{\Psi}^{-1}]_{2,2} & \dots & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & \dots & [\boldsymbol{\Psi}^{-1}]_{L,L}\end{bmatrix}\\
&=\frac{1}{2}(M\boldsymbol{\alpha}^{-1}-\mathrm{diag}[\bar{\boldsymbol{A}}^T\boldsymbol{\Lambda}\bar{\boldsymbol{A}}]-M\,\mathrm{diag}[\boldsymbol{\Psi}^{-1}]).
\end{aligned}\tag{5.78}$$

[4] Here computing the derivative of a scalar X with a diagonal matrix $\boldsymbol{A}$ is equal to creating a diagonal matrix whose (j,j)th diagonal element is equal to $\partial X/\partial A_{j,j}$ where $A_{j,j}$ is the (j,j)th diagonal element of $\boldsymbol{A}$.

Therefore, setting the right-hand side of the equation above to zero, we have the update equation for $\boldsymbol{\alpha}$, such that,

$$\boldsymbol{\alpha}^{-1} = \text{diag}\left[\frac{1}{M}\bar{\boldsymbol{A}}^T \boldsymbol{\Lambda}\bar{\boldsymbol{A}} + \boldsymbol{\Psi}^{-1}\right]. \tag{5.79}$$

5.3.2.4 Update Equation for the Noise Precision $\boldsymbol{\Lambda}$

We next derive the update equation for $\boldsymbol{\Lambda}$. To do so, we maximize the free energy in Eq. (5.71) with respect to $\boldsymbol{\Lambda}$. On the right-hand-side of Eq. (5.71), the terms containing $\boldsymbol{\Lambda}$ are $\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A})$ and $\log p(\boldsymbol{A})$. Thus, omitting the terms not containing $\boldsymbol{\Lambda}$, the free energy is expressed as

$$\begin{aligned}\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}] &= E_{(\boldsymbol{A},\boldsymbol{u})}\left[\log p(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{A}) + \log p(\boldsymbol{A})\right] \\ &= E_{(\boldsymbol{A},\boldsymbol{u})}\left[\frac{K}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)\right. \\ &\qquad \left. + \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\log(\lambda_j\alpha_\ell) - \frac{1}{2}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\lambda_j\alpha_\ell A_{j,\ell}^2\right].\end{aligned} \tag{5.80}$$

To maximize $\mathcal{F}[\boldsymbol{\alpha}, \boldsymbol{\Lambda}]$ with respect to $\boldsymbol{\Lambda}$, we consider the relationship

$$\begin{aligned}&\frac{1}{\partial\lambda_j}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\left[\frac{1}{2}\log(\lambda_j\alpha_\ell) - \frac{1}{2}\lambda_j\alpha_\ell A_{j,\ell}^2\right] \\ &\quad = \sum_{\ell=1}^{L}\left[\frac{1}{2}\frac{1}{\lambda_j} - \frac{1}{2}\alpha_\ell A_{j,\ell}^2\right] = \frac{L}{2}\frac{1}{\lambda_j} - \frac{1}{2}\sum_{\ell=1}^{L}\alpha_\ell A_{j,\ell}^2,\end{aligned}$$

and obtain

$$\begin{aligned}&\frac{1}{\partial\boldsymbol{\Lambda}}\sum_{j=1}^{M}\sum_{\ell=1}^{L}\left[\frac{1}{2}\log(\lambda_j\alpha_\ell) - \frac{1}{2}\lambda_j\alpha_\ell A_{j,\ell}^2\right] \\ &\quad = \frac{L}{2}\begin{bmatrix} 1/\lambda_1 & 0 & \dots & 0 \\ 0 & 1/\lambda_2 & \dots & 0 \\ 0 & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 1/\lambda_M \end{bmatrix} \\ &\qquad - \frac{1}{2}\begin{bmatrix} \sum_{\ell=1}^{L}\alpha_\ell A_{1,\ell}^2 & 0 & \dots & 0 \\ 0 & \sum_{\ell=1}^{L}\alpha_\ell A_{2,\ell}^2 & \dots & 0 \\ 0 & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sum_{\ell=1}^{L}\alpha_\ell A_{M,\ell}^2 \end{bmatrix}\end{aligned}$$

$$= \frac{L}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}\,\mathrm{diag}[\boldsymbol{A}\boldsymbol{\alpha}\boldsymbol{A}^T]. \tag{5.81}$$

Then, using Eq. (5.80), we derive the derivative of $\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha})$ with respect to $\boldsymbol{\Lambda}$ such that

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\Lambda}}\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}) \\
&= E_{(\boldsymbol{A},\boldsymbol{u})}\left[\frac{K}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T + \frac{L}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}\boldsymbol{A}\boldsymbol{\alpha}\boldsymbol{A}^T\right] \\
&= \frac{K}{2}\boldsymbol{\Lambda}^{-1} + \frac{1}{2}E_{\boldsymbol{A}}[-\boldsymbol{R}_{yy} + \boldsymbol{R}_{yu}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{R}_{uy} - \boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A}^T + L\boldsymbol{\Lambda}^{-1} - \boldsymbol{A}\boldsymbol{\alpha}\boldsymbol{A}^T] \\
&= \frac{K}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}(\boldsymbol{R}_{yy} - \boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T - \bar{\boldsymbol{A}}\boldsymbol{R}_{uy}) + \frac{L}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\alpha}\boldsymbol{A}^T].
\end{aligned} \tag{5.82}$$

The following relationship holds:

$$E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\alpha}\boldsymbol{A}^T] = E_{\boldsymbol{A}}[\boldsymbol{A}(\boldsymbol{R}_{uu} + \boldsymbol{\alpha})\boldsymbol{A}^T] = E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T] \tag{5.83}$$

Also, we can prove the relationship

$$E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T] = \bar{\boldsymbol{A}}\boldsymbol{\Psi}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1} = \boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1}. \tag{5.84}$$

The proof of this equation is provided in Sect. 5.7.1. Thus, the relationship

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\Lambda}}\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}) &= \frac{K}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}(\boldsymbol{R}_{yy} - \boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T - \bar{\boldsymbol{A}}\boldsymbol{R}_{uy}) \\
&\quad + \frac{L}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}(\boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1}) = \frac{K}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}(\boldsymbol{R}_{yy} - \bar{\boldsymbol{A}}\boldsymbol{R}_{uy})
\end{aligned} \tag{5.85}$$

holds. Setting the right-hand side of the equation above zero, we obtain the update equation for $\boldsymbol{\Lambda}$ as

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{K}\,\mathrm{diag}(\boldsymbol{R}_{yy} - \bar{\boldsymbol{A}}\boldsymbol{R}_{uy}). \tag{5.86}$$

5.3.3 Computation of Free Energy

Although we cannot compute the exact marginal likelihood in the VBEM algorithm, we can compute its lower bound using the free energy. The free energy after the E-step of the VBEM algorithm is expressed as

$$\begin{aligned}\mathcal{F} &= E_{(\boldsymbol{A},\boldsymbol{u})}[\log p(\boldsymbol{y},\boldsymbol{u},\boldsymbol{A}) - \log \widehat{p}(\boldsymbol{u}|\boldsymbol{y}) - \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})] \\ &= E_{(\boldsymbol{A},\boldsymbol{u})}\left[\log p(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{A}) + \log p(\boldsymbol{u}) + \log p(\boldsymbol{A})\right] \\ &\quad + \mathcal{H}(\widehat{p}(\boldsymbol{u}|\boldsymbol{y})) + \mathcal{H}(\widehat{p}(\boldsymbol{A}|\boldsymbol{y})).\end{aligned} \tag{5.87}$$

Substituting

$$\log p(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{A}) = \frac{K}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k), \tag{5.88}$$

$$\log p(\boldsymbol{u}) = -\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{u}_k\boldsymbol{u}_k^T, \tag{5.89}$$

$$\log p(\boldsymbol{A}) = \frac{1}{2}\sum_{j=1}^{M}\log|\lambda_j\boldsymbol{\alpha}| - \frac{1}{2}\sum_{j=1}^{M}\boldsymbol{a}_j^T\lambda_j\boldsymbol{\alpha}\boldsymbol{a}_j, \tag{5.90}$$

and

$$\mathcal{H}(\widehat{p}(\boldsymbol{u}|\boldsymbol{y})) = -\frac{K}{2}\log|\boldsymbol{\Gamma}|, \tag{5.91}$$

$$\mathcal{H}(\widehat{p}(\boldsymbol{A}|\boldsymbol{y})) = -\frac{1}{2}\sum_{j=1}^{M}\log|\lambda_j\boldsymbol{\Psi}|, \tag{5.92}$$

into Eq. (5.87), we obtain

$$\begin{aligned}\mathcal{F} = \frac{K}{2}\log|\boldsymbol{\Lambda}| + E_{(\boldsymbol{A},\boldsymbol{u})}\left[-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) - \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{u}_k\boldsymbol{u}_k^T\right] \\ + \frac{1}{2}\sum_{j=1}^{M}\log|\lambda_j\boldsymbol{\alpha}| - \frac{1}{2}E_{\boldsymbol{A}}\left[\sum_{j=1}^{M}\boldsymbol{a}_j^T\lambda_j\boldsymbol{\alpha}\boldsymbol{a}_j\right] - \frac{K}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\sum_{j=1}^{M}\log|\lambda_j\boldsymbol{\Psi}|.\end{aligned} \tag{5.93}$$

Note that the following relationships:

$$\begin{aligned}&E_{(\boldsymbol{A},\boldsymbol{u})}\left[-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k) - \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{u}_k\boldsymbol{u}_k^T\right] \\ &\quad = -\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k + \frac{1}{2}\sum_{k=1}^{K}\bar{\boldsymbol{u}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k\end{aligned} \tag{5.94}$$

and

$$\sum_{j=1}^{M} \log |\lambda_j \boldsymbol{\alpha}| - \sum_{j=1}^{M} \log |\lambda_j \boldsymbol{\Psi}| = \sum_{j=1}^{M} \log \frac{\lambda_j^M |\boldsymbol{\alpha}|}{\lambda_j^M |\boldsymbol{\Psi}|} = M \log \frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\Psi}|} \tag{5.95}$$

hold. The proof of Eq. (5.94) is presented in Sect. 5.7.2. Also, we can show

$$\begin{aligned} E_{\boldsymbol{A}} \left[\sum_{j=1}^{M} \boldsymbol{a}_j^T \lambda_j \boldsymbol{\alpha} \boldsymbol{a}_j \right] &= E_{\boldsymbol{A}} \left[\sum_{j=1}^{M} \sum_{\ell=1}^{L} \lambda_j \alpha_\ell A_{j,\ell}^2 \right] \\ &= \sum_{j=1}^{M} \sum_{\ell=1}^{L} \lambda_j \alpha_\ell \bar{A}_{j,\ell}^2 + \sum_{j=1}^{M} \sum_{\ell=1}^{L} \lambda_j \alpha_\ell \frac{1}{\lambda_j} [\boldsymbol{\Psi}^{-1}]_{\ell,\ell} \\ &= \sum_{j=1}^{M} \sum_{\ell=1}^{L} \lambda_j \alpha_\ell \bar{A}_{j,\ell}^2 + M \sum_{\ell=1}^{L} \alpha_\ell [\boldsymbol{\Psi}^{-1}]_{\ell,\ell} \\ &= \sum_{j=1}^{M} \bar{\boldsymbol{a}}_j^T \lambda_j \boldsymbol{\alpha} \bar{\boldsymbol{a}}_j + M \operatorname{tr} \left(\boldsymbol{\alpha} \boldsymbol{\Psi}^{-1} \right). \end{aligned} \tag{5.96}$$

By substituting the above three equations into Eq. (5.93), we obtain the expression for the free energy:

$$\begin{aligned} \mathcal{F} = \frac{K}{2} \log \frac{|\boldsymbol{\Lambda}|}{|\boldsymbol{\Gamma}|} - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Lambda} \boldsymbol{y}_k + \frac{1}{2} \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma} \bar{\boldsymbol{u}}_k \\ + \frac{M}{2} \log \frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\Psi}|} - \frac{1}{2} \sum_{j=1}^{M} \bar{\boldsymbol{a}}_j^T \lambda_j \boldsymbol{\alpha} \bar{\boldsymbol{a}}_j - \frac{M}{2} \operatorname{tr} \left(\boldsymbol{\alpha} \boldsymbol{\Psi}^{-1} \right). \end{aligned} \tag{5.97}$$

Note here that

$$\sum_{j=1}^{M} \bar{\boldsymbol{a}}_j^T \lambda_j \boldsymbol{\alpha} \bar{\boldsymbol{a}}_j = \operatorname{tr} \left[\bar{\boldsymbol{A}} \boldsymbol{\alpha} \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \right] = \operatorname{tr} \left[\boldsymbol{\alpha} \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} \right]. \tag{5.98}$$

Thus, we have

$$\begin{aligned} \sum_{j=1}^{M} \bar{\boldsymbol{a}}_j^T \lambda_j \boldsymbol{\alpha} \bar{\boldsymbol{a}}_j + M \operatorname{tr} \left(\boldsymbol{\alpha} \boldsymbol{\Psi}^{-1} \right) &= \operatorname{tr} \left[\boldsymbol{\alpha} \left(\bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M \boldsymbol{\Psi}^{-1} \right) \right] \\ &= \operatorname{tr} \left[\boldsymbol{\alpha} M \boldsymbol{\alpha}^{-1} \right] = ML. \end{aligned} \tag{5.99}$$

In the equation above, we use the update equation for $\boldsymbol{\alpha}$ in Eq. (5.79). Therefore, ignoring constant terms, the free energy is finally expressed as

$$\mathcal{F} = \frac{K}{2} \log \frac{|\boldsymbol{\Lambda}|}{|\boldsymbol{\Gamma}|} - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{y}_k^T \boldsymbol{\Lambda} \boldsymbol{y}_k + \frac{1}{2} \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k^T \boldsymbol{\Gamma} \bar{\boldsymbol{u}}_k + \frac{M}{2} \log \frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\Psi}|}. \tag{5.100}$$

Since the free energy is limited (i.e., upper bounded) by the likelihood $\log p(\boldsymbol{y}|\boldsymbol{\theta})$ (where $\boldsymbol{\theta}$ collectively expresses the hyperparameters), increasing the free energy increases the likelihood.

5.3.4 Summary of the VBFA Algorithm

Let us summarize the VBFA algorithm. The algorithm is based on the factor analysis model in Eq. (5.2). The algorithm estimates the factor activity $\boldsymbol{u}_k$, the mixing matrix $\boldsymbol{A}$, and the sensor-noise precision matrix $\boldsymbol{\Lambda}$. In the estimation, an overspecified value is set for the model order L, and appropriate initial values are set for $\bar{\boldsymbol{A}}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Lambda}$. In the E-step, $\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{u}}_k$ $(k = 1, \ldots, K)$ are updated according to Eqs. (5.58) and (5.59). In the M-step, values of $\bar{\boldsymbol{A}}$, and $\boldsymbol{\Psi}$ are updated using Eqs. (5.68), and (5.70).

The hyperparameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$ are updated using Eqs. (5.79) and (5.86). Since we cannot compute the marginal likelihood, the free energy in Eq. (5.100)—which is the lower limit of the marginal likelihood—is used for monitoring the progress of the VBEM iteration. That is, if the increase of $\mathcal{F}$ in Eq. (5.100) becomes very small with respect to the iteration count, the VBEM iteration can be terminated.

Using the VBFA algorithm, the estimate of the signal component in the sensor data, $\widehat{\boldsymbol{y}}_k^S$, is given by

$$\widehat{\boldsymbol{y}}_k^S = E_{(\boldsymbol{A},\boldsymbol{u})}[\boldsymbol{A}\boldsymbol{u}] = E_{\boldsymbol{A}}[\boldsymbol{A}] E_{\boldsymbol{u}}[\boldsymbol{u}] = \bar{\boldsymbol{A}} \bar{\boldsymbol{u}}_k. \tag{5.101}$$

In the equation above, we use the values of $\bar{\boldsymbol{u}}_k$ and $\bar{\boldsymbol{A}}$ obtained when the VBEM iteration is terminated. The sample covariance matrix can be computed using only the signal component, and such a covariance matrix is derived as

$$\begin{aligned} \bar{\boldsymbol{R}} &= \frac{1}{K} \sum_{k=1}^{K} E_{(\boldsymbol{u},\boldsymbol{A})} \left[(\boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{A}\boldsymbol{u}_k)^T \right] = \frac{1}{K} \sum_{k=1}^{K} E_{\boldsymbol{u}} E_{\boldsymbol{A}} \left[\left[\boldsymbol{A}\boldsymbol{u}_k \boldsymbol{u}_k^T \boldsymbol{A}^T \right] \right] \\ &= \frac{1}{K} \sum_{k=1}^{K} E_{\boldsymbol{A}} \left[\boldsymbol{A} E_{\boldsymbol{u}} \left[\boldsymbol{u}_k \boldsymbol{u}_k^T \right] \boldsymbol{A}^T \right] = \frac{1}{K} E_{\boldsymbol{A}} \left[\boldsymbol{A} \boldsymbol{R}_{uu} \boldsymbol{A}^T \right], \end{aligned} \tag{5.102}$$

where $\boldsymbol{R}_{uu}$ is defined in Eq. (5.15). We can show

$$E_{\boldsymbol{A}} \left[\boldsymbol{A} \boldsymbol{R}_{uu} \boldsymbol{A}^T \right] = \bar{\boldsymbol{A}} \boldsymbol{R}_{uu} \bar{\boldsymbol{A}}^T + \boldsymbol{\Lambda}^{-1} \operatorname{tr} \left(\boldsymbol{R}_{uu} \boldsymbol{\Psi}^{-1} \right), \tag{5.103}$$

where the proof is presented in Sect. 5.7.3. Using Eq. (5.103), we have

$$\bar{\boldsymbol{R}} = \frac{1}{K}\bar{\boldsymbol{A}}\boldsymbol{R}_{uu}\bar{\boldsymbol{A}}^T + \frac{1}{K}\boldsymbol{\Lambda}^{-1}\,\mathrm{tr}\left(\boldsymbol{R}_{uu}\boldsymbol{\Psi}^{-1}\right). \tag{5.104}$$

This $\bar{\boldsymbol{R}}$ can be used in source imaging algorithms such as the adaptive beamformers.

5.4 Partitioned Factor Analysis (PFA)

5.4.1 Factor Analysis Model

The bioelectromagnetic data is often contaminated not only by sensor noise but also by various types of interference of biological and nonbiological origins. A simple modification of the VBFA algorithm enables the removal of such interferences. The modified algorithm is called the partitioned factor analysis (PFA) [1]. The prerequisite for the PFA algorithm is that a control measurement, which contains the interferences but does not contain the signal of interest, be available. The factor analysis model for PFA is expressed as

$$\boldsymbol{y}_k = \boldsymbol{B}\boldsymbol{v}_k + \varepsilon \qquad \text{for control data,} \tag{5.105}$$

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{B}\boldsymbol{v}_k + \varepsilon \quad \text{for target data.} \tag{5.106}$$

In the equations above, $L \times 1$ column vector $\boldsymbol{u}_k$ is the factor activity that represents the signal of interest and $\boldsymbol{A}$ is an $M \times L$ mixing matrix. Also, $L_v \times 1$ column vector $\boldsymbol{v}_k$ is the factor activity that represents the interference and $\boldsymbol{B}$ is an $M \times L_v$ interference mixing matrix.

The PFA algorithm has a two-step procedure. The first step applies the VBFA algorithm to the control data and estimates the mixing matrix $\boldsymbol{B}$ and the sensor-noise precision $\boldsymbol{\Lambda}$. The second step estimates the mixing matrix $\boldsymbol{A}$ and the signal factor activity $\boldsymbol{u}_k$. In the second step, the interference mixing matrix $\boldsymbol{B}$ and the noise precision $\boldsymbol{\Lambda}$ are fixed, and the VBFA algorithm is applied to the target data, which is expressed as

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{B}\boldsymbol{v}_k + \varepsilon = [\boldsymbol{A},\ \boldsymbol{B}]\begin{bmatrix}\boldsymbol{u}_k \\ \boldsymbol{v}_k\end{bmatrix} + \varepsilon = \boldsymbol{A}_c\boldsymbol{z}_k + \varepsilon, \tag{5.107}$$

where

$$\boldsymbol{A}_c = [\boldsymbol{A}, \boldsymbol{B}] \quad \text{and} \quad \boldsymbol{z}_k = \begin{bmatrix}\boldsymbol{u}_k \\ \boldsymbol{v}_k\end{bmatrix}.$$

The equation above indicates that the second step can also be expressed by the factor analysis model using the augmented factor vector $\boldsymbol{z}_k$ and the augmented mixing matrix $\boldsymbol{A}_c$.

5.4.2 Probability Model

The prior probability distributions for the factor vectors are:

$$P(\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{u}_k|\mathbf{0}, \boldsymbol{I}), \tag{5.108}$$
$$P(\boldsymbol{v}_k) = \mathcal{N}(\boldsymbol{v}_k|\mathbf{0}, \boldsymbol{I}). \tag{5.109}$$

Thus, we have the prior probability distribution for the augmented factor vector $\boldsymbol{z}_k$, such that

$$\begin{aligned} p(\boldsymbol{z}_k) &= p(\boldsymbol{u}_k)p(\boldsymbol{v}_k) = \mathcal{N}(\boldsymbol{u}_k|\mathbf{0}, \boldsymbol{I})\mathcal{N}(\boldsymbol{v}_\mathbf{k}|\mathbf{0}, \boldsymbol{I}) \\ &= \left|\frac{\boldsymbol{I}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}\boldsymbol{u}_k^T\boldsymbol{u}_k\right] \left|\frac{\boldsymbol{I}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}\boldsymbol{v}_k^T\boldsymbol{v}_k\right] \\ &= \left|\frac{\boldsymbol{I}}{2\pi}\right|^{1/2} \exp\left[-\frac{1}{2}\boldsymbol{z}_k^T\boldsymbol{z}_k\right] = \mathcal{N}(\boldsymbol{z}_k|\mathbf{0}, \boldsymbol{I}). \end{aligned} \tag{5.110}$$

The noise is assumed to be Gaussian,

$$P(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0}, \boldsymbol{\Lambda}^{-1}).$$

Therefore, we have the conditional probability, $P(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{v}_k)$, such that

$$P(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{v}_k) = P(\boldsymbol{y}_k|\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{B}\boldsymbol{v}_k, \boldsymbol{\Lambda}^{-1}) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{A}_c\boldsymbol{z}_k, \boldsymbol{\Lambda}^{-1}). \tag{5.111}$$

We assume the same prior for $\boldsymbol{A}$ as in Eq. (5.42).

5.4.3 VBEM Algorithm for PFA

5.4.3.1 E-Step

The E-step estimates the posterior probability $\widehat{p}(\boldsymbol{z}_k|\boldsymbol{y}_k)$. Using the same arguments in Sect. 5.3.2.1, we have,

$$\begin{aligned} \log \widehat{p}(\boldsymbol{z}_k|\boldsymbol{y}_k) &= E_{\boldsymbol{A}}\left[\log p(\boldsymbol{z}_k, \boldsymbol{y}_k, \boldsymbol{A}_c)\right] \\ &= E_{\boldsymbol{A}}\left[\log p(\boldsymbol{y}_k|\boldsymbol{z}_k, \boldsymbol{A}_c) + \log p(\boldsymbol{z}_k) + \log p(\boldsymbol{A}_c)\right]. \end{aligned} \tag{5.112}$$

Note that, since $\boldsymbol{B}$ is fixed, $p(\boldsymbol{A}_c)$ is the same as $p(\boldsymbol{A})$, which is expressed in Eq. (5.42). Omitting the terms not containing $\boldsymbol{z}_k$, and using Eqs. (5.110) and (5.111), we obtain

$$\log \widehat{p}(\boldsymbol{z}_k|\boldsymbol{y}_k) = E_{\boldsymbol{A}}\left[-\frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{A}_c\boldsymbol{z}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}_c\boldsymbol{z}_k)\right] - \frac{1}{2}\boldsymbol{z}_k^T\boldsymbol{z}_k. \tag{5.113}$$

Since $\widehat{p}(z_k|\boldsymbol{y}_k)$ is Gaussian, we assume that

$$\widehat{p}(z_k|\boldsymbol{y}_k) = \mathcal{N}(z_k|\bar{z}_k, \boldsymbol{\Gamma}^{-1}).$$

The mean $\bar{z}_k$ is obtained as z_k that makes $\frac{\partial}{\partial z_k} \log \widehat{p}(z_k|\boldsymbol{y}_k)$ equal to zero, and the precision $\boldsymbol{\Gamma}$ is obtained from the coefficient of z_k in this derivative. Using Eq. (5.113), the derivative is given by

$$\begin{aligned}\frac{\partial}{\partial z_k} \log \widehat{p}(z_k|\boldsymbol{y}_k) &= E_{\boldsymbol{A}}[\boldsymbol{A}_c^T \boldsymbol{\Lambda} \left(\boldsymbol{y}_k - \boldsymbol{A}_c z_k\right)] - z_k \\ &= \bar{\boldsymbol{A}}_c^T \boldsymbol{\Lambda} \boldsymbol{y}_k - E_{\boldsymbol{A}}[\boldsymbol{A}_c^T \boldsymbol{\Lambda} \boldsymbol{A}_c] z_k - z_k. \end{aligned} \tag{5.114}$$

In the expression above, using Eq. (5.56), $E_{\boldsymbol{A}}[\boldsymbol{A}_c^T \boldsymbol{\Lambda} \boldsymbol{A}_c]$ is expressed as

$$\begin{aligned} E_{\boldsymbol{A}}[\boldsymbol{A}_c^T \boldsymbol{\Lambda} \boldsymbol{A}_c] &= E_{\boldsymbol{A}}\left[\begin{bmatrix} \boldsymbol{A}^T \\ \boldsymbol{B}^T \end{bmatrix} \boldsymbol{\Lambda} [\boldsymbol{A}^T \ \boldsymbol{B}^T]\right] = \begin{bmatrix} E_{\boldsymbol{A}}[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}] & \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{B} \\ \boldsymbol{B}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} & \boldsymbol{B}^T \boldsymbol{\Lambda} \boldsymbol{B} \end{bmatrix} \\ &= \begin{bmatrix} \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M \boldsymbol{\Psi}^{-1} & \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{B} \\ \boldsymbol{B}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} & \boldsymbol{B}^T \boldsymbol{\Lambda} \boldsymbol{B} \end{bmatrix} = \bar{\boldsymbol{A}}_c^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_c + M \boldsymbol{\Psi}_c^{-1}, \end{aligned} \tag{5.115}$$

where

$$\boldsymbol{\Psi}_c^{-1} = \begin{bmatrix} \boldsymbol{\Psi}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{5.116}$$

and

$$\bar{\boldsymbol{A}}_c = \left[\bar{\boldsymbol{A}}, \boldsymbol{B}\right]. \tag{5.117}$$

Therefore, we finally have

$$\frac{\partial}{\partial z_k} \log \widehat{p}(z_k|\boldsymbol{y}_k) = \bar{\boldsymbol{A}}_c^T \boldsymbol{\Lambda} \boldsymbol{y}_k - (\bar{\boldsymbol{A}}_c^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_c z_k + M \boldsymbol{\Psi}_c^{-1} + \boldsymbol{I}) z_k. \tag{5.118}$$

The precision matrix of the posterior distribution is obtained as

$$\boldsymbol{\Gamma} = \bar{\boldsymbol{A}}_c^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_c + M \boldsymbol{\Psi}_c^{-1} + \boldsymbol{I}, \tag{5.119}$$

and the mean of the posterior as

$$\bar{z}_k = \begin{bmatrix} \bar{\boldsymbol{u}}_k \\ \bar{\boldsymbol{v}}_k \end{bmatrix} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \bar{\boldsymbol{A}}^T \\ \boldsymbol{B}^T \end{bmatrix} \boldsymbol{\Lambda} \boldsymbol{y}_k. \tag{5.120}$$

Equations (5.119) and (5.120) are the E-step update equations in the PFA algorithm.

5.4.3.2 M-Step

The M-step estimates the posterior probability $\widehat{p}(\boldsymbol{A}_c|\boldsymbol{y}_k)$. Using the same arguments as in Sect. 5.3.2.2, we have

$$\begin{aligned}
&\log \widehat{p}(\boldsymbol{A}_c|\boldsymbol{y}) \\
&= E_z\left[\log p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{A}_c)\right] = E_z\left[\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{A}_c) + \log p(\boldsymbol{z}) + \log p(\boldsymbol{A}_c)\right] \\
&= E_z\left[\sum_{k=1}^{K} \log p(\boldsymbol{y}_k|\boldsymbol{z}_k, \boldsymbol{A}_c) + \log p(\boldsymbol{A}_c)\right] \\
&= E_z\left[-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k - \boldsymbol{B}\boldsymbol{v}_k)^T \boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k - \boldsymbol{B}\boldsymbol{v}_k) - \frac{1}{2}\sum_{j=1}^{M} \boldsymbol{a}_j^T \lambda_j \boldsymbol{\alpha}\boldsymbol{a}_j\right],
\end{aligned} \tag{5.121}$$

where $\boldsymbol{z}$ collectively expresses $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$, and terms not containing $\boldsymbol{A}$ are omitted. The form of the posterior distribution in Eq. (5.44) is also assumed:

$$p(\boldsymbol{A}_c|\boldsymbol{y}) = p(\boldsymbol{A}|\boldsymbol{y}) = \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{a}_j|\bar{\boldsymbol{a}}_j, (\lambda_j \boldsymbol{\Psi})^{-1}).$$

The mean $\bar{\boldsymbol{A}}$ is obtained as the $\boldsymbol{A}$ that makes $\frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})$ equal to zero, and the precision $\lambda_j \boldsymbol{\Psi}$ as the coefficient of $\boldsymbol{a}_j$ in that derivative. Using Eqs. (5.65) and (5.66) the derivative is computed as

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}_k) &= E_z\left[\boldsymbol{\Lambda}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{u}_k - \boldsymbol{B}\boldsymbol{v}_k)\boldsymbol{u}_k^T\right] - \boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{\alpha} \\
&= \boldsymbol{\Lambda}\boldsymbol{R}_{yu} - \boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{R}_{vu} - \boldsymbol{\Lambda}\boldsymbol{A}(\boldsymbol{R}_{uu} + \boldsymbol{\alpha}).
\end{aligned} \tag{5.122}$$

In Eq. (5.122), the coefficient of $\boldsymbol{a}_j$ is $\lambda_j(\boldsymbol{R}_{uu} + \boldsymbol{\alpha})$, and thus the matrix $\boldsymbol{\Psi}$ is equal to

$$\boldsymbol{\Psi} = \boldsymbol{R}_{uu} + \boldsymbol{\alpha}, \tag{5.123}$$

and $\bar{\boldsymbol{A}}$ is obtained as

$$\bar{\boldsymbol{A}} = (\boldsymbol{R}_{yu} - \boldsymbol{B}\boldsymbol{R}_{vu})\boldsymbol{\Psi}^{-1}, \tag{5.124}$$

where $\boldsymbol{R}_{yu}$ is obtained as

$$\boldsymbol{R}_{yu} = \sum_{k=1}^{K} \boldsymbol{y}_k \bar{\boldsymbol{u}}_k^T. \tag{5.125}$$

Also, $\boldsymbol{R}_{vu}$ and $\boldsymbol{R}_{uu}$ are obtained in the following manner. We first define

$$\boldsymbol{\Gamma}^{-1} = \begin{bmatrix} \widetilde{\boldsymbol{\Gamma}}_{uu} & \widetilde{\boldsymbol{\Gamma}}_{uv} \\ \widetilde{\boldsymbol{\Gamma}}_{vu} & \widetilde{\boldsymbol{\Gamma}}_{vv} \end{bmatrix}, \tag{5.126}$$

and have

$$\begin{aligned} \begin{bmatrix} \boldsymbol{R}_{uu} & \boldsymbol{R}_{uv} \\ \boldsymbol{R}_{vu} & \boldsymbol{R}_{vv} \end{bmatrix} &= E_z \left[\sum_{k=1}^{K} \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix} [\boldsymbol{u}_k^T, \boldsymbol{v}_k^T] \right] \\ &= \begin{bmatrix} \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T & \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{v}}_k^T \\ \sum_{k=1}^{K} \bar{\boldsymbol{v}}_k \bar{\boldsymbol{u}}_k^T & \sum_{k=1}^{K} \bar{\boldsymbol{v}}_k \bar{\boldsymbol{v}}_k^T \end{bmatrix} + K \begin{bmatrix} \widetilde{\boldsymbol{\Gamma}}_{uu} & \widetilde{\boldsymbol{\Gamma}}_{uv} \\ \widetilde{\boldsymbol{\Gamma}}_{vu} & \widetilde{\boldsymbol{\Gamma}}_{vv} \end{bmatrix}. \end{aligned} \tag{5.127}$$

Thus, we obtain,

$$\boldsymbol{R}_{uu} = \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T + K \widetilde{\boldsymbol{\Gamma}}_{uu}, \tag{5.128}$$

$$\boldsymbol{R}_{vu} = \sum_{k=1}^{K} \bar{\boldsymbol{v}}_k \bar{\boldsymbol{u}}_k^T + K \widetilde{\boldsymbol{\Gamma}}_{vu}. \tag{5.129}$$

The hyperparameter $\boldsymbol{\alpha}$ is obtained by maximizing the free energy, which is expressed as

$$\begin{aligned} \mathcal{F} = E_{(\boldsymbol{A},\boldsymbol{u})}[&\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{A}_c) + \log p(\boldsymbol{z}) + \log p(\boldsymbol{A}_c) \\ &- \log \widehat{p}(\boldsymbol{z}|\boldsymbol{y}) - \log \widehat{p}(\boldsymbol{A}_c|\boldsymbol{y})]. \end{aligned} \tag{5.130}$$

However, since $\boldsymbol{\alpha}$ is contained only in $\log p(\boldsymbol{A}_c)$, (which is equal to $\log p(\boldsymbol{A})$), the update equation for $\boldsymbol{\alpha}$ is exactly the same as that in Eq. (5.79).

5.4.4 Summary of the PFA Algorithm

The PFA algorithm is summarized as follows. The first step estimates the interference mixing matrix $\boldsymbol{B}$ and the diagonal noise precision matrix $\boldsymbol{\Lambda}$ by applying the VBFA algorithm to the control data. The second step applies the PFA-VBEM algorithm to the target data, and estimates the signal factor vector $\boldsymbol{u}_k$ and the signal mixing matrix $\boldsymbol{A}$. In the second step, $\boldsymbol{B}$ and $\boldsymbol{\Lambda}$ are fixed at the values obtained in the first step. There is a different version of the PFA algorithm in which $\boldsymbol{B}$ and $\boldsymbol{\Lambda}$ are also updated in the second step. The details of the algorithm are given in [1].

The free energy is computed in exactly the same manner as in Eq. (5.100) except that $\bar{\boldsymbol{u}}_k$ is replaced by $\bar{\boldsymbol{z}}_k$. Thus, the free energy is expressed as

$$\mathcal{F} = \frac{K}{2}\log\frac{|\boldsymbol{\Lambda}|}{|\boldsymbol{\Gamma}|} - \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k + \frac{1}{2}\sum_{k=1}^{K}\bar{\boldsymbol{z}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{z}}_k + \frac{M}{2}\log\frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\Psi}|}. \tag{5.131}$$

This free energy is used for monitoring the progress of the PFA-VBEM iteration.

Using the PFA algorithm, the estimate of the signal of interest, $\widehat{\boldsymbol{y}}_k^S$, is given by.

$$\widehat{\boldsymbol{y}}_k^S = E_{(\boldsymbol{A},\boldsymbol{u})}[\boldsymbol{A}\boldsymbol{u}_k] = E_{\boldsymbol{A}}[\boldsymbol{A}]E_{\boldsymbol{u}}[\boldsymbol{u}_k] = \bar{\boldsymbol{A}}\bar{\boldsymbol{u}}_k. \tag{5.132}$$

The sample covariance matrix computed only using the signal of interest is given by

$$\bar{\boldsymbol{R}} = \frac{1}{K}\bar{\boldsymbol{A}}\boldsymbol{R}_{uu}\bar{\boldsymbol{A}}^T + \frac{1}{K}\boldsymbol{\Lambda}^{-1}\,\mathrm{tr}\left(\boldsymbol{R}_{uu}\boldsymbol{\Psi}^{-1}\right), \tag{5.133}$$

where $\boldsymbol{R}_{uu}$ is defined in Eq. (5.128). This $\bar{\boldsymbol{R}}$ can be used in source imaging algorithms such as the adaptive beamformers, and resultant images can be free from the influence of interferences.

5.5 Saketini: Source Localization Algorithm Based on the VBFA Model

5.5.1 Data Model

This section describes a virtual-sensor type source localization algorithm, called Saketini [2]. The Saketini algorithm is based on Bayesian factor analysis, and enables the estimation of source activity. In the Saketini algorithm, the sensor data $\boldsymbol{y}(t)$ is modeled using

$$\boldsymbol{y}(t) = \boldsymbol{L}(\boldsymbol{r})\boldsymbol{s}(\boldsymbol{r},t) + \boldsymbol{A}\boldsymbol{u}(t) + \varepsilon, \tag{5.134}$$

where $\boldsymbol{s}(\boldsymbol{r},t)$ is the source vector defined in Eq. (2.3). An $M \times 3$ matrix, $\boldsymbol{L}(\boldsymbol{r})$, is the lead field matrix at $\boldsymbol{r}$, which is defined in Eq. (2.4), and ε is the additive sensor noise. In Eq. (5.134), the signal from the source activity at $\boldsymbol{r}$ is represented by $\boldsymbol{L}(\boldsymbol{r})\boldsymbol{s}(\boldsymbol{r},t)$. On the basis of the factor analysis model, the interference is modeled using $\boldsymbol{A}\boldsymbol{u}(t)$ where $\boldsymbol{A}$ is an $M \times L$ factor mixing matrix, and $\boldsymbol{u}(t)$ is an $L \times 1$ factor vector. Here, $\boldsymbol{A}\boldsymbol{u}(t)$ represents all interference and source activities except the source activity at $\boldsymbol{r}$.

As in the preceding sections, $\boldsymbol{y}(t_k)$, $\boldsymbol{s}(\boldsymbol{r},t_k)$, and $\boldsymbol{u}(t_k)$ are denoted $\boldsymbol{y}_k$, $\boldsymbol{s}_k$, and $\boldsymbol{u}_k$, respectively. $\boldsymbol{L}(\boldsymbol{r})$ is denoted $\boldsymbol{L}$ for simplicity. Then, Eq. (5.134) is rewritten as

$$\boldsymbol{y}_k = \boldsymbol{L}\boldsymbol{s}_k + \boldsymbol{A}\boldsymbol{u}_k + \varepsilon. \tag{5.135}$$

The goal of this algorithm is to estimate $\boldsymbol{s}_k$ from the data $\boldsymbol{y}_k$. Once $\boldsymbol{s}_k$ is obtained, the algorithm pointing location $\boldsymbol{r}$ is scanned over the whole brain region to obtain the spatiotemporal reconstruction of the source activity over the whole brain. To formulate

the estimation problem based on the VBFA algorithm, Eq. (5.135) is rewritten as

$$\boldsymbol{y}_k = \boldsymbol{L}\boldsymbol{s}_k + \boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{\varepsilon} = [\boldsymbol{L}, \boldsymbol{A}] \begin{bmatrix} \boldsymbol{s}_k \\ \boldsymbol{u}_k \end{bmatrix} + \boldsymbol{\varepsilon} = \boldsymbol{A}_s \boldsymbol{z}_k + \boldsymbol{\varepsilon}, \tag{5.136}$$

where $\boldsymbol{z}_k = [\boldsymbol{s}_k^T, \boldsymbol{u}_k^T]^T$, and $\boldsymbol{A}_s = [\boldsymbol{L}, \boldsymbol{A}]$. This equation indicates that the problem of estimating $\boldsymbol{s}_k$ can be expressed by the factor analysis model using the augmented factor vector $\boldsymbol{z}_k$ and the augmented mixing matrix $\boldsymbol{A}_s$. Equation (5.136) is, in principle, the same as Eq. (5.107). Therefore, the algorithm developed here is very similar to the PFA algorithm described in Sect. 5.4.

5.5.2 Probability Model

We assume the prior distribution for $\boldsymbol{s}_k$ to be zero-mean Gaussian, such that

$$p(\boldsymbol{s}_k) = \mathcal{N}(\boldsymbol{s}_k | \boldsymbol{0}, \boldsymbol{\Phi}^{-1}), \tag{5.137}$$

where $\boldsymbol{\Phi}$ is a 3×3 (non-diagonal) precision matrix of this prior distribution. The prior distribution of the factor vector $\boldsymbol{u}_k$ is assumed to be:

$$p(\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{u}_k | \boldsymbol{0}, \boldsymbol{I}). \tag{5.138}$$

Then, the prior distribution for $\boldsymbol{z}_k$ is derived as

$$\begin{aligned} p(\boldsymbol{z}_k) &= p(\boldsymbol{s}_k) p(\boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{s}_k | \boldsymbol{0}, \boldsymbol{\Phi}^{-1}) \mathcal{N}(\boldsymbol{u}_\mathbf{k} | \boldsymbol{0}, \boldsymbol{I}) \\ &= \left| \frac{\boldsymbol{\Phi}}{2\pi} \right|^{1/2} \exp\left[-\frac{1}{2} \boldsymbol{s}_k^T \boldsymbol{\Phi} \boldsymbol{s}_k \right] \left| \frac{\boldsymbol{I}}{2\pi} \right|^{1/2} \exp\left[-\frac{1}{2} \boldsymbol{u}_k^T \boldsymbol{u}_k \right] \\ &= \left| \frac{\widetilde{\boldsymbol{\Phi}}}{2\pi} \right|^{1/2} \exp\left[-\frac{1}{2} \boldsymbol{z}_k^T \widetilde{\boldsymbol{\Phi}} \boldsymbol{z}_k \right] = \mathcal{N}(\boldsymbol{z}_k | \boldsymbol{0}, \widetilde{\boldsymbol{\Phi}}^{-1}), \end{aligned} \tag{5.139}$$

where

$$\widetilde{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}.$$

Equation (5.139) indicates that the prior distribution of $\boldsymbol{z}_k$ is the mean-zero Gaussian with the precision matrix of $\widetilde{\boldsymbol{\Phi}}$. The sensor noise is also assumed to be zero-mean Gaussian with a diagonal precision matrix $\boldsymbol{\Lambda}$. Hence, we have

$$p(\boldsymbol{y}_k | \boldsymbol{s}_k, \boldsymbol{u}_k) = \mathcal{N}(\boldsymbol{y}_k | \boldsymbol{L}\boldsymbol{s}_k + \boldsymbol{A}\boldsymbol{u}_k, \boldsymbol{\Lambda}^{-1}) = \mathcal{N}(\boldsymbol{y}_k | \boldsymbol{A}_s \boldsymbol{z}_k, \boldsymbol{\Lambda}^{-1}). \tag{5.140}$$

We assume the same prior distribution for the mixing matrix $\boldsymbol{A}$, as shown in Eq. (5.42).

5.5.3 VBEM Algorithm

5.5.3.1 E-Step

The posterior $p(z|y)$ is Gaussian, and assumed to be,

$$p(z|y) = \prod_{k=1}^{K} p(z_k|y_k), \quad \text{and} \quad p(z_k|y_k) = \mathcal{N}(z_k|\bar{z}_k, \boldsymbol{\Gamma}^{-1}), \tag{5.141}$$

where $\bar{z}_k$ and $\boldsymbol{\Gamma}$ are, respectively, the mean and precision of $p(z_k|y_k)$. Using similar arguments as for the E-step of the PFA algorithm, the estimate of $p(z_k|y_k)$, $\widehat{p}(z_k|y_k)$, is given by

$$\log \widehat{p}(z_k|y_k) = E_{\boldsymbol{A}}[\log p(y_k|z_k, \boldsymbol{A})] + \log p(z_k)], \tag{5.142}$$

where we omit terms that do not contain z_k. The precision $\boldsymbol{\Gamma}$ of the posterior is derived from the coefficient of $\frac{\partial}{\partial z_k}\widehat{p}(z_k|y_k)$, and the mean $\bar{z}_k$ is derived as the z_k that makes this derivative zero.

Defining $\bar{\boldsymbol{A}}_s = E_{\boldsymbol{A}}([\boldsymbol{L}, \boldsymbol{A}]) = [\boldsymbol{L}, \bar{\boldsymbol{A}}]$, we obtain

$$\begin{aligned}\frac{\partial}{\partial z_k}\log \widehat{p}(z_k|y_k) &= E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda}(y_k - \boldsymbol{A}_s z_k)] - \widetilde{\boldsymbol{\Phi}} z_k \\ &= E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda} y_k] - E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda} \boldsymbol{A}_s] z_k - \widetilde{\boldsymbol{\Phi}} z_k,\end{aligned} \tag{5.143}$$

where

$$E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda} y_k] = \bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} y_k$$

and

$$\begin{aligned}E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda} \boldsymbol{A}_s] &= \begin{bmatrix} E_{\boldsymbol{A}}[\boldsymbol{L}^T \boldsymbol{\Lambda} \boldsymbol{L}] & E_{\boldsymbol{A}}[\boldsymbol{L}^T \boldsymbol{\Lambda} \boldsymbol{A}] \\ E_{\boldsymbol{A}}[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{L}] & E_{\boldsymbol{A}}[\boldsymbol{A}^T \boldsymbol{\Lambda} \boldsymbol{A}] \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{L}^T \boldsymbol{\Lambda} \boldsymbol{L} & \boldsymbol{L}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \boldsymbol{L} & \bar{\boldsymbol{A}}^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}} + M \boldsymbol{\Psi}^{-1} \end{bmatrix} = \bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_s + M \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}^{-1} \end{bmatrix}.\end{aligned} \tag{5.144}$$

Hence, we have

$$E_{\boldsymbol{A}}[\boldsymbol{A}_s^T \boldsymbol{\Lambda} \boldsymbol{A}_s] z_k = (\bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_s + M \boldsymbol{\Psi}_s^{-1}) z_k, \tag{5.145}$$

where

$$\boldsymbol{\Psi}_s^{-1} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}^{-1} \end{bmatrix}. \tag{5.146}$$

Thus, we derive

$$\frac{\partial}{\partial \boldsymbol{z}_k} \log \widehat{p}(\boldsymbol{z}_k|\boldsymbol{y}_k) = \bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} \boldsymbol{y}_k - (\bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_s + M\boldsymbol{\Psi}_s^{-1} + \widetilde{\boldsymbol{\Phi}})\boldsymbol{z}_k. \tag{5.147}$$

Consequently, the coefficient of $\boldsymbol{z}_k$ gives

$$\boldsymbol{\Gamma} = \bar{\boldsymbol{A}}_s^T \boldsymbol{\Lambda} \bar{\boldsymbol{A}}_s + M\boldsymbol{\Psi}_s^{-1} + \widetilde{\boldsymbol{\Phi}}. \tag{5.148}$$

The mean $\bar{\boldsymbol{z}}_k$ is obtained as

$$\bar{\boldsymbol{z}}_k = \begin{bmatrix} \bar{\boldsymbol{s}}_k \\ \bar{\boldsymbol{u}}_k \end{bmatrix} = \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \boldsymbol{L}^T \\ \bar{\boldsymbol{A}}^T \end{bmatrix} \boldsymbol{\Lambda} \boldsymbol{y}_k. \tag{5.149}$$

5.5.3.2 M-Step

The derivation for the M step is also very similar to the M step of the PFA-VBEM algorithm in Sect. 5.4.3.2. The estimation of $\widehat{p}(\boldsymbol{A}|\boldsymbol{y})$ is based on

$$\begin{aligned} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) = &-\frac{1}{2} E_z \left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k) \right] \\ &- \frac{1}{2} \sum_{j=1}^{M} \boldsymbol{a}_j^T \lambda_j \boldsymbol{\alpha} \boldsymbol{a}_j. \end{aligned} \tag{5.150}$$

The derivative $\frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})$ is obtained as,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{A}} \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y}) &= E_z \left[\boldsymbol{\Lambda} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k) \boldsymbol{u}_k^T \right] - \boldsymbol{\Lambda} \boldsymbol{A} \boldsymbol{\alpha} \\ &= \boldsymbol{\Lambda} \boldsymbol{R}_{yu} - \boldsymbol{\Lambda} \boldsymbol{L} \boldsymbol{R}_{su} - \boldsymbol{\Lambda} \boldsymbol{A} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}). \end{aligned} \tag{5.151}$$

The precision, $\lambda_j \boldsymbol{\Psi}$, is equal to the coefficient of $\boldsymbol{a}_j$, and it is obtained as $\lambda_j (\boldsymbol{R}_{uu} + \boldsymbol{\alpha})$. Hence we have

$$\boldsymbol{\Psi} = (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}). \tag{5.152}$$

Setting the derivative to zero gives $\bar{\boldsymbol{A}}$, i.e.,

$$\bar{\boldsymbol{A}} = (\boldsymbol{R}_{yu} - \boldsymbol{L}\boldsymbol{R}_{su})(\boldsymbol{R}_{uu} + \boldsymbol{\alpha})^{-1} = (\boldsymbol{R}_{yu} - \boldsymbol{L}\boldsymbol{R}_{su}) \boldsymbol{\Psi}^{-1}. \tag{5.153}$$

In the equations above, $\boldsymbol{R}_{su}$ and $\boldsymbol{R}_{uu}$ are obtained in the following manner. Defining

$$\boldsymbol{\Gamma}^{-1} = \begin{bmatrix} \widetilde{\boldsymbol{\Gamma}}_{ss} & \widetilde{\boldsymbol{\Gamma}}_{su} \\ \widetilde{\boldsymbol{\Gamma}}_{us} & \widetilde{\boldsymbol{\Gamma}}_{uu} \end{bmatrix}, \tag{5.154}$$

we have

$$\begin{aligned} \begin{bmatrix} \boldsymbol{R}_{ss} & \boldsymbol{R}_{su} \\ \boldsymbol{R}_{us} & \boldsymbol{R}_{uu} \end{bmatrix} &= E_z \left[\sum_{k=1}^{K} \begin{bmatrix} \boldsymbol{s}_k \\ \boldsymbol{u}_k \end{bmatrix} [\boldsymbol{s}_k^T, \boldsymbol{u}_k^T] \right] \\ &= \begin{bmatrix} \sum_{k=1}^{K} \bar{\boldsymbol{s}}_k \bar{\boldsymbol{s}}_k^T & \sum_{k=1}^{K} \bar{\boldsymbol{s}}_k \bar{\boldsymbol{u}}_k^T \\ \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{s}}_k^T & \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T \end{bmatrix} + K \begin{bmatrix} \widetilde{\boldsymbol{\Gamma}}_{ss} & \widetilde{\boldsymbol{\Gamma}}_{su} \\ \widetilde{\boldsymbol{\Gamma}}_{us} & \widetilde{\boldsymbol{\Gamma}}_{uu} \end{bmatrix}. \end{aligned} \tag{5.155}$$

Therefore, we obtain:

$$\boldsymbol{R}_{ss} = \sum_{k=1}^{K} \bar{\boldsymbol{s}}_k \bar{\boldsymbol{s}}_k^T + K \widetilde{\boldsymbol{\Gamma}}_{ss}, \tag{5.156}$$

$$\boldsymbol{R}_{su} = \sum_{k=1}^{K} \bar{\boldsymbol{s}}_k \bar{\boldsymbol{u}}_k^T + K \widetilde{\boldsymbol{\Gamma}}_{su}, \tag{5.157}$$

$$\boldsymbol{R}_{uu} = \sum_{k=1}^{K} \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k^T + K \widetilde{\boldsymbol{\Gamma}}_{uu}. \tag{5.158}$$

Also, $\boldsymbol{R}_{yu}$ is obtained as

$$\boldsymbol{R}_{yu} = \sum_{k=1}^{K} \boldsymbol{y}_k \bar{\boldsymbol{u}}_k^T. \tag{5.159}$$

5.5.3.3 Update Equations for Hyperparameters

The update equations for $\boldsymbol{\Phi}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\alpha}$ are derived by maximizing the free energy, which is rewritten as

$$\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Phi}) = E_{(z,A)}[\log p(\boldsymbol{y}|z, \boldsymbol{A}_s) + \log p(z) + \log p(\boldsymbol{A})], \tag{5.160}$$

where the terms $-\log \widehat{p}(z|\boldsymbol{y}) - \log \widehat{p}(\boldsymbol{A}|\boldsymbol{y})$ are ignored because these terms contain none of these hyperparameters. To compute $\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Phi})$, we use the relationships,

$$\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{A}_s) = \sum_{k=1}^{K} \log p(\boldsymbol{y}_k|\boldsymbol{z}_k, \boldsymbol{A}_s)$$
$$= \frac{K}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k), \tag{5.161}$$

$$\log p(\boldsymbol{z}) = \sum_{k=1}^{K} \log p(\boldsymbol{z}_k) = \frac{K}{2} \log |\widetilde{\boldsymbol{\Phi}}| - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{z}_k^T \widetilde{\boldsymbol{\Phi}} \boldsymbol{z}_k$$
$$= \frac{K}{2} \log |\boldsymbol{\Phi}| - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{s}_k^T \boldsymbol{\Phi} \boldsymbol{s}_k - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{u}_k^T \boldsymbol{u}_k, \tag{5.162}$$

and

$$\log p(\boldsymbol{A}) = \frac{1}{2} \sum_{j=1}^{M} \log |\lambda_j \boldsymbol{\alpha}| - \frac{1}{2} \sum_{j=1}^{M} \boldsymbol{a}_j^T \lambda_j \boldsymbol{\alpha} \boldsymbol{a}_j. \tag{5.163}$$

Let us derive the update equation for $\boldsymbol{\Phi}$. In Eq. (5.160), the only term that contains $\boldsymbol{\Phi}$ is $\log p(\boldsymbol{z})$. Thus, we have

$$\frac{1}{\partial \boldsymbol{\Phi}} \mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Phi}) = \frac{1}{\partial \boldsymbol{\Phi}} \frac{K}{2} \log |\boldsymbol{\Phi}| - \frac{1}{\partial \boldsymbol{\Phi}} \frac{1}{2} E_{(z,A)} \left[\sum_{k=1}^{K} \boldsymbol{s}_k^T \boldsymbol{\Phi} \boldsymbol{s}_k \right]$$
$$= \frac{K}{2} \boldsymbol{\Phi}^{-1} - \frac{1}{2} E_s \left[\sum_{k=1}^{K} \boldsymbol{s}_k \boldsymbol{s}_k^T \right] = \frac{K}{2} \boldsymbol{\Phi}^{-1} - \frac{1}{2} \boldsymbol{R}_{ss}, \tag{5.164}$$

and setting the right-hand side to zero gives the update equation,

$$\boldsymbol{\Phi}^{-1} = \frac{1}{K} \boldsymbol{R}_{ss}, \tag{5.165}$$

where $\boldsymbol{R}_{ss}$ is obtained in Eq. (5.156). We can derive the update equation for $\boldsymbol{\alpha}$ in a similar manner. However, Since in Eq. (5.160), the only term that contains $\boldsymbol{\alpha}$ is $\log p(\boldsymbol{A})$, the update equation for $\boldsymbol{\alpha}$ is exactly the same as that in Eq. (5.79). The update equation for $\boldsymbol{\Lambda}$ is given by

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{K} \text{diag}[\boldsymbol{R}_{yy} - \boldsymbol{R}_{ys} \boldsymbol{L}^T - \boldsymbol{L} \boldsymbol{R}_{sy} + \boldsymbol{L} \boldsymbol{R}_{ss} \boldsymbol{L}^T - \bar{\boldsymbol{A}} \boldsymbol{\Psi} \bar{\boldsymbol{A}}^T]. \tag{5.166}$$

Since the derivation of the equation above is lengthy, it is presented in Sect. 5.7.4.

5.5.4 Summary of the Saketini Algorithm

The Saketini algorithm is summarized as follows. It estimates the source time course at a pointing location, based on the data model in Eq. (5.134). It uses the VBEM algorithm in which the E-step updates the posterior precision $\boldsymbol{\Gamma}$ and the posterior mean $\bar{\boldsymbol{z}}_k$ by using Eqs. (5.148) and (5.149). The M-step updates $\boldsymbol{\Psi}$ and $\bar{\boldsymbol{A}}$ using Eqs. (5.152) and (5.153). The hyperparameter $\boldsymbol{\Phi}$ is updated using Eq. (5.165). The hyperparameter $\boldsymbol{\alpha}$ is updated using (5.79). The noise precision $\boldsymbol{\Lambda}$ is updated using Eq. (5.166). The free energy can be computed using the same equation as in Eq. (5.131) with the sole modification of adding $\log|\boldsymbol{\Phi}|$. That is, the free energy is expressed as

$$\mathcal{F} = \frac{K}{2}\log\frac{|\boldsymbol{\Lambda}||\boldsymbol{\Phi}|}{|\boldsymbol{\Gamma}|} - \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k + \frac{1}{2}\sum_{k=1}^{K}\bar{\boldsymbol{z}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{z}}_k + \frac{M}{2}\log\frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\Psi}|}. \tag{5.167}$$

This free energy forms the lower bound of the marginal likelihood, and it is used for monitoring the progress of the VBEM iteration.

5.6 Numerical Examples

Computer simulation was performed to present typical results from the methods mentioned in this chapter. An alignment of the 275-sensor array from the Omega™ (VMS Medtech, Coquitlam, Canada) neuromagnetometer was used. The coordinate system and source-sensor configuration used in the computer simulation are depicted in Fig. 5.1. A vertical plane was assumed at the middle of the whole-head sensor array ($x = 0\,\text{cm}$), and three sources were assumed to exist on this plane. The time courses shown in the upper three panels of Fig. 5.2 were assigned to the three sources.

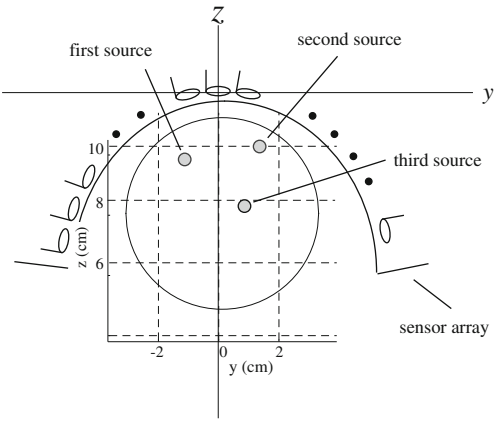

Fig. 5.1 The coordinate system and source-sensor configuration used in the computer simulation. A vertical plane was assumed at the *middle* of the whole-head sensor array ($x = 0\,\text{cm}$), and three sources were assumed to exist on this plane at $x = 0\,\text{cm}$. The *large circle* shows the source space and the *small filled circles* show the locations of the three sources

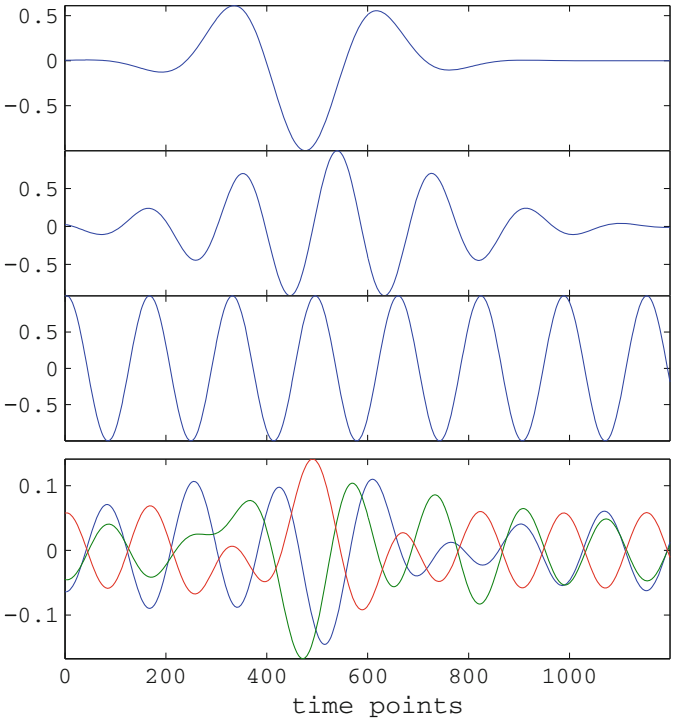

Fig. 5.2 The time courses of the three sources are shown in the *upper* three panels. The *bottom panel* shows the signal-only magnetic recordings (before noise is added). The selected three time courses from sensor #1, #101, and #201 are shown. The ordinates show relative values, and the abscissa shows the time points. A total of 1,200 time point data was generated

Simulated magnetic recordings were computed by projecting the time courses of the three sources onto the sensor space using the sensor lead field at the source locations. These recordings are shown for three selected sensor channels in the bottom panel of Fig. 5.2.

We first performed denoising experiments using the BFA and VBFA algorithms. We generated simulated MEG data by adding the sensor noise onto the computed magnetic recordings shown in the bottom panel of Fig. 5.2. Here, the noise generation was performed using a Gaussian random number, and the signal-to-noise ratio (SNR) was set to one. The noise-added simulated sensor data is shown in the top panel of Fig. 5.3 for the same selected sensor channels.

The BFA algorithm was applied to these simulated MEG data and the signal component was estimated by computing $\boldsymbol{A}\bar{\boldsymbol{u}}_k$. The results of this experiment are presented in Fig. 5.3. The results with the number of factors L set to 3 are shown in the second panel from the top, and the results with L set to 20 are in the third panel. Since we assume that three sources exist in this numerical experiment, the correct value of L is three.

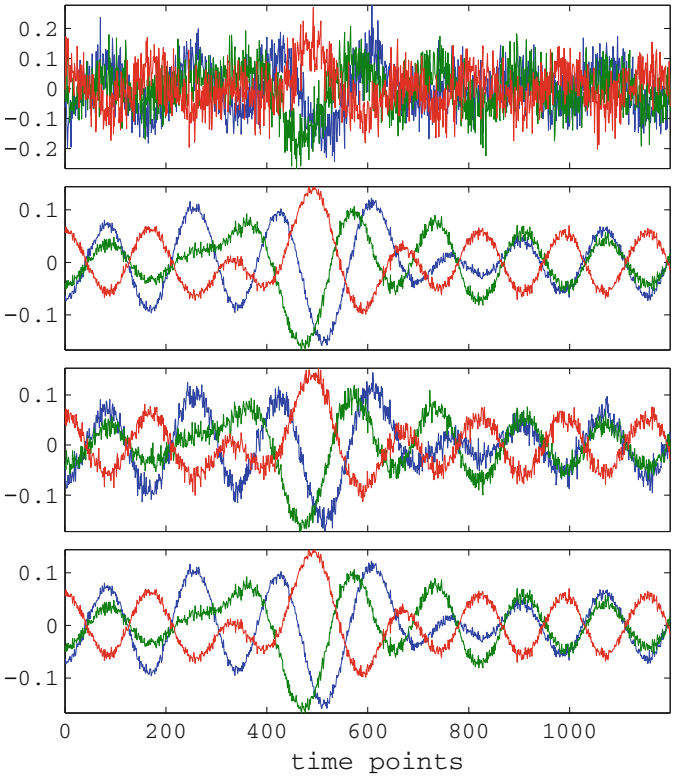

Fig. 5.3 Results of denoising experiments using BFA and VBFA. The *top panel* shows the sensor recordings before denoising. The *second panel* from the *top* shows the denoising results by BFA with the model order L set to three. The *third panel* shows the denoising results by BFA with L set to twenty. The *bottom panel* shows the denoising results by VBFA with L set to twenty. The time courses of the selected three sensor channels, (channel #1, channel #101, and channel #201), are shown. The ordinates show relative values, and abscissa shows the time points

Results in the second panel (results with the correct L) are better than the results in the third panel (results with a significantly overespecified L), and we can see that the overspecification of the number of factors, L, reduces the denoising capability of the algorithm. The VBFA algorithm was then applied, and the results are shown in the bottom panel of Fig. 5.3. Here, although the number of factors L was set to 20, results close to those obtained using the BFA algorithm with the correct value of L were obtained. The results demonstrate the fact that the VBFA algorithm incorporates the model order determination and is insensitive to the overspecification of the number of factors.

We next performed experiments on the interference removal using the PFA algorithm. In this experiment, the simulated MEG recordings were generated with 2,400 time points where the first half period does not contain the signal activity, which is contained in the second half period. The interference data was computed by

generating fifty interference sources in the source space. The interference sources had Gaussian random time courses, and the interference magnetic recordings were generated by projecting these time courses onto the sensor space using the sensor lead field at the interference source locations. The simulated MEG recordings were generated by adding the interference (and a small amount of sensor noise) onto the same simulated signal-only sensor data in the previous denoting experiment.

In the PFA application, the first half period was used as the control period and the mixing matrix of the interference $\boldsymbol{B}$ and the diagonal noise precision matrix $\boldsymbol{\Lambda}$ were estimated using this control period. We then estimated the signal mixing matrix $\boldsymbol{A}$ and the signal factor activity $\boldsymbol{u}_k$ using the second half period. The signal component was estimated using $\widehat{\boldsymbol{y}}_k^S$ in Eq. (5.132).

Results are shown in Fig. 5.4. In this figure, the top panel shows the control period (the first half period) of the original interference-added sensor data, and the middle panel shows the target period (the second half period) of the original interference-added sensor data. The bottom panel shows the interference-removed sensor data, $\widehat{\boldsymbol{y}}_k^S$. Here, the number of factors for the interference was set to 100, and that for the signal of interest was set to 10. Note that the true number of independent interference activities is 50, and that for the signal activity is 3. The results in Fig. 5.4 demonstrate that

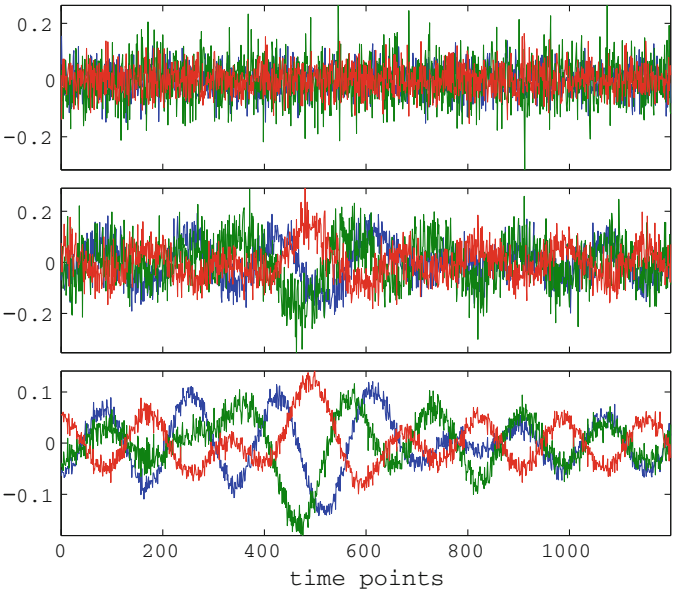

Fig. 5.4 Results of interference-removal experiments using the PFA algorithm. The *top panel* shows the control period (the first 1,200 time points) of the interference-added simulated sensor data. The *middle panel* shows the target period (the second 1,200 time points) of the interference-added simulated sensor data. The *bottom panel* shows the results of interference removal experiments. Here, results of $\widehat{\boldsymbol{y}}_k^S$ in Eq. (5.132) are plotted. The selected three sensor time courses (channel #1, channel #101, and channel #201) are shown in these three panels

the PFA algorithm successfully removes the interference, even though the number of factors for the signal and interference components are significantly overspecified.

We next performed a source reconstruction experiment. We computed the interference-removed covariance matrix $\bar{\boldsymbol{R}}$ in Eq. (5.133), and used it with the adaptive beamformer algorithm. The results are shown in Fig. 5.5. In Fig. 5.5a, the results of source reconstruction, obtained using the signal-only data (data before adding the interferences), are shown. The results of source reconstruction obtained with the interference-added data are shown in Fig. 5.5b. We can observe a certain amount of distortion in Fig. 5.5b. The results of reconstruction using $\bar{\boldsymbol{R}}$ are shown in Fig. 5.5c. The distortion is significantly reduced, demonstrating the effectiveness of the PFA algorithm.

Finally, a source reconstruction experiment using the Saketini algorithm was performed. Simulated MEG data was generated in which the signal-to-noise ratio (SNR) was set equal to two. We computed the power map using $\mathrm{tr}(\boldsymbol{\Phi}^{-1})$ with the number of factors L set to 20. The results are shown in Fig. 5.6. The reconstructed source power map on the plane of $x = 0$ is shown in Fig. 5.6a. The reconstructed time courses

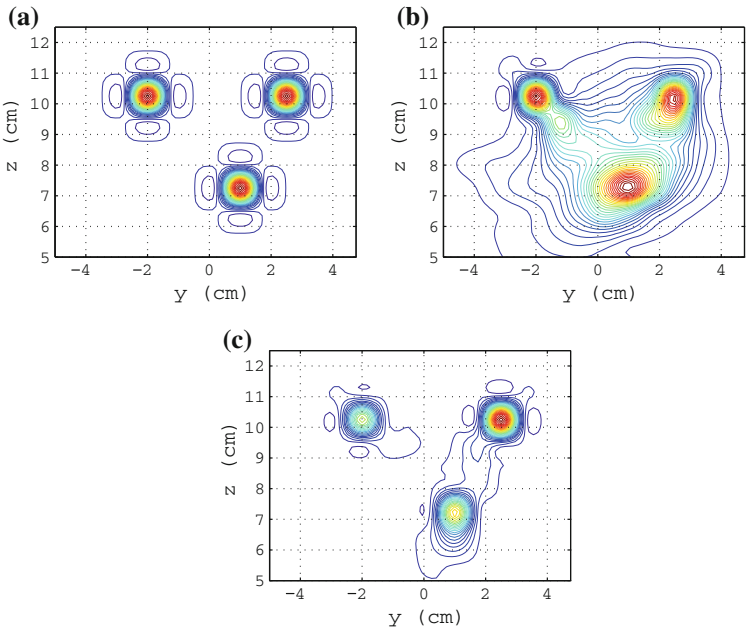

Fig. 5.5 Results of source reconstruction experiments using the interference-removed covariance matrix $\bar{\boldsymbol{R}}$ in Eq. (5.133). The adaptive beamformer algorithm was used for source reconstruction, and the reconstructed source distribution at $x = 0$ cm is shown. **a** The results obtained with a sample covariance matrix computed from signal-only simulated recordings. **b** The results obtained with a sample covariance matrix computed from the interference-added sensor data. **c** The results of the interference-removal experiment, obtained with $\bar{\boldsymbol{R}}$ in Eq. (5.133). The (y, z) coordinates of the three sources were set to $(-2.0, 10.2)$ cm, $(2.5, 10.2)$ cm, and $(1.0, 7.2)$ cm in this experiment

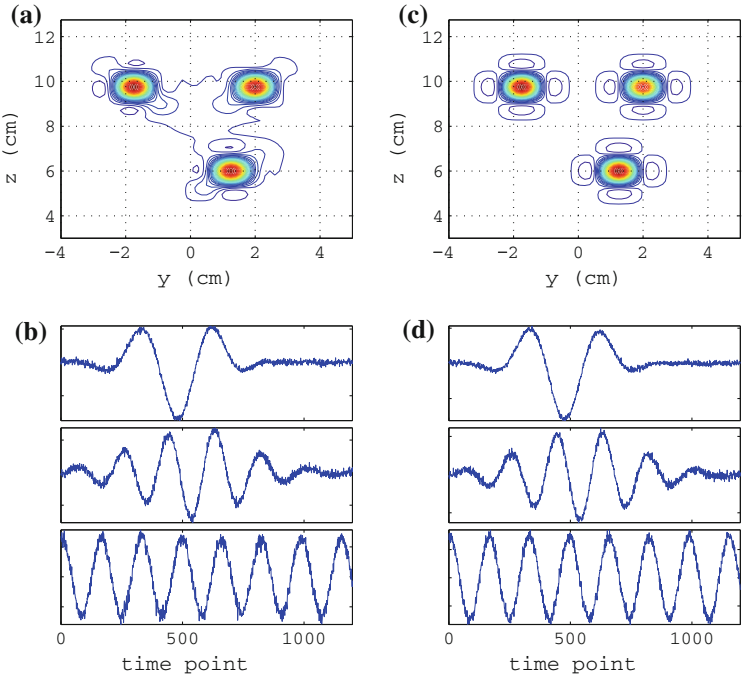

Fig. 5.6 Results of source reconstruction experiments using the Saketini algorithm. **a** The reconstructed source power map on the plane at $x = 0$ cm. **b** The reconstructed time courses of the three sources: The time courses of the first, second, and third sources are shown in the *top*, *middle*, and *bottom panels*, respectively. **c** The reconstructed source power map on the plane at $x = 0$ cm, obtained using the Champagne algorithm. **d** The reconstructed time courses of the three sources, obtained using the Champagne algorithm. The locations (y, z) of the first, second, and third sources were set to $(-1.8, 9.8)$, $(2.0, 9.8)$, and $(1.3, 6.0)$, respectively

of the voxels closest to the three sources are shown in Fig. 5.6b. The source power map and source time courses obtained using the Champagne algorithm are shown in Fig. 5.6c, d for comparison. The results show that both algorithms are capable of providing accurate spatiotemporal reconstruction of the three sources.

5.7 Appendix to This Chapter

5.7.1 Proof of Eq. (5.84)

We present the proof of Eq. (5.84), which is rewritten as

$$E_A[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T] = \bar{\boldsymbol{A}}\boldsymbol{\Psi}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1} = \boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1}.$$

First, we have

$$[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T]_{i,j} = \sum_{k,\ell} A_{i,k}[\boldsymbol{\Psi}]_{k,\ell}A_{j,\ell} = \sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}A_{i,k}A_{j,\ell}.$$

Equation (5.44) expresses the posterior distribution of the mixing matrix $\boldsymbol{A}$ where the two elements, $A_{i,k}$ and $A_{j,\ell}$, are independent when $i \neq j$ and when $i = j$, the covariance of these elements is equal to $[\lambda_j^{-1}\boldsymbol{\Psi}^{-1}]_{k,\ell}$. Thus, we have

$$E_{\boldsymbol{A}}[A_{i,k}A_{j,\ell}] = \bar{A}_{i,k}\bar{A}_{j,\ell} + \delta_{i,j}\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{k,\ell}. \tag{5.168}$$

Therefore, we get

$$E_{\boldsymbol{A}}\left[[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T]_{i,j}\right] = \sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}\bar{A}_{i,k}\bar{A}_{j,\ell} + \sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}\delta_{i,j}\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{k,\ell}. \tag{5.169}$$

The first term on the right-hand side of the equation above is rewritten as

$$\sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}\bar{A}_{i,k}\bar{A}_{j,\ell} = [\bar{\boldsymbol{A}}\boldsymbol{\Psi}\bar{\boldsymbol{A}}^T]_{i,j}.$$

Considering

$$\sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}[\boldsymbol{\Psi}^{-1}]_{k,\ell} = \mathrm{tr}\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^{-1}\right) = \mathrm{tr}\,(\boldsymbol{I}) = L, \tag{5.170}$$

the second term in Eq. (5.169) is rewritten as

$$\sum_{k,\ell}[\boldsymbol{\Psi}]_{k,\ell}\delta_{i,j}\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{k,\ell} = L\delta_{i,j}\frac{1}{\lambda_j} = L[\boldsymbol{\Lambda}^{-1}]_{i,j}, \tag{5.171}$$

because $\boldsymbol{\Lambda}$ is diagonal. Consequently, we get the following relationship:

$$E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T] = \bar{\boldsymbol{A}}\boldsymbol{\Psi}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1}. \tag{5.172}$$

Using Eqs. (5.68) and (5.70), we obtain the relationship

$$\bar{\boldsymbol{A}} = \boldsymbol{R}_{yu}\boldsymbol{\Psi}^{-1} \quad \text{or} \quad \bar{\boldsymbol{A}}\boldsymbol{\Psi} = \boldsymbol{R}_{yu}.$$

Thus, we finally derive Eq. (5.84), such that

$$E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{A}^T] = \boldsymbol{R}_{yu}\bar{\boldsymbol{A}}^T + L\boldsymbol{\Lambda}^{-1}. \tag{5.173}$$

5.7.2 *Proof of Eq.* (5.94)

We provide a proof of Eq. (5.94), which is repeated here for convenience.

$$E_{(\boldsymbol{A},\boldsymbol{u})}\left[-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{u}_k)-\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{u}_k\boldsymbol{u}_k^T\right]$$
$$=-\frac{1}{2}\sum_{j=1}^{K}\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k+\frac{1}{2}\sum_{j=1}^{K}\bar{\boldsymbol{u}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k.$$

The left-hand side of the equation above is changed to:

$$\begin{aligned}&E_{(\boldsymbol{A},\boldsymbol{u})}\left[-\frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{u}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{u}_k)-\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{u}_k\boldsymbol{u}_k^T\right]\\&=-\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k\\&\quad-\frac{1}{2}\sum_{k=1}^{K}E_{(\boldsymbol{A},\boldsymbol{u})}\left[-\boldsymbol{u}_k^T\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{y}_k-\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k+\boldsymbol{u}_k^T\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k+\boldsymbol{u}_k^T\boldsymbol{u}_k\right].\end{aligned}\tag{5.174}$$

On the right-hand side of Eq. (5.174), the following relationship:

$$\begin{aligned}E_{(\boldsymbol{A},\boldsymbol{u})}\left[\boldsymbol{u}_k^T\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k+\boldsymbol{u}_k^T\boldsymbol{u}_k\right]&=E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T\left[E_{\boldsymbol{A}}\left(\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A}\right)+\boldsymbol{I}\right]\boldsymbol{u}_k\right]\\&=E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T\left[\bar{\boldsymbol{A}}^T\boldsymbol{\Lambda}\bar{\boldsymbol{A}}+M\boldsymbol{\Psi}^{-1}+\boldsymbol{I}\right]\boldsymbol{u}_k\right]\\&=E_{\boldsymbol{u}}\left[\boldsymbol{u}_k^T\boldsymbol{\Gamma}\boldsymbol{u}_k\right]=E_{\boldsymbol{u}}\left[\mathrm{tr}\left(\boldsymbol{u}_k\boldsymbol{u}_k^T\boldsymbol{\Gamma}\right)\right]\\&=\mathrm{tr}\left[E_{\boldsymbol{u}}\left(\boldsymbol{u}_k\boldsymbol{u}_k^T\right)\boldsymbol{\Gamma}\right]=\mathrm{tr}\left[(\bar{\boldsymbol{u}}_k\bar{\boldsymbol{u}}_k^T+\boldsymbol{\Gamma}^{-1})\boldsymbol{\Gamma}\right]\\&=\bar{\boldsymbol{u}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k\end{aligned}\tag{5.175}$$

holds, where constant terms are omitted. In the equation above, we use Eq. (5.56). Also, the relationship,

$$\begin{aligned}E_{(\boldsymbol{A},\boldsymbol{u})}\left[\boldsymbol{y}_k^T\boldsymbol{\Lambda}\boldsymbol{A}\boldsymbol{u}_k+\boldsymbol{u}_k^T\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{y}_k\right]&=\boldsymbol{y}_k^T\boldsymbol{\Lambda}\bar{\boldsymbol{A}}\bar{\boldsymbol{u}}_k+\bar{\boldsymbol{u}}_k^T\bar{\boldsymbol{A}}^T\boldsymbol{\Lambda}\boldsymbol{y}_k\\&=\boldsymbol{y}_k^T\boldsymbol{\Lambda}\bar{\boldsymbol{A}}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k+\bar{\boldsymbol{u}}_k^T\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{A}}^T\boldsymbol{\Lambda}\boldsymbol{y}_k\\&=2\bar{\boldsymbol{u}}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{u}}_k\end{aligned}\tag{5.176}$$

holds. Substituting Eqs. (5.175) and (5.176) into (5.174), we get Eq. (5.94).

5.7.3 Proof of Eq. (5.103)

Next, the proof of Eq. (5.103) is presented, which is

$$E_{\boldsymbol{A}}\left[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A}^T\right] = \bar{\boldsymbol{A}}\boldsymbol{R}_{uu}\bar{\boldsymbol{A}}^T + \boldsymbol{\Lambda}^{-1}\,\mathrm{tr}\,(\boldsymbol{R}_{uu}\boldsymbol{\Psi})\,.$$

The proof is quite similar to the one provided in Sect. 5.7.1. First, we have

$$[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A}^T]_{i,j} = \sum_{k,\ell} A_{i,k}[\boldsymbol{R}_{uu}]_{k,\ell}A_{j,\ell} = \sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}A_{i,k}A_{j,\ell}.$$

Using Eq. (5.168),we get

$$E_{\boldsymbol{A}}\left[[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A}^T]_{i,j}\right] = \sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}\bar{A}_{i,k}\bar{A}_{j,\ell} + \sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}\delta_{i,j}\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{k,\ell}. \tag{5.177}$$

The first term on the right-hand side of the equation above is rewritten as

$$\sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}\bar{A}_{i,k}\bar{A}_{j,\ell} = [\bar{\boldsymbol{A}}\boldsymbol{R}_{uu}\bar{\boldsymbol{A}}^T]_{i,j}.$$

Considering

$$\sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}[\boldsymbol{\Psi}^{-1}]_{k,\ell} = \mathrm{tr}(\boldsymbol{R}_{uu}\boldsymbol{\Psi}^{-1}), \tag{5.178}$$

the second term in Eq. (5.177) is rewritten as

$$\sum_{k,\ell}[\boldsymbol{R}_{uu}]_{k,\ell}\delta_{i,j}\frac{1}{\lambda_j}[\boldsymbol{\Psi}^{-1}]_{k,\ell} = [\boldsymbol{\Lambda}^{-1}]_{i,j}\,\mathrm{tr}(\boldsymbol{R}_{uu}\boldsymbol{\Psi}^{-1}). \tag{5.179}$$

Consequently, we get the following relationship:

$$E_{\boldsymbol{A}}[\boldsymbol{A}\boldsymbol{R}_{uu}\boldsymbol{A}^T] = \bar{\boldsymbol{A}}\boldsymbol{R}_{uu}\bar{\boldsymbol{A}}^T + \boldsymbol{\Lambda}^{-1}\,\mathrm{tr}(\boldsymbol{R}_{uu}\boldsymbol{\Psi}^{-1}). \tag{5.180}$$

5.7.4 Proof of Eq. (5.166)

Next, the proof of Eq. (5.166) is presented. To derive this equation, we should consider $\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{A})$ and $\log p(\boldsymbol{A})$, because $\boldsymbol{\Lambda}$ is contained in these two terms. We have

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\Lambda}} \log p(\boldsymbol{y}|z, \boldsymbol{A}) \\
&= \frac{\partial}{\partial \boldsymbol{\Lambda}} \left[\frac{K}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \boldsymbol{\Lambda} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k) \right] \\
&= \frac{K}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)^T. \qquad (5.181)
\end{aligned}$$

According to Eq. (5.81), we have

$$\frac{\partial}{\partial \boldsymbol{\Lambda}} \log p(\boldsymbol{A}) = \frac{L}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} \operatorname{diag} \left[\boldsymbol{A} \boldsymbol{\alpha} \boldsymbol{A}^T \right]. \qquad (5.182)$$

We can thus compute the derivative of $\mathcal{F}(\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Phi})$ with respect to $\boldsymbol{\Lambda}$, which is

$$\begin{aligned}
\frac{\partial \mathcal{F}}{\partial \boldsymbol{\Lambda}} &= E_{(A,z)} \Bigg[\frac{K}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)(\boldsymbol{y}_k - \boldsymbol{L}\boldsymbol{s}_k - \boldsymbol{A}\boldsymbol{u}_k)^T \\
&\qquad + \frac{L}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} \boldsymbol{A} \boldsymbol{\alpha} \boldsymbol{A}^T \Bigg] \\
&= \frac{K}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} [\boldsymbol{R}_{yy} - \boldsymbol{R}_{yu} \bar{\boldsymbol{A}}^T - \bar{\boldsymbol{A}} \boldsymbol{R}_{uy} - \boldsymbol{R}_{ys} \boldsymbol{L}^T - \boldsymbol{L} \boldsymbol{R}_{sy} \\
&\qquad + \boldsymbol{L} \boldsymbol{R}_{su} \bar{\boldsymbol{A}}^T + \bar{\boldsymbol{A}} \boldsymbol{R}_{us} \boldsymbol{L}^T + \boldsymbol{L} \boldsymbol{R}_{ss} \boldsymbol{L}^T] \\
&\quad - \frac{1}{2} E_A [\boldsymbol{A} \boldsymbol{R}_{uu} \boldsymbol{A}^T] + \frac{L}{2} \boldsymbol{\Lambda}^{-1} - \frac{1}{2} E_A [\boldsymbol{A} \boldsymbol{\alpha} \boldsymbol{A}^T]. \qquad (5.183)
\end{aligned}$$

According to Eqs. (5.83) and (5.84), we get

$$\begin{aligned}
E_A [\boldsymbol{A} \boldsymbol{R}_{uu} \boldsymbol{A}^T + \boldsymbol{A} \boldsymbol{\alpha} \boldsymbol{A}^T] &= E_A [\boldsymbol{A} (\boldsymbol{R}_{uu} + \boldsymbol{\alpha}) \boldsymbol{A}^T] \\
&= E_A [\boldsymbol{A} \boldsymbol{\Psi} \boldsymbol{A}^T] = \bar{\boldsymbol{A}} \boldsymbol{\Psi} \bar{\boldsymbol{A}}^T + L \boldsymbol{\Lambda}^{-1}. \qquad (5.184)
\end{aligned}$$

Using the equation above and the relationship $\bar{\boldsymbol{A}} \boldsymbol{\Psi} = (\boldsymbol{R}_{yu} - \boldsymbol{L} \boldsymbol{R}_{su})$, and setting the right-hand side of Eq. (5.183) equal to zero, we derive the update equation for $\boldsymbol{\Lambda}$, such that,

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{K} \operatorname{diag} [\boldsymbol{R}_{yy} - \boldsymbol{R}_{ys} \boldsymbol{L}^T - \boldsymbol{L} \boldsymbol{R}_{sy} + \boldsymbol{L} \boldsymbol{R}_{ss} \boldsymbol{L}^T - \bar{\boldsymbol{A}} \boldsymbol{\Psi} \bar{\boldsymbol{A}}^T].$$

The equation above is equal to Eq. (5.166).

References

1. S.S. Nagarajan, H.T. Attias, K.E. Hild, K. Sekihara, A probabilistic algorithm for robust interference suppression in bioelectromagnetic sensor data. Stat. Med. **26**, 3886–3910 (2007)
2. J.M. Zumer, H.T. Attias, K. Sekihara, S.S. Nagarajan, A probabilistic algorithm integrating source localization and noise suppression for MEG and EEG data. NeuroImage **37**, 102–115 (2007)
3. H. Attias, Inferring parameters and structure of latent variable models by variational bayes, in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, (Morgan Kaufmann Publishers Inc, 1999), pp. 21–30
4. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodological) **39**, 1–38 (1977)
5. R.H. Shumway, D.S. Stoffer, An approach to time series smoothing and forecasting using the EM algorithm. J. Time Ser. Anal. **3**, 253–263 (1982)
6. H.T. Attias, A variational Bayesian framework for graphical models, in *Advances in Neural Information Processing* (MIT Press, Cambridge, 2000), pp. 209–215

Chapter 6
A Unified Bayesian Framework for MEG/EEG Source Imaging

6.1 Introduction

Magnetoencephalography (MEG) and related electroencephalography (EEG) use an array of sensors to take electromagnetic field (or voltage) measurements from on or near the scalp surface with excellent temporal resolution. In both MEG and EEG, the observed field can in many cases be explained by synchronous, compact current sources located within the brain. Although useful for research and clinical purposes, accurately determining the spatial distribution of these unknown sources is a challenging inverse problem. The relevant estimation problem can be posed as follows: The measured electromagnetic signal is $\boldsymbol{y} \in \mathbb{R}^{d_y \times d_t}$, where d_y equals the number of sensors and d_t is the number of time points at which measurements are made. Each unknown source $\boldsymbol{s}_r \in \mathbb{R}^{d_c \times d_t}$ is a d_c-dimensional neural current dipole, at d_t timepoints, projecting from the rth (discretized) voxel or candidate location distributed throughout the brain. These candidate locations can be obtained by segmenting a structural MR scan of a human subject and tessellating the brain volume with a set of vertices, $\boldsymbol{y}$ and each $\boldsymbol{s}_r$ are related by the likelihood model

$$\boldsymbol{y} = \sum_{r=1}^{d_s} \boldsymbol{L}_r \boldsymbol{s}_r + \boldsymbol{\varepsilon}, \tag{6.1}$$

where d_s is the number of voxels under consideration and $\boldsymbol{L}_r \in \mathbb{R}^{d_y \times d_c}$ is the so-called lead-field matrix for the rth voxel. The kth column of $\boldsymbol{L}_r$ represents the signal vector that would be observed at the scalp given a unit current source/dipole at the rth vertex with a orientation in the kth direction. It is common to assume $d_c = 2$ (for MEG with a single spherical shell model) or $d_c = 3$ (for EEG), which allows flexible source orientations to be estimated in 2D or 3D space. Multiple methods based on the physical properties of the brain and Maxwell's equations are available for the computation of each L_i, as described in Appendix A. Finally, $\boldsymbol{\varepsilon}$ is a noise-plus-interference term where we assume, for simplicity, that the columns

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_6

are drawn independently from $\mathcal{N}(0, \Sigma_\epsilon)$. However, temporal correlations can easily be incorporated if desired using a simple transformation outlined in [1] or using the spatio-temporal framework introduced in [2]. In this chapter, we will mostly assume that Σ_ϵ is known; however, robust procedures for its estimation can be found in Chap. 5, and can naturally be incorporated into the proposed model. However, it should be noted that joint estimation of brain source activity and Σ_ϵ potentially leads to identifiability issues, and remains an unsolved problem to date.

To obtain reasonable spatial resolution, the number of candidate source locations will necessarily be much larger than the number of sensors ($d_s \gg d_y$). The salient inverse problem then becomes the ill-posed estimation of regions with significant brain activity, which are reflected by voxels i such that $|\boldsymbol{s}_i| > 0$; we refer to these as *active* dipoles or sources. The severe underdetermine nature of this MEG (or related EEG) source localization problem (since the mapping from source activity configuration $\boldsymbol{s} \triangleq [\boldsymbol{s}_1^T, \ldots, \boldsymbol{s}_{d_s}^T]^T$ to sensor measurement $\boldsymbol{y}$ is many to one), requires the incorporation of prior assumptions when choosing an appropriate solution out of an infinite set of candidates.

Bayesian approaches are useful in this capacity because they allow these assumptions to be explicitly quantified using postulated prior distributions. However, the means by which these priors are chosen, as well as the estimation and inference procedures that are subsequently adopted to affect localization, have led to a daunting array of algorithms with seemingly very different properties and assumptions.

While seemingly quite different in many respects, we present a generalized framework that encompasses all of these methods and points to intimate connections between algorithms. The underlying motivation here is to leverage analytical tools and ideas from machine learning, Bayesian inference, and convex analysis that have not as of yet been fully exploited in the context of MEG/EEG source localization. Specifically, here we address how a simple Gaussian scale mixture prior with flexible covariance components underlie and generalize all of the above. This process demonstrates a number of surprising similarities or out-right equivalences between what might otherwise appear to be very different methodologies. Theoretical properties related to convergence, global and local minima, and localization bias are analyzed and fast algorithms are derived that improve upon existing methods. This perspective leads to explicit connections between many established algorithms and suggests natural extensions for handling unknown dipole orientations, extended source configurations, correlated sources, temporal smoothness, and computational expediency. Specific imaging methods elucidated under this paradigm include weighted minimum $\ell 2$-norm, FOCUSS [3], MCE [4], VESTAL [5], sLORETA [6], ReML [7] and covariance component estimation, beamforming, variational Bayes, the Laplace approximation, and automatic relevance determination (ARD) [8, 9]. Perhaps surprisingly, all of these methods can be formulated as particular cases of covariance component estimation using different concave regularization terms and optimization rules, making general theoretical analyses and algorithmic extensions/improvements particularly relevant. By providing a unifying theoretical perspective and comprehensive analyses, neuroelectromagnetic imaging practitioners will be better able to

assess the relative strengths of many Bayesian strategies with respect to particular applications; it will also help ensure that different methods are used to their full potential and not underutilized.

6.2 Bayesian Modeling Framework

In a Bayesian framework all prior assumptions are embedded in the distribution $p(\boldsymbol{s})$. If under a given experimental or clinical paradigm this $p(\boldsymbol{s})$ were somehow known exactly, then the posterior distribution $p(\boldsymbol{s}|\boldsymbol{y})$ can be computed via Bayes rule:

$$p(\boldsymbol{s}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s})}{p(\boldsymbol{y})}. \tag{6.2}$$

This distribution contains all possible information about the unknown $\boldsymbol{s}$ conditioned on the observed data $\boldsymbol{y}$. Two fundamental problems prevent using $p(\boldsymbol{s}|\boldsymbol{y})$ for source localization. First, for most priors $p(\boldsymbol{s})$, the distribution $p(\boldsymbol{y})$ given by:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s})\mathrm{d}\boldsymbol{s}. \tag{6.3}$$

cannot be computed. Because this quantity, which is sometimes referred to as the model evidence or marginal likelihood, is required to compute posterior moments and is also sometimes used to facilitate model selection, this deficiency can be very problematic. Of course if only a point estimate for $\boldsymbol{s}$ is desired, then this normalizing distribution may not be needed. For example, a popular estimator involves finding the value of $\boldsymbol{s}$ that maximizes the posterior distribution, often called the maximum a posteriori or MAP estimate, and is invariant to $p(\boldsymbol{y})$. However MAP estimates may be unrepresentative of posterior mass and are unfortunately intractable to compute for most $p(\boldsymbol{s})$ given reasonable computational resources. Secondly, we do not actually know the prior $p(\boldsymbol{s})$ and so some appropriate distribution must be assumed, perhaps based on neurophysiological constraints or computational considerations. In fact, it is this choice, whether implicitly or explicitly, that differentiates a wide variety of localization methods at a very high level.

Such a prior is often considered to be fixed and known, as in the case of minimum $\ell 2$-norm approaches [2], minimum current estimation (MCE), FOCUSS, sLORETA, and minimum variance beamformers. Alternatively, a number of empirical Bayesian approaches have been proposed that attempt a form of model selection by using the data to guide the search for an appropriate $p(\boldsymbol{s})$. In this scenario, candidate priors are distinguished by a set of flexible hyperparameters γ that must be estimated via a variety of data-driven iterative procedures including hierarchical covariance component models, automatic relevance determination (ARD), and several related variational Bayesian methods (for a more comprehensive list of citations see Wipf et al. [10]).

6.3 Bayesian Modeling Using General Gaussian Scale Mixtures and Arbitrary Covariance Components

In this section, we present a general-purpose Bayesian framework for source localization and discuss a central distinction between fixed-prior MAP estimation schemes and empirical Bayesian approaches that adopt a flexible, parameterized prior. While often derived using different assumptions and methodology, they can be related via a simple hierarchical structure based on general Gaussian scale mixture distributions with arbitrary covariance components. Numerous special cases of this model have been considered previously in the context of MEG and EEG source localization and related problems as will discussed in subsequent sections.

6.3.1 The Generative Model

To begin we invoke the noise model from (6.1), which fully defines the assumed data likelihood

$$p(\boldsymbol{y}|\boldsymbol{s}) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{y}-\sum_{i=1}^{d_s}\boldsymbol{L}_i\boldsymbol{s}_i\right\|^2_{\Sigma_\epsilon^{-1}}\right), \tag{6.4}$$

where $\|X\|^2_W$ denotes the weighted matrix norm $\sqrt{\text{trace}[X^T W X]}$.

While the unknown noise covariance can also be parameterized and seamlessly estimated from the data via the proposed paradigm, for simplicity we assume that Σ_ϵ is known, estimated from the data using a variational Bayesian factor analysis (VBFA) model as discussed in Sect. 5.3 and that it is fixed. Next we adopt the following source prior for $\boldsymbol{s}$:

$$p\left(\boldsymbol{s}|\gamma\right) \propto \exp\left(-\frac{1}{2}\text{trace}[\boldsymbol{s}^T\Sigma_s^{-1}\boldsymbol{s}]\right), \Sigma_s = \sum_{i=1}^{d_\gamma}\gamma_i C_i. \tag{6.5}$$

This is equivalent to applying independently, at each time point, a zero-mean Gaussian distribution with covariance Σ_s to each column of $\boldsymbol{s}$. Here $\gamma \triangleq [\gamma_1, \ldots, \gamma_{d_\gamma}]$ is a vector of d_γ nonnegative hyperparameters that control the relative contribution of each covariance basis matrix C_i. While the hyperparameters are unknown, the set of components $C \triangleq \{C_i : i = 1, \ldots, d_\gamma\}$ is assumed to fixed and known. Such a formulation is extremely flexible however, because a rich variety of candidate covariance bases can be proposed as will be discussed in more detail later. Moreover, this structure has been advocated by a number of others in the context of neuroelectromagnetic source imaging [7, 11]. Finally, we assume a hyperprior on γ of the form

$$p(\gamma) = \prod_{i=1}^{d_\gamma}\frac{1}{2}\exp[-f_i(\gamma_i)] \tag{6.6}$$

where each $f_i(.)$ is an unspecified function that is assumed to be known. The implicit prior on $\boldsymbol{s}$, obtained by integrating out (marginalizing) the unknown γ, is known as a Gaussian scale mixture.

$$p(\boldsymbol{s}) = \int p(\boldsymbol{s}|\gamma)p(\gamma)\mathrm{d}\gamma. \tag{6.7}$$

6.3.2 Estimation and Inference

Estimation and inference using the proposed model can be carried out in multiple ways depending how the unknown quantities $\boldsymbol{s}$ and γ are handled. This leads to a natural partitioning of a variety of inverse methods.We briefly summarize three possibilities before discussing the details and close interrelationships.

6.3.2.1 Hyperparameter MAP or Empirical Bayesian

The first option is if γ were somehow known (and assuming Σ_s is known as well), then the conditional distribution $p(\boldsymbol{s}|\boldsymbol{y}, \gamma) \propto p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s}|\gamma)$ is a fully specified Gaussian distribution with mean and covariance given by

$$\begin{aligned} \mathrm{E}_{p(\boldsymbol{s}|\boldsymbol{y},\gamma)}[\boldsymbol{s}] &= \gamma \boldsymbol{L}^T \left(\Sigma_\epsilon + \boldsymbol{L}\Sigma_s \boldsymbol{L}^T\right)^{-1} B \\ \mathrm{Cov}_{p(\boldsymbol{s}_j|\boldsymbol{y},\gamma)}\left[\boldsymbol{s}_j\right] &= \Sigma_s - \Sigma_s \boldsymbol{L}^T \left(\Sigma_\epsilon + \boldsymbol{L}\Sigma_s \boldsymbol{L}^T\right)^{-1} \boldsymbol{L}\Sigma_s, \quad \forall j, \end{aligned} \tag{6.8}$$

where $\boldsymbol{L} \triangleq [\boldsymbol{L}_1, \ldots, \boldsymbol{L}_{d_s}]$ and $\boldsymbol{s}_j$ denotes the jth column of $\boldsymbol{s}$ (i.e., the sources at the jth time point) and individual columns are uncorrelated.

It is then common to use the simple estimator $\hat{\boldsymbol{s}} = E_{p(\boldsymbol{s}|\boldsymbol{y},\gamma)}[\boldsymbol{s}]$ for the unknown sources. However, since γ is actually not known, a suitable approximation $\hat{\gamma} \approx \gamma$ must first be found. One principled way to accomplish this is to integrate out the sources $\boldsymbol{s}$ and then solve

$$\hat{\gamma} = \arg\max_{\gamma} \int p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s}|\gamma)p(\gamma)\mathrm{d}\boldsymbol{s} \tag{6.9}$$

This treatment is sometimes referred to as empirical Bayes because the γ-dependent prior on $\boldsymbol{s}$, $p(\boldsymbol{s}|\gamma)$, is empirically learned from the data, often using expectation-maximization (EM) algorithms which treat $\boldsymbol{s}$ as hidden data. Additionally, the process of marginalization provides a natural regularizing mechanism that can shrink many elements of γ, to exactly zero, in effect pruning the associated covariance component from the model, with only the relevant components remaining. Consequently, estimation under this model is sometimes called automatic relevance determination (ARD). This procedure can also be leveraged to obtain a rigorous lower bound on

$\log p(\boldsymbol{y})$. While knowing $p(\boldsymbol{s}|\boldsymbol{y})$ is useful for source estimation given a particular model, access to $p(\boldsymbol{y})$ (or equivalently $\log p(\boldsymbol{y})$) can assist model selection.

γ-MAP obtains a point estimate for the unknown γ by first integrating out the unknown sources $\boldsymbol{s}$ producing the hyperparameter likelihood equation

$$p(\boldsymbol{y}|\gamma) = \int p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s}|\gamma)\mathrm{d}\boldsymbol{s} \propto \exp\left(-\frac{1}{2}\boldsymbol{y}^T \Sigma_y^{-1}\boldsymbol{y}\right), \tag{6.10}$$

where

$$\Sigma_y = \Sigma_\epsilon + \boldsymbol{L}\Sigma_s\boldsymbol{L}^T. \tag{6.11}$$

To estimate γ we then solve:

$$\hat{\gamma} = \arg\max_{\gamma} p(\gamma|\boldsymbol{y}) = \arg\max_{\gamma} p(\boldsymbol{y}|\gamma)p(\gamma), \tag{6.12}$$

which is equivalent to minimizing the cost function

$$\mathcal{L}(\gamma) \triangleq -2\log p(\boldsymbol{y}|\gamma)p(\gamma) \equiv \mathrm{trace}[C_y\Sigma_y^{-1}] + \log|\Sigma_y| + \frac{1}{n}\left[\sum_{i=1}^{n} f_i(\gamma_i)\right], \tag{6.13}$$

where $C_y \triangleq n^{-1}\boldsymbol{y}\boldsymbol{y}^T$ is the empirical covariance. This cost function is composed of three parts. The first is a data fit term based on the dissimilarity between the empirical covariance and the model covariance; in general, this factor encourages γ to be large. The second term provides the primary regularizing or sparsifying effect, penalizing a measure of the volume formed by the model covariance. Since the volume of any high dimensional space is more effectively reduced by collapsing individual dimensions as close to zero as possible (as opposed to reducing all dimensions isometrically), this penalty term promotes a model covariance that is maximally degenerate (or non-spherical), which pushes elements of γ to exactly zero.

Finally, the third term follows directly from the hyperprior, which we have thus far assumed to be arbitrary. This term can be useful for incorporating specific prior information, perhaps from fMRI data or some other imaging modality. It can also be used to expedite hyperparameter pruning or conversely, to soften the pruning process or prevent pruning altogether. One popular choice is the inverse-Gamma hyperprior [12] given by

$$p_i(\gamma_i^{-1}) = \mathrm{Gam}(a_i)^{-1}b_i^{a_i}\gamma_i^{1-a_i}\exp\left(-\frac{b_i}{\gamma_i}\right) \tag{6.14}$$

for the ith hyper parameter, where $\mathrm{Gam}(x) = \int_0^{\inf} z^{x-1}\exp(-z)dz$ is the standard Gamma function and a_i, b_i are the shape parameters. In the limit as $a_i, b_i \to 0$ then the prior converges to a non-informative Jeffreys prior, when optimization is performed on $\log\gamma_i$ space as is customary in some applications, the effective prior is flat and can therefore be ignored. In contrast, $a_i, b_i \to \infty$, all hyperparameters

are constrained to have equal value and essentially no learning (or pruning) occurs. Consequently, the standard weighted minimum $\ell 2$-norm solution can be seen as a special case.

We will often concern ourselves with flat hyperpriors when considering the γ-*MAP* option. In this case, the third term in Eq. (6.13) vanishes and the only regularization will come from the log $|.|$ term. In this context, the optimization problem with respect to the unknown hyperparameters is sometimes referred to as type-II maximum likelihood. It is also equivalent to the restricted maximum likelihood (ReML) cost function, discussed by Friston et al. [7] Regardless of how γ is optimized, once some $\hat{\gamma}$ is obtained, we compute $\hat{\Sigma}_s$ which fully specifies our assumed empirical prior on $\boldsymbol{s}$. To the extent that the "learned" prior $p(\boldsymbol{s}|\hat{\gamma})$ is realistic, this posterior quantifies regions of significant current density and point estimates for the unknown sources can be obtained by evaluating the posterior mean.

6.3.2.2 Optimization of γ-*MAP*

The primary objective of this section is to minimize Eq. (6.13) with respect to γ. For simplicity, we will first present updates with $f_i(\gamma_i) = 0$ (i.e. a flat hyper prior). We then address natural adaptations to the more general cases (e.g., not just conjugate priors). Of course one option is to treat the problem as a general nonlinear optimization task and perform gradient descent or some other generic procedure. In contrast, here we will focus on methods specifically tailored for minimizing Eq. (6.13) using principled methodology. We begin with methods based directly on the EM algorithm and then diverge to alternatives that draw on convex analysis to achieve faster convergence. One approach to minimizing Eq. (6.13) is the restricted likelihood method (ReML) which utilizes what amounts to EM-based updates treating $\boldsymbol{s}$ as hidden data.

For the E-step, the mean and covariance of $\boldsymbol{s}$ are computed given some estimate of the hyperparameters $\hat{\gamma}$. For the M-step, we then must update $\hat{\gamma}$ using these moments as the true values. Unfortunately, the optimal value of $\hat{\gamma}$ cannot be obtained in closed form for arbitrary covariance component sets, so a second-order Fisher scoring procedure is adopted to approximate the desired solution. While effective for estimating small numbers of hyperparameters, this approach requires inverting a $d_\gamma \times d_\gamma$ Fisher information matrix, which is not computationally feasible for large d_γ. Moreover, unlike exact EM implementations, there is no guarantee that such a Fisher scoring method will decrease the likelihood function Eq. (6.13) at each iteration.

Consequently, here we present alternative optimization procedures that apply to the arbitrary covariance model discussed above, and naturally guarantee that $\gamma_i \geqslant 0$ for all i. All of these methods rely on reparameterizing the generative model such that the implicit M-step can be solved in closed form. First, we note that $\mathcal{L}(\gamma)$ only depends on the data $\boldsymbol{y}$ through the $d_y \times d_y$ sample correlation matrix C_y. Therefore, to reduce the computational burden, we replace $\boldsymbol{y}$ with a matrix $\widetilde{\boldsymbol{y}} \in \mathbb{R}^{d_y \times \text{rank}(\boldsymbol{y})}$ such that $\boldsymbol{y}\boldsymbol{y}^T = \widetilde{\boldsymbol{y}}\widetilde{\boldsymbol{y}}^T$. This removes any per-iteration dependency on n, which can potentially be large, without altering that actual cost function. It also implies that, for purposes of computing γ, the number of columns of $\boldsymbol{s}$ is reduced to match $\text{rank}(\boldsymbol{y})$.

Next we introduce the decomposition

$$\boldsymbol{s} = \sum_{i=1}^{d_\gamma} A_i \widetilde{\boldsymbol{s}}_i = A\widetilde{\boldsymbol{s}}, \tag{6.15}$$

where each A_i is selected such that $A_i A_i^T = C_i$ and $A \triangleq [A_1 \dots A_{d_\gamma}]$, $\boldsymbol{s} \triangleq [\boldsymbol{s}_1^T \dots \boldsymbol{s}_{d_\gamma}^T]^T$. Letting $\widetilde{L} \triangleq [\widetilde{L}_1, \dots, \widetilde{L}_{d_\gamma}] = \boldsymbol{L}[A_1, \dots A_{d_\gamma}]$, this allows us to re-express the original hierarchical Bayesian model as

$$p(\widetilde{\boldsymbol{y}}|\widetilde{\boldsymbol{s}}) \propto \exp(-\frac{1}{2}||\boldsymbol{y} - \widetilde{L}\widetilde{\boldsymbol{s}}||^2_{\Sigma_\epsilon^{-1}}) \tag{6.16}$$

$$p(\widetilde{\boldsymbol{s}}_i|\gamma_i) \propto \exp(-\frac{1}{2\gamma_i}||\widetilde{\boldsymbol{s}}_i||^2_{\mathcal{F}}), \quad \forall i = 1, \dots, d_\gamma, \tag{6.17}$$

where $||X||_{\mathcal{F}}$ is the standard Frobenius norm $\sqrt{\text{trace}[XX^T]}$. The hyperprior remains unaltered. It is easily verified by the rules for transformation of random variables that Eq. (6.16) and the original model are consistent. It also follows that

$$\Sigma_y = \Sigma_\epsilon + \boldsymbol{L}(\sum_{i=1}^{d_\gamma} \gamma_i C_i)\boldsymbol{L}^T = \Sigma_\epsilon + \sum_{i=1}^{d_\gamma} \gamma_i \widetilde{L}_i \widetilde{L}_i i^T = \Sigma_\epsilon + \widetilde{L}\Sigma_{\widetilde{s}}\widetilde{L}^T \tag{6.18}$$

where $\Sigma_{\widetilde{s}}$ is the diagonal, γ-dependent prior covariance of the pseudo sources.

There are three ways to derive update rules for γ: EM iteration, Fixed-point (MacKay) updates, and Convexity based updates. Although the EM iteration update rules are the most straightforward to derive, they have empirically slow convergence rates. In contrast, the fixed-point updates have fast convergence rates but no convergence guarantees. In contrast, the convexity based updates have both fast convergence properties as well as guaranteed convergence properties. Details of derivations and properties of these update rules can be found elsewhere, but the three types of update rules are listed here for completeness.

1. EM-Updates

$$\gamma_i^{(k+1)} \to \frac{1}{nr_i}||\gamma_i^{(k)}\widetilde{\boldsymbol{L}}_i^T(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{y}}||^2_{\mathcal{F}} + \frac{1}{r_i}\text{trace}\Big[\gamma_i^{(k)}I - \gamma_i^{(k)}\widetilde{\boldsymbol{L}}_i^T(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{L}}_i\gamma_i^{(k)}\Big] \tag{6.19}$$

2. MacKay update:

$$\gamma_i^{(k+1)} \to \frac{1}{n}||\gamma_i^{(k)}\widetilde{\boldsymbol{L}}_i^T(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{y}}||^2_{\mathcal{F}}(\text{trace}[\gamma_i^{(k)}\widetilde{\boldsymbol{L}}_i^T(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{L}}_i])^{-1} \tag{6.20}$$

3. Convexity-based update

$$\gamma_i^{(k+1)} \to \frac{\gamma_i^{(k)}}{n}||\widetilde{\boldsymbol{L}}_i(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{y}}||_{\mathcal{F}}(\text{trace}[\widetilde{\boldsymbol{L}}_i^T(\Sigma_y^{(k)})^{-1}\widetilde{\boldsymbol{L}}_i])^{-1/2} \tag{6.21}$$

6.3.2.3 Analysis of γ-*MAP*

Previously, we have claimed that the γ-*MAP* process naturally forces excessive/irrelevant hyperparameters to converge to zero, thereby reducing model complexity. Note that, somewhat counterintuitively, this occurs even when a flat hyperprior is assumed. While this observation has been verified empirically by ourselves and others in various application settings, there has been relatively little corroborating theoretical evidence, largely because of the difficulty in analyzing the potentially multimodal, non-convex γ-*MAP* cost-function. We can then show that: Every local minimum of the generalized γ-*MAP* cost function, is achieved at a solution with utmost $\text{rank}(\boldsymbol{y})d_y \leqslant d_y^2$ non-zero hyper parameters if $f_i(\gamma_i)$ is concave and non-decreasing for all i, including flat hyper priors. Therefore, we can be confident that the pruning mechanism of γ-*MAP* is not merely an empirical phenomena. Nor is it dependent on a particular sparse hyperprior, the result holds when a flat (uniform) hyperprior is assumed.

The number of observation vectors n also plays an important role in shaping γ-*MAP* solutions. Increasing n has two primary benefits: (i) it facilitates convergence to the global minimum (as opposed to getting stuck in a suboptimal extrema) and (ii), it improves the quality of this minimum by mitigating the effects of noise. Finally, a third benefit to using $n > 1$ is that it leads to temporal smoothing of estimated time courses (i.e., rows of $\hat{\boldsymbol{s}}$. This occurs because the selected covariance components do not change across time, as would be the case if a separate set of hyperparameters were estimated at each time point. For purposes of model selection, a rigorous bound on $log\ p(\boldsymbol{y})$ can be derived using principles from convex analysis that have been successfully applied in general-purpose probabilistic graphical models (see Wipf and Nagarajan [13]).

6.3.3 Source MAP or Penalized Likelihood Methods

The second option is to integrate out the unknown γ, we can treat $p(\boldsymbol{s})$ as the effective prior and attempt to compute a MAP estimate of $\boldsymbol{s}$ via

$$\hat{\boldsymbol{s}} = \arg\max_{s} \int p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s}|\gamma)p(\gamma)d\gamma = \arg\max_{s} p(\boldsymbol{y}|\boldsymbol{s})p(\boldsymbol{s}) \tag{6.22}$$

While it may not be immediately transparent, solving $\boldsymbol{s}$-MAP also leads to a shrinking and pruning of superfluous covariance components. In short, this occurs because the hierarchical model upon which it is based leads to a convenient, iterative EM algorithm-based implementation, which treats the hyperparameters γ as hidden data and computes their expectation for the E-step. Over the course of learning, this expectation collapses to zero for many of the irrelevant hyperparameters, removing them from the model in much the same way as γ-*MAP*.

In a general setting Eq. (6.22) can be a difficult optimization problem and furthermore, the nature of the underlying cost function is not immediately transparent. Consequently, we advocate an indirect alternative utilizing the pseudo-source decomposition given by $\boldsymbol{s}$ described previously, which leads to an efficient EM implementation and a readily interpretable cost function. It also demonstrates that both FOCUSS and MCE can be viewed as EM algorithms that are readily generalized to handle more complex spatio-temporal constraints. Explicitly, we will minimize

$$\begin{aligned}\mathcal{L}(\widetilde{\boldsymbol{s}}) &\triangleq -2\log\int p(\widetilde{\boldsymbol{y}}|\widetilde{\boldsymbol{s}})\,p(\widetilde{\boldsymbol{s}}|\gamma)\,p(\gamma)\mathrm{d}\gamma \\ &= -2\log p(\widetilde{\boldsymbol{y}}|\widetilde{\boldsymbol{s}})\,p(\widetilde{\boldsymbol{s}}) \\ &\equiv ||\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{s}}||^2_{\Sigma_\epsilon^{-1}} + \sum_{i=1}^{d_\gamma} g_i(||\widetilde{\boldsymbol{s}}||^2_{\mathcal{F}}),\end{aligned} \tag{6.23}$$

where $g_i(.)$ is defined as

$$g_i(||\widetilde{\boldsymbol{s}}||^2_{\mathcal{F}}) \triangleq -2\log\int p(\widetilde{\boldsymbol{s}}_i|\gamma_i)\,p_i(\gamma_i)\mathrm{d}\gamma_i\,. \tag{6.24}$$

For many choices of the hyperprior, the associated $g_i(.)$ may not be available in closed form. Moreover, it is often more convenient and transparent to directly assume the form of $g_i(.)$ rather than infer its value from some postulated hyperprior. Virtually any non-decreasing, concave function $g_i(.)$ of interest can be generated by the proposed hierarchical model. In other words, there will always exist some $p_i(\gamma_i)$, possibly improper, such that the stated Gaussian mixture representation will produce any desired concave $g_i(.)$. For example, a generalized version of MCE and FOCUSS can be produced from the selection $g_i(z) = c_i z^{p/2}$, which is concave and amenable to a Gaussian scale-mixture representation for any $p \in (0, 2]$ and constant $c_i > 0$.

Presumably, there are a variety of ways to optimize Eq. (6.23). One particularly straightforward and convenient method exploits the hierarchical structure inherent in the assumed Bayesian model. This leads to simple and efficient EM-based update rules. It also demonstrates that the canonical FOCUSS iterations are equivalent to principled EM updates. Likewise, regularized MCE solutions can also be obtained in the same manner.

Ultimately, we would like to estimate each $\widetilde{\boldsymbol{s}}_i$, which in turn gives us the true sources S. If we knew the values of the hyperparameters γ this would be straightforward; however, these are of course unknown. Consequently, in the EM framework, γ is treated as hidden data whose distribution (or relevant expectation) is computed during the E-step. The M-step then computes the MAP estimate of $\widetilde{\boldsymbol{s}}$ assuming that γ equals the appropriate expectation. For the $(k+1)$th E-step, the expected value of each γ_i^{-1} under the distribution $p(\gamma|\boldsymbol{y}, \widetilde{\boldsymbol{s}}^{(k)})$ is required (see the M-step below) and can be computed analytically assuming $g_i(.)$ is differentiable, regardless of the underlying form of $p(\gamma)$. Assuming $g_i(z) \propto z^{p/2}$, it can be shown that

$$\gamma_i^{(k+1)} \triangleq E_{p(\gamma_i|\widetilde{\boldsymbol{y}},\widetilde{\boldsymbol{s}}_i^{(k)})}[\gamma_i^{-1}]^{-1} = \left[\frac{1}{r_i n}||\widetilde{\boldsymbol{s}}_i^{(k)}||_{\mathcal{F}}^2\right]^{\frac{2-p}{2}} \tag{6.25}$$

The associated M-step is then readily computed using

$$\widetilde{\boldsymbol{s}}_i^{(k+1)} \rightarrow \gamma_i \widetilde{\boldsymbol{L}}_i^T (\Sigma_\epsilon + \sum_{i=1}^{d_\gamma} \gamma_i \widetilde{\boldsymbol{L}}_i \widetilde{\boldsymbol{L}}_i^T)^{-1} \widetilde{\boldsymbol{y}}. \tag{6.26}$$

$\boldsymbol{s}^{(k+1)} = A\widetilde{\boldsymbol{s}}^{(k+1)}$ is then the $(k+1)$th estimate of the source activity. Each iteration of this procedure decreases the cost function of Eq. (6.23) and converges to some fixed point is guaranteed. From a computational stand point, the $\boldsymbol{s}$-*MAP* updates are of the same per-iteration complexity as the γ-*MAP* updates. Roughly speaking, the EM iterations above can be viewed as coordinate descent over a particular auxiliary cost function dependent on both $\boldsymbol{s}$ and γ. There exists a formal duality between $\boldsymbol{s}$-*MAP* and γ-*MAP* procedures that is beyond the scope of this chapter, but details of which are elaborate in [10, 14].

Given $n = 1$, and $A_i = \boldsymbol{e}_i$, where each $\boldsymbol{e}_i$ is a standard indexing vector of zeros with a "1" for the ith element, we get,

$$\gamma_i^{(k+1)} \rightarrow [\gamma_i^k \boldsymbol{L}_i^T (\lambda I + \boldsymbol{L}\,\mathrm{diag}[\gamma^k]\boldsymbol{L}^T)^{-1} \boldsymbol{y}]^{(2-p)} \tag{6.27}$$

where $\boldsymbol{L}_i$ is the ith column of L. We then recover the exact FOCUSS updates when $p \rightarrow 0$ and a FOCUSS-like update for MCE when $p = 1$. Note however, that while previous applications of MCE and FOCUSS to electromagnetic imaging using $n = 1$ require a separate iterative solution to be computed at each time point in isolation, here the entire $\boldsymbol{s}$ can be computed at once with $n > 1$ for about the same computational cost as a single FOCUSS run.

The nature of the EM updates for $\boldsymbol{s}$-*MAP*, where an estimate of γ is obtained via the E-step, suggest that this approach is indirectly performing some form of covariance component estimation. But if this is actually the case, it remains unclear exactly what cost function these covariance component estimates are minimizing. This is unlike the case of γ-*MAP* where it is more explicit. The fundamental difference between $\boldsymbol{s}$-*MAP* and γ-*MAP* lies in the regularization mechanism of the covariance components.Unlike γ-*MAP* the penalty term in Eq. (6.23), is a separable summation that depends on the value of p to affect hyperparameter pruning; and importantly there is no volume-based penalty. For $p < 1$ the penalty is concave in γ and hence, every local minimum of Eq. (6.23) is achieved at a solution with at most $\mathrm{rank}(\boldsymbol{y})d_y \leqslant d_y^2$ non-zero hyper parameters. Rather than promoting sparsity at the level of individual source elements at a given voxel and time (as occurs with standard MCE and FOCUSS when $n = 1$, $C_i = \boldsymbol{e}_i$, here sparsity is encouraged at the level of the pseudo-sources, $\widetilde{\boldsymbol{s}}$. The function $g_i(.)$ operates on the Frobenius norm of each $\widetilde{\boldsymbol{S}}_i$ and favors solutions with many $||\widetilde{\boldsymbol{s}}||_{\mathcal{F}} = 0$ for many indices of i. Notably though, by virtue of the Frobenius norm over time, within a non-zero $\widetilde{\boldsymbol{s}}_i$ temporally smooth (non-sparse)

solutions are favored. This is why, one obtains continuous source estimates over time, as opposed to application of FOCUSS or MCE at each individual time-point.

Even though sparsity bounds for ***s**-MAP* and γ-*MAP* are somewhat similar, the actual performance in practical situations is quite distinct. This is because of global minimum convergence properties of ***s**-MAP* that are impacted by the choice of $g_i(.)$ or p. For instance, although setting $p = 1$ leads to a convex cost function devoid of non-global minima, and the update rules for generalized-MCE will converge to the a global minimum, the resultant estimate can have problems recovering the true source estimate, especially for conditions related to MEG and EEG source reconstructions. This is because lead-fields L required for MEG and EEG are highly correlated, and violate the restricted isometric properties (RIP) that are required for accurate performance of $\ell 1$-norm procedures. More details on this point can be found in [14]. These theoretical restrictions essentially render the conditions for MCE performance to reconstruction of 1–2 dipoles at best, with no localization bias guarantees. The problem with MCE is not the existence of local minima, rather, it is that the global minimum may be unrepresentative of the true source distribution even for simple dipolar source configurations. In this situation, and especially when lead field columns are highly correlated, the MCE solution may fail to find sufficiently sparse source representations consistent with the assumption of a few equivalent current dipoles, and this issue will also persist with more complex covariance components and source configurations. Nevertheless, the unimodal, convex nature of the generalized MCE like procedures are its attractive advantage.

Furthermore, if $p < 1$ (which implies that $g_i(z^2)$ is concave in z) then more pseudo sources will be pruned at any global solution, which often implies that those which remain may be more suitable than the MCE estimate. Certainly this is true when estimating dipolar sources, but it likely holds in more general situations as well. However, local minima can be an unfortunate menace with $p < 1$. For instance when $p \to 0$, we get the the canonical FOCUSS algorithm, and it has a combinatorial number of local minima satisfying

$$d_s - 1 C_{d_y} + 1 \leqslant \#\text{ of FOCUSS Local Minima} \leqslant_{d_s} C_{d_y}. \tag{6.28}$$

which is a huge number for practical lead-field matrices, and this property largely explains the sensitivity of FOCUSS to initialization and noise. While the FOCUSS cost function can be shown to have zero localization bias at the global solution, because of the tendency to become stuck at local optima, in practice a bias can be observed when recovering even a single dipolar source. Other selections of p between zero and one can lead to a similar fate. In the general case, a natural trade-off exists with ***s**-MAP* procedures: greater sparsity of solutions at the global minimum the less possibility that this minimum is biased, but the higher the chance of suboptimal convergence to a biased local minimum, and the optimal balance could be application dependent. Nevertheless, ***s**-MAP* is capable of successfully handling large numbers of diverse covariance components, and therefore simultaneous constraints on the source space.

6.3.4 Variational Bayesian Approximation

From the perspective of a Bayesian purist however, the pursuit of MAP estimates for unknown quantities of interest, whether parameters $\boldsymbol{s}$ or hyperparameters γ, can be misleading since these estimates discount uncertainty and may not reflect regions of significant probability mass, unlike (for example) the posterior mean. Variational Bayesian methods, which have successfully been applied to a wide variety of hierarchical Bayesian models in the machine learning literature offer an alternative to $\boldsymbol{s}$-*MAP* and γ-*MAP*. Therefore, a third possibility involves finding formal approximations to $p(\boldsymbol{s}|\boldsymbol{y})$ as well as the marginal $p(\boldsymbol{y})$ using an intermediary approximation for $p(\boldsymbol{y})$. However, because of the intractable integrations involved in obtaining either distribution, practical implementation requires additional assumptions leading to different types of approximation strategies. The principle idea here is that all unknown quantities should either be marginalized (integrated out) when possible or approximated with tractable distributions that reflect underlying uncertainty and have computable posterior moments. Practically, we would like to account for ambiguity regarding γ when estimating $p(\boldsymbol{s}|\boldsymbol{y})$, and potentially, we would like a good approximation for $p(\boldsymbol{y})$, or a bound on the model evidence $\log p(\boldsymbol{y})$ for application to model selection. The only meaningful difference between VB and γ-*MAP*, at least in the context of the proposed generative model, involves approximations to the model evidence $\log p(\boldsymbol{y})$, with VB and γ-*MAP* giving different estimates.

In this section, we discuss two types of variational approximations germane to the source localization problem: the mean field approximation (VB-MF), and a fixed-form, Laplace approximation (VB-LA). It turns out that both are related to γ-*MAP* but with important distinctions. A mean-field approximation makes the simplifying assumption that the joint distribution over unknowns $\boldsymbol{s}$ and γ and factorizes, meaning $p(\boldsymbol{s}, \gamma|\boldsymbol{y}) \approx \hat{p}(\boldsymbol{s}|\boldsymbol{y})\hat{p}(\gamma|\boldsymbol{y})$ where $\hat{p}(\boldsymbol{s}|\boldsymbol{y})$ and $\hat{p}(\gamma|\boldsymbol{y})$ are chosen to minimize the Kullback-Leibler divergence between the factorized and full posterior. This is accomplished via an iterative process akin to EM, effectively using two E-steps (one for $\boldsymbol{s}$ and one for γ). It also produces a rigorous lower bound on $\log p(\boldsymbol{y})$ similar to γ-*MAP*. A second possibility applies a second-order Laplace approximation to the posterior on the hyperparameters (after marginalizing over the sources $\boldsymbol{s}$), which is then iteratively matched to the true posterior; the result can then be used to approximate $p(\boldsymbol{s}|\boldsymbol{y})$ and $\log\, p(\boldsymbol{y})$. Both of these VB methods lead to posterior approximations $\hat{p}(\boldsymbol{s}|\boldsymbol{y}) = p(\boldsymbol{s}|\boldsymbol{y}, \gamma = \hat{\gamma})$, $\hat{\gamma}$ is equivalently computed via γ-*MAP*. Consequently, VB has the same level of component pruning as γ-*MAP*.

6.3.4.1 Mean-Field Approximation

The basic strategy here is to replace intractable posterior distributions with approximate ones that, while greatly simplified and amenable to simple inference procedures, still retain important characteristics of the full model. In the context of our presumed model structure, both the posterior source distribution $p(\boldsymbol{s}|\boldsymbol{y})$, which is maximized

with ***s*-*MAP***, as well as the hyperparameter posterior $(\gamma|\boldsymbol{y})$, which is maximized via γ-*MAP*, are quite complex and can only be expressed up to some unknown scaling factor (the integration required for normalization is intractable). Likewise, the joint posterior $p(\boldsymbol{s}, \gamma|\boldsymbol{y})$ over all unknowns is likewise intractable and complex. VB attempts to simplify this situation by finding an approximate joint posterior that factorizes as

$$p(\boldsymbol{s}, \gamma|\boldsymbol{y}) \approx \hat{p}(\boldsymbol{s}, \gamma|\boldsymbol{y}) = \hat{p}(\boldsymbol{s}|\boldsymbol{y})\hat{p}(\gamma|\boldsymbol{y}), \tag{6.29}$$

where $\hat{p}(\boldsymbol{s}|\boldsymbol{y})$ and $\hat{p}(\gamma|\boldsymbol{y})$ are amenable to closed-form computation of posterior quantities such as means and variances (unlike the full posteriors upon which our model is built). This is possible because the enforced factorization, often called the mean-field approximation reflecting its origins in statistical physics, simplifies things significantly. The cost function optimized to find this approximate distribution is

$$\hat{p}(\boldsymbol{s}|\boldsymbol{y}), \hat{p}(\gamma|\boldsymbol{y}) = \underset{q(\boldsymbol{s}),q(\gamma)}{\operatorname{argmin}} \mathrm{KL}[q(\boldsymbol{s})q(\gamma)||p(\boldsymbol{s}, \gamma|\boldsymbol{y})], \tag{6.30}$$

where $q(\boldsymbol{s})$ and $q(\gamma)$ are arbitrary probability distributions and KL[.||.] indicates the Kullback-Leibler divergence measure.

Recall that γ-*MAP* iterations effectively compute an approximate distribution for $\boldsymbol{s}$ (E-step) and then a point estimate for γ (M-step); ***s*-*MAP*** does the exact opposite. In contrast, here an approximating distribution is required for both parameters $\boldsymbol{s}$ and hyperparameters γ. While it is often convenient that conjugate hyperpriors must be employed such that Eq. (6.30) is solvable, in fact this problem can be solved by coordinate descent over $q(\boldsymbol{s})$ and $q(\gamma)$ for virtually any hyperprior. It can be shown that γ-*MAP* and mean-field approximations can be equivalent in terms of the cost function being optimized and the source activity estimates obtained, given certain choice of hyperpiors [10]. Nevertheless offer a whole class of algorithms within this framework with different covariance component sets, and possible hyperpriors selected, and how the optimization is performed. The main advantage of VB is that strict lower bounds on $\log p(\boldsymbol{y})$ automatically fall out of the VB framework, given by:

$$\begin{aligned} \log p(\boldsymbol{y}) \geqslant F &\triangleq \log p(\boldsymbol{y}) - \mathrm{KL}[q(\boldsymbol{s})q(\gamma)||p(\boldsymbol{s}, \gamma|\boldsymbol{y})] \\ &= \int \hat{p}(\boldsymbol{s}|\boldsymbol{y})\hat{p}(\gamma|\boldsymbol{y}) \log \frac{p(\boldsymbol{y}, \boldsymbol{s}, \gamma)}{\hat{p}(\boldsymbol{s}|\boldsymbol{y})\hat{p}(\gamma|\boldsymbol{y})} \mathrm{d}_{\gamma}, \end{aligned} \tag{6.31}$$

where the inequality follows by the non-negativity of the Kullback-Leibler divergence. The quantity F is sometimes referred to the variational free energy. Evaluation of F requires the full distribution $\hat{p}(\gamma|\boldsymbol{y})$ and therefore necessitates using conjugate priors or further approximations.

6.3.4.2 Laplace Approximation

The Laplace approximation has been advocated to finding a tractable posterior distribution on the hyperparameters and then using this $\hat{p}(\gamma|\boldsymbol{y})$ to find approximations

to $p(\boldsymbol{s}|\boldsymbol{y})$ and $\log p(\boldsymbol{y})$. To facilitate this process, the hyperparameters are re-parameterized via the transformation $\lambda_i \triangleq \log \gamma_i, \forall i$, which now allows them to have positive or negative values. The Laplace approximation then involves the assumption that the posterior distribution of these new hyperparameters $\boldsymbol{\lambda} \triangleq [\lambda_1, \ldots, \lambda_{d_\gamma}]^T$ satisfies $p(\boldsymbol{\lambda}|\boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{\lambda}|\mu_{\boldsymbol{\lambda}}, \Sigma_{\boldsymbol{\lambda}})$. There are a variety of ways $\mu_{\boldsymbol{\lambda}}$ and $\Sigma_{\boldsymbol{\lambda}}$ can be chosen to form the best approximation. For instance, a MAP estimate of $\boldsymbol{\lambda}$ can be computed by maximizing $\log p(\boldsymbol{y}, \boldsymbol{\lambda})$ using a second-order Fisher scoring procedure [1], although this method not guaranteed to increase the cost function and requires a very expensive $O(d_\gamma^3)$ inverse computation. Once the mode $\hat{\boldsymbol{\lambda}}$ is obtained, setting $\mu_\lambda = \hat{\boldsymbol{\lambda}}$ and match the second-order statistics of the true and approximate distributions at this mode using $\Sigma_\lambda = -[\ell''(\mu_{\boldsymbol{\lambda}})]^{-1}$ where $\ell(\mu_{\boldsymbol{\lambda}}) \triangleq \log p(\boldsymbol{y}, \boldsymbol{\lambda} = \hat{\boldsymbol{\lambda}})$ and $\ell''(\mu_{\boldsymbol{\lambda}})$ is the corresponding Hessian matrix evaluated at $\mu_{\boldsymbol{\lambda}}$. Furthermore, because of the non-negativity of the KL divergence, it always holds that

$$\begin{aligned} \log p(\boldsymbol{y}) \geqslant F &\triangleq \log p(\boldsymbol{y}) - \mathrm{KL}[\mathcal{N}(\boldsymbol{\lambda}|\mu_{\boldsymbol{\lambda}}, \Sigma_{\boldsymbol{\lambda}})||p(\boldsymbol{\lambda}|\boldsymbol{y})] \\ &= \int \mathcal{N}(\boldsymbol{\lambda}|\mu_{\boldsymbol{\lambda}}, \Sigma_{\boldsymbol{\lambda}}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\lambda})}{\mathcal{N}(\boldsymbol{\lambda}|\mu_{\boldsymbol{\lambda}}, \Sigma_{\boldsymbol{\lambda}})} \mathrm{d}\boldsymbol{\lambda}. \end{aligned} \tag{6.32}$$

Unfortunately however, this bound cannot be computed in closed form, so the computable approximation is used as a surrogate instead [1].

In summary, the approximate distributions $\hat{p}(\boldsymbol{s}|\boldsymbol{y})$ obtained from both variational methods can be shown to be equivalent to a γ-*MAP* estimate given the appropriate equalizing hyperprior. Consequently, the optimization procedures for γ-*MAP* can be adapted for VB. Furthermore, the sparsifying properties of the underlying γ-*MAP* cost functions also apply to VB. Whereas VB-MF gives more principled bounds on the likelihood, the Laplace approximation methods do not. However, some limitations of the Laplace's approximation method can potentially be ameliorated by combining the Laplace and mean-field approximations as outline in Appendix D of [13].

6.4 Selection of Covariance Components C

While details and assumptions may differ γ-*MAP*, $\boldsymbol{s}$-*MAP*, and *VB* can all be leveraged to obtain a point estimate of S expressed in the Tikhonov regularized form.

$$\hat{S} = \hat{\Sigma}_S \boldsymbol{L}^T (\Sigma_\epsilon + \boldsymbol{L}\hat{\Sigma}_S \boldsymbol{L}^T)^{-1} \boldsymbol{y} \quad \text{with} \quad \hat{\Sigma}_S = \sum_i \hat{\gamma}_i C_i. \tag{6.33}$$

Importantly, since each method intrinsically sets $\hat{\gamma}_i = 0$ for superfluous covariance components, we can in principle allow the cardinality of C to be very large, and rely on a data-learning process to select the most appropriate subset. Specific choices for C lead to a variety of established algorithms as discussed next.

In the simplest case, the single-component assumption $\Sigma_S = \gamma_1 C_1 = \gamma_1 I$, where I is the Identity matrix, leads to a weighted minimum-$\ell 2$-norm algorithm. More

interesting covariance component terms have been used to effect spatial smoothness, depth bias compensation, and candidate locations of likely activity. With regard to the latter, it has been suggested that prior information about a source location can be coded by including a second term C_2 i.e. $C = C_1 C_2$, with all zeros except a patch of 1's along the diagonal signifying a location of probable source activity, perhaps based on fMRI data. For ***s*-*MAP***, γ-*MAP*, *VB*, we will obtain a source estimate representing a trade-off (modulated by the relative values of the associated $\hat{\gamma}_1$ and $\hat{\gamma}_2$) between honoring the prior information imposed by C_2 and the smoothness implied by C_1. The limitation of this proposal is that we generally do not know, a priori, the regions where activity is occurring with both high spatial and temporal resolution. Therefore, we cannot reliably known how to choose an appropriate location-prior term C_2 in many situations. A potential solution to this dilemma is to try out many different (or even all possible) combinations of location priors. For example, if we assume the underlying source currents are formed from a collection of dipolar point sources located at each vertex of the leadfield grid, then we may choose $C = \boldsymbol{e}_i \boldsymbol{e}_i^T \; i = 1, \ldots, d_S$, where each $\boldsymbol{e}_i$ is a standard indexing vector of zeros with a "1" for the ith element (and so $C_i = \boldsymbol{e}_i \boldsymbol{e}_i^T$ encodes a prior preference for a dipolar source at location i). This specification for the prior involves the counterintuitive addition of an unknown hyperparameter for every candidate source location which, on casual analysis may seem prone to severe overfitting. As suggested previously however, ***s*-*MAP***, γ-*MAP*, *VB* all possess an intrinsic, sparsity-based regularization mechanism. This ameliorates the overfitting problem substantially and effectively reduces the space of possible active source locations by choosing a small relevant subset of active dipolar locations. In general, the methodology is quite flexible and other prior specifications can be included as well, such as temporal and spectral constraints.

In summary, there are two senses with which to understand the notion of covariance component selection. First, there is the selection of which components to include in the model before any estimation takes place, i.e., the choice of C. Second, there is the selection that occurs within C as a natural byproduct of many hyperparameters being driven to zero during the learning process. Such components are necessarily pruned by the model; those that remain have therefore been 'selected' in some sense. Here we have argued that in many cases the later, data-driven selection can be used to ease the burden of often ad hoc user-specified selections.

6.5 Discussion

The efficacy of modern Bayesian techniques for quantifying uncertainty and explicitly accounting for prior assumptions make them attractive candidates for source localization. However, it is not always transparent how these methods relate, nor how they can be extended to handle more challenging problems, nor which ones should be expected to perform best in various situations relevant to MEG/EEG source imaging.APtarting from a hierarchical Bayesian model constructed using Gaussian scale mixtures with flexible covariance components, we analyze and, where pos-

sible, extend three broad classes of Bayesian inference methods: γ-*MAP*, which involves integrating out the unknown sources and optimizing the hyperparameters, and $\boldsymbol{s}$-*MAP*, which integrates out the hyperparameters and directly optimizes over the sources, and variational approximation methods, which attempt to account for uncertainty in all unknowns. Together, these three encompass a surprisingly wide range of existing source reconstruction approaches, which makes general theoretical analyses and algorithmic extensions/improvements pertaining to the particularly relevant. Thus far, we have attempted to relate and extend three large classes of Bayesian inverse methods, all of which turn out to be performing covariance component estimation/pruning using different sparsity promoting regularization procedures. We now provide some summary points related to connections to existing methods.

1. $\boldsymbol{s}$-*MAP*, γ-*MAP*, and *VB* can be viewed as procedures for learning a source covariance model using a set of predetermined symmetric, positive semi-definite covariance components. The number of components in this set, each of which acts as a constraint on the source space, can be extremely large, potentially much larger than the number of sensors. However, a natural pruning mechanism effectively discards components that are unsupported by the data. This occurs because of an intrinsic sparsity preference in the Gaussian scale mixture model, which is manifested in an explicit sparsityinducing regularization term. Consequently, it is not crucial that the user/analyst manually determine an optimal set of components a priori; many components can be included initially allowing the learning process to remove superfluous ones.
2. The wide variety of Bayesian source localization methods that fall under this framework can be differentiated by the following factors: (1) Selection of covariance component regularization term; (2) Choice of initial covariance component set C; (3) Optimization method/ Update rules; and (4) Approximation to $\log p(\boldsymbol{y})$; this determines whether we are ultimately performing $\boldsymbol{s}$-*MAP*, γ-*MAP*, or *VB*.
3. Covariance component possibilities include geodesic neural basis functions for estimating distributed sources [34], spatial smoothing factors [24], indicator matrices to couple dipole components or learn flexible orientations [36], fMRI-based factors [31], and temporal and spectral constraints [21].
4. With large numbers of covariance components, $\boldsymbol{s}$-*MAP*, γ-*MAP*, and *VB* provably remove or prune a certain number of components which are not necessary for representing the observed data.
5. In principle, the noise-plus-interference covariance can be jointly estimated as well, competing with all the other components to model the data. However, identifiability issues can be a concern here and so we consider it wiser to estimate Σ_ϵ via other means (e.g., using VBFA applied to prestimulus data as described in Chap. 5).
6. The latent structure inherent to the Gaussian scale-mixture model leads to an efficient, principled family of update rules for $\boldsymbol{s}$-*MAP*, γ-*MAP*, and *VB*. This facilitates the estimation of complex covariance structures modulated by very large numbers of hyperparameters (e.g., 10^5+) with relatively little difficulty.

7. Previous focal source imaging techniques such as FOCUSS and MCE display undesirable discontinuities across time as well significant biases in estimating dipole orientations. Consequently, various heuristics have been proposed to address these deficiencies. However, the general spatiotemporal framework of ***s*-*MAP***, γ-*MAP*, and *VB* handles both of these concerns in a robust, principled fashion by the nature of their underlying cost function. The standard weighted minimum norm can be seen as a limiting case of γ_{MAP}.
8. As described in other chapters here, adaptive beamformers are spatial filters that pass source signals in particular focused locations while suppressing interference from elsewhere. The widely-used minimum variance adaptive beamformer (MVAB) creates such filters using a sample covariance estimate; however, the quality of this estimate deteriorates when the sources are correlated or the number of samples n is small. The simpler γ-*MAP* strategy can also be used to enhance beamforming in a way that is particularly robust to source correlations and limited data [15]. Specifically, the estimated γ-*MAP* data covariance matrix $\hat{\Sigma}_y = \Sigma_\epsilon + \sum_i \hat{\gamma}_i \boldsymbol{L} C_i \boldsymbol{L}^T$ can be used to replace the problematic sample covariance $C_y = \boldsymbol{y}\boldsymbol{y}^T$. This substitution has the natural ability to remove the undesirable effects of correlations or limited data. When n becomes large and assuming uncorrelated sources, this method reduces to the exact MVAB. Additionally, the method can potentially enhance a variety of traditional signal processing methods that rely on robust sample covariance estimates.
9. It can be shown that sLORETA is equivalent to performing a single iteration of a particular γ-MAP optimization procedure. Consequently, the latter can be viewed as an iterative refinement of sLORETA. This is exactly analogous to the view of FOCUSS as an iterative refinement of a weighted minimum $\ell 2$-norm estimate.
10. γ-MAP and VB have theoretically zero localization bias estimating perfectly uncorrelated dipoles given the appropriate hyperprior and initial set of covariance component.
11. The role of the hyperprior $p(\gamma)$ is heavily dependent on the estimation algorithm being performed. In the ***s*-*MAP*** framework, the hyperprior functions through its role in creating the concave regularization function $g_i(.)$. In practice, it is much more transparent to formulate a model directly based on a desired $g_i(.)$ as opposed to working with some supposedly plausible hyperprior $p(\gamma)$ and then inferring the what the associated $g_i(.)$ would be. In contrast, with γ-*MAP* and *VB* the opposite is true. Choosing a model based on the desirability of some $g_i(.)$ can lead to a model with an underlying hyperprior $p(\gamma)$ that performs poorly. Both *VB* and γ-*MAP* give rigorous bounds on the model evidence $\log p(\boldsymbol{y})$.

In summary, we hope that these ideas help to bring an insightful perspective to Bayesian source imaging methods, reduce confusion about how different techniques relate, expand the range of feasible applications of these methods. We have observed a number of surprising similarities or out-right equivalences between what might otherwise appear to be very different methodologies. Additionally, there are numerous promising directions for future research, including time-frequency extensions,

alternative covariance component parameterizations, and robust interference suppression that inspire our own current body of work in this area, and we hope will also inspire other researchers!

References

1. K. Friston, L. Harrison, J. Daunizeau, S. Kiebel, C. Phillips, N. Trujillo-Barreto, R. Henson, G. Flandin, J. Mattout, Multiple sparse priors for the M/EEG inverse problem. NeuroImage **39**(3), 1104–1120 (2008)
2. A. Bolstad, B.V. Veen, R. Nowak, Space-time event sparse penalization for magneto-/electroencephalography. NeuroImage **46**(4), 1066–1081 (2009)
3. I.F. Gorodnitsky, J.S. George, B.D. Rao, Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm. Electroencephalogr. Clin. Neurophysiol. **95**, 231–251 (1995)
4. K. Uutela, M. Hämäläinen, E. Somersalo, Visualization of magnetoencephalographic data using minimum current estimate. NeuroImage **10**, 173–180 (1999)
5. M.-X. Huang, A.M. Dale, T. Song, E. Halgren, D.L. Harrington, I. Podgorny, J.M. Canive, S. Lewis, R.R. Lee, Vector-based spatial-temporal minimum l1-norm solution for MEG. NeuroImage **31**(3), 1025–1037 (2006)
6. R.D. Pascual-Marqui, Standardized low resolution brain electromagnetic tomography (sLORETA): technical details. Methods Find. Exp. Clin. Pharmacol. **24**, 5–12 (2002)
7. K.J. Friston, W. Penny, C. Phillips, S. Kiebel, G. Hinton, J. Ashburner, Classical and Bayesian inference in neuroimaging: theory. NeuroImage **16**(2), 465–483 (2002)
8. D.J.C. MacKay, Bayesian interpolation. Neural Comput. **4**, 415–447 (1992)
9. R.R. Ramírez, S. Makeig, Neuroelectromagnetic source imaging using multiscale geodesic neural bases and sparse Bayesian learning, in *12th Annual Meeting of the Organization for Human Brain Mapping* (Florence, Italy, 2006)
10. D.P. Wipf, J.P. Owen, H.T. Attias, K. Sekihara, S.S. Nagarajan, Robust Bayesian estimation of the location, orientation, and time course of multiple correlated neural sources using MEG. NeuroImage **49**, 641–655 (2010)
11. J. Mattout, C. Phillips, W.D. Penny, M.D. Rugg, K.J. Friston, MEG source localization under multiple constraints: an extended Bayesian framework. NeuroImage **30**(3), 753–767 (2006)
12. M.-A. Sato, T. Yoshioka, S. Kajihara, K. Toyama, N. Goda, K. Doya, M. Kawato, Hierarchical Bayesian estimation for MEG inverse problem. NeuroImage **23**(3), 806–826 (2004)
13. D. Wipf, S. Nagarajan, A unified Bayesian framework for MEG/EEG source imaging. NeuroImage **44**(3), 947–966 (2009)
14. D.P. Wipf, B.D. Rao, S. Nagarajan, Latent variable Bayesian models for promoting sparsity. IEEE Trans. Inf. Theory **57**(9), 6236–6255 (2011)
15. D. Wipf, S. Nagarajan, Beamforming using the relevance vector machine, in *Proceedings of the 24th International Conference on Machine Learning* (ACM, 2007), pp. 1023–1030

Chapter 7
Source-Space Connectivity Analysis Using Imaginary Coherence

7.1 Introduction

There has been tremendous interest in estimating the functional connectivity of neuronal activities across different brain regions using electromagnetic brain imaging. Functional connectivity analysis has traditionally been implemented in the sensor space, but lately, a number of studies have begun to use source-space analysis, in which voxel time courses are first estimated by an inverse algorithm, and brain interactions are then analyzed using those estimated voxel time courses [1–3]. Although a certain degree of inaccuracy exists in the source estimation process, the source-space analysis has the potential of providing more accurate information regarding which brain regions are functionally coupled.

Source-space connectivity analysis computes a metric of brain interaction called a connectivity metric, using the voxel time courses. Among existing connectivity metrics, a representative metric is the coherence [2–5]. However, in the source-space coherence analysis, a serious problem arises from spurious coherence caused by the leakage of an inverse algorithm. Such leakages are more or less inevitable in all types of inverse algorithms [6]. One representative ramification of this spurious coherence is an artifactual large peak around the seed voxel, called seed blur, in the resulting coherence image. To remove such spurious coherence, the use of the imaginary part of coherence, which is called the imaginary coherence, has been proven to be effective [7]. It should be mentioned that the use of imaginary coherence was originally proposed by Nolte et al. [8] to remove the spurious coherence caused by the volume conduction in EEG sensor-space analysis.

This chapter reviews the source-space imaginary coherence analysis and related methods. We first provide a detailed theoretical analysis on how the use of imaginary coherence leads to the removal of the spurious coherence caused by the algorithm leakage. We then discuss several related methods of imaginary coherence, including corrected imaginary coherence, canonical coherence, and envelope-to-envelope correlation/coherence. We present numerical examples that confirm our arguments.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_7

7.2 Source-Space Coherence Imaging

In the source-space analysis, the first step estimates voxel time courses using an inverse algorithm. Since the source is a three-dimensional vector with three (x, y, and z) components,[1] most source reconstruction algorithms produce component-wise, multiple time courses at each voxel. Accordingly, we should compute a "representative" single time course by projecting the multiple time courses onto the direction of the source orientation.

However, in practical applications, the source orientation is generally unknown, and it must be estimated from the data. One quick and easy way to estimate the source orientation is to use the direction that maximizes the reconstructed voxel power. The reconstructed source-time courses at the jth voxel is expressed as

$$\widehat{\boldsymbol{S}}_j = \begin{bmatrix} \widehat{s}_x(\boldsymbol{r}_j, t_1)\ \widehat{s}_x(\boldsymbol{r}_j, t_2) \cdots \widehat{s}_x(\boldsymbol{r}_j, t_K) \\ \widehat{s}_y(\boldsymbol{r}_j, t_1)\ \widehat{s}_y(\boldsymbol{r}_j, t_2) \cdots \widehat{s}_y(\boldsymbol{r}_j, t_K) \\ \widehat{s}_z(\boldsymbol{r}_j, t_1)\ \widehat{s}_z(\boldsymbol{r}_j, t_2) \cdots \widehat{s}_z(\boldsymbol{r}_j, t_K) \end{bmatrix}, \tag{7.1}$$

where we assume that the data is collected at time points, $t_1, t_2, \ldots, t_K$, and $\boldsymbol{r}_j$ indicates the location of the jth voxel. Denoting the orientation of the source at the jth voxel as $\boldsymbol{\eta}_j$, the estimate of $\boldsymbol{\eta}_j$, $\widehat{\boldsymbol{\eta}}_j$, is obtained using the following maximization:

$$\widehat{\boldsymbol{\eta}}_j = \underset{\boldsymbol{\eta}_j}{\operatorname{argmax}}\, \boldsymbol{\eta}_j^T \left(\widehat{\boldsymbol{S}}_j \widehat{\boldsymbol{S}}_j^T \right) \boldsymbol{\eta}_j.$$

The optimum estimate is obtained as the eigenvector corresponding to the maximum eigenvalue of a matrix $\widehat{\boldsymbol{S}}_j \widehat{\boldsymbol{S}}_j^T$, i.e.,

$$\widehat{\boldsymbol{\eta}}_j = \vartheta_{\max}\{\widehat{\boldsymbol{S}}_j \widehat{\boldsymbol{S}}_j^T\}, \tag{7.2}$$

where the notation $\vartheta_{\max}\{\cdot\}$, defined in Sect. C.9, indicates the eigenvector corresponding to the maximum eigenvalue of a matrix between the parentheses. The representative time course at the jth voxel, $u_j(t)$, is obtained using

$$\left[u_j(t_1), u_j(t_2), \ldots, u_j(t_K)\right] = \widehat{\boldsymbol{\eta}}_j^T \widehat{\boldsymbol{S}}_j. \tag{7.3}$$

Once a representative voxel time course is obtained at each voxel, the next step is to compute a voxel-pairwise coherence. This step involves first setting a reference point, called the seed point, and computing the coherence between the time course from the seed point and that from another voxel's location, referred to as the target location. By scanning through all target locations in a brain, a three-dimensional

[1] Although the source has two components when the homogeneous spherical conductor model is used, for arguments in this chapter, we assume that the source vector has three x, y, and z components.

mapping of source coherence—a source coherence image with respect to the seed location—can be obtained. We can further scan not only the target location but also the seed location to obtain a six-dimensional coherence image, called the voxel-to-voxel coherence matrix.

Let us define the spectra of the seed and the target voxels as $\sigma_S(f)$ and $\sigma_T(f)$, respectively. The coherence $\phi(f)$ is obtained by computing the correlation of these spectra,

$$\phi(f) = \frac{\langle \sigma_T(f)\sigma_S^*(f)\rangle}{\sqrt{\langle|\sigma_T(f)|^2\rangle\langle|\sigma_S(f)|^2\rangle}}, \tag{7.4}$$

where the superscript $*$ indicates the complex conjugate, and the brackets $\langle\cdot\rangle$ indicate the ensemble average. In practical applications, this ensemble average is computed by averaging across multiple trials. When only a single continuous data set is measured, the single data set is divided into many trials and coherence is obtained by averaging across these trials.

It is apparent in Eq. (7.4) that if the seed and target spectra contain common components that do not result from true brain interactions, then the coherence may contain spurious components. In source coherence imaging, the leakage of the imaging algorithm is a major source of such spurious coherence.

7.3 Real and Imaginary Parts of Coherence

We here take a look at the nature of real and imaginary parts of coherence before proceeding with the arguments of the leakage influence on the coherence imaging. Let us define the time course from the seed voxel as $u_S(t)$ and the time course from the target voxel as $u_T(t)$. The cross correlation of the seed and target time courses, $R(\tau)$, is then defined such that

$$R(\tau) = \langle u_T(t+\tau)u_S(t)\rangle = \int_{-\infty}^{\infty} u_T(t+\tau)u_S(t)\,\mathrm{d}t. \tag{7.5}$$

The cross correlation is related to the cross spectrum density through

$$R(\tau) = \int_{-\infty}^{\infty} \Psi(f)\mathrm{e}^{-i2\pi f\tau}\,\mathrm{d}f, \tag{7.6}$$

where the cross spectrum is

$$\Psi(f) = \langle \sigma_T(f)\sigma_S^*(f)\rangle. \tag{7.7}$$

According to Eq. (7.6), we have

$$R(0) = \int_{-\infty}^{\infty} \Psi(f)\,\mathrm{d}f = \int_{-\infty}^{\infty} \Re\left[\Psi(f)\right]\,\mathrm{d}f = \int_{-\infty}^{\infty} \Re\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right]\,\mathrm{d}f, \tag{7.8}$$

where $\Re[\cdot]$ indicates the real part of the complex number in the square brackets. The right-hand side of the above equation holds because the real part of $\Psi(f)$ is an even function and the imaginary part is an odd function. (This is because $R(\tau)$ is a real-valued function.) Therefore, for a narrow-band signal,

$$R(0) = \int_{-\infty}^{\infty} \Re\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right]\,\mathrm{d}f \approx \Re\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right]. \tag{7.9}$$

Here, $R(0)$, which represents the zero-time-lag correlation, is related to the real part of the cross spectrum, and thus, the real part of coherence represents the instantaneous interaction.

Applying Parseval's theorem to the Fourier transform relationship in Eq. (7.6), we get

$$\int_{-\infty}^{\infty} R(\tau)^2\,\mathrm{d}\tau = \int_{-\infty}^{\infty} |\Psi(f)|^2\,\mathrm{d}f = \int_{-\infty}^{\infty} |\langle \sigma_T(f)\sigma_S^*(f)\rangle|^2\,\mathrm{d}f. \tag{7.10}$$

Thus, for a narrow-band signal, the following relationship holds:

$$\int_{-\infty}^{\infty} R(\tau)^2\,\mathrm{d}\tau \approx |\langle \sigma_T(f)\sigma_S^*(f)\rangle|^2 = \Re\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right] + \Im\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right], \tag{7.11}$$

where $\Im[\cdot]$ indicate the imaginary part of the complex number in the square brackets. Using Eq. (7.9), we have

$$\Im\left[\langle \sigma_T(f)\sigma_S^*(f)\rangle\right] \approx \int_{-\infty}^{\infty} R(\tau)^2\,\mathrm{d}\tau - R(0)^2 = \sum_{\tau\neq 0} R(\tau)^2. \tag{7.12}$$

The imaginary part of the cross spectrum is equal to the sum of nonzero-lag correlations. Therefore, we can see that the imaginary part of coherence represents the non-instantaneous interaction.

7.4 Effects of the Leakage

7.4.1 Leakage Effects on the Magnitude Coherence

We analyze the effects of leakage of imaging algorithms in source coherence imaging. When the leakage is taken into account, the estimated source time courses at the seed and the target voxels, $\widehat{u}_S(t)$ and $\widehat{u}_S(t)$, can be expressed as

$$\widehat{u}_S(t) = u_S(t) + d_1 u_T(t) + c_S(t), \tag{7.13}$$

$$\widehat{u}_T(t) = u_T(t) + d_2 u_S(t) + c_T(t). \tag{7.14}$$

where $u_S(t)$ and $u_T(t)$ are the true source time courses of the seed and the target locations. In the above equations, $d_1 u_T(t)$ indicates the leakage of the target signal in the estimated seed signal, and $d_2 u_S(t)$ indicates the leakage of the seed signal in the estimated target signal. The real-valued constants d_1 and d_2 express the relative amount of these leakages.

In the equations above, $c_S(t)$ and $c_T(t)$ express the interference terms, which may include the leakage from the third source, contributions from external disturbances, and that of the sensor noise. The influence of these interference terms can be considered separately from the influence of the leakage terms [7]. In the arguments here, ignoring $c_S(t)$ and $c_T(t)$, the estimated spectra at the seed and target voxels, $\widehat{\sigma}_S(f)$ and $\widehat{\sigma}_T(f)$, are expressed as

$$\widehat{\sigma}_S = \sigma_S + d_1 \sigma_T, \tag{7.15}$$

$$\widehat{\sigma}_T = \sigma_T + d_2 \sigma_S. \tag{7.16}$$

where we omit the explicit notation of (f) for simplicity.

The magnitude coherence between the seed and the target voxels is expressed as

$$|\widehat{\phi}| = \left| \frac{\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle}{\sqrt{\langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle}} \right|. \tag{7.17}$$

Using Eqs. (7.15) and (7.16), we have

$$\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle = \langle \sigma_T \sigma_S^* \rangle + d_1 \langle |\sigma_T|^2 \rangle + d_2 \langle |\sigma_S|^2 \rangle + d_1 d_2 \langle \sigma_S \sigma_T^* \rangle, \tag{7.18}$$

$$\langle |\widehat{\sigma}_T|^2 \rangle = \langle |\sigma_S|^2 \rangle + d_1^2 \langle |\sigma_T|^2 \rangle + 2 d_1 \Re \left(\langle \sigma_T \sigma_S^* \rangle \right), \tag{7.19}$$

$$\langle |\widehat{\sigma}_S|^2 \rangle = \langle |\sigma_T|^2 \rangle + d_2^2 \langle |\sigma_S|^2 \rangle + 2 d_2 \Re \left(\langle \sigma_T \sigma_S^* \rangle \right). \tag{7.20}$$

The equations above show that even when there is no true source interaction, i.e., even when $\langle \sigma_T \sigma_S^* \rangle = 0$ and $\langle \sigma_S \sigma_T^* \rangle = 0$, $|\widehat{\phi}|$ has a nonzero value, which is equal to

$$|\widehat{\phi}| = \frac{|d_1\langle|\sigma_T|^2\rangle + d_2\langle|\sigma_S|^2\rangle|}{\sqrt{\left(\langle|\sigma_S|^2\rangle + d_1^2\langle|\sigma_T|^2\rangle\right)\left(\langle|\sigma_T|^2\rangle + d_2^2\langle|\sigma_S|^2\rangle\right)}}. \tag{7.21}$$

7.4.2 Leakage Effects on the Imaginary Coherence

We next analyze the effects of the algorithm leakage on the imaginary coherence. Using Eq. (7.18) and the relationship

$$\langle\sigma_T\sigma_S^*\rangle + \langle\sigma_S\sigma_T^*\rangle = 2\Re\left(\langle\sigma_S\sigma_T^*\rangle\right),$$

the cross spectrum $\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle$ can be expressed as

$$\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle = (1 - d_1d_2)\langle\sigma_T\sigma_S^*\rangle + d_1\langle|\sigma_T|^2\rangle + d_2\langle|\sigma_S|^2\rangle + 2d_1d_2\Re\left(\langle\sigma_T\sigma_S^*\rangle\right). \tag{7.22}$$

By taking the imaginary part of Eq. (7.22), we can derive

$$\Im\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right) = (1 - d_1d_2)\Im\left(\langle\sigma_T\sigma_S^*\rangle\right). \tag{7.23}$$

We can then obtain the imaginary part of the estimated coherence $\Im(\widehat{\phi})$ as

$$\Im(\widehat{\phi}) = \frac{\Im\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right)}{\sqrt{\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle}} = \frac{(1 - d_1d_2)\Im\left(\langle\sigma_T\sigma_S^*\rangle\right)}{\sqrt{\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle}} = \Lambda\,\Im\left(\phi\right), \tag{7.24}$$

where $\Im\left(\phi\right)$ indicates the true value of the imaginary coherence. Using Eqs. (7.19) and (7.20), Λ is obtained as

$$\Lambda = \frac{(1 - d_1d_2)}{\sqrt{\tau_1\tau_2}} \tag{7.25}$$

where

$$\tau_1 = 1 + d_1^2\frac{\langle|\sigma_T|^2\rangle}{\langle|\sigma_S|^2\rangle} + 2d_1\frac{\Re\left(\langle\sigma_T\sigma_S^*\rangle\right)}{\langle|\sigma_S|^2\rangle}, \tag{7.26}$$

and

$$\tau_2 = 1 + d_2^2\frac{\langle|\sigma_S|^2\rangle}{\langle|\sigma_T|^2\rangle} + 2d_2\frac{\Re\left(\langle\sigma_T\sigma_S^*\rangle\right)}{\langle|\sigma_T|^2\rangle}. \tag{7.27}$$

Equation (7.24) shows that when $\Im\left(\phi\right) = 0$, we have $\Im(\widehat{\phi}) = 0$, indicating that no spurious imaginary coherence has been caused. However, Eq. (7.24) also indicates that the value of $\Im(\widehat{\phi})$ differs from the true value $\Im\left(\phi\right)$, i.e., the intensity of the estimated imaginary coherence is biased and the bias is represented by Λ in Eq. (7.25).

Let us calculate the intensity bias factor Λ assuming a simple scenario in which $d_1 = d_2 = d$ and $\langle|\sigma_S|^2\rangle = \langle|\sigma_T|^2\rangle$. Under these assumptions, the bias factor is simplified to

$$\frac{1+|d|}{1-|d|} \geq \Lambda = \frac{1-d^2}{1+d^2+2\Re(\phi)d} \geq \frac{1-|d|}{1+|d|}. \tag{7.28}$$

The equation above shows how the amount of leakage $|d|$ affects the bias factor Λ. It shows that if $|d|$ is as small as $|d| < 0.1$, the intensity bias is less than 10 %. However, when $|d|$ is as large as 0.4, we have $2.3 \geq \Lambda \geq 0.4$ and the intensity value of the imaginary coherence may have a very large bias. In the following, we introduce a metric referred to as the corrected imaginary coherence, which avoids this intensity bias.

7.5 Corrected Imaginary Coherence

7.5.1 Modification of Imaginary Coherence

The imaginary coherence can avoid spurious results caused by the algorithm leakage. However, its intensity value is affected by the algorithm leakage. In this section, we introduce a modified form of the imaginary coherence whose intensity values are unaffected by the algorithm leakage. Let us define the corrected imaginary coherence as

$$\xi = \frac{\Im(\phi)}{\sqrt{1-\Re(\phi)^2}}. \tag{7.29}$$

Let us express the estimate of ξ, $\widehat{\xi}$, such that

$$\widehat{\xi} = \frac{\Im(\widehat{\phi})}{\sqrt{1-\Re(\widehat{\phi})^2}} = \frac{\Im\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right)}{\sqrt{\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle - \Re\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right)}}.$$

Then, considering

$$\Re\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right) = d_1^2\langle|\sigma_T|^2\rangle + d_2^2\langle|\sigma_S|^2\rangle + (1+d_1d_2)\Re\left(\langle\sigma_T\sigma_S^*\rangle\right), \tag{7.30}$$

and using Eqs. (7.19) and (7.20), we derive

$$\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle - \Re\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right) = (1-d_1d_2)^2\left(\langle|\sigma_T|^2\rangle\langle|\sigma_S|^2\rangle - \Re\left(\langle\sigma_T\sigma_S^*\rangle\right)\right). \tag{7.31}$$

Combining this with Eq. (7.23), we finally obtain

$$\begin{aligned}\widehat{\xi} &= \frac{\Im(\widehat{\phi})}{\sqrt{1-\Re(\widehat{\phi})^2}} \\ &= \frac{\Im\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right)}{\sqrt{\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle - \Re\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^*\rangle\right)}} \\ &= \frac{(1-d_1d_2)\Im\left(\langle\sigma_T\sigma_S^*\rangle\right)}{(1-d_1d_2)\sqrt{\langle|\sigma_T|^2\rangle\langle|\sigma_S|^2\rangle - \Re\left(\langle\sigma_T\sigma_S^*\rangle\right)}} \\ &= \frac{\Im(\phi)}{\sqrt{1-\Re(\phi)^2}}. \end{aligned} \tag{7.32}$$

The above equation indicates that the corrected imaginary coherence computed using voxel spectra is exactly equal to the true corrected imaginary coherence. This implies that the corrected imaginary coherence is unaffected by the algorithm leakage.

In the above arguments, the corrected imaginary coherence is introduced somewhat in an ad hoc manner. In the following, we derive the corrected imaginary coherence in two different manners: factorization of the mutual coherence and regression of the target signal with the seed signal. These derivations provide some insights into the nature of corrected imaginary coherence.

7.5.2 Factorization of Mutual Information

Here we show that the corrected imaginary coherence is derived from the factorization of the mutual information into the instantaneous and non-instantaneous components. We assume that $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ are Gaussian distributed, circular complex random variables. Concise explanations on the complex Gaussian random variable are found in Sect. C.2 in the Appendix.

To derive the mutual information in the frequency domain, we define the entropy of voxel spectra $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$. To do so, we first define real-valued 2×1 vectors $\boldsymbol{\zeta}_T = [\Re(\widehat{\sigma}_T), \Im(\widehat{\sigma}_T)]^T$ and $\boldsymbol{\zeta}_S = [\Re(\widehat{\sigma}_S), \Im(\widehat{\sigma}_S)]^T$. Using these vectors, the entropy is defined for $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ such that,

$$\mathcal{H}(\widehat{\sigma}_T) = -\int p(\boldsymbol{\zeta}_T)\log p(\boldsymbol{\zeta}_T)\,\mathrm{d}\boldsymbol{\zeta}_T, \tag{7.33}$$

$$\mathcal{H}(\widehat{\sigma}_S) = -\int p(\boldsymbol{\zeta}_S)\log p(\boldsymbol{\zeta}_S)\,\mathrm{d}\boldsymbol{\zeta}_S. \tag{7.34}$$

The entropy is a metric for uncertainty. $\mathcal{H}(\widehat{\sigma}_T)$ represents the uncertainty when $\widehat{\sigma}_T$ is unknown, and $\mathcal{H}(\widehat{\sigma}_S)$ represents the uncertainty when $\widehat{\sigma}_S$ is unknown. The joint entropy is defined as

$$\mathcal{H}(\widehat{\sigma}_T, \widehat{\sigma}_S) = -\iint p(\zeta_T, \zeta_S) \log p(\zeta_T, \zeta_S) d\zeta_T d\zeta_S. \tag{7.35}$$

The mutual information between $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ is then defined as

$$\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S) = \mathcal{H}(\widehat{\sigma}_T) + \mathcal{H}(\widehat{\sigma}_S) - \mathcal{H}(\widehat{\sigma}_T, \widehat{\sigma}_S). \tag{7.36}$$

When $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ are independent, we have $\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S) = 0$.

Under the assumption that $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ are complex Gaussian distributed, the entropy is expressed as

$$\mathcal{H}(\widehat{\sigma}_T) = \log\left\langle |\widehat{\sigma}_T|^2 \right\rangle, \tag{7.37}$$

$$\mathcal{H}(\widehat{\sigma}_S) = \log\left\langle |\widehat{\sigma}_S|^2 \right\rangle. \tag{7.38}$$

The joint entropy is given by

$$\begin{aligned}
\mathcal{H}(\widehat{\sigma}_T, \widehat{\sigma}_S) &= \log \left| \left\langle \begin{bmatrix} \widehat{\sigma}_T \\ \widehat{\sigma}_S \end{bmatrix} \widehat{\phi} \begin{bmatrix} \widehat{\sigma}_T^* & \widehat{\sigma}_S^* \end{bmatrix} \right\rangle \right| \\
&= \log \begin{vmatrix} \langle |\widehat{\sigma}_T|^2 \rangle & \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle \\ \langle \widehat{\sigma}_S \widehat{\sigma}_T^* \rangle & \langle |\widehat{\sigma}_S|^2 \rangle \end{vmatrix} \\
&= \log \left(\left\langle |\widehat{\sigma}_T|^2 \right\rangle \left\langle |\widehat{\sigma}_S|^2 \right\rangle - \left| \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle \right|^2 \right)
\end{aligned} \tag{7.39}$$

Therefore, the mutual information between $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$ is obtained as

$$\begin{aligned}
\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S) &= \mathcal{H}(\widehat{\sigma}_T) + \mathcal{H}(\widehat{\sigma}_S) - \mathcal{H}(\widehat{\sigma}_T, \widehat{\sigma}_S) \\
&= \log \frac{\langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle}{\langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle - \left| \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle \right|^2}.
\end{aligned} \tag{7.40}$$

It is easy to see that the mutual information $\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S)$ is related to the magnitude coherence $|\widehat{\phi}|$ such that

$$\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S) = -\log(1 - |\widehat{\phi}|^2), \tag{7.41}$$

where

$$|\widehat{\phi}|^2 = \frac{\left| \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle \right|^2}{\langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle}. \tag{7.42}$$

Let us consider the factorization of the mutual information using the real and imaginary parts of coherence. Using $|\widehat{\phi}|^2 = \Re(\widehat{\phi})^2 + \Im(\widehat{\phi})^2$, we have

$$\begin{aligned}\mathcal{I}(\widehat{\sigma}_T, \widehat{\sigma}_S) &= -\log(1 - |\widehat{\phi}|^2) = -\log(1 - \Re(\widehat{\phi})^2 - \Im(\widehat{\phi})^2) \\ &= -\log\left[\left(1 - \Re(\widehat{\phi})^2\right)\left[(1 - \frac{\Im(\widehat{\phi})^2}{1 - \Re(\widehat{\phi})^2}\right]\right] \\ &= -\log(1 - \Re(\widehat{\phi})^2) - \log\left[1 - \frac{\Im(\widehat{\phi})^2}{1 - \Re(\widehat{\phi})^2}\right]. \end{aligned} \tag{7.43}$$

On the right-hand side, the first term,

$$\mathcal{I}_R(\widehat{\sigma}_T, \widehat{\sigma}_S) = -\log(1 - \Re(\widehat{\phi})^2)$$

represents a component of the mutual information corresponding to the real part of coherence. This $\mathcal{I}_R(\widehat{\sigma}_T, \widehat{\sigma}_S)$ can be interpreted as the instantaneous component, which corresponds to the zero-lag correlation between the target and the seed time courses, as discussed in Sect. 7.3.

The second term,

$$\mathcal{I}_I(\widehat{\sigma}_T, \widehat{\sigma}_S) = -\log\left[(1 - \frac{\Im(\widehat{\phi})^2}{1 - \Re(\widehat{\phi})^2}\right]$$

represents a component of the mutual information corresponding to the imaginary part of coherence. It is interpreted as the non-instantaneous component, which corresponds to the nonzero-lag correlation between $u_T(t)$ and $u_S(t)$. It is easy to see

$$\mathcal{I}_I(\widehat{\sigma}_T, \widehat{\sigma}_S) = -\log\left[(1 - \frac{\Im(\widehat{\phi})^2}{1 - \Re(\widehat{\phi})^2}\right] = -\log(1 - \widehat{\xi}^2), \tag{7.44}$$

where $\widehat{\xi}$ is the corrected imaginary coherence. The equation above indicates that the corrected imaginary coherence can be interpreted as a coherence-domain expression of the non-instantaneous component of the mutual information. Arguments similar to those in this section are found in [9].

7.5.3 Residual Coherence

We next show that the corrected imaginary coherence can be derived using regression analysis. Let us regress the target spectrum $\widehat{\sigma}_T$ using the seed spectrum $\widehat{\sigma}_S$, such that

$$\widehat{\sigma}_T = \alpha\widehat{\sigma}_S + v, \tag{7.45}$$

where α is a real-valued constant, and v is a residual signal of this regression. The value for α is determined using the least-squares fit:

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \langle |\widehat{\sigma}_T - \alpha \widehat{\sigma}_S|^2 \rangle, \tag{7.46}$$

and is derived as

$$\alpha = \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle}. \tag{7.47}$$

Thus, the residual signal is expressed as

$$v = \widehat{\sigma}_T - \alpha \widehat{\sigma}_S = \widehat{\sigma}_T - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle} \widehat{\sigma}_S. \tag{7.48}$$

We then define the residual coherence, $\widehat{\phi}_R$, such that

$$\widehat{\phi}_R = \frac{\langle v \widehat{\sigma}_S^* \rangle}{\sqrt{\langle |v|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle}}. \tag{7.49}$$

Let us show that this residual coherence is equal to the corrected imaginary coherence. First, the cross spectrum between the residual and seed signals is expressed as

$$\begin{aligned} \langle v \widehat{\sigma}_S^* \rangle &= \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle} \langle |\widehat{\sigma}_S|^2 \rangle \\ &= \langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle - \Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle] \\ &= i \Im[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]. \end{aligned} \tag{7.50}$$

The equation above shows that the cross spectrum between v and $\widehat{\sigma}_S$ is equal to the imaginary part of the cross spectrum between $\widehat{\sigma}_T$ and $\widehat{\sigma}_S$. Using Eq. (7.48), we express $\langle |v|^2 \rangle$ such that

$$\begin{aligned} \langle |v|^2 \rangle &= \left\langle \left[\widehat{\sigma}_T - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle} \widehat{\sigma}_S \right] \left[\widehat{\sigma}_T - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle} \widehat{\sigma}_S \right]^* \right\rangle \\ &= \langle |\widehat{\sigma}_T|^2 \rangle - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\langle |\widehat{\sigma}_S|^2 \rangle} \left(\widehat{\sigma}_S \widehat{\sigma}_T^* + \widehat{\sigma}_T \widehat{\sigma}_S^* \right) + \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]^2}{\langle |\widehat{\sigma}_S|^2 \rangle^2} \langle |\widehat{\sigma}_S|^2 \rangle \\ &= \langle |\widehat{\sigma}_T|^2 \rangle - \frac{\Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]^2}{\langle |\widehat{\sigma}_S|^2 \rangle}. \end{aligned} \tag{7.51}$$

Therefore, we get

$$\langle |v|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle = \langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle - \Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]^2. \tag{7.52}$$

Substituting Eqs. (7.52) and (7.50) into Eq. (7.49), and neglecting the constant i, the residual coherence is expressed as

$$\widehat{\phi}_R = \frac{\Im[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]}{\sqrt{\langle |\widehat{\sigma}_T|^2 \rangle \langle |\widehat{\sigma}_S|^2 \rangle - \Re[\langle \widehat{\sigma}_T \widehat{\sigma}_S^* \rangle]^2}} = \frac{\Im(\widehat{\phi})}{\sqrt{1 - \Re(\widehat{\phi})^2}} = \widehat{\xi}. \tag{7.53}$$

The above equation shows that the residual coherence is exactly equal to the corrected imaginary coherence $\widehat{\xi}$. That is, the corrected imaginary coherence is derived as the coherence between the seed signal and the residual signal, which is obtained by regressing out the seed signal from the target signal.

7.5.4 Phase Dependence of the Corrected Imaginary Coherences

The coherence ϕ is complex valued, and it is expressed as the amplitude and phase, such that

$$\phi = |\phi| e^{i\theta}, \tag{7.54}$$

where θ is the phase of the coherence, and the notation (f) is omitted for simplicity. When the coherence is caused by true brain interactions, we naturally interpret that the amplitude $|\phi|$, which is equal to the magnitude coherence, represents the strength of the brain interactions.

The imaginary coherence is expressed as

$$\Im(\phi) = |\phi| \sin\theta. \tag{7.55}$$

The value of the imaginary coherence depends on the phase θ. This is one weak point of the imaginary coherence, because even if there is a strong brain interaction, the imaginary coherence becomes very small when the phase θ has a value close to one of multiples of π.

Let us see how the corrected imaginary coherence ξ depends on the phase. Using Eq. (7.54), we have

$$\xi = \frac{|\phi| \sin\theta}{\sqrt{1 - |\phi|^2 \cos^2\theta}}. \tag{7.56}$$

The plots of ξ with respect to the phase are shown in Fig. 7.1 with four values of the magnitude coherence $|\phi|$. This figure shows that the corrected imaginary coherence becomes asymptotically independent of the phase with the limit of $|\phi| \to 1$. However, the corrected imaginary coherence has almost the same phase dependence as the imaginary coherence, when the magnitude coherence $|\phi|$ is less than 0.6. When $|\phi|$ becomes closer to 1, the corrected imaginary coherence ξ becomes significantly greater than the imaginary coherence $\Im(\phi)$.

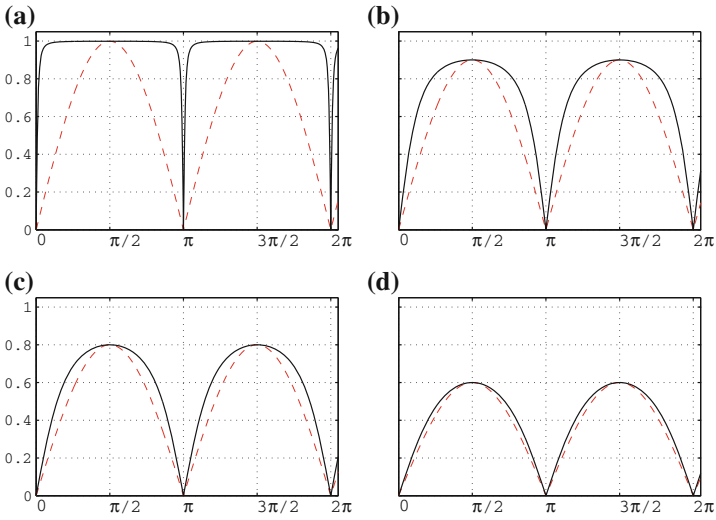

Fig. 7.1 Plots of the imaginary coherence and corrected imaginary coherence with respect to the phase for four values of magnitude coherence. **a** $|\phi| = 0.99$. **b** $|\phi| = 0.9$. **c** $|\phi| = 0.8$. **d** $|\phi| = 0.6$. The *solid line* shows the corrected imaginary coherence and the *broken line* shows the imaginary coherence

7.6 Canonical Coherence

7.6.1 Canonical Magnitude Coherence

So far, we have discussed methods to compute voxel-based coherence. However, when the number of voxels is large and our target is to analyze all-voxel-to-all-voxel connections, the interpretation and visualization of analysis results may not be easy because the number of voxel connections becomes huge. One way to reduce this difficulty is to compute coherence between regions determined based on the neurophysiology and/or neuroanatomy of a brain. However, in this case, since a brain region contains multiple voxels, we must compute the coherence between groups of multiple time courses. We here extend the theory of canonical correlation described in Sect. C.3 in the Appendix to compute the canonical coherence, which should be effective for region-based connectivity analysis. The explanation here follows those in [10, 11].

In this section, random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are complex-valued column vectors in the frequency domain, i.e.,

$$\boldsymbol{x}(f) = \begin{bmatrix} x_1(f) \\ x_2(f) \\ \vdots \\ x_p(f) \end{bmatrix} \quad \text{and} \quad \boldsymbol{y}(f) = \begin{bmatrix} y_1(f) \\ y_2(f) \\ \vdots \\ y_q(f) \end{bmatrix}. \tag{7.57}$$

In the following arguments, the notation (f) is omitted for simplicity. Using the same arguments as those for the canonical correlation in Sect. C.3, the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are, respectively, projected in the directions of $\boldsymbol{a}$ and $\boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are complex-valued $p \times 1$ and $q \times 1$ column vectors. That is, defining $\widehat{x} = \boldsymbol{a}^H \boldsymbol{x}$ and $\widehat{y} = \boldsymbol{b}^H \boldsymbol{y}$, the coherence between $\widehat{x}$ and $\widehat{y}$, ϕ, is given by

$$\begin{aligned} \phi = \frac{\langle \widehat{x}\,\widehat{y}^* \rangle}{\sqrt{\langle \widehat{x}\,\widehat{x}^* \rangle}\sqrt{\langle \widehat{y}\,\widehat{y}^* \rangle}} &= \frac{\boldsymbol{a}^H \langle \boldsymbol{x}\boldsymbol{y}^H \rangle \boldsymbol{b}}{\sqrt{[\boldsymbol{a}^H \langle \boldsymbol{x}\boldsymbol{x}^H \rangle \boldsymbol{a}][\boldsymbol{b}^H \langle \boldsymbol{y}\boldsymbol{y}^H \rangle \boldsymbol{b}]}} \\ &= \frac{\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}}{\sqrt{[\boldsymbol{a}^H \boldsymbol{\Sigma}_{xx} \boldsymbol{a}][\boldsymbol{b}^H \boldsymbol{\Sigma}_{yy} \boldsymbol{b}]}}, \end{aligned} \tag{7.58}$$

where the superscript H indicates the Hermitian transpose. Here the cross-spectral matrices are defined as $\boldsymbol{\Sigma}_{xy} = \langle \boldsymbol{x}\boldsymbol{y}^H \rangle$, $\boldsymbol{\Sigma}_{xx} = \langle \boldsymbol{x}\boldsymbol{x}^H \rangle$, and $\boldsymbol{\Sigma}_{yy} = \langle \boldsymbol{y}\boldsymbol{y}^H \rangle$. We derive a formula to compute the canonical magnitude coherence. Using Eq. (7.58), the magnitude coherence between $\widehat{x}$ and $\widehat{y}$ is expressed as

$$|\phi|^2 = \frac{\left|\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right|^2}{[\boldsymbol{a}^H \boldsymbol{\Sigma}_{xx} \boldsymbol{a}][\boldsymbol{b}^H \boldsymbol{\Sigma}_{yy} \boldsymbol{b}]}. \tag{7.59}$$

The canonical squared magnitude coherence $|\psi|^2$ is defined as the maximum of $|\phi|^2$ (with respect to $\boldsymbol{a}$ and $\boldsymbol{b}$), which is obtained by solving the optimization problem

$$|\psi|^2 = \max_{\boldsymbol{a},\boldsymbol{b}} \left|\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right|^2 \quad \text{subject to } \boldsymbol{a}^H \boldsymbol{\Sigma}_{xx} \boldsymbol{a} = 1 \quad \text{and} \quad \boldsymbol{b}^H \boldsymbol{\Sigma}_{yy} \boldsymbol{b} = 1. \tag{7.60}$$

This constrained optimization problem can be solved in the following manner. Since $\left|\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right|^2$ is a scalar, we have the relationship,

$$\left|\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right|^2 = \left(\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right)^H \left(\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}\right) = \boldsymbol{b}^H \boldsymbol{\Sigma}_{xy}^H \boldsymbol{a}\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b}. \tag{7.61}$$

We first fix $\boldsymbol{a}$, and solve the maximization problem with respect to $\boldsymbol{b}$. The maximization problem is

$$\max_{\boldsymbol{b}} \boldsymbol{b}^H \boldsymbol{\Sigma}_{xy}^H \boldsymbol{a}\boldsymbol{a}^H \boldsymbol{\Sigma}_{xy} \boldsymbol{b} \quad \text{subject to} \quad \boldsymbol{b}^H \boldsymbol{\Sigma}_{yy} \boldsymbol{b} = 1. \tag{7.62}$$

Using the same derivation mentioned in Sect. C.9, the maximum value is equal to the largest eigenvalue of the matrix

$$\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{H}\boldsymbol{a}\boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xy}.$$

The matrix above is a rank-one matrix and it only has a single nonzero eigenvalue. According to Sect. C.8 (Property No. 10), this eigenvalue is equal to

$$\boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{H}\boldsymbol{a}, \tag{7.63}$$

which is a scalar.

Thus, the optimization

$$|\psi|^2 = \max_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{b}^{H}\boldsymbol{\Sigma}_{xy}^{H}\boldsymbol{a}\boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xy}\boldsymbol{b} \quad \text{subject to } \boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xx}\boldsymbol{a} = 1 \quad \text{and} \quad \boldsymbol{b}^{H}\boldsymbol{\Sigma}_{yy}\boldsymbol{b} = 1, \tag{7.64}$$

is now rewriten as

$$|\psi|^2 = \max_{\boldsymbol{a}} \boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{H}\boldsymbol{a} \quad \text{subject to} \quad \boldsymbol{a}^{H}\boldsymbol{\Sigma}_{xx}\boldsymbol{a} = 1. \tag{7.65}$$

Using again the derivation described in Sect. C.9, the solution of this maximization is obtained as the maximum eigenvalue of the matrix

$$\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{H}.$$

That is, denoting the eigenvalues of this matrix as γ_j where $j = 1, \ldots, d$ and $d = \min\{p, q\}$, the canonical squared magnitude coherence is derived as

$$|\psi|^2 = \mathcal{S}_{\max}\{\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{H}\} = \gamma_1, \tag{7.66}$$

where the notation $\mathcal{S}_{\max}\{\cdot\}$ indicates the maximum eigenvalue of a matrix between the parentheses, as is defined in Sect. C.9.

This canonical squared magnitude coherence is considered the best overall magnitude coherence measure between the two sets of multiple spectra $x_1, \ldots, x_p$ and $y_1, \ldots, y_q$, and it is equal to the maximum eigenvalue γ_1 in Eq. (7.66). However, other eigenvalues may have information complementary to γ_1, and therefore, a metric that uses all the eigenvalues may be preferable. Let us assume the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to be complex Gaussian. According to Eq. (C.52), we can then define the mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ such that

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{d} \log \frac{1}{1-\gamma_j}, \tag{7.67}$$

which is a metric using all the eigenvalues. In the arguments in Sect. 7.5.2, the relationship between the coherence and the mutual information is given in Eq. (7.41). This relationship can lead to an alternative definition of canonical magnitude coherence based on the mutual information, such that

$$|\widetilde{\psi}|^2 = 1 - \exp[-\mathcal{I}(\boldsymbol{x}, \boldsymbol{y})] = 1 - \prod_{j=1}^{d}(1 - \gamma_j). \tag{7.68}$$

This metric uses all the eigenvalues, and since its value is normalized between 0 and 1, the interpretation is intuitively easy, compared to the original mutual information $\mathcal{I}(\boldsymbol{x}, \boldsymbol{y})$.

Also, the mutual information $\mathcal{I}(\boldsymbol{x}, \boldsymbol{y})$ in Eq. (7.67) depends on d, which is the sizes of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Such property may not be appropriate for a brain interaction metric, and the metric independent of the sizes of vectors can be defined such that,

$$\widetilde{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{d}\sum_{j=1}^{d} \log \frac{1}{1 - \gamma_j}. \tag{7.69}$$

The alternative definition of canonical magnitude coherence in this case is expressed as

$$|\breve{\psi}|^2 = 1 - \exp[-\widetilde{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{y})] = 1 - \left[\prod_{j=1}^{d}(1 - \gamma_j)\right]^{1/d}. \tag{7.70}$$

The above metric may be more effective than the one in Eq. (7.68).

7.6.2 Canonical Imaginary Coherence

We next derive a formula to compute the canonical imaginary coherence. Following the arguments of Evald et al. [12] we define

$$\boldsymbol{\Sigma}_{xx} = \Re(\boldsymbol{\Sigma}_{xx}) + i\Im(\boldsymbol{\Sigma}_{xx}) = \boldsymbol{\Gamma}_{xx} + i\boldsymbol{\Upsilon}_{xx}, \tag{7.71}$$

$$\boldsymbol{\Sigma}_{yy} = \Re(\boldsymbol{\Sigma}_{yy}) + i\Im(\boldsymbol{\Sigma}_{yy}) = \boldsymbol{\Gamma}_{yy} + i\boldsymbol{\Upsilon}_{yy}, \tag{7.72}$$

$$\boldsymbol{\Sigma}_{xy} = \Re(\boldsymbol{\Sigma}_{xy}) + i\Im(\boldsymbol{\Sigma}_{xy}) = \boldsymbol{\Gamma}_{xy} + i\boldsymbol{\Upsilon}_{xy}. \tag{7.73}$$

We use real-valued $\boldsymbol{a}$ and $\boldsymbol{b}$ and express the imaginary part of the coherence between $\boldsymbol{a}^T\boldsymbol{x}$ and $\boldsymbol{b}^T\boldsymbol{y}$:

$$\Im(\phi) = \frac{\boldsymbol{a}^T\Im\left(\boldsymbol{\Sigma}_{xy}\right)\boldsymbol{b}}{\sqrt{\left[\boldsymbol{a}^T\boldsymbol{\Sigma}_{xx}\boldsymbol{a}\right]\left[\boldsymbol{b}^T\boldsymbol{\Sigma}_{yy}\boldsymbol{b}\right]}} = \frac{\boldsymbol{a}^T\boldsymbol{\Upsilon}_{xy}\boldsymbol{b}}{\sqrt{\left[\boldsymbol{a}^T\boldsymbol{\Sigma}_{xx}\boldsymbol{a}\right]\left[\boldsymbol{b}^T\boldsymbol{\Sigma}_{yy}\boldsymbol{b}\right]}}. \tag{7.74}$$

The canonical imaginary coherence ψ_I is defined as the maximum of $\Im(\phi)$, which is obtained using the maximization,

$$\psi_I = \max_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{a}^T \boldsymbol{\Upsilon}_{xy} \boldsymbol{b}, \quad \text{subject to} \quad \boldsymbol{a}^T \boldsymbol{\Sigma}_{xx} \boldsymbol{a} = 1 \quad \text{and} \quad \boldsymbol{b}^T \boldsymbol{\Sigma}_{yy} \boldsymbol{b} = 1. \tag{7.75}$$

The formulation above is exactly the same as that for the canonical correlation in Eq. (C.31). Thus, as in Sect. C.3.1, this maximization problem might seem to be rewritten as the eigenvalue problem,

$$\left(\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Upsilon}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Upsilon}_{xy}^T \right) \boldsymbol{a} = \psi_I^2 \boldsymbol{a}, \tag{7.76}$$

and ψ_I^2 is obtained as its maximum eigenvalue. However, in this eigenproblem, the eigenvector $\boldsymbol{a}$ is generally complex-valued, conflicting with the assumption that the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are real-valued.

We instead use real-valued $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ obtained such that

$$\boldsymbol{\alpha} = \boldsymbol{\Gamma}_{xx}^{1/2} \boldsymbol{a}, \tag{7.77}$$

$$\boldsymbol{\beta} = \boldsymbol{\Gamma}_{yy}^{1/2} \boldsymbol{b}. \tag{7.78}$$

Then, using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we have

$$\begin{aligned} \boldsymbol{a}^T \boldsymbol{\Sigma}_{xx} \boldsymbol{a} &= \boldsymbol{\alpha}^T \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xx} \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^T \boldsymbol{\Gamma}_{xx}^{-1/2} \left(\boldsymbol{\Gamma}_{xx} + i \boldsymbol{\Upsilon}_{xx} \right) \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\alpha} \\ &= \|\boldsymbol{\alpha}\|^2 + i \boldsymbol{\alpha}^T \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\Upsilon}_{xx} \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|^2 \end{aligned} \tag{7.79}$$

and

$$\begin{aligned} \boldsymbol{b}^T \boldsymbol{\Sigma}_{yy} \boldsymbol{b} &= \boldsymbol{\beta}^T \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\Sigma}_{yy} \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\beta} \\ &= \boldsymbol{\beta}^T \boldsymbol{\Gamma}_{yy}^{-1/2} \left(\boldsymbol{\Gamma}_{yy} + i \boldsymbol{\Upsilon}_{yy} \right) \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\beta} \\ &= \|\boldsymbol{\beta}\|^2 + i \boldsymbol{\beta}^T \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\Upsilon}_{yy} \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\beta} = \|\boldsymbol{\beta}\|^2. \end{aligned} \tag{7.80}$$

In the equations above, we use the fact that the matrices $\boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\Upsilon}_{xx} \boldsymbol{\Gamma}_{xx}^{-1/2}$ and $\boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\Upsilon}_{yy} \boldsymbol{\Gamma}_{yy}^{-1/2}$ are skew-symmetric.[2]

Using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the optimization in Eq. (7.75) can be rewritten as

$$\psi_I = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\alpha}^T \boldsymbol{\Gamma}_{xx}^{-1/2} \boldsymbol{\Upsilon}_{xy} \boldsymbol{\Gamma}_{yy}^{-1/2} \boldsymbol{\beta} \quad \text{subject to} \quad \boldsymbol{\alpha}^T \boldsymbol{\alpha} = 1 \quad \text{and} \quad \boldsymbol{\beta}^T \boldsymbol{\beta} = 1. \tag{7.81}$$

[2] If $\boldsymbol{A}^T = -\boldsymbol{A}$ holds, matrix $\boldsymbol{A}$ is skew-symmetric. Since $\boldsymbol{\alpha}^T \boldsymbol{A} \boldsymbol{\alpha}$ is a scalar, if $\boldsymbol{A}$ is skew-symmetric, $\boldsymbol{\alpha}^T \boldsymbol{A} \boldsymbol{\alpha} = 0$.

Following the derivation in Sect. C.3.1, the solution of the above optimization problem is expressed as the eigenvalue problem,

$$\left(\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T\right)\boldsymbol{\alpha} = \psi_I^2\boldsymbol{\alpha}, \tag{7.82}$$

where $\boldsymbol{\Pi}_{xy} = \boldsymbol{\Gamma}_{xx}^{-1/2}\boldsymbol{\Upsilon}_{xy}\boldsymbol{\Gamma}_{yy}^{-1/2}$. In the above equation, since the matrix, $\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T$ is a real symmetric matrix, the eigenvector $\boldsymbol{\alpha}$ is real-valued. The vector $\boldsymbol{\beta}$ is obtained using

$$\boldsymbol{\beta} = \frac{1}{\psi_I}\boldsymbol{\Upsilon}_{xy}^T\boldsymbol{\alpha},$$

which is also real-valued.

The matrices $\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T$ and $\boldsymbol{\Gamma}_{xx}^{-1}\boldsymbol{\Upsilon}_{xy}\boldsymbol{\Gamma}_{yy}^{-1}\boldsymbol{\Upsilon}_{xy}^T$ have the same eigenvalues, according to Sect. C.8 (Property No. 9). Let us define the eigenvalues of the matrix $\boldsymbol{\Gamma}_{xx}^{-1}\boldsymbol{\Upsilon}_{xy}\boldsymbol{\Gamma}_{yy}^{-1}\boldsymbol{\Upsilon}_{xy}^T$ as ζ_j $(j = 1, \ldots, d)$. The canonical imaginary coherence ψ_I^2 is derived as

$$\psi_I^2 = \mathcal{S}_{\max}\{\boldsymbol{\Gamma}_{xx}^{-1}\boldsymbol{\Upsilon}_{xy}\boldsymbol{\Gamma}_{yy}^{-1}\boldsymbol{\Upsilon}_{xy}^T\} = \zeta_1. \tag{7.83}$$

Using the same arguments in the preceding section, the imaginary-coherence-based mutual information is given by

$$\mathcal{I}_s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{d} \log\frac{1}{1-\zeta_j}. \tag{7.84}$$

The alternative definition of canonical imaginary coherence, based on $\mathcal{I}_s(\boldsymbol{x}, \boldsymbol{y})$, is obtained as

$$\widetilde{\psi}_I^2 = 1 - \exp[-\mathcal{I}_s(\boldsymbol{x}, \boldsymbol{y})] = 1 - \prod_{j=1}^{d}(1-\zeta_j), \tag{7.85}$$

which uses all the eigenvalues $\zeta_1, \ldots, \zeta_d$. When we can assume $\zeta_j \ll 1$ (where $j = 1, \ldots, d$), we have

$$\widetilde{\psi}_I^2 \approx \sum_{j=1}^{d}\zeta_j. \tag{7.86}$$

The connectivity metric above has been proposed and called the multivariate interaction measure (MIM) in [12].

The mutual information independent of the sizes of vectors is defined such that,

$$\widetilde{\mathcal{I}}_s(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{d}\sum_{j=1}^{d} \log\frac{1}{1-\zeta_j}. \tag{7.87}$$

The alternative canonical imaginary coherence, based on $\widetilde{\mathcal{I}}_s(\boldsymbol{x}, \boldsymbol{y})$, is obtained as

$$\breve{\psi}_I^2 = 1 - \exp[-\mathcal{I}_s(\boldsymbol{x}, \boldsymbol{y})] = 1 - \left[\prod_{j=1}^{d}(1-\zeta_j)\right]^{1/d}. \tag{7.88}$$

When we can assume $\zeta_j \ll 1$ (where $j = 1, \ldots, d$), we have

$$\breve{\psi}_I^2 \approx \frac{1}{d}\sum_{j=1}^{d}\zeta_j. \tag{7.89}$$

The connectivity metric above is called the global interaction measure (GIM) [12].

7.6.3 Canonical Residual Coherence

We can compute the canonical residual coherence, which is a multivariate version of the residual coherence described in Sect. 7.5.3. Let us assume that $\boldsymbol{y}$ is the target spectra, $\boldsymbol{x}$ is the seed spectra, and consider the regression

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}. \tag{7.90}$$

The real-valued regression coefficient matrix $\boldsymbol{A}$ is obtained by the least-squares fit. The optimum $\boldsymbol{A}$ is obtained as

$$\widehat{\boldsymbol{A}} = \Re\left(\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\right). \tag{7.91}$$

Thus, the residual signal $\boldsymbol{v}$ is expressed as

$$\boldsymbol{v} = \boldsymbol{y} - \widehat{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{y} - \Re\left(\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\right)\boldsymbol{x}. \tag{7.92}$$

Let us define $\boldsymbol{\Sigma}_{vv}$ and $\boldsymbol{\Sigma}_{vx}$ such that $\boldsymbol{\Sigma}_{vv} = \langle \boldsymbol{v}\boldsymbol{v}^H \rangle$ and $\boldsymbol{\Sigma}_{vx} = \langle \boldsymbol{v}\boldsymbol{x}^H \rangle$. They are obtained as

$$\boldsymbol{\Sigma}_{vv} = \langle (\boldsymbol{y} - \widehat{\boldsymbol{A}}\boldsymbol{x})(\boldsymbol{y} - \widehat{\boldsymbol{A}}\boldsymbol{x})^H \rangle = \boldsymbol{\Sigma}_{yy} - \widehat{\boldsymbol{A}}\boldsymbol{\Sigma}_{xy} - \boldsymbol{\Sigma}_{yx}\widehat{\boldsymbol{A}}^T + \widehat{\boldsymbol{A}}\boldsymbol{\Sigma}_{xx}\widehat{\boldsymbol{A}}^T, \tag{7.93}$$

and

$$\boldsymbol{\Sigma}_{vx} = \langle (\boldsymbol{y} - \widehat{\boldsymbol{A}}\boldsymbol{x})\boldsymbol{x}^H \rangle = \boldsymbol{\Sigma}_{yx} - \widehat{\boldsymbol{A}}\boldsymbol{\Sigma}_{xx}. \tag{7.94}$$

We just follow the arguments in Sect. 7.6.1, and derive an expression for the canonical magnitude coherence between $\boldsymbol{v}$ and $\boldsymbol{x}$. Using complex-valued $\boldsymbol{a}$ and $\boldsymbol{b}$, we define $\widehat{v} = \boldsymbol{a}^H\boldsymbol{v}$ and $\widehat{x} = \boldsymbol{b}^H\boldsymbol{x}$, the magnitude coherence between $\widehat{v}$ and $\widehat{x}$, $|\phi_R|^2$,

is given by

$$|\phi_R|^2 = \frac{|\langle \widehat{v}\,\widehat{x}^* \rangle|^2}{\sqrt{\langle \widehat{v}\,\widehat{v}^* \rangle}\sqrt{\langle \widehat{x}\,\widehat{x}^* \rangle}} = \frac{|\boldsymbol{a}^H \boldsymbol{\Sigma}_{vx} \boldsymbol{b}|^2}{[\boldsymbol{a}^H \boldsymbol{\Sigma}_{vv} \boldsymbol{a}][\boldsymbol{b}^H \boldsymbol{\Sigma}_{xx} \boldsymbol{b}]}. \tag{7.95}$$

The canonical squared residual coherence $|\psi_R|^2$ is defined as the maximum of $|\phi_R|^2$, which is obtained by solving the optimization problem,

$$|\psi_R|^2 = \max_{\boldsymbol{a},\boldsymbol{b}} \left| \boldsymbol{a}^H \boldsymbol{\Sigma}_{vx} \boldsymbol{b} \right|^2, \text{ subject to } \boldsymbol{a}^H \boldsymbol{\Sigma}_{vv} \boldsymbol{a} = 1 \text{ and } \boldsymbol{b}^H \boldsymbol{\Sigma}_{xx} \boldsymbol{b} = 1. \tag{7.96}$$

This maximization problem is exactly the same as that in Eq. (7.60), and the solution of this maximization is known to be the maximum eigenvalue of the matrix $\boldsymbol{\Sigma}_{vv}^{-1} \boldsymbol{\Sigma}_{vx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{vx}^H$. That is, the canonical residual coherence is derived as

$$|\psi_R|^2 = \mathcal{S}_{\max}\{\boldsymbol{\Sigma}_{vv}^{-1} \boldsymbol{\Sigma}_{vx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{vx}^H\}, \tag{7.97}$$

where $\boldsymbol{\Sigma}_{vv}$ is derived using Eq. (7.93), and $\boldsymbol{\Sigma}_{vx}$ is derived using Eq. (7.94).

7.6.4 Computing Coherence When Each Voxel has Multiple Time Courses

When the source vector has x, y, and z components, most source imaging algorithms generate three time courses corresponding to the x, y, and z components at each voxel. A usual way is to obtain a single representative time course at each voxel and compute coherence using such representative time courses, as described in Sect. 7.2. Here, we describe an alternative method to compute voxel coherence when each voxel has multiple time courses [13].

The method is a straightforward application of the canonical coherence. Let us assume that the voxel time courses are expressed as in Eq. (7.1). The voxel spectra at the target and seed voxels are respectively denoted using 3×1 vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, which are given by

$$\boldsymbol{x}(f) = \begin{bmatrix} x_1(f) \\ x_2(f) \\ x_3(f) \end{bmatrix} \quad \text{and} \quad \boldsymbol{y}(f) = \begin{bmatrix} y_1(f) \\ y_2(f) \\ y_3(f) \end{bmatrix}, \tag{7.98}$$

where $x_1(f)$, $x_2(f)$, and $x_3(f)$ are the spectra obtained from the source time courses in the x, y and z directions at the target voxel, and $y_1(f)$, $y_2(f)$, and $y_3(f)$ are the spectra obtained from the source time courses in the x, y and z directions at the seed voxel. Using these $\boldsymbol{x}$ and $\boldsymbol{y}$, we compute canonical magnitude coherence $|\psi|$ in Eq. (7.66) or $|\widetilde{\psi}|$ in Eq. (7.68), and the canonical imaginary coherence ψ_I in Eq. (7.83) or $\widetilde{\psi}_I$ in Eq. (7.85). This method is equivalent to determining the source

orientations at the seed and target voxels by simultaneously maximizing the canonical (magnitude/imaginary) coherence between these voxels.

7.7 Envelope Correlation and Related Connectivity Metrics

7.7.1 Envelope Correlation

Coherence is a metric that measures the phase relationship between two spectra, and naturally it is sensitive to phase jitters. This property becomes problematic when we try to estimate brain interactions at higher frequencies, such as gamma and high-gamma activities. This is because, at high frequencies, a small time jitter could cause a large phase jitter, and coherence may not be able to detect connectivity relationships. That is, the coherence may not be appropriate as a connectivity metric at high frequencies. The envelope-to-envelope correlation [14] is considered an appropriate metric for such cases.

To compute the envelope correlation, we first convert the seed and target time courses into their analytic signals, such that

$$\mathcal{A}[u(t)] = u(t) + \frac{i}{\pi} \int \frac{u(t')}{t - t'} \, \mathrm{d}t'. \tag{7.99}$$

On the left-hand side of the equation above, $\mathcal{A}[\,\cdot\,]$ indicates an operator that creates an analytic signal of the real-valued time signal in the parentheses. Let us define analytic signals from the seed- and target-voxel time courses $u_S(t)$ and $u_T(t)$, such that

$$\mathcal{A}[u_S(t)] = A_S(t)\mathrm{e}^{i\theta_S(t)}, \tag{7.100}$$

$$\mathcal{A}[u_T(t)] = A_T(t)\mathrm{e}^{i\theta_T(t)}, \tag{7.101}$$

where $A_S(t)$ and $A_T(t)$ are the amplitudes of the seed and target analytic signals, and $\theta_S(t)$ and $\theta_T(t)$ are their instantaneous phases. The envelope correlation is the correlation between the amplitudes, $A_S(t)$ and $A_T(t)$, and is computed such that

$$\Theta = \frac{\sum_{j=1}^{K} A_T(t_j) A_S(t_j)}{\sqrt{\left[\sum_{j=1}^{K} A_T(t_j)^2\right]\left[\sum_{j=1}^{K} A_S(t_j)^2\right]}}. \tag{7.102}$$

It is obvious in the above equation that common interferences such as the algorithm leakage cause spurious correlation, and the seed blur should exist in an image of envelope correlation.

7.7.2 Residual Envelope Correlation

Residual envelope correlation, which is free from the problem of seed blur, is proposed in [15]. To compute the residual envelope correlation, we first compute the residual time course $u_R(t)$ using

$$u_R(t) = \int_{-\infty}^{\infty} v(f) \mathrm{e}^{i2\pi ft} \, \mathrm{d}f. \tag{7.103}$$

Here $v(f)$ is the residual spectrum obtained using Eq. (7.48). Since $v(f)$ is defined only for $f \geq 0$, we create the residual spectrum for $f < 0$, such that

$$v(-f) = v(f)^*,$$

where the superscript $*$ indicates the complex conjugation. Since the Hermitian symmetry holds for $v(f)$, $u_R(t)$ is guaranteed to be a real-valued time course. The envelope of $u_R(t)$ is computed using

$$\mathcal{A}[u_R(t)] = A_R(t) \mathrm{e}^{i\theta_R(t)}, \tag{7.104}$$

where $A_R(t)$ is the envelope of $u_R(t)$ called the residual envelope. Once $A_R(t)$ is computed, the residual envelope correlation Θ_R is computed using

$$\Theta_R = \frac{\sum_{j=1}^{K} A_R(t_j) A_S(t_j)}{\sqrt{\left[\sum_{j=1}^{K} A_R(t_j)^2\right]\left[\sum_{j=1}^{K} A_S(t_j)^2\right]}}. \tag{7.105}$$

Since in the residual time course the seed signal is regressed out, the residual envelope correlation is free from the spurious correlation caused by the algorithm leakage. Note that a method similar to the one mentioned here was reported in [16].

7.7.3 Envelope Coherence

The coherence can be computed between the envelope time courses $A_T(t)$ and $A_S(t)$. Let us define the spectra obtained from the envelope $A_S(t)$ and $A_T(t)$ as $\Sigma_S(f)$ and $\Sigma_T(f)$, respectively. The complex envelope coherence is computed using

$$\phi_E(f) = \frac{\langle \Sigma_T(f) \Sigma_S^*(f) \rangle}{\sqrt{\langle |\Sigma_T(f)|^2 \rangle \langle |\Sigma_S(f)|^2 \rangle}}. \tag{7.106}$$

The imaginary and corrected imaginary coherences are obtained as $\Im(\phi_E(f))$ and $\Im(\phi_E(f))/\sqrt{(1-\Re(\phi_E(f))^2}$.

7.8 Statistical Thresholding of Coherence Images

In practical applications, we need to assess the statistical significance of the obtained coherence images. The surrogate data method [17, 18] has been used for this assessment. In this method, the surrogate voxel spectra are created by multiplying random phases with the original voxel spectra. The surrogate spectra for the seed and target voxels, $\widetilde{\sigma}_S(f)$ and $\widetilde{\sigma}_T(f)$, are expressed as

$$\widetilde{\sigma}_S = \widehat{\sigma}_S e^{i2\pi\delta_S} \quad \text{and} \quad \widetilde{\sigma}_T = \widehat{\sigma}_T e^{i2\pi\delta_T}, \tag{7.107}$$

where δ_S and δ_T are uniform random numbers between 0 and 1, and the explicit notation of (f) is again omitted for simplicity. Note that the surrogate spectra have the same original power spectra but the phase relationship is destroyed by multiplying the random phases to the original spectra.

Any coherence-based metric, such as the magnitude/imaginary coherence or corrected imaginary coherence, can be computed using the surrogate spectra, $\widetilde{\sigma}_S$ and $\widetilde{\sigma}_T$. The metric computed using the surrogate spectra is denoted ω. As an example, the imaginary coherence is computed using the surrogate spectra. It is expressed as[3]

$$\omega = \frac{|\Im\left(\langle\widetilde{\sigma}_T\widetilde{\sigma}_S^*\rangle\right)|}{\sqrt{\langle|\widetilde{\sigma}_T|^2\rangle\langle|\widetilde{\sigma}_S|^2\rangle}} = \frac{|\Im\left(\langle\widehat{\sigma}_T\widehat{\sigma}_S^* e^{i2\pi\Delta\delta}\rangle\right)|}{\sqrt{\langle|\widehat{\sigma}_T|^2\rangle\langle|\widehat{\sigma}_S|^2\rangle}}, \tag{7.108}$$

where $\Delta\delta = \delta_T - \delta_S$. The generation of ω is repeated B times, and a total of B values of ω, denoted $\omega^1, \omega^2, \ldots, \omega^B$, are obtained. These $\omega^1, \ldots, \omega^B$ can form an empirical null distribution at each voxel.

We could derive a voxel-by-voxel statistical threshold using this empirical null distribution. However, the statistical threshold derived in this manner does not take the multiple comparisons into account and it generally leads to a situation in which many false-positive voxels arise, i.e., many voxels that contain no brain interaction are found to be interacting. To avoid this problem, the statistical significance is determined using a procedure that takes multiple comparisons into account. For this purpose, we can use the maximal statistics[4] [19, 20].

To utilize maximum statistics, the values ω^ℓ $(\ell = 1, \ldots, B)$ are first standardized and converted into pseudo-t values, such that

[3] The absolute value of the imaginary coherence is usually computed, because its sign has no meaning in expressing the connectivity.

[4] We can apply another method such as the false discovery rate to this multiple comparison problem.

$$T^{\ell} = \frac{\omega^{\ell} - \langle \omega \rangle}{\sigma_{\omega}}, \tag{7.109}$$

where $\langle \omega \rangle$ and σ_{ω}^2 are the average and the variance of ω^{ℓ} ($\ell = 1, \ldots, B$). Since these $\langle \omega \rangle$ and σ_{ω}^2 are obtained at each voxel, the values at the jth voxel are denoted $\langle \omega(j) \rangle$ and $\sigma_{\omega}^2(j)$. The maximum value of T^{ℓ} obtained at the jth voxel is denoted $T^{\max}(j)$. Defining a total number of voxels as N_V, we have $T^{\max}(1), \ldots, T^{\max}(N_V)$ to form a null distribution. We then sort these values in an increasing order:

$$T^{\max}(\tilde{1}) \leq T^{\max}(\tilde{2}) \leq \cdots \leq T^{\max}(\tilde{N}_V),$$

where $T^{\max}(\tilde{k})$ is the kth minimum value.

We set the level of the statistical significance to α, and choose $T^{\max}(\tilde{p})$ where $\tilde{p} = \lceil \alpha N_V \rceil$ and $\lceil \alpha N_V \rceil$ indicates the maximum integer not greater than αN_V. The threshold value for the jth voxel, $\omega^{\text{th}}(j)$, is finally derived as

$$\omega^{\text{th}}(j) = T^{\max}(\tilde{p}) \sigma_{\omega}(j) + \langle \omega(j) \rangle. \tag{7.110}$$

At the jth voxel, we evaluate the statistical significance of the imaginary coherence value by comparing it with $\omega^{\text{th}}(j)$. When the metric value is greater than $\omega^{\text{th}}(j)$, it is considered to be statistically significant; if not, it is considered to be statistically insignificant.

7.9 Mean Imaginary Coherence (MIC) Mapping

Guggisberg et al. [3] have proposed to compute a metric called the mean imaginary coherence. Defining the coherence computed between the jth and kth voxels as $\widehat{\phi}_{j,k}(f)$, the mean imaginary coherence for the jth voxel, $\mathcal{M}_j(f)$, is obtained using

$$\mathcal{M}_j(f) = \tanh\left[\frac{1}{N_V} \sum_{k=1}^{N_V} \tanh^{-1}\left(|\Im(\widehat{\phi}_{j,k}(f))| \right) \right]. \tag{7.111}$$

On the right-hand side, the absolute value of the imaginary coherence $|\Im(\widehat{\phi}_{j,k}(f))|$ is averaged across all voxel connections. In Eq. (7.111),

$$z = \tanh^{-1}(r) = \frac{1}{2} \log\left(\frac{1+r}{1-r} \right) \quad \text{and} \quad r = \tanh(z) = \frac{\mathrm{e}^{2z} - 1}{\mathrm{e}^{2z} + 1}$$

are the inverse hyperbolic and hyperbolic functions, respectively. The idea of using these functions is to average the voxel coherence values in the Fisher's Z-transform domain. We may use the corrected imaginary coherence instead of using the imaginary coherence in Eq. (7.111).

The mean imaginary coherence $\mathcal{M}_j(f)$ is hypothesized to express the communication capability (healthiness) of the brain tissue at the jth voxel location. This hypothesis has been tested by applying the mean imaginary coherence (MIC) mapping to various types of clinical data, and results positively confirming the validity of the hypothesis have been obtained. Such investigations include the MIC mapping for stroke-recovery patients [21], and patients with traumatic brain lesions [22]. In these applications, mapping of $\mathcal{M}_j(f)$ can predict the degree of a patient's recovery from stroke or brain injury. The MIC mapping has also been applied to MEG data from patients with schizophrenia [23] and Alzheimer's disease [24], and the values of $\mathcal{M}_j(f)$ are found to be significantly correlated with patient symptom scores at particular brain areas.

7.10 Numerical Examples

Computer simulation was performed to illustrate results of our arguments in this chapter. A sensor alignment of the 275-sensor array from the Omega™ (VMS Medtech, Coquitlam, Canada) neuromagnetometer was used. The coordinate system and source-sensor configuration used in the computer simulation are depicted in Fig. 5.1 in Chap. 5. A vertical plane ($x = 0\,\text{cm}$) was assumed at the middle of the whole-head sensor array, and three sources were assumed to exist on this plane.

Multiple-trial measurements were simulated, in which a total of 120 trials were generated with each trial consisting of 600 time points. The data were assumed to be collected with 2 ms sampling. We first conducted numerical experiments with

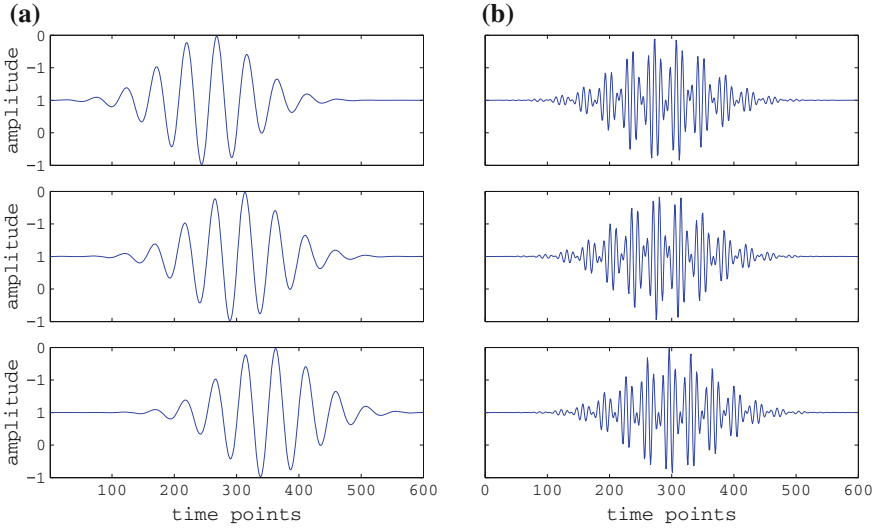

Fig. 7.2 Examples of the time courses **a** for the alpha-band experiments and **b** for the gamma-band experiments. The *top*, *middle*, and *bottom panels*, respectively, show the time courses for the first, second, and third sources

alpha-band activity. The time courses of the three sources for the first trial are shown in Fig. 7.2a. The time courses had trial-to-trial time jitters. The time jitters for the three time courses were generated using Gaussian random numbers with the same standard deviation of 20 time points. These time courses were projected onto the sensor space using the sensor lead field to obtain the signal magnetic recordings. The simulated sensor recordings were computed by adding spontaneous MEG data to the signal magnetic recordings. The signal-to-interference ratio was set to 0.5.

The voxel time courses were reconstructed using the vector-type narrow-band adaptive beamformer described in Chap. 3. The data-covariance matrix was tuned to the alpha frequency band. Reconstructed source power images on the plane $x = 0\,\text{cm}$ are shown in Fig. 7.3a. The three sources are clearly resolved. Since the spherical homogeneous conductor [25] was used for forward modeling, the vector adaptive beamformer reconstructed two time courses corresponding to the two tangential directions. We estimated a single representative time course at each voxel using the method described in Sect. 7.2, and computed voxel spectra. Coherence images were then computed with respect to the seed voxel, which was set at the second

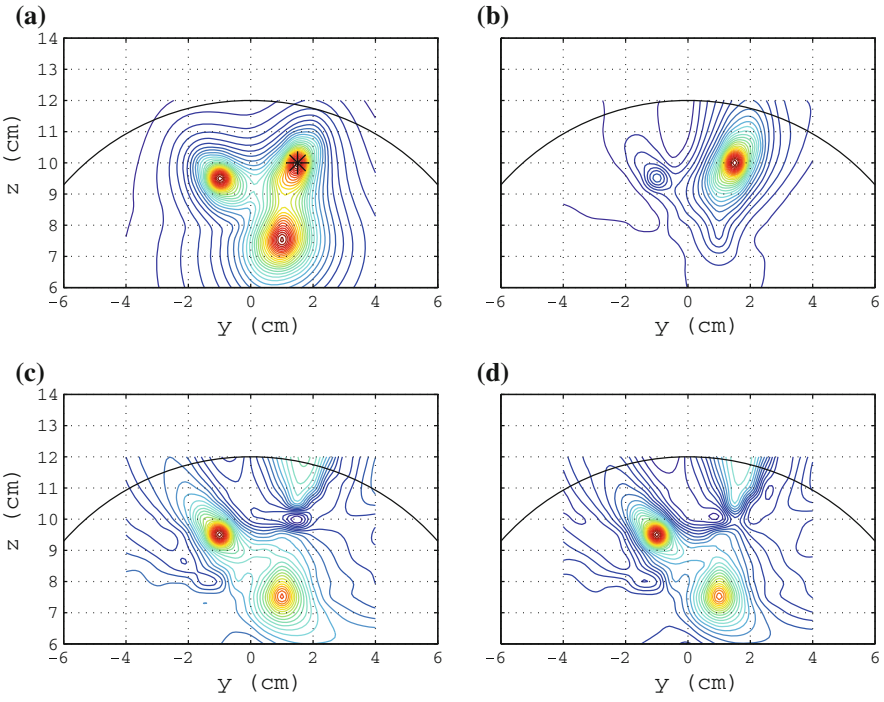

Fig. 7.3 Results of imaging the alpha-band voxel coherence on the plane $x = 0\,\text{cm}$. **a** Results of source reconstruction. The *asterisk* shows the location of the seed voxel. **b** Magnitude coherence image. **c** Imaginary coherence image. **d** Corrected imaginary coherence image

source location. The magnitude coherence image, imaginary coherence image, and corrected imaginary coherence image are respectively shown in Fig. 7.3b–d.

In the magnitude coherence image (Fig. 7.3b), the seed blur dominates and obscures the other sources. On the contrary, in the imaginary coherence image (Fig. 7.3c), and in the corrected imaginary coherence image (7.3d), the intensity of the seed blur is much reduced and the two sources that interact with the second source can clearly be observed. Also, the imaginary and the corrected imaginary coherence images are very similar, because the magnitude coherence is as small as 0.26–0.35 in this computer simulation.

We next implemented the method described in Sect. 7.6.4 in which a single representative voxel time course is not computed, but instead, the canonical coherence is directly computed by using multiple voxel time courses. The image of canonical imaginary coherence ψ_I in Eq. (7.83) is shown in Fig. 7.4a. The image of mutual-information-based canonical imaginary coherence, $\widetilde{\psi}_I$ in Eq. (7.85) is shown in Fig. 7.4b. The canonical magnitude coherence $|\psi|$ in Eq. (7.66) is shown in Fig. 7.4c. An image of mutual-information-based canonical magnitude coherence

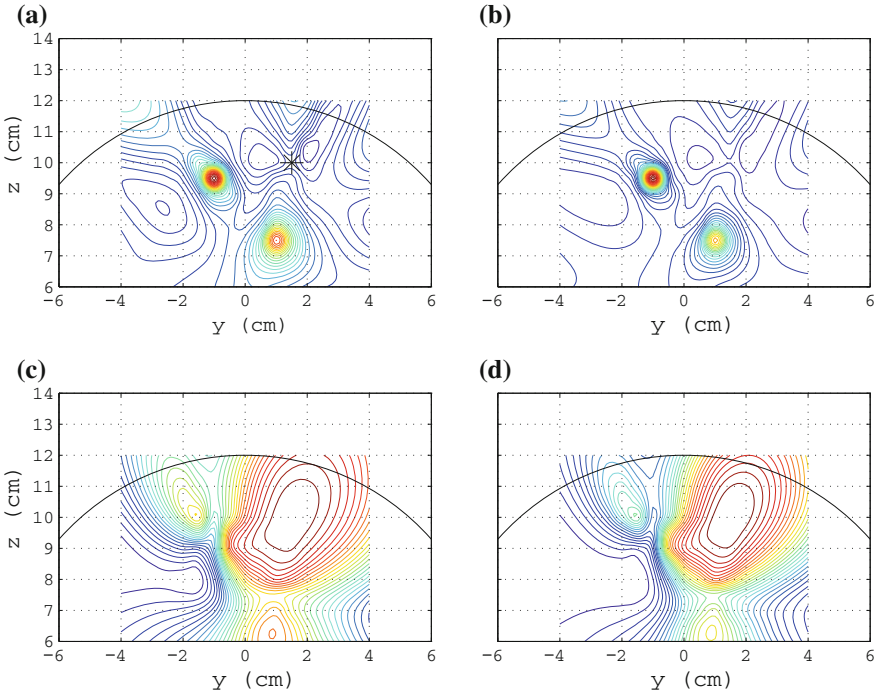

Fig. 7.4 Results of imaging canonical coherence on the plane $x = 0$ cm. **a** Image of canonical imaginary coherence ψ_I in Eq. (7.83). The *asterisk* shows the location of the seed voxel. **b** Image of mutual-information-based canonical imaginary coherence, $\widetilde{\psi}_I$ in Eq. (7.85). **c** Image of canonical magnitude coherence $|\psi|$ in Eq. (7.66). **d** Image of mutual-information-based canonical magnitude coherence $|\widetilde{\psi}|$ in Eq. (7.68)

$|\widetilde{\psi}|$ in Eq. (7.68) is shown in Fig. 7.4d. In these results, imaginary-coherence-based images ((a) and (b)), clearly detect the two interacting sources but the seed blur dominates in the magnitude-coherence-based images ((c) and (d)).

We carried out the numerical experiment simulating gamma-band signals with theta-band envelopes. The time courses of the three sources for the first trial are shown in Fig. 7.2b. The time courses had trial-to-trial time jitters generated using Gaussian random numbers with the same standard deviation of 20 time points.

The voxel time courses were estimated using the narrow-band adaptive beamformer with a data-covariance tuned to the gamma frequency band, and the coherence images were computed. The seed was set at the second source location. The coherence images are shown in Fig. 7.5a. Here, the magnitude, imaginary, and corrected imaginary coherence images are, respectively, shown in the top, middle, and bottom panels. In the magnitude coherence image, the seed blur dominates and only the seed source is detected. However, since the time jitter created a large phase jitter for the gamma-band signals, neither the imaginary nor the corrected imaginary coherence image contain meaningful information on the source connectivity.

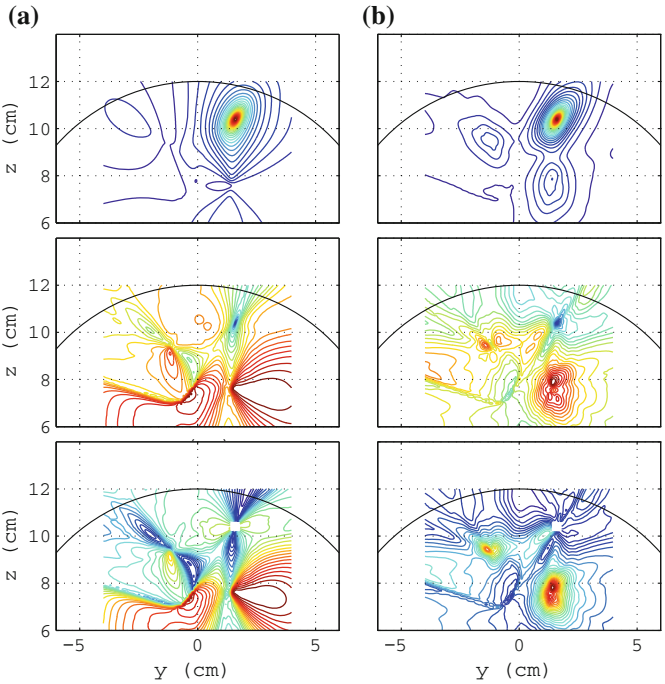

Fig. 7.5 Results of imaging the gamma-band source coherence on the plane $x = 0$ cm. The seed was set at the second source location. The coherence images were computed using the gamma-band voxel time courses. The *top*, *middle*, and *bottom panels* in **a**, respectively, show the magnitude, imaginary, and corrected imaginary coherence images. The *top*, *middle*, and *bottom panels* in **b**, respectively, show the magnitude, imaginary, and corrected imaginary envelope coherence images

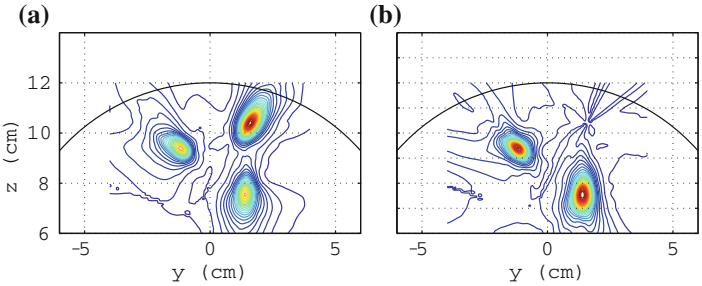

Fig. 7.6 Results of imaging envelope correlation on the plane $x = 0$ cm. The seed was set at the second source location. **a** Image of envelope correlation. **b** Image of residual envelope correlation

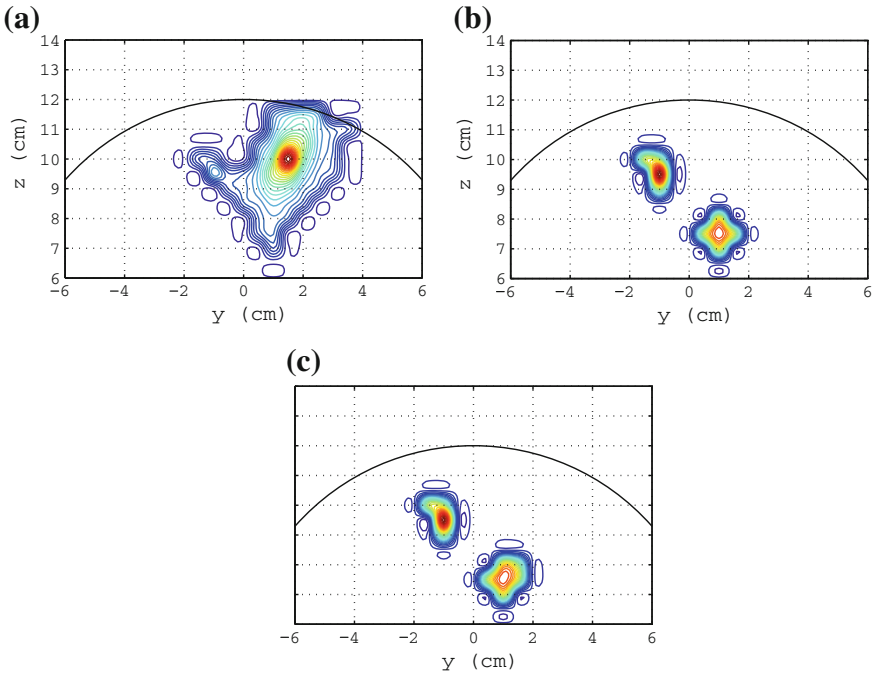

Fig. 7.7 Results of statistical thresholding of the alpha-band source coherence images. **a** Magnitude coherence image. **b** Imaginary coherence image. **c** Corrected imaginary coherence image. Statistical thresholding using the surrogate data method described in Sect. 7.8 was applied with the number of surrogate data set to 200 and the level of the statistical significance set to 0.99

We then computed the Hilbert envelopes of the reconstructed voxel time courses, and computed the envelope-to-envelope coherence. The results of the envelope-to-envelope coherence are shown in Fig. 7.5b in which the top, middle, and bottom panels show the magnitude, imaginary, and corrected imaginary coherence images, respectively. The magnitude coherence image detects only the seed source. In the imaginary coherence image, the interacting sources are not very clearly detected, but in the corrected imaginary coherence image, the first and the third sources are detected with reasonable clarity. In these results, the difference between the imaginary coherence and corrected imaginary coherence images is significantly large because the magnitude envelope coherence between the source time courses is as large as 0.7.

We computed the image of the envelope-to-envelope correlation, and the results are shown in Fig. 7.6. In this figure, the image of the envelope correlation is shown in (a) and the image of the residual envelope correlation is in (b). In both images, the first and the third sources that were interacting with the second (seed) source can be observed. While the original envelope correlation image contains seed blur, the residual envelope correlation image is free from such spurious activity.

We finally show the results of statistical thresholding method described in Sect. 7.8. The method was applied to the alpha-band coherence imaging results in Fig. 7.3. The thresholded images are shown in Fig. 7.7. The spurious baseline activity existing in the unthresholded images in Fig. 7.3 are removed, and it is much easier to interpret the results in the thresholded images in Fig. 7.7.

References

1. J.-M. Schoffelen, J. Gross, Source connectivity analysis with MEG and EEG. Hum. Brain Mapp. **30**, 1857–1865 (2009)
2. J. Gross, J. Kujara, M. Hämäläinen, L. Timmermann, A. Schnitzler, R. Salmelin, Dynamic imaging of coherent sources: studying neural interactions in the human brain. Proc. Natl. Acad. Sci. U.S.A. **98**, 694–699 (2001)
3. A.G. Guggisberg, S.M. Honma, A.M. Findlay, S.S. Dalal, H.E. Kirsch, M.S. Berger, S.S. Nagarajan, Mapping functional connectivity in patients with brain lesions. Ann. Neurol. **63**, 193–203 (2007)
4. P. Belardinelli, L. Ciancetta, M. Staudt, V. Pizzella, A. Londei, N.B.G.L. Romani, C. Braun, Cerebro-muscular and cerebro-cerebral coherence in patients with pre- and perinatally acquired unilateral brain lesions. NeuroImage **37**, 1301–1314 (2007)
5. W.H.R. Miltner, C. Braun, M. Arnold, H. Witte, E. Taub, Coherence of gamma-band EEG activity as a basis for associative learning. Nature **397**, 434–436 (1999)
6. K. Sekihara, S.S. Nagarajan, *Adaptive Spatial Filters for Electromagnetic Brain Imaging* (Springer, Berlin, 2008)
7. K. Sekihara, J.P. Owen, S. Trisno, S.S. Nagarajan, Removal of spurious coherence in MEG source-space coherence analysis. IEEE Trans. Biomed. Eng. **58**, 3121–3129 (2011)
8. G. Nolte, O.B.L. Wheaton, Z. Mari, S. Vorbach, M. Hallett, Identifying true brain interaction from EEG data using the imaginary part of coherency. Clin. Neurophysiol. **115**, 2292–2307 (2004)
9. R.D. Pascual-Marqui, Instantaneous and lagged measurements of linear and nonlinear dependence between groups of multivariate time series: frequency decomposition (2007). arXiv preprint arXiv:0711.1455

10. E.J. Hannan, *Multiple Time Series*, vol. 38 (Wiley, New York, 2009)
11. D.R. Brillinger, *Time Series: Data Analysis and Theory*, vol. 36 (Siam, Philadelphia, 2001)
12. A. Ewald, L. Marzetti, F. Zappasodi, F.C. Meinecke, G. Nolte, Estimating true brain connectivity from EEG/MEG data invariant to linear and static transformations in sensor space. NeuroImage **60**(1), 476–488 (2012)
13. F. Shahbazi Avarvand, A. Ewald, G. Nolte, Localizing true brain interactions from EEG and MEG data with subspace methods and modified beamformers. Comput. Math. Methods Med. **2012**, 402341 (2012)
14. A. Bruns, R. Eckhorn, Task-related coupling from high- to low-frequency signals among visual cortical areas in human subdural recordings. Int. J. Psychophysiol. **51**, 97–116 (2004)
15. K. Sekihara, S.S. Nagarajan, Residual coherence and residual envelope correlation in MEG/EEG source-space connectivity analysis, in *Conference of Proceedings of the IEEE Engineering in Medicine and Biology Society*, pp. 4417–7 (2013)
16. J.F. Hipp, D.J. Hawellek, M. Corbetta, M. Siegel, A.K. Engel, Large-scale cortical correlation structure of spontaneous oscillatory activity. Nat. Neurosci. **15**(6), 884–890 (2012)
17. J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, J.D. Farmer, Testing for nonlinearity in time series: the method of surrogate data. Phys. D **58**, 77–94 (1992)
18. L. Faes, G.D. Pinna, A. Porta, R. Maestri, G. Nollo, Surrogate data analysis for assessing the significance of the coherence function. IEEE Trans. Biomed. Eng. **51**, 1156–1166 (2004)
19. Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. B **57**, 289–300 (1995)
20. T.E. Nichols, S. Hayasaka, Controlling the familywise error rate in functional neuroimaging: Aj neural eng. comparative review. Stat. Methods Med. Res. **12**, 419–446 (2003)
21. K.P. Westlake, L.B. Hinkley, M. Bucci, A.G. Guggisberg, A.M. Findlay, R.G. Henry, S.S. Nagarajan, N. Byl, Resting state alpha-band functional connectivity and recovery after stroke. Exp. Neurol. **237**(1), 160–169 (2012)
22. P.E. Tarapore, A.M. Findlay, S.C. LaHue, H. Lee, S.M. Honma, D. Mizuiri, T.L. Luks, G.T. Manley, S.S. Nagarajan, P. Mukherjee, Resting state magnetoencephalography functional connectivity in traumatic brain injury: clinical article. J. Neurosurg. **118**(6), 1306–1316 (2013)
23. L.B. Hinkley, J.P. Owen, M. Fisher, A.M. Findlay, S. Vinogradov, S.S. Nagarajan, Cognitive impairments in schizophrenia as assessed through activation and connectivity measures of magnetoencephalography (MEG) data. Front. Hum. Neurosci. **3**, 73 (2009)
24. K.G. Ranasinghe, L.B. Hinkley, A.J. Beagle, D. Mizuiri, A.F. Dowling, S.M. Honma, M.M. Finucane, C. Scherling, B.L. Miller, S.S. Nagarajan et al., Regional functional connectivity predicts distinct cognitive impairments in Alzheimers disease spectrum. NeuroImage: Clin. **5**, 385–395 (2014)
25. J. Sarvas, Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. Phys. Med. Biol. **32**, 11–22 (1987)

Chapter 8
Estimation of Causal Networks: Source-Space Causality Analysis

8.1 Introduction

This chapter reviews the methodology for estimating causal relationships among cortical activities in MEG/EEG source space analysis. The source space causality analysis computes some types of causality measures using the estimated time series of the source activities of interest. Commonly-used measures are the Granger causality related measures, which are based on the MVAR modeling of the voxel time series. We first describe these Granger causality based measures, including the original definition in the temporal domain [1, 2], Geweke's extension to the spectral domain [3, 4], and related measures such as the directed transfer function (DTF) [5] and the partial directed coherence (PDC) [6]. The transfer entropy [7], which is a representative non-parametric measure, is also discussed, with a proof of its equivalence to the Granger causality under the Gaussianity assumption. The last section of this chapter describes methods of estimating the MVAR coefficients. Here we present a conventional least-square-based algorithm and a method based on the sparse Bayesian inference for this MVAR estimation problem. The sparse Bayesian algorithm outperforms the conventional method when the sensor data contains a large amount of interferences.

8.2 Multivariate Vector Autoregressive (MVAR) Process

8.2.1 MVAR Modeling of Time Series

The Granger-causality measures make use of multivariate vector auto regressive (MVAR) process of the source time series. We first explain the modeling of time series using the MVAR process. A general approach to the source-space causality analysis first chooses a relatively small number of voxels corresponding to the source activities of interest, and the causal relationships among time series of the selected voxels are estimated.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_8

Denoting the number of selected voxels q, we express the q-channel voxel time series using the vector $\boldsymbol{y}(t)$: $\boldsymbol{y}(t) = [y_1(t), \ldots, y_q(t)]^T$ where $y_j(t)$ is the time series of the jth selected voxel at time t. Here the time t is expressed using a unit-less value. We impose multivariate vector autoregressive (MVAR) modeling on the time series $\boldsymbol{y}(t)$, such that

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{A}(p)\boldsymbol{y}(t-p) + \boldsymbol{e}(t). \tag{8.1}$$

Here, $\boldsymbol{A}(p)$ is the AR coefficient matrix, P is the model order, and $\boldsymbol{e}(t)$ is the residual vector. The MVAR process is expressed in the frequency domain. By computing the Fourier transform of Eq. (8.1), we get

$$\boldsymbol{y}(f) = \sum_{p=1}^{P} \boldsymbol{A}(p)e^{-2\pi ipf}\boldsymbol{y}(f) + \boldsymbol{e}(f), \tag{8.2}$$

where the Fourier transforms of $\boldsymbol{y}(t)$ and $\boldsymbol{e}(t)$ are expressed in $\boldsymbol{y}(f)$ and $\boldsymbol{e}(f)$. We here use the relationship,

$$\int \boldsymbol{y}(t-p)\exp(-2\pi ift)\,\mathrm{d}t = e^{-2\pi ipf}\boldsymbol{y}(f). \tag{8.3}$$

Equation (8.2) is also expressed as

$$\left[\boldsymbol{I} - \sum_{p=1}^{P} \boldsymbol{A}(p)e^{-2\pi ipf}\right]\boldsymbol{y}(f) = \boldsymbol{e}(f). \tag{8.4}$$

Defining a $q \times q$ matrix $\bar{\boldsymbol{A}}(f)$ such that

$$\bar{\boldsymbol{A}}(f) = \boldsymbol{I} - \sum_{p=1}^{P} \boldsymbol{A}(p)e^{-2\pi ipf}, \tag{8.5}$$

we can obtain

$$\bar{\boldsymbol{A}}(f)\boldsymbol{y}(f) = \boldsymbol{e}(f). \tag{8.6}$$

Also, defining $\boldsymbol{H}(f) = \bar{\boldsymbol{A}}(f)^{-1}$, the relationship

$$\boldsymbol{y}(f) = \boldsymbol{H}(f)\boldsymbol{e}(f) \tag{8.7}$$

can be obtained. According to the equation above, $\boldsymbol{e}(f)$ and $\boldsymbol{y}(f)$ can be interpreted as the input and the output of a linear-system whose transfer function is $\boldsymbol{H}(f)$.

8.2.2 Coherence and Partial Coherence of the MVAR Process

Using Eq. (8.7), we can derive the relationship

$$\boldsymbol{S}(f) = \boldsymbol{H}(f)\boldsymbol{\Sigma}\boldsymbol{H}^H(f), \tag{8.8}$$

where the superscript H indicates the Hermitian transpose.[1] In the equation above, $\boldsymbol{S}(f)$ is the cross spectrum matrix, which is given by $\boldsymbol{S}(f) = \langle \boldsymbol{y}(f)\boldsymbol{y}^H(f)\rangle$ where $\langle\cdot\rangle$ indicates the ensemble average. Also, $\boldsymbol{\Sigma}$ is the covariance matrix of the residual, which is equal to $\boldsymbol{\Sigma} = \langle \boldsymbol{e}(f)\boldsymbol{e}^H(f)\rangle$. Using these definitions, the coherence between the time series of the jth and the kth channels is expressed as

$$\phi_{j,k}(f) = \frac{S_{j,k}}{\sqrt{S_{j,j}\,S_{k,k}}} = \frac{\boldsymbol{h}_j^H\boldsymbol{\Sigma}\boldsymbol{h}_k}{\sqrt{[\boldsymbol{h}_j^H\boldsymbol{\Sigma}\boldsymbol{h}_j][\boldsymbol{h}_k^H\boldsymbol{\Sigma}\boldsymbol{h}_k]}}, \tag{8.9}$$

where $\boldsymbol{h}_j$ is the jth column of $\boldsymbol{H}^H$ such that $\boldsymbol{H}^H = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_q]$.

The partial coherence between the jth and the kth channels, $\kappa_{j,k}$, is expressed as [8, 9]

$$\kappa_{j,k}(f) = \frac{\boldsymbol{M}_{j,k}}{\sqrt{\boldsymbol{M}_{j,j}\,\boldsymbol{M}_{k,k}}}, \tag{8.10}$$

where $\boldsymbol{M}_{j,k}$ is the minor of the matrix $\boldsymbol{S}(f)$; the minor is a determinant value of a matrix formed by the jth row and the kth column removed from $\boldsymbol{S}(f)$. The partial coherence $\kappa_{j,k}(f)$ is a measure of the coherence between the time series of the jth and the kth channels where the influence of other channels is removed. In other words, it expresses the direct interaction between these two channels.

Let us derive a convenient formula for computing the partial coherence. We use the relationship called Cramer's rule [10]:

$$\left[\boldsymbol{S}^{-1}(f)\right]_{j,k} = \frac{(-1)^{j+k}\boldsymbol{M}_{k,j}}{|\boldsymbol{S}(f)|},$$

where $[\boldsymbol{S}^{-1}(f)]_{j,k}$ indicates the (j,k) component of the matrix $\boldsymbol{S}^{-1}(f)$. Considering the fact that $\boldsymbol{S}(f)$ is a positive semidefinite Hermitian matrix, the equation above can be changed to

$$\boldsymbol{M}_{j,k} = (-1)^{j+k}\left[\boldsymbol{S}^{-1}(f)\right]_{j,k}|\boldsymbol{S}(f)|. \tag{8.11}$$

[1] The Hermitian transpose is the matrix transpose with the complex conjugation.

We then rewrite Eq. (8.10) into

$$\kappa_{j,k}(f) = \frac{(-1)^{j+k}\left[\boldsymbol{S}^{-1}(f)\right]_{j,k}|\boldsymbol{S}(f)|}{\sqrt{(-1)^{2j}\left[\boldsymbol{S}^{-1}(f)\right]_{j,j}|\boldsymbol{S}(f)|(-1)^{2k}\left[\boldsymbol{S}^{-1}(f)\right]_{k,k}|\boldsymbol{S}(f)|}} = \frac{\left[\boldsymbol{S}^{-1}(f)\right]_{j,k}}{\sqrt{\left[\boldsymbol{S}^{-1}(f)\right]_{j,j}\left[\boldsymbol{S}^{-1}(f)\right]_{k,k}}}. \tag{8.12}$$

On the other hand, using Eq. (8.8), we have the relationship

$$\boldsymbol{S}^{-1}(f) = \left[\boldsymbol{H}(f)\boldsymbol{\Sigma}\boldsymbol{H}^{H}(f)\right]^{-1} = \left[\boldsymbol{H}^{-1}(f)\right]^{H}\boldsymbol{\Sigma}^{-1}\boldsymbol{H}^{-1}(f) = \bar{\boldsymbol{A}}^{H}(f)\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{A}}(f). \tag{8.13}$$

The kth column vector of $\bar{\boldsymbol{A}}(f)$ is denoted $\bar{\boldsymbol{a}}_k$, i.e., $\bar{\boldsymbol{A}}(f) = \left[\bar{\boldsymbol{a}}_1, \ldots, \bar{\boldsymbol{a}}_q\right]$. We can then obtain

$$\left[\boldsymbol{S}^{-1}(f)\right]_{j,k} = \bar{\boldsymbol{a}}_j^H \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{a}}_k,$$

and thus derive

$$\kappa_{j,k}(f) = \frac{\bar{\boldsymbol{a}}_j^H \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{a}}_k}{\sqrt{[\bar{\boldsymbol{a}}_j^H \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{a}}_j][\bar{\boldsymbol{a}}_k^H \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{a}}_k]}}. \tag{8.14}$$

The equation above is used for deriving partial directed coherence in Sect. 8.5.3.

8.3 Time-Domain Granger Causality

8.3.1 Granger Causality for a Bivariate Process

Let us first consider bivariate cases in which only a pair of two time series $y_1(t)$ and $y_2(t)$ are considered. In the first case, $y_1(t)$ and $y_2(t)$ are determined by using only their own past values, i.e., the following relationships hold:

$$y_1(t) = \sum_{p=1}^{\infty} A_{1,1}(p)y_1(t-p) + e_1(t), \tag{8.15}$$

$$y_2(t) = \sum_{p=1}^{\infty} A_{2,2}(p)y_2(t-p) + e_2(t). \tag{8.16}$$

We then consider the second case in which the following relationships hold:

$$y_1(t) = \sum_{p=1}^{\infty} A_{1,1}(p) y_1(t-p) + \sum_{p=1}^{\infty} A_{1,2}(p) y_2(t-p) + \epsilon_1(t), \tag{8.17}$$

$$y_2(t) = \sum_{p=1}^{\infty} A_{2,1}(p) y_1(t-p) + \sum_{p=1}^{\infty} A_{2,2}(p) y_2(t-p) + \epsilon_2(t). \tag{8.18}$$

The difference between the first and the second cases above is that in the second case, time series $y_1(t)$ and $y_2(t)$ are determined by the past values of both the time series of $y_1(t)$ and $y_2(t)$, while in the first case, the time series $y_1(t)$ is determined only by the past values of $y_1(t)$ and the time series $y_2(t)$ is determined only by the past values of $y_2(t)$.

We define the variances of the residual terms appearing above, such that

$$V(e_1(t)) = \Sigma_e^1, \tag{8.19}$$

$$V(e_2(t)) = \Sigma_e^2, \tag{8.20}$$

$$V(\epsilon_1(t)) = \Sigma_\epsilon^1, \tag{8.21}$$

$$V(\epsilon_2(t)) = \Sigma_\epsilon^2, \tag{8.22}$$

where $V(\cdot)$ indicates the variance. Then, the Granger causality, $\mathcal{G}_{2\to 1}$, is defined as

$$\mathcal{G}_{2\to 1} = \log\left[\frac{\Sigma_e^1}{\Sigma_\epsilon^1}\right]. \tag{8.23}$$

The meaning of $\mathcal{G}_{2\to 1}$ is that, if the residual variance is reduced by using both the past values of $y_1(t)$ and $y_2(t)$ when estimating the current value of $y_1(t)$, the past of $y_2(t)$ affects the current value of $y_1(t)$. In such a case, we can conclude that there is an information flow from the time series $y_2(t)$ to the time series $y_1(t)$. The above $\mathcal{G}_{2\to 1}$ can quantify the amount of this information flow. Similarly, we can define $\mathcal{G}_{1\to 2}$ such that

$$\mathcal{G}_{1\to 2} = \log\left[\frac{\Sigma_e^2}{\Sigma_\epsilon^2}\right]. \tag{8.24}$$

This $\mathcal{G}_{1\to 2}$ expresses the information flow from the time series $y_1(t)$ to the time series $y_2(t)$.

8.3.2 Multivariate Granger Causality

The idea of Granger causality is extended to a general multivariate case. Let us define q-dimensional time series as $\boldsymbol{y}(t)$ and r-dimensional time series as $\boldsymbol{x}(t)$. We consider

a case in which $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ obey the MVAR process such that

$$\boldsymbol{x}(t) = \sum_{p=1}^{P} \boldsymbol{A}_x(p)\boldsymbol{x}(t-p) + \boldsymbol{e}_x(t), \tag{8.25}$$

and

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{A}_y(p)\boldsymbol{y}(t-p) + \boldsymbol{e}_y(t). \tag{8.26}$$

We define covariance matrices of the residuals in this case, such that

$$\boldsymbol{\Sigma}_e^x = \langle \boldsymbol{e}_x(t)\boldsymbol{e}_x^T(t)\rangle, \tag{8.27}$$
$$\boldsymbol{\Sigma}_e^y = \langle \boldsymbol{e}_y(t)\boldsymbol{e}_y^T(t)\rangle, \tag{8.28}$$

where $\langle\cdot\rangle$ indicates the ensemble average.

We next assume that $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ obey the following MVAR process

$$\boldsymbol{x}(t) = \sum_{p=1}^{P} \boldsymbol{A}_x(p)\boldsymbol{x}(t-p) + \sum_{p=1}^{P} \boldsymbol{B}_y(p)\boldsymbol{y}(t-p) + \boldsymbol{\epsilon}_x(t), \tag{8.29}$$

and

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{A}_y(p)\boldsymbol{y}(t-p) + \sum_{p=1}^{P} \boldsymbol{B}_x(p)\boldsymbol{x}(t-p) + \boldsymbol{\epsilon}_y(t). \tag{8.30}$$

We can define covariance matrices of the residuals, such that

$$\boldsymbol{\Sigma}_\epsilon^x = \langle \boldsymbol{\epsilon}_x(t)\boldsymbol{\epsilon}_x^T(t)\rangle, \tag{8.31}$$
$$\boldsymbol{\Sigma}_\epsilon^y = \langle \boldsymbol{\epsilon}_y(t)\boldsymbol{\epsilon}_y^T(t)\rangle. \tag{8.32}$$

Using the same idea for Eqs. (8.23) and (8.24), the multivariate Granger causality $\mathcal{G}_{\boldsymbol{x}\to\boldsymbol{y}}$ and $\mathcal{G}_{\boldsymbol{y}\to\boldsymbol{x}}$ are given by

$$\mathcal{G}_{\boldsymbol{x}\to\boldsymbol{y}} = \log \frac{|\boldsymbol{\Sigma}_e^y|}{|\boldsymbol{\Sigma}_\epsilon^y|}, \tag{8.33}$$
$$\mathcal{G}_{\boldsymbol{y}\to\boldsymbol{x}} = \log \frac{\boldsymbol{\Sigma}_e^x}{|\boldsymbol{\Sigma}_\epsilon^x|}, \tag{8.34}$$

where $|\cdot|$ indicates the matrix determinant.

8.3.3 Total Interdependence

Let us define the $(q+r)$-dimensional augmented time series $\boldsymbol{z}(t) = [\boldsymbol{x}^T(t), \boldsymbol{y}^T(t)]^T$, which obeys

$$\boldsymbol{z}(t) = \sum_{p=1}^{P} \boldsymbol{A}_z(p)\boldsymbol{z}(t-p) + \boldsymbol{\epsilon}_z(t). \tag{8.35}$$

The covariance matrix of the residual vector $\boldsymbol{\epsilon}_z(t)$ is defined as $\boldsymbol{\Sigma}_\epsilon^z$. When the time series $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are independent, $\boldsymbol{\epsilon}_z(t)$ is expressed as $\boldsymbol{\epsilon}_z(t) = [\boldsymbol{e}_x(t), \boldsymbol{e}_y(t)]^T$, and $\boldsymbol{\Sigma}_\epsilon^z$ is expressed as

$$\boldsymbol{\Sigma}_\epsilon^z = \langle \boldsymbol{\epsilon}_z(t)\boldsymbol{\epsilon}_z^T(t)\rangle = \begin{bmatrix} \boldsymbol{\Sigma}_e^x & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_e^y \end{bmatrix}. \tag{8.36}$$

When $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are not independent, it is expressed as

$$\boldsymbol{\Sigma}_\epsilon^z = \begin{bmatrix} \boldsymbol{\Sigma}_\epsilon^x & \boldsymbol{\Sigma}_\epsilon^{xy} \\ \boldsymbol{\Sigma}_\epsilon^{yx} & \boldsymbol{\Sigma}_\epsilon^y \end{bmatrix}, \tag{8.37}$$

where

$$\boldsymbol{\Sigma}_\epsilon^{xy} = \langle \boldsymbol{\epsilon}_x(t)\boldsymbol{\epsilon}_y^T(t)\rangle, \tag{8.38}$$

$$\boldsymbol{\Sigma}_\epsilon^{yx} = \langle \boldsymbol{\epsilon}_y(t)\boldsymbol{\epsilon}_x^T(t)\rangle = (\boldsymbol{\Sigma}_\epsilon^{xy})^T. \tag{8.39}$$

The total interdependence between $\boldsymbol{x}$ and $\boldsymbol{y}$, $\mathcal{I}_{\{\boldsymbol{x},\boldsymbol{y}\}}$, is defined such that

$$\mathcal{I}_{\{\boldsymbol{x},\boldsymbol{y}\}} = \log \frac{\begin{vmatrix} \boldsymbol{\Sigma}_e^x & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_e^y \end{vmatrix}}{\left|\boldsymbol{\Sigma}_\epsilon^z\right|} = \log \frac{\left|\boldsymbol{\Sigma}_e^x\right|\left|\boldsymbol{\Sigma}_e^y\right|}{\left|\boldsymbol{\Sigma}_\epsilon^z\right|}. \tag{8.40}$$

This total interdependence expresses the deviation of the value of $\left|\boldsymbol{\Sigma}_\epsilon^z\right|$ from its value obtained when $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are independent. Using Eqs. (8.33) and (8.34), it is easy to show the relationship

$$\mathcal{I}_{\{\boldsymbol{x},\boldsymbol{y}\}} - \mathcal{G}_{\boldsymbol{x}\to\boldsymbol{y}} - \mathcal{G}_{\boldsymbol{y}\to\boldsymbol{x}} = \log \frac{\left|\boldsymbol{\Sigma}_\epsilon^x\right|\left|\boldsymbol{\Sigma}_\epsilon^y\right|}{\left|\boldsymbol{\Sigma}_\epsilon^z\right|}. \tag{8.41}$$

According to Geweke [3], the right-hand side of the equation above is defined as $\mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}}$:

$$\mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}} = \log \frac{\left|\boldsymbol{\Sigma}_\epsilon^x\right|\left|\boldsymbol{\Sigma}_\epsilon^y\right|}{\left|\boldsymbol{\Sigma}_\epsilon^z\right|}, \tag{8.42}$$

and this $\mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}}$ is called the instantaneous dependence. The following relationship holds among the causality measures mentioned above,

$$\mathcal{I}_{\{\boldsymbol{x},\boldsymbol{y}\}} = \mathcal{G}_{\boldsymbol{x}\to\boldsymbol{y}} + \mathcal{G}_{\boldsymbol{y}\to\boldsymbol{x}} + \mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}}.$$

That is, the total interdependence between the two time series $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ is expressed as the summation of the Granger causality from $\boldsymbol{x}(t)$ to $\boldsymbol{y}(t)$, the Granger causality from $\boldsymbol{y}(t)$ to $\boldsymbol{x}(t)$, and the instantaneous dependence.

The rationale of this instantaneous dependence can be explained in the following manner. Using the determinant identity in Eq. (C.94), Eq. (8.42) is rewritten as

$$\mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}} = \log\left|\boldsymbol{\Sigma}_{\epsilon}^{x}\right| - \log|\boldsymbol{\Delta}|, \tag{8.43}$$

where

$$\boldsymbol{\Delta} = \boldsymbol{\Sigma}_{\epsilon}^{x} - \boldsymbol{\Sigma}_{\epsilon}^{xy}(\boldsymbol{\Sigma}_{\epsilon}^{y})^{-1}\boldsymbol{\Sigma}_{\epsilon}^{yx}. \tag{8.44}$$

Let us consider the linear regression in which the residual signal $\boldsymbol{\epsilon}_x(t)$ is regressed by the other residual signal $\boldsymbol{\epsilon}_y(t)$. In Eq. (8.44), $\boldsymbol{\Delta}$ expresses the covariance matrix of the residual of this linear regression, according to the arguments in Sect. C.3.3.

Let us suppose a case where $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ contain a common instantaneous interaction $\boldsymbol{\nu}(t)$, which does not exist in the past values of $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$. In such a case, both the residual signals $\boldsymbol{\epsilon}_x(t)$ and $\boldsymbol{\epsilon}_y(t)$ contain the common component $\boldsymbol{\nu}(t)$, and because of this, $\boldsymbol{\nu}(t)$ is regressed out when $\boldsymbol{\epsilon}_x(t)$ is regressed by $\boldsymbol{\epsilon}_y(t)$. Therefore, $\log|\boldsymbol{\Delta}|$ does not contain the influence of this common instantaneous component, while $\log\left|\boldsymbol{\Sigma}_{\epsilon}^{x}\right|$ does. Accordingly, $\mathcal{I}_{\{\boldsymbol{x}\cdot\boldsymbol{y}\}}$ represents the influence of the instantaneous component alone.

8.4 Spectral Granger Causality: Geweke Measures

8.4.1 Basic Relationships in the Frequency Domain

Since brain activities have spectral dependence, we naturally like to perform causality analysis in the spectral domain. The extension of Granger causality analysis into the spectral domain has been investigated by Geweke [3, 4]. The arguments in this section are according to Ding [11], and we restrict our arguments to a bivariate process[2]:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{p=1}^{\infty} \boldsymbol{A}(p) \begin{bmatrix} x(t-p) \\ y(t-p) \end{bmatrix} + \begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix}, \tag{8.45}$$

[2] The extension of the arguments here to a general multivariate case remains unknown.

where $\boldsymbol{A}(p)$ is a (2×2) coefficient matrix. We can derive Eq. (8.7), which is explicitly written as

$$\begin{bmatrix} x(f) \\ y(f) \end{bmatrix} = \begin{bmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{bmatrix} \begin{bmatrix} e_x(f) \\ e_y(f) \end{bmatrix}. \tag{8.46}$$

In the expressions above, the explicit notation of the frequency f is omitted from the components of $\boldsymbol{H}(f)$. Denoting the residual covariance matrix such that

$$\begin{bmatrix} \langle |e_x(f)|^2 \rangle & \langle e_x(f) e_y(f)^* \rangle \\ \langle e_y(f) e_x(f)^* \rangle & \langle |e_y(f)|^2 \rangle \end{bmatrix} = \begin{bmatrix} \Sigma_x & \gamma \\ \gamma^* & \Sigma_y \end{bmatrix}, \tag{8.47}$$

where the superscript $*$ indicates the complex conjugation. Equation (8.8) is explicitly written as

$$\begin{bmatrix} \langle |x(f)|^2 \rangle & \langle x(f) y(f)^* \rangle \\ \langle y(f) x(f)^* \rangle & \langle |y(f)|^2 \rangle \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} \Sigma_x & \gamma \\ \gamma^* & \Sigma_y \end{bmatrix} \boldsymbol{H}^H. \tag{8.48}$$

8.4.2 Total Interdependence and Coherence

We first define the total interdependence between x and y, which corresponds to $\mathcal{I}_{\{\boldsymbol{x},\boldsymbol{y}\}}$ in Eq. (8.40). Using the arguments in Sect. 8.3.3, the total interdependence between x and y in the spectral domain is defined as

$$\begin{aligned} f_{\{x,y\}} &= \log \frac{\begin{vmatrix} \langle |x(f)|^2 \rangle & 0 \\ 0 & \langle |y(f)|^2 \rangle \end{vmatrix}}{|\boldsymbol{S}(f)|} \\ &= \log \frac{\langle |x(f)|^2 \rangle \langle |y(f)|^2 \rangle}{\langle |x(f)|^2 \rangle \langle |y(f)|^2 \rangle - |\langle x(f) y(f)^* \rangle|^2}. \end{aligned} \tag{8.49}$$

Comparing the equation above to Eq. (7.40) in Chap. 7, it can immediately be seen that under the Gaussianity assumption, the spectral domain total interdependence is equal to the mutual information discussed in Sect. 7.5.2. Therefore, using the same arguments, the total interdependence $f_{\{x,y\}}$ is related to the magnitude coherence $|\phi(f)|$ through

$$f_{\{x,y\}}(f) = -\log(1 - |\phi(f)|^2), \tag{8.50}$$

where

$$|\phi(f)|^2 = \frac{|\langle x(f) y(f)^* \rangle|^2}{\langle |x(f)|^2 \rangle \langle |y(f)|^2 \rangle}. \tag{8.51}$$

8.4.3 Deriving Causal Relationships in the Frequency Domain

In order to derive causal relationships between two channels, we should distinguish between the amount of the signal power coming from its own past and the amount of the power coming from the past of the other channel. On the basis of Eq. (8.46), we are able to do this separation. That is, using Eq. (8.46), we have

$$x(f) = H_{xx}e_x(f) + H_{xy}e_y(f). \tag{8.52}$$

On the right-hand side of the equation above, the first term is the intrinsic term representing the influence of the past of $x(t)$, and the second term represents the influence of the past of $y(t)$. We then decompose the power of the signal $x(f)$, such that

$$\langle |x(f)|^2 \rangle = H_{xx}\Sigma_x H_{xx}^* + H_{xy}\Sigma_y H_{xy}^* + 2\Re\left[\gamma H_{xx}H_{xy}^*\right]. \tag{8.53}$$

On the right-hand side above, the first term represents the intrinsic term and the second term expresses the influence from the past of $y(f)$, i.e., the causal term. However, it is unclear how much amount of the information represented by the third term is attributed to the intrinsic or the causal terms. Therefore, to attain the intrinsic/causal factorization, we first transform Eq. (8.46) into a domain where the innovation terms are uncorrelated, and, in this domain, we factorize the total signal power into the intrinsic and causal terms.

Let us factorize $\langle |x(f)|^2 \rangle$ in this manner. The transformation is performed in this case using $\boldsymbol{\Pi}$ defined such that

$$\boldsymbol{\Pi} = \begin{bmatrix} 1 & 0 \\ -\gamma^*/\Sigma_x & 1 \end{bmatrix}. \tag{8.54}$$

Using Eq. (8.46) and

$$\boldsymbol{\Pi}^{-1} = \begin{bmatrix} 1 & 0 \\ \gamma^*/\Sigma_x & 1 \end{bmatrix},$$

we have

$$\begin{bmatrix} x(f) \\ y(f) \end{bmatrix} = \begin{bmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma^*/\Sigma_x & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\gamma^*/\Sigma_x & 1 \end{bmatrix} \begin{bmatrix} e_x(f) \\ e_y(f) \end{bmatrix}. \tag{8.55}$$

We then have

$$\begin{bmatrix} x(f) \\ y(f) \end{bmatrix} = \begin{bmatrix} H_{xx} + \frac{\gamma^*}{\Sigma_x}H_{xy} & H_{xy} \\ H_{yx} + \frac{\gamma^*}{\Sigma_x}H_{yy} & H_{yy} \end{bmatrix} \begin{bmatrix} e_x(f) \\ -\frac{\gamma^*}{\Sigma_x}e_x(f) + e_y(f) \end{bmatrix}. \tag{8.56}$$

The cross correlation of the two innovation components, $e_x(f)$ and $-\frac{\gamma^*}{\Sigma_x}e_x(f)+e_y(f)$ is zero, because

$$\left\langle\left[-\frac{\gamma^*}{\Sigma_x}e_x(f)+e_y(f)\right]e_x(f)^*\right\rangle = -\frac{\gamma^*}{\Sigma_x}\langle|e_x(f)|^2\rangle+\langle e_y(f)e_x(f)^*\rangle = -\frac{\gamma^*}{\Sigma_x}\Sigma_x+\gamma^*=0 \tag{8.57}$$

We can thus derive

$$\begin{bmatrix}\langle|x(f)|^2\rangle & \langle x(f)y(f)^*\rangle\\ \langle y(f)x(f)^*\rangle & \langle|y(f)|^2\rangle\end{bmatrix} = \begin{bmatrix}H_{xx}+\frac{\gamma^*}{\Sigma_x}H_{xy} & H_{xy}\\ H_{yx}+\frac{\gamma^*}{\Sigma_x}H_{yy} & H_{yy}\end{bmatrix}\begin{bmatrix}\Sigma_x & 0\\ 0 & \Sigma_y-\frac{|\gamma|^2}{\Sigma_x}\end{bmatrix}\begin{bmatrix}H^*_{xx}+\frac{\gamma}{\Sigma_x}H^*_{xy} & H^*_{yx}+\frac{\gamma}{\Sigma_x}H^*_{yy}\\ H^*_{xy} & H^*_{yy}\end{bmatrix}. \tag{8.58}$$

Therefore, the signal power $\langle|x(f)|^2\rangle$ is factorized to

$$\langle|x(f)|^2\rangle = \left(H_{xx}+\frac{\gamma^*}{\Sigma_x}H_{xy}\right)\Sigma_x\left(H^*_{xx}+\frac{\gamma}{\Sigma_x}H^*_{xy}\right)+H_{xy}\left(\Sigma_y-\frac{|\gamma|^2}{\Sigma_x}\right)H^*_{xy}. \tag{8.59}$$

On the right-hand side of this equation, the first term is interpreted as the intrinsic term, which represents the influence of the past of $x(t)$ on the power spectrum $\langle|x(f)|^2\rangle$. The second term is interpreted as the causal influence term, which represents the influence of the past of $y(t)$ on $\langle|x(f)|^2\rangle$. Thus, the spectral Granger causality, $f_{y\to x}(f)$, is defined as

$$\begin{aligned} f_{y\to x}(f) &= \log\frac{\langle|x(f)|^2\rangle}{(H_{xx}+\frac{\gamma^*}{\Sigma_x}H_{xy})\Sigma_x(H^*_{xx}+\frac{\gamma}{\Sigma_x}H^*_{xy})}\\ &= -\log\frac{\langle|x(f)|^2\rangle-H_{xy}(\Sigma_y-\frac{|\gamma|^2}{\Sigma_x})H^*_{xy}}{\langle|x(f)|^2\rangle}\\ &= -\log\left[1-\frac{(\Sigma_y-\frac{|\gamma|^2}{\Sigma_x})|H_{xy}(f)|^2}{\langle|x(f)|^2\rangle}\right]. \end{aligned} \tag{8.60}$$

The spectral Granger causality, $f_{x\to y}(f)$, can be derived in exactly the same manner, using

$$\boldsymbol{\Pi} = \begin{bmatrix}1 & -\gamma^*/\Sigma_y\\ 0 & 1\end{bmatrix},$$

and the results are given by

$$f_{x\to y}(f) = -\log\left[1 - \frac{(\Sigma_x - \frac{|\gamma|^2}{\Sigma_y})|H_{yx}(f)|^2}{\langle |y(f)|^2\rangle}\right]. \tag{8.61}$$

8.5 Other MVAR-Modeling-Based Measures

8.5.1 Directed Transfer Function (DTF)

The directed transfer function (DTF) [5] is derived from a definition of causality that is somewhat different from the Granger causality. For a bivariate AR process, we have the relationships

$$y_1(t) = \sum_{p=1}^{P} A_{1,1}(p)y_1(t-p) + \sum_{p=1}^{P} A_{1,2}(p)y_2(t-p) + \epsilon_1(t)$$
$$y_2(t) = \sum_{p=1}^{P} A_{2,1}(p)y_1(t-p) + \sum_{p=1}^{P} A_{2,2}(p)y_2(t-p) + \epsilon_2(t).$$

It can be intuitively clear that, for example, the influence of the past values of $y_2(t)$ on the current value of $y_1(t)$ can be assessed by the values of the AR coefficients $A_{1,2}(p)$, (where $p = 1, \ldots, P$). Namely, the causal influence of $y_2(t)$ on $y_1(t)$ can be assessed by using

$$\sqrt{\sum_{p=1}^{P} A_{1,2}^2(p)}. \tag{8.62}$$

The spectral domain quantities that play a role similar to the above quantity are

$$|\bar{A}_{1,2}(f)| \quad \text{and} \quad |H_{1,2}(f)|, \tag{8.63}$$

where the matrix $\bar{\boldsymbol{A}}(f)$ is defined in Eq. (8.5) and the transfer matrix $\boldsymbol{H}(f)$ is defined as $\boldsymbol{H}(f) = \bar{\boldsymbol{A}}(f)^{-1}$. The directed transfer function (DTF) makes use of $\boldsymbol{H}(f)$ to express the causal relationship.

We define the (unnormalized) directed transfer function (DTF) using the elements of the transfer function $\boldsymbol{H}(f)$. Namely, the unnormalized directed transfer function that represents the causal influence of $y_2(t)$ on $y_1(t)$ is defined as $H_{1,2}(f)$. For a bivariate case, The relationship,

$$H_{1,2}(f) = \frac{\bar{A}_{1,2}(f)}{|\bar{\boldsymbol{A}}(f)|}$$

indicates that the directed transfer function is equal to $\bar{A}_{1,2}(f)$ aside from the scaling constant $|\bar{\boldsymbol{A}}(f)|$.

One advantage of this definition is that the extension to a general multivariate case is straightforward. In a general multivariate case, the unnormalized transfer function is equal to

$$|H_{1,2}(f)| = \frac{|\boldsymbol{M}_{2,1}(f)|}{|\bar{\boldsymbol{A}}(f)|}, \tag{8.64}$$

where $\boldsymbol{M}_{2,1}(f)$ is a minor of the (2, 1) element of the matrix $\bar{\boldsymbol{A}}(f)$. In a trivariate case, using Eq. (8.64), the unnormalized transfer function is derived as

$$|H_{1,2}(f)| = \frac{|\bar{A}_{1,2}\bar{A}_{3,3} - \bar{A}_{3,1}\bar{A}_{2,3}|}{|\bar{\boldsymbol{A}}(f)|}, \tag{8.65}$$

where the explicit notation (f) is omitted in the numerator. On the right hand side, $\bar{A}_{1,2}$ represents the direct influence from channel #1 to channel #2. The equation above shows that even when $\bar{A}_{1,2}$ is equal to zero, $|H_{1,2}|$ cannot equal zero because the term $\bar{A}_{3,1}\bar{A}_{2,3}$ can also not be zero. This term represents the indirect influence of channel #1 on channel #2 via channel #3. That is, DTF contains the indirect causal influence, as well as the direct causal influence.

The normalized DTF can be defined by normalizing the causal influence on the jth channel from all other channels. That is, when total q channels exist, the normalized DTF is defined as

$$\mu_{k\to j} = \frac{H_{j,k}(f)}{\sqrt{[\boldsymbol{h}_j^H \boldsymbol{h}_j]}} = \frac{H_{j,k}(f)}{\sqrt{\sum_{m=1}^{q} |H_{j,m}(f)|^2}}. \tag{8.66}$$

8.5.2 Relationship Between DTF and Coherence

The coherence can be expressed as a result of factorization of the normalized DTF. To show this, we define the directed coherence $\gamma_{j,k}$ such that

$$\gamma_{j,k} = \frac{H_{j,k}(f)}{\sqrt{\boldsymbol{h}_j^H \boldsymbol{\Sigma} \boldsymbol{h}_j}}. \tag{8.67}$$

We then define the column vector

$$\boldsymbol{\gamma}_j(f) = [\gamma_{j,1}^*, \ldots, \gamma_{j,q}^*]^T, \tag{8.68}$$

and using this column vector, coherence is expressed as[3]

$$\phi_{j,k}(f) = \frac{S_{j,k}}{\sqrt{S_{j,j}S_{k,k}}} = \frac{\boldsymbol{h}_j^H \boldsymbol{\Sigma} \boldsymbol{h}_k}{\sqrt{[\boldsymbol{h}_j^H \boldsymbol{\Sigma} \boldsymbol{h}_j][\boldsymbol{h}_k^H \boldsymbol{\Sigma} \boldsymbol{h}_k]}} = \boldsymbol{\gamma}_j^H(f) \boldsymbol{\Sigma} \boldsymbol{\gamma}_k(f). \tag{8.69}$$

This directed coherence $\gamma_{j,k}$ is known to represent the directional influence from the kth to the jth channels. It contains the residual covariance matrix $\boldsymbol{\Sigma}$ whose off-diagonal terms may represent the instantaneous interaction between two channels, according to the arguments in Sect. 8.3.3. Setting $\boldsymbol{\Sigma}$ equal to $\boldsymbol{I}$, we have

$$\gamma_{j,k} = \frac{H_{j,k}(f)}{\sqrt{|\boldsymbol{h}_j^H \boldsymbol{h}_j|}}, \tag{8.70}$$

which is exactly equal to the normalized DTF in Eq. (8.66). Namely, the DTF is a measure for directional influences obtained by removing instantaneous components from the directed coherence. Under the assumption that $\boldsymbol{\Sigma} = \boldsymbol{I}$, coherence $\phi_{j,k}(f)$ can be decomposed of the sum of the product of DTFs, such that

$$\phi_{j,k}(f) = \frac{\sum_{m=1}^{q} H_{j,m}(f) H_{k,m}^*(f)}{\sqrt{[\boldsymbol{h}_j^H \boldsymbol{h}_j][\boldsymbol{h}_k^H \boldsymbol{h}_k]}} = \sum_{m=1}^{q} \mu_{m \to j} \, \mu_{m \to k}^*. \tag{8.71}$$

8.5.3 Partial Directed Coherence (PDC)

In Sect. 8.5.2, DTF is expressed using a factorization of coherence. We apply similar factorization to partial coherence to obtain the partial directed coherence (PDC) [6, 12]. The starting point is Eq. (8.14), which is re-written as,

$$\kappa_{j,k}(f) = \frac{\bar{\boldsymbol{a}}_j^H \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{a}}_k}{\sqrt{[\bar{\boldsymbol{a}}_j^H \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{a}}_j][\bar{\boldsymbol{a}}_k^H \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{a}}_k]}}. \tag{8.72}$$

Here, recall that $\bar{\boldsymbol{a}}_j$ is the jth column of the matrix $\bar{\boldsymbol{A}}(f)$ and $\bar{\boldsymbol{a}}_j = [\bar{A}_{1,j}, \ldots, \bar{A}_{q,j}]$. We define

$$\bar{\pi}_{j,k}(f) = \frac{\bar{A}_{j,k}(f)}{\sqrt{\bar{\boldsymbol{a}}_k^H \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{a}}_k}}, \tag{8.73}$$

[3] Note that since $\boldsymbol{h}_j$ is the jth column vector of $\boldsymbol{H}^H$, $\boldsymbol{h}_j$ is equal to $\boldsymbol{h}_j = [H_{j,1}^*, \ldots, H_{j,q}^*]^T$.

and a column vector $\bar{\boldsymbol{\pi}}_k(f)$ as

$$\bar{\boldsymbol{\pi}}_k(f) = [\bar{\pi}_{1,k}, \ldots, \bar{\pi}_{q,k}]^T. \tag{8.74}$$

We can then express the partial coherence as

$$\kappa_{j,k}(f) = \bar{\boldsymbol{\pi}}_j^H(f)\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{\pi}}_k(f). \tag{8.75}$$

The equation above indicates that the partial coherence $\kappa_{j,k}(f)$ is factorized, and the form of the factorization is very similar to the factorization of the coherence in Eq. (8.69).

The partial directed coherence (PDC) is defined using $\bar{\pi}_{j,k}$ in Eq. (8.73) with replacing $\boldsymbol{\Sigma}$ with $\boldsymbol{I}$ for removing the instantaneous interaction. That is, the PDC from the kth to the jth channels, $\pi_{k\to j}$, is defined as

$$\pi_{k\to j} = \frac{\bar{A}_{j,k}(f)}{\sqrt{\bar{\boldsymbol{a}}_k^H \bar{\boldsymbol{a}}_k}}. \tag{8.76}$$

Assuming $\boldsymbol{\Sigma}^{-1} = \boldsymbol{I}$, the partial coherence $\kappa_{j,k}(f)$ is factorized using PDC, such that

$$\kappa_{j,k}(f) = \frac{\sum_{m=1}^{q} \bar{A}_{m,j}^*(f)\bar{A}_{m,k}(f)}{\sqrt{[\bar{\boldsymbol{a}}_j^H \bar{\boldsymbol{a}}_j][\bar{\boldsymbol{a}}_k^H \bar{\boldsymbol{a}}_k]}} = \sum_{m=1}^{q} \pi_{j\to m}^* \pi_{k\to m}. \tag{8.77}$$

As discussed in the previous sections, the non-diagonal elements of the MVAR coefficient matrix $\bar{\boldsymbol{A}}(f)$ should contain information on causal interaction. The PDC directly makes use of this information. The DTF also uses this information but it does so in an indirect manner through the transfer matrix $\boldsymbol{H}(f)$, which is the inverse of $\bar{\boldsymbol{A}}(f)$. Although PDC is derived in a somewhat heuristic manner, it represents causality through $\bar{A}_{j,k}(f)$. The major difference between PDC and DTF is that PDC only represents the direct causal influence and is not affected by the indirect influence.

8.6 Transfer Entropy

8.6.1 Definition

Granger causality and its related measures such as DTF and PDC relies on the MVAR modeling of voxel time series. In that sense, these measures are model-based. In this section, we introduce transfer entropy [7, 13], which does not use the MVAR modeling of voxel time series. Explanations on fundamentals of entropy and mutual information are found in Sect. C.3.2 in the Appendix.

We denote two random vector time series $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$. Let us define vectors $\widetilde{\boldsymbol{x}}(t)$ and $\widetilde{\boldsymbol{y}}(t)$ as those made by concatenating their past values such that

$$\widetilde{\boldsymbol{x}}(t) = \begin{bmatrix} \boldsymbol{x}(t-1) \\ \boldsymbol{x}(t-2) \\ \vdots \\ \boldsymbol{x}(t-P) \end{bmatrix} \quad \text{and} \quad \widetilde{\boldsymbol{y}}(t) = \begin{bmatrix} \boldsymbol{y}(t-1) \\ \boldsymbol{y}(t-2) \\ \vdots \\ \boldsymbol{y}(t-P) \end{bmatrix}. \tag{8.78}$$

We then define the conditional entropy of $\boldsymbol{y}(t)$, given its past values $\widetilde{\boldsymbol{y}}(t)$, such that

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) = -\iint p(\boldsymbol{y}, \widetilde{\boldsymbol{y}}) \log p(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\widetilde{\boldsymbol{y}}. \tag{8.79}$$

Similarly, we define the conditional entropy of $\boldsymbol{y}(t)$, given the past values $\widetilde{\boldsymbol{x}}(t)$ and $\widetilde{\boldsymbol{y}}(t)$, such that

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) = -\iint p(\widetilde{\boldsymbol{x}}, \boldsymbol{y}, \widetilde{\boldsymbol{y}}) \log p(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\widetilde{\boldsymbol{x}} \, \mathrm{d}\widetilde{\boldsymbol{y}}. \tag{8.80}$$

The transfer entropy $\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}}$ is defined as

$$\begin{aligned} \mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}} &= \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) - \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) \\ &= \iint p(\widetilde{\boldsymbol{x}}, \boldsymbol{y}, \widetilde{\boldsymbol{y}}) \log \frac{\log p(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})}{\log p(\boldsymbol{y}|\widetilde{\boldsymbol{y}})} \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\widetilde{\boldsymbol{x}} \, \mathrm{d}\widetilde{\boldsymbol{y}}. \end{aligned} \tag{8.81}$$

In the equations above, the explicit time notation (t) is omitted for simplicity. In Eq. (8.81), $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}})$ represents the uncertainty on the current value of $\boldsymbol{y}$, when we know $\widetilde{\boldsymbol{y}}$, which is the past values of $\boldsymbol{y}$. Also, $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})$ represents the uncertainty on the current value of $\boldsymbol{y}$, when we know both $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{y}}$, which are the past values of $\boldsymbol{x}$ and $\boldsymbol{y}$. Therefore, the transfer entropy is equal to the reduction of uncertainty of the current value of $\boldsymbol{y}$ as a result of knowing the past values of $\boldsymbol{x}$.

8.6.2 Transfer Entropy Under Gaussianity Assumption

Assuming that the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ follow a Gaussian distribution, using Eq. (C.61), the conditional entropy $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}})$ is expressed as [14]

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) = \frac{1}{2} \log \left| \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{y}} \boldsymbol{\Sigma}_{\tilde{y}\tilde{y}}^{-1} \boldsymbol{\Sigma}_{y\tilde{y}}^{T} \right|, \tag{8.82}$$

where[4]

$$\boldsymbol{\Sigma}_{yy} = \langle \boldsymbol{y}\boldsymbol{y}^T \rangle, \tag{8.83}$$

$$\boldsymbol{\Sigma}_{y\tilde{y}} = \langle \boldsymbol{y}\widetilde{\boldsymbol{y}}^T \rangle, \tag{8.84}$$

$$\boldsymbol{\Sigma}_{\tilde{y}\tilde{y}} = \langle \widetilde{\boldsymbol{y}}\widetilde{\boldsymbol{y}}^T \rangle. \tag{8.85}$$

Similarly, the conditional entropy $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})$ is expressed as

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) = \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{z}}) = \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{z}}\boldsymbol{\Sigma}_{\tilde{z}\tilde{z}}^{-1}\boldsymbol{\Sigma}_{y\tilde{z}}^T\right|, \tag{8.86}$$

where $\widetilde{\boldsymbol{z}} = [\widetilde{\boldsymbol{y}}^T,\ \widetilde{\boldsymbol{x}}^T]^T$ and

$$\boldsymbol{\Sigma}_{y\tilde{z}} = \langle \boldsymbol{y}\widetilde{\boldsymbol{z}}^T \rangle, \tag{8.87}$$

$$\boldsymbol{\Sigma}_{\tilde{z}\tilde{z}} = \langle \widetilde{\boldsymbol{z}}\widetilde{\boldsymbol{z}}^T \rangle. \tag{8.88}$$

Thus, the transfer entropy $\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}}$ is given by

$$\begin{aligned}\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}} &= \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) - \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) \\ &= \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) - \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{z}}) = \frac{1}{2}\log\frac{\left|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{y}}\boldsymbol{\Sigma}_{\tilde{y}\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y\tilde{y}}^T\right|}{\left|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{z}}\boldsymbol{\Sigma}_{\tilde{z}\tilde{z}}^{-1}\boldsymbol{\Sigma}_{y\tilde{z}}^T\right|}.\end{aligned} \tag{8.89}$$

8.6.3 *Equivalence Between Transfer Entropy and Granger Causality*

We will show that the transfer entropy and Granger causality are equivalent under the Gaussianity assumption. The arguments here follow those in [14]. As discussed in Sect. 8.3.2, we consider the two forms of the regression to define Granger causality. In the first regression, $\boldsymbol{y}(t)$ is regressed using only its past values, such that

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{A}(p)\boldsymbol{y}(t-p) + \boldsymbol{e} = \boldsymbol{A}\widetilde{\boldsymbol{y}}(t) + \boldsymbol{e}, \tag{8.90}$$

where $\boldsymbol{A} = [\boldsymbol{A}(1), \ldots, \boldsymbol{A}(P)]$, $\widetilde{\boldsymbol{y}}(t)$ is defined in Eq. (8.78), and $\boldsymbol{e}$ is a residual vector. In the second regression, $\boldsymbol{y}(t)$ is regressed using not only its past values but also the past values of $\boldsymbol{x}(t)$, such that

[4] Note that in Sect. C.3.3 the expectation operator $E[\,\cdot\,]$ is used, instead of the averaging operator $\langle\,\cdot\,\rangle$. They have the same meaning in the arguments here.

$$\begin{aligned}\boldsymbol{y}(t) &= \sum_{p=1}^{P} \boldsymbol{A}(p)\boldsymbol{y}(t-p) + \sum_{p=1}^{P} \boldsymbol{B}(p)\boldsymbol{x}(t-p) + \boldsymbol{\epsilon} \\ &= \boldsymbol{A}\widetilde{\boldsymbol{y}}(t) + \boldsymbol{B}\widetilde{\boldsymbol{x}}(t) + \boldsymbol{\epsilon} = \boldsymbol{C}\,\widetilde{\boldsymbol{z}} + \boldsymbol{\epsilon}. \end{aligned} \tag{8.91}$$

where $\boldsymbol{B} = [\boldsymbol{B}(1), \ldots, \boldsymbol{B}(P)]$, $\boldsymbol{C} = [\boldsymbol{A},\ \boldsymbol{B}]$, and $\boldsymbol{\epsilon}$ is a residual vector.

The Granger causality from the time series $\boldsymbol{x}$ to $\boldsymbol{y}$, $\mathcal{G}_{x\to y}$ is defined as

$$\mathcal{G}_{x\to y} = \log \frac{|\boldsymbol{\Sigma}_e|}{|\boldsymbol{\Sigma}_\epsilon|}, \tag{8.92}$$

where $\boldsymbol{\Sigma}_e$ is the covariance matrix of the residual $\boldsymbol{e}$ in Eq. (8.90), and $\boldsymbol{\Sigma}_\epsilon$ is the covariance matrix of the residual $\boldsymbol{\epsilon}$ in Eq. (8.91). Using Eq. (C.57), we have

$$|\boldsymbol{\Sigma}_e| = |\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{y}}\boldsymbol{\Sigma}_{\tilde{y}\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y\tilde{y}}^{T}|, \tag{8.93}$$

and

$$|\boldsymbol{\Sigma}_\epsilon| = |\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{z}}\boldsymbol{\Sigma}_{\tilde{z}\tilde{z}}^{-1}\boldsymbol{\Sigma}_{y\tilde{z}}^{T}|. \tag{8.94}$$

Substituting Eqs. (8.93) and (8.94) into (8.92), we get

$$\mathcal{G}_{x\to y} = \log \frac{|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{y}}\boldsymbol{\Sigma}_{\tilde{y}\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y\tilde{y}}^{T}|}{|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\tilde{z}}\boldsymbol{\Sigma}_{\tilde{z}\tilde{z}}^{-1}\boldsymbol{\Sigma}_{y\tilde{z}}^{T}|}. \tag{8.95}$$

The right-hand side of the equation above is exactly equal to that of Eq. (8.89) except for the multiplicative constant $1/2$, indicating that these two measures are equivalent.

8.6.4 Computation of Transfer Entropy

In Sect. C.3.2 in the Appendix, we show that, assuming the Gaussian processes for the $r \times 1$ real random vector $\boldsymbol{x}$ and the $q \times 1$ real random vector $\boldsymbol{y}$, the mutual information between these two vectors is expressed as

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\log \frac{1}{\left|\boldsymbol{I} - \boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}^{T}\right|} = \frac{1}{2}\sum_{j=1}^{d}\log\frac{1}{1-\lambda_j}, \tag{8.96}$$

where we assume $d = \min\{q, r\}$, and λ_j is the jth eigenvalue of the matrix,

$$\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}^{T}.$$

We can derive a similar formula to compute the transfer entropy. To do so, we use the similarity between the transfer entropy and mutual information. Using Eq. (C.45), the conditional entropy $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})$ is rewritten as

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) = \mathcal{H}(\boldsymbol{y}, \widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}) - \mathcal{H}(\widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}). \tag{8.97}$$

Substituting this equation into Eq. (8.81), we get

$$\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}} = \mathcal{H}(\widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}) + \mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) - \mathcal{H}(\boldsymbol{y}, \widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}). \tag{8.98}$$

Comparing the equation above with Eq. (C.46), it can be seen that the transfer entropy is equal to the mutual information between $\boldsymbol{y}(t)$ and $\widetilde{\boldsymbol{x}}(t)$, when $\widetilde{\boldsymbol{y}}(t)$ is given.

We define $\boldsymbol{\Sigma}_{u,v|w}$ as

$$\boldsymbol{\Sigma}_{u,v|w} = \boldsymbol{\Sigma}_{uv} - \boldsymbol{\Sigma}_{uw}\boldsymbol{\Sigma}_{ww}^{-1}\boldsymbol{\Sigma}_{vw}^{T}. \tag{8.99}$$

Then, using Eq. (C.61), we can express $\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}})$, and $\mathcal{H}(\widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}})$, such that

$$\mathcal{H}(\boldsymbol{y}|\widetilde{\boldsymbol{y}}) = \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{y,y|\tilde{y}}\right|, \tag{8.100}$$

$$\mathcal{H}(\widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}) = \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{\tilde{x},\tilde{x}|\tilde{y}}\right|. \tag{8.101}$$

On the basis of Eq. (C.59), we also derive

$$\mathcal{H}(\boldsymbol{y}, \widetilde{\boldsymbol{x}}|\widetilde{\boldsymbol{y}}) = \frac{1}{2}\log\begin{vmatrix} \boldsymbol{\Sigma}_{y,y|\tilde{y}} & \boldsymbol{\Sigma}_{\tilde{x},y|\tilde{y}}^{T} \\ \boldsymbol{\Sigma}_{\tilde{x},y|\tilde{y}} & \boldsymbol{\Sigma}_{\tilde{x},\tilde{x}|\tilde{y}} \end{vmatrix}. \tag{8.102}$$

Thus, substituting the equations above into Eq. (8.98), we obtain

$$\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}} = \frac{1}{2}\log\frac{1}{\left|\boldsymbol{I} - \boldsymbol{\Sigma}_{y,y|\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y,\tilde{x}|\tilde{y}}\boldsymbol{\Sigma}_{\tilde{x},\tilde{x}|\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y,\tilde{x}|\tilde{y}}^{T}\right|}. \tag{8.103}$$

Accordingly, defining the eigenvalues of a matrix,

$$\boldsymbol{\Sigma}_{y,y|\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y,\tilde{x}|\tilde{y}}\boldsymbol{\Sigma}_{\tilde{x},\tilde{x}|\tilde{y}}^{-1}\boldsymbol{\Sigma}_{y,\tilde{x}|\tilde{y}}^{T}, \tag{8.104}$$

as χ_j $(j = 1, \ldots, d)$, the transfer entropy is given by

$$\mathcal{H}_{\boldsymbol{x}\to\boldsymbol{y}} = \frac{1}{2}\sum_{j=1}^{d}\log\frac{1}{1-\chi_j}. \tag{8.105}$$

8.7 Estimation of MVAR Coefficients

8.7.1 Least-Squares Algorithm

The Granger-causality and related measures rely on the modeling of the multivariate vector auto-regressive (MVAR) process of the source time series. Use of these measures requires estimating the MVAR coefficient matrices. This section deals with the estimation of MVAR coefficients. A $q \times 1$ random vector $\boldsymbol{y}(t)$ is modeled using the MVAR process, such that

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{A}(p)\boldsymbol{y}(t-p) + \boldsymbol{e}(t). \tag{8.106}$$

We can estimate the MVAR coefficients $A_{i,j}(p)$ where $i, j = 1, \ldots, q$ and $p = 1, \ldots, P$ based on the least-squares principle. To derive the least-squares equation, let us explicitly write the MVAR process for the ℓth component $y_\ell(t)$ as

$$\begin{aligned} y_\ell(t) = &\sum_{j=1}^{q} A_{\ell,j}(1)y_j(t-1) + \sum_{j=1}^{q} A_{\ell,j}(2)y_j(t-2) \\ &+ \cdots + \sum_{j=1}^{q} A_{\ell,j}(P)y_j(t-P) + e_\ell(t). \end{aligned} \tag{8.107}$$

We assume that the source time series are obtained at $t = 1, \ldots, K$ where $K \gg q \times P$. Then, since Eq. (8.107) holds for $t = (P+1), \ldots, K$, a total of $K - P$ linear equations are obtained by setting $t = (P+1), \ldots, K$ in Eq. (8.107).

These equations are formulated in a matrix form,

$$\boldsymbol{y}_\ell = \boldsymbol{G}\boldsymbol{x}_\ell + \boldsymbol{e}_\ell. \tag{8.108}$$

Here, the $(K - P) \times 1$ column vector $\boldsymbol{y}_\ell$ is defined as

$$\boldsymbol{y}_\ell = [y_\ell(P+1), y_\ell(P+2), \ldots, y_\ell(K)]^T. \tag{8.109}$$

In Eq. (8.108), $\boldsymbol{G}$ is a $(K - P) \times Pq$ matrix expressed as

$$\boldsymbol{G} = \begin{bmatrix} y_1(P) & \cdots & y_q(P) & \cdots & y_1(1) & \cdots & y_q(1) \\ y_1(P+1) & \cdots & y_q(P+1) & \cdots & y_1(2) & \cdots & y_q(2) \\ & & & \vdots & & & \\ y_1(K-1) & \cdots & y_q(K-1) & \cdots & y_1(K-P) & \cdots & y_q(K-P) \end{bmatrix}. \tag{8.110}$$

The column vector $\boldsymbol{x}_\ell$ is expressed as

$$\boldsymbol{x}_\ell = \left[A_{\ell,1}(1), \ldots, A_{\ell,q}(1), \ldots, A_{\ell,1}(P), \ldots, A_{\ell,q}(P)\right]^T . \tag{8.111}$$

The residual vector $\boldsymbol{e}_\ell$ is given by

$$\boldsymbol{e}_\ell = [e_\ell(P+1), \ldots, e_\ell(K)]^T . \tag{8.112}$$

Equation (8.108) is called the Yule-Walker equation. The least-squares estimate of $\boldsymbol{x}_\ell$, $\widehat{\boldsymbol{x}}_\ell$, is then obtained using,

$$\widehat{\boldsymbol{x}}_\ell = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{y}_\ell. \tag{8.113}$$

8.7.2 Sparse Bayesian (Champagne) Algorithm

Since causality analysis is generally performed using the source time series estimated from non-averaged data, the estimated time series inevitably contains a large influence of noise, which may cause errors in the MVAR-coefficient estimation. One approach to reduce such errors is to impose a sparsity constraint when estimating the MVAR coefficients. The key assumption here is that true brain interaction causes a small number of MVAR coefficients to have non-zero values, and most of the MVAR coefficients remain zero. If this is true, the sparsity constraint should be able to prevent MVAR coefficients that must be equal to zero from having erroneous non-zero values.

We can apply a simpler version of the Champagne algorithm described in Chap. 4 to this MVAR coefficient estimation. In the Champagne algorithm, the prior probability distribution of $\boldsymbol{x}_\ell$, $p(\boldsymbol{x}_\ell)$, is assumed to be Gaussian:

$$p(\boldsymbol{x}_\ell) = \mathcal{N}(\boldsymbol{x}_\ell|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}), \tag{8.114}$$

where $\boldsymbol{\Phi}$ is a diagonal precision matrix. The probability distribution of $\boldsymbol{y}_\ell$ given $\boldsymbol{x}_\ell$, $p(\boldsymbol{y}_\ell|\boldsymbol{x}_\ell)$, is also assumed to be Gaussian:

$$p(\boldsymbol{y}_\ell|\boldsymbol{x}_\ell) = \mathcal{N}(\boldsymbol{y}_\ell|\boldsymbol{G}\boldsymbol{x}_\ell, \boldsymbol{\Lambda}^{-1}), \tag{8.115}$$

where $\boldsymbol{\Lambda}$ is a diagonal noise precision matrix. Then, the posterior distribution of $\boldsymbol{x}_\ell$, $p(\boldsymbol{x}_\ell|\boldsymbol{y}_\ell)$, is shown to be Gaussian, and it is expressed as

$$p(\boldsymbol{x}_\ell|\boldsymbol{y}_\ell) = \mathcal{N}(\boldsymbol{x}_\ell|\bar{\boldsymbol{x}}_\ell, \boldsymbol{\Gamma}^{-1}). \tag{8.116}$$

Unlike the Champagne algorithm in Chap. 4, the update equations for $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ can be derived using the expectation-maximization (EM) algorithm. This is because,

since $(K - P) \gg Pq$ holds, the estimation problem is not an ill-posed problem. The EM algorithm for the Gaussian model is described in Sect. B.5 in the Appendix. According to Eqs. (B.24) and (B.25), the E-step update equations are given by

$$\boldsymbol{\Gamma} = \boldsymbol{G}^T \boldsymbol{\Lambda} \boldsymbol{G} + \boldsymbol{\Phi} \tag{8.117}$$

$$\bar{\boldsymbol{x}}_\ell = \boldsymbol{\Gamma}^{-1} \boldsymbol{G}^T \boldsymbol{\Lambda} \boldsymbol{y}_\ell \tag{8.118}$$

The parameters, $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$, are estimated in the M step of the EM algorithm. The update equation for $\boldsymbol{\Phi}$ is obtained from Eq. (B.42) with setting $K = 1$ in that equation, resulting in

$$\boldsymbol{\Phi}^{-1} = \mathrm{diag}\left[\bar{\boldsymbol{x}}_\ell \bar{\boldsymbol{x}}_\ell^T + \boldsymbol{\Gamma}^{-1}\right], \tag{8.119}$$

where diag[$\cdot$] indicates a diagonal matrix whose diagonal elements are equal to those of a matrix in the parentheses. According to Eq. (B.45), the noise precision matrix $\boldsymbol{\Lambda}$ is given by

$$\boldsymbol{\Lambda}^{-1} = \mathrm{diag}\left[\|\boldsymbol{y}_\ell - \boldsymbol{G}\bar{\boldsymbol{x}}_\ell\|^2 + \boldsymbol{G}\boldsymbol{\Gamma}^{-1}\boldsymbol{G}^T\right]. \tag{8.120}$$

The estimate of the MVAR coefficients is obtained as $\bar{\boldsymbol{x}}_\ell$, after the EM iteration is terminated. This algorithm is similar to, but considerably simpler than the one proposed in [15].

8.8 Numerical Examples

8.8.1 Experiments Using Bivariate Causal Time Series

Numerical experiments were performed to illustrate the properties of the causality measures described in this chapter. The source-space causality analysis was applied in which source time series are first estimated from simulated MEG recordings, and the MVAR coefficients are then computed using the time series at selected voxels. Here, we use a sensor alignment of the 275 whole-head MEG sensor array from Omega^{TM} (VMS Medtech, Coquitlam, Canada) neuromagnetometer. Three sources are assumed to exist on the vertical single plane of $x = 0$ cm, as in the numerical experiments in the previous chapters. The source-sensor configuration and the coordinate system are depicted in Fig. 8.1. As shown in this figure, we assume three sources and the time series of these three sources are denoted $s_1(t)$, $s_2(t)$, and $s_3(t)$. The first experiments assume a causal relationship between bivariate time series, and an information flow exists from $s_1(t)$ to $s_2(t)$. The time series $s_1(t)$ and $s_2(t)$ are generated by using the MVAR process [11]

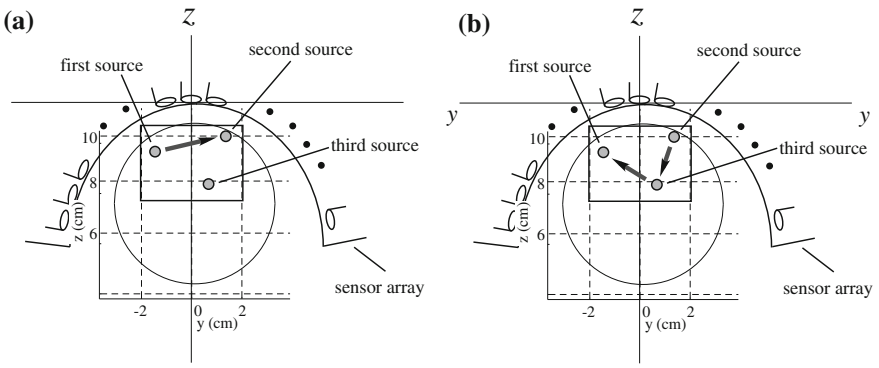

Fig. 8.1 The coordinate system and source-sensor configuration used in the numerical experiments. The plane at $x = 0$ cm is shown. The *small circles* show the locations of the three sources, and the *bold arrows* schematically show their causal relationships assumed in the experiments. **a** The first experiment with bivariate causal time series. **b** The second experiments with trivariate causal time series

$$\begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = \begin{bmatrix} 0.9 & 0 \\ 0.16 & 0.8 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \end{bmatrix} + \begin{bmatrix} -0.5 & 0 \\ -0.2 & -0.5 \end{bmatrix} \begin{bmatrix} s_1(t-2) \\ s_2(t-2) \end{bmatrix} + \boldsymbol{e}(t). \tag{8.121}$$

The time series of the third source, $s_3(t)$, was generated using Gaussian random numbers. Thus, the third source activity was independent from either the first or the second source activities. The causal relationship assumed in this experiment is depicted in Fig. 8.1a.

Then, simulated sensor recordings were computed by projecting the time series of the three sources onto the sensor space by using the sensor lead field. A small amount of simulated sensor noise was added. The Champagne source reconstruction algorithm was applied to the simulated sensor recordings. Here, three-dimensional reconstruction was performed on a region defined as $-4 \leq x \leq 4$, $-4 \leq x \leq 4$, and $6 \leq z \leq 12$ cm with a voxel interval equal to 0.5 cm. Reconstructed source time series $\widehat{s}_1(t)$, $\widehat{s}_2(t)$, and $\widehat{s}_3(t)$ were obtained as the time series at voxels nearest to the assumed source locations.

Once $\widehat{s}_1(t)$, $\widehat{s}_2(t)$ were obtained, the MVAR coefficients between these time series were estimated by using the least-squares method in Sect. 8.7.1. Using the estimated MVAR coefficients, we computed the spectral Geweke causality described in Sect. 8.4, as well as coherence. The results are shown in Fig. 8.2a.

In these results, coherence (Eq. (8.51)) can detect an interaction between the first and second source activities. The spectral Geweke causality in Eqs. (8.60) and (8.61) detects the unidirectional information flow from the first source activity to the second source activity. We also computed the partial directed coherence (PDC) and the directed transfer function (DTF), with results shown in Fig. 8.2b. The PDC and DTF

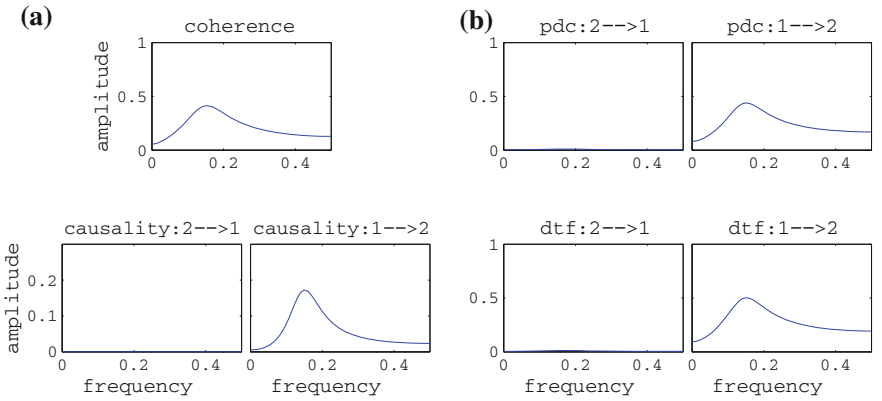

Fig. 8.2 Results of computing bivariate measures between $\widehat{s}_1(t)$ and $\widehat{s}_2(t)$ in the first experiment. **a** Results of computing spectral causality measures and the coherence. Results for the coherence are shown in the *top panel*. Results for the spectral Geweke causality are shown on the *bottom-left*, and *bottom-right* panels. **b** Results of computing the partial directed coherence (PDC) (*top panels*) and the directed transfer function (DTF) (*bottom panels*). The abscissas of plots indicate the normalized frequency. (The frequency is normalized by the sampling frequency in which the value 0.5 indicates the Nyquist frequency.) The ordinates indicate the relative amplitudes of corresponding measures

can also detect the information flow from the first to the second sources. In this two-channel experiment, the PDC and DTF provide identical results, because, aside from the scaling, both measures are equal to $|\bar{A}_{1,2}(f)|$ in a bivariate case.

The causality measures were computed between $\widehat{s}_1(t)$ and $\widehat{s}_3(t)$, and these results are shown in Fig. 8.3. Since there was no interaction between $s_1(t)$ and $s_3(t)$, all the causal measures used in the experiment, as well as the coherence, are close to zero in these results.

8.8.2 Experiments Using Trivariate Causal Time Series

Next, we conducted experiments using trivariate causal time series. In these experiments, the time series of the three sources are generated using the MVAR process introduced in [11],

$$\begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix} = \begin{bmatrix} 0.8 & 0 & 0.4 \\ 0 & 0.9 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \\ s_3(t-1) \end{bmatrix} + \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & -0.8 & 0 \\ 0 & 0 & -0.2 \end{bmatrix} \begin{bmatrix} s_1(t-2) \\ s_2(t-2) \\ s_3(t-2) \end{bmatrix} + \boldsymbol{e}(t). \tag{8.122}$$

This MVAR process represents the causal relationship depicted in Fig. 8.1b, in which the second source has a directional causal influence on the third source and the third

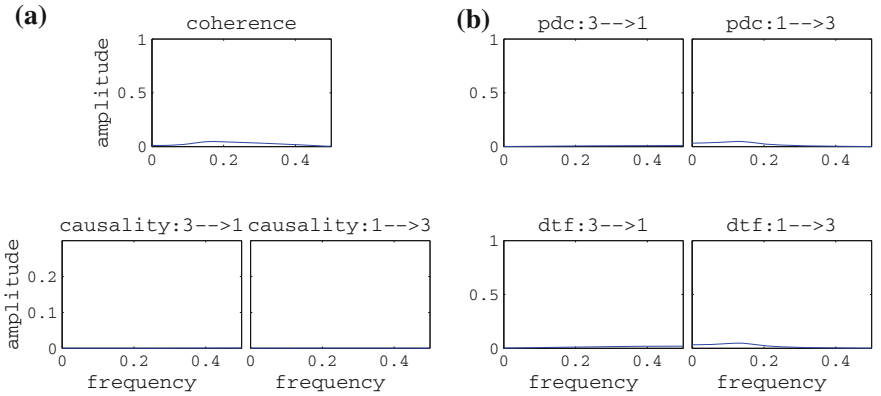

Fig. 8.3 Results of computing bivariate measures between $\widehat{s}_1(t)$ and $\widehat{s}_3(t)$ in the first experiments. **a** Results of computing spectral causality measures and the coherence. Results for the coherence are shown in the *top panel*. Results for the spectral Geweke causality are shown in the *bottom panels*. **b** Results of computing the partial directed coherence (PDC) (*top panels*) and the directed transfer function (DTF) (*bottom panels*). Explanations on the ordinate and abscissa variables are found in the caption for Fig. 8.2

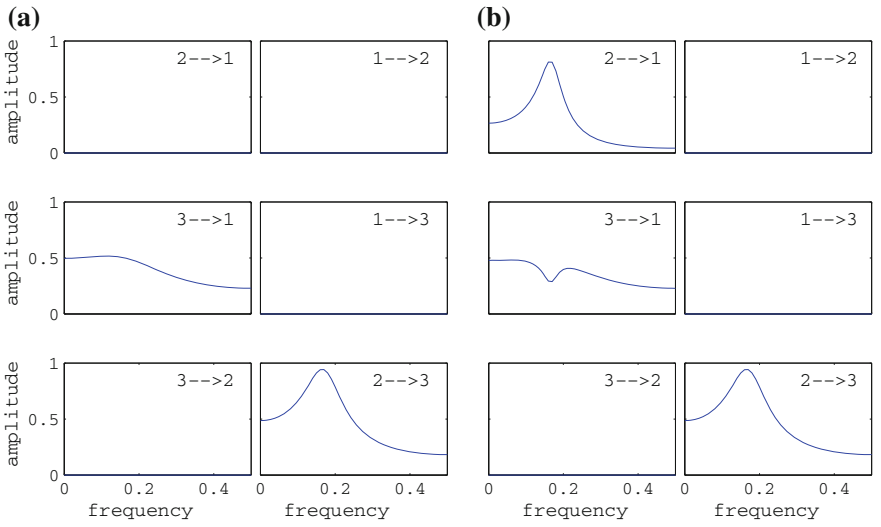

Fig. 8.4 **a** The plot of partial directed coherence (PDC) and **b** directed transfer function (DTF) computed using the model MVAR coefficients in Eq. (8.122). Explanations on the ordinate and abscissa variables are found in the caption for Fig. 8.2

source has a directional influence on the first source. The PDC and DTF computed using the model MVAR coefficients which appear in Eq. (8.122) are shown in Fig. 8.4. The results in Fig. 8.4 show the ground truth in the following experiments.

The signal part of simulated sensor recordings, $\boldsymbol{b}_s(t)$, was generated by projecting the time series of the three sources onto the sensor space using the sensor lead field. The final form of simulated MEG recordings $\boldsymbol{b}(t)$ was computed by adding spontaneous MEG to $\boldsymbol{b}_s(t)$, such that $\boldsymbol{b}(t) = \boldsymbol{b}_s(t) + \varrho \boldsymbol{b}_I(t)$, where $\boldsymbol{b}_I(t)$ is the spontaneous MEG measured using the same sensor array, and ϱ is a constant that controls the signal-to-interference ratio(SIR) of the generated sensor recordings. We first set ϱ in order for SIR to be equal to 8, and computed the simulated MEG recordings $\boldsymbol{b}(t)$. The Champagne source reconstruction algorithm was applied to $\boldsymbol{b}(t)$, and the estimated time series of the three sources, $\widehat{s}_1(t)$, $\widehat{s}_2(t)$, and $\widehat{s}_3(t)$ were obtained. The MVAR coefficients were estimated using the least-squares method. The results of computing PDC and DTF are shown in Fig. 8.5.

In Fig. 8.5, results very close to the ground truth in Fig. 8.4 were obtained. The DTF detects the information flow from the second to the first sources, which is the indirect causal coupling via $s_3(t)$, but the PDC does not detect this indirect coupling. These results are consistent with the explanation in Sect. 8.5.1.

We next generated the simulated sensor recordings with setting SIR at 2. The results of computing the PDC and DTF are shown in Fig. 8.6. Here, the MVAR coefficients were estimated using the least-squares method. The figure shows that large spurious causal relationships exist, due to the low SIR of the generated data. We then applied the sparse Bayesian algorithm described in Sect. 8.7.2 for estimating the MVAR coefficients, and computed the PDC and DTF. The results are shown in

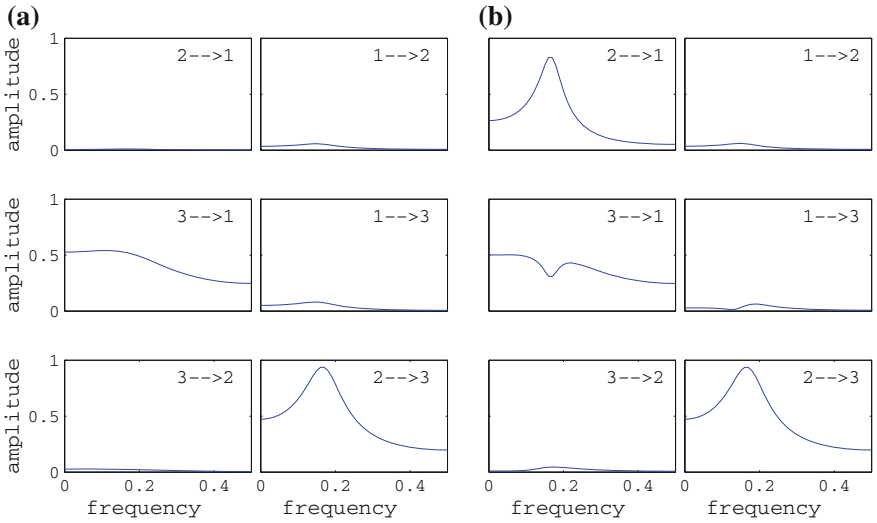

Fig. 8.5 **a** The plot of partial directed coherence (PDC) and **b** the directed transfer function (DTF) computed using MVAR coefficients estimated from the simulated MEG data with SIR equal to 8. The least-squares method in Sect. 8.7.1 was used for estimating MVAR coefficients. Explanations on the ordinate and abscissa variables are found in the caption for Fig. 8.2

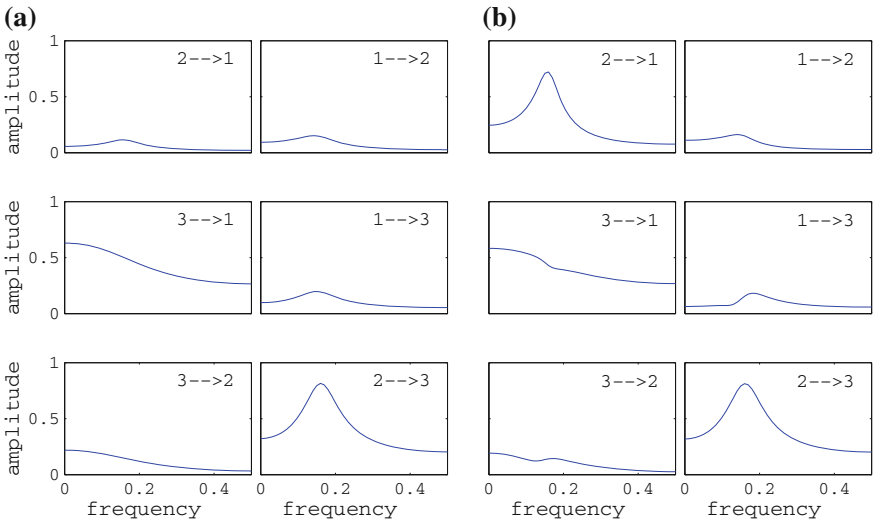

Fig. 8.6 **a** Plots of partial directed coherence (PDC) and **b** directed transfer function (DTF) computed using MVAR coefficients estimated from the simulated MEG data with SIR equal to 2. The least-squares method in Sect. 8.7.1 was used for estimating MVAR coefficients. Explanations on the ordinate and abscissa variables are found in the caption for Fig. 8.2

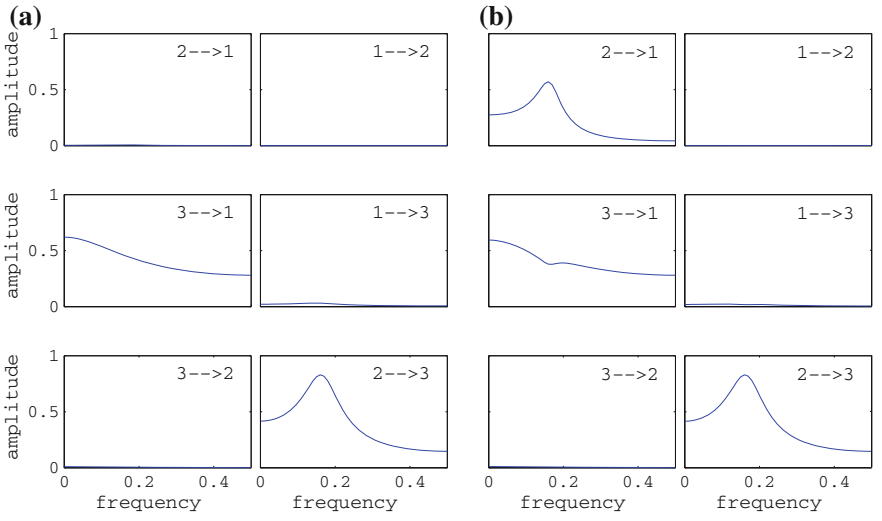

Fig. 8.7 **a** Plots of partial directed coherence (PDC) and **b** the directed transfer function (DTF) computed using MVAR coefficients estimated from the simulated MEG data with SIR equal to 2. The sparse Bayesian algorithm described in Sect. 8.7.2 was used for estimating MVAR coefficients. Explanations on the ordinate and abscissa variables are found in the caption for Fig. 8.2

Fig. 8.7. The results are very close to the ground truth in Fig. 8.4, demonstrating the effectiveness of the sparse Bayesian method in the MVAR estimation.

References

1. C.W. Granger, Investigating causal relations by econometric models and cross-spectral methods. Econom.: J. Econom. Soc. **37**(3), 424–438 (1969)
2. C. Granger, Investigating causal relations by econometric models and cross-spectral methods. Econom. Soc. Monogr. **33**, 31–47 (2001)
3. J. Geweke, Measures of conditional linear dependence and feedback between time series. J. Am. Stat. Assoc. **77**, 304–313 (1982)
4. J. Geweke, Inference and causality in economic time series models, in *Handbook of Econometrics*, vol. 2 (Elsevier, Amsterdam, 1984), pp. 1101–1144
5. M.J. Kamiński, K.J. Blinowska, A new method of the description of the information flow in the brain structure. Biol. Cybern. **65**, 203–210 (1991)
6. L.A. Baccalá, K. Sameshima, Partial directed coherence: a new concept in neural structure determination. Biol. Cybern. **84**, 463–474 (2001)
7. R. Vicente, M. Wibral, M. Lindner, G. Pipa, Transfer entropy—a model-free measure of effective connectivity for the neurosciences. J. Comput. Neurosci. **30**, 45–67 (2011)
8. K.J. Blinowska, M. Kamiński, Multivariate signal analysis by parametric models, in *Handbook of Time Series Analysis: Recent Theoretical Developments and Applications* (2006), p. 373
9. P. Franaszczuk, K. Blinowska, M. Kowalczyk, The application of parametric multichannel spectral estimates in the study of electrical brain activity. Biol. Cybern. **51**(4), 239–247 (1985)
10. C.D. Meyer, *Matrix Analysis and Applied Linear Algebra* (Society for Industrial and Applied Mathematics, Philadelphia, 2000)
11. M. Ding, Y. Chen, S.L. Bressler, Granger causality: basic theory and application to neuroscience, in *Handbook of Time Series Analysis*, ed. by B. Schelter et al. (Wiley-VCH, Weinheim, 2006), pp. 500–600
12. K. Sameshima, L.A. Baccalá, Using partial directed coherence to describe neuronal ensemble interactions. J. Neurosci. Methods **94**(1), 93–103 (1999)
13. T. Schreiber, Measuring information transfer. Phys. Rev. Lett. **85**, 461–464 (2000)
14. L. Barnett, A.B. Barrett, A.K. Seth, Granger causality and transfer entropy are equivalent for Gaussian variables. Phys. Rev. Lett. **103**, 238701 (2009)
15. W.D. Penny, S.J. Roberts, Bayesian multivariate autoregressive models with structured priors. IEE Proc. **149**, 33–41 (2002)

Chapter 9
Detection of Phase–Amplitude Coupling in MEG Source Space: An Empirical Study

9.1 Introduction

Neural oscillations across multiple frequency bands have consistently been observed in EEG and MEG recordings. The alpha rhythm (8–13 Hz), the best known brain oscillation, is observed throughout the brain. Other well-known oscillations include the delta (1–3 Hz), theta (4–7 Hz), beta (13–30 Hz), and gamma (>30 Hz) bands. Gamma-band oscillations are considered to be those most closely associated with firing of cortical neurons [1].

Recently, neural substrates responsible for the genesis of such brain rhythms and their functional role in brain information processing have been the subject of intense investigations. As a result, a number of hypotheses for explaining the role of neural oscillations have emerged, some including the concept that sensory input information is coded using multiple temporal scales [2–6]. One of these hypotheses claims that temporal coding through regulation of oscillators' firing is essential for the effective exchange of information in the mammalian brain [7]. Another hypothesis claims that temporal coding organized by coupled alpha and gamma oscillations prioritizes visual processing [8].

A recent study [9] suggests that information gating via selective attention is closely related to temporal encoding. Another study suggests that temporal encoding combined with the encoding by the phase of firing is expected to carry more information than temporal encoding by spike rates or spike patterns [10]. Therefore, temporal encoding due to phase of firing is considered to be one of the most plausible explanations behind mechanisms of information processing in the brain.

On the other hand, if the brain uses temporal encoding based on the phase of firing, there should be an oscillator that regulates neural firing based on its phase. Therefore, the hypothesis for temporal encoding based on phase of firing suggests a possible coupling between oscillators with different frequencies. That is, the amplitude

---

The authors of this chapter are Eiichi Okumura, Ph.D. and Takashi Asakawa, Ph.D. who are with Neuroscience Project Office, Innovation Headquaters, Yokogawa Electric Corporation.

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9_9

of the high-frequency oscillator increases at a certain phase of the low-frequency oscillator. Such relationship between oscillators with different frequencies is called cross-frequency coupling [11], which is the main topic of this chapter.

The aim of this chapter is to demonstrate that MEG source space analysis is able to detect such cross-frequency interactions. It is a widely accepted notion that low-frequency oscillations, such as alpha, are "encoders" of information processing, and high-frequency oscillations, such as gamma, represent particular neuronal activities. Therefore, analysis of cross-frequency interactions between different cortical regions would be extremely important for the investigation of information processing mechanisms in a human brain.

9.2 Types of Cross-Frequency Coupling

Connections between brain oscillators at the same frequency, termed as same-frequency coupling (SFC) in this chapter, have been investigated in various functional connectivity studies. In such studies, the coherence (or phase-locking value) between target oscillations is commonly used as a measure of connectivity.[1] In contrast to SFC, the cross-frequency coupling (CFC) is a phenomenon in which oscillations with different frequencies interact with each other. In general, CFC is classified into the following coupling types between the oscillators:

- Phase–amplitude coupling (PAC): The phase of a low-frequency (LF) oscillation drives the amplitude of a high-frequency (HF) oscillation with the highest amplitude of the HF oscillation occurring at a specific phase of the LF oscillation, which is referred to as the "preferred coupling phase".
- Amplitude–amplitude coupling (AAC): The amplitude of an oscillation correlates the amplitude of other oscillations. Even if the oscillators are not directly coupled, their amplitude can be co-modulated.
- Phase–phase coupling (PPC): The phase of an LF oscillation is related to the phase of an HF oscillation. This is sometimes referred to as "n:m phase locking" where phase locking occurs between n cycles of the HF oscillation and m cycles of the LF oscillation.

Figure 9.1 depicts typical time courses of oscillations when a coupling exists. When a pair of oscillations is interacting with each other, the interaction causes a constant phase relationship between the two HF oscillations, resulting in same-frequency coupling. This situation is depicted in Fig. 9.1a. When phase–amplitude coupling exists between two oscillators, the phase of the LF oscillation regulates the amplitude of the HF oscillation. In this case, the HF oscillation increases in amplitude at a certain phase of the LF oscillation. Such a situation is depicted in Fig. 9.1b. One special case of the PAC is that, when a common LF oscillation simultaneously drives the amplitudes of two HF oscillators, amplitude–amplitude coupling (AAC) between

[1] Detection of SFC in the MEG source space is the scope of Chap. 7.

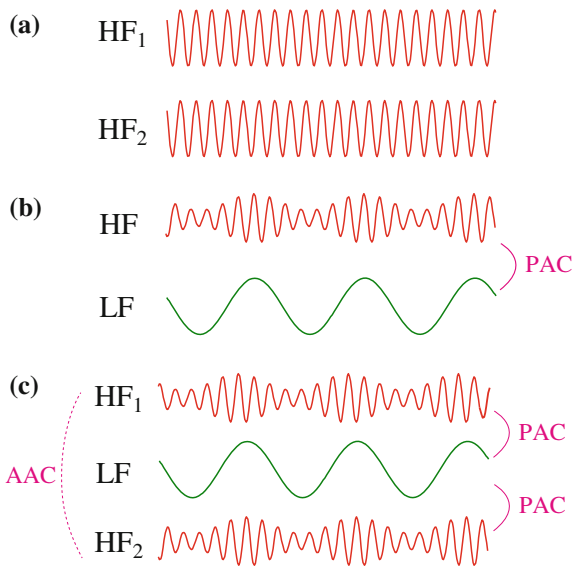

Fig. 9.1 **a** Typical time courses of two oscillators when same-frequency coupling (SFC) exists. **b** Typical time courses of two oscillators when phase–amplitude coupling (PAC) exists. **c** Typical time courses of two oscillators when PAC exists and a common lower frequency (LF) component simultaneously drives the amplitudes of two higher frequency (HF) oscillators. In this case, an amplitude–amplitude coupling (AAC) between the two HF oscillations exists even when they are not directly coupled

the two oscillators exists. Namely, AAC can exist, even when two HF oscillators are not directly coupled. This situation is depicted in Fig. 9.1c.

This chapter is based on the hypothesis that, in brain information processing, the phases of the low-frequency oscillations encode the amplitudes of the high-frequency oscillations that represent particular neuronal activities. If this hypothesis is true, analyzing the relationship between the phases of the LF oscillations and the amplitudes of the HF oscillations, i.e., analyzing the phase–amplitude coupling (PAC) between the LF and HF signals is extremely important in investigating brain information processing. Therefore, we focus on the PAC and related topics in this chapter.

9.3 Local PAC and Cross-Location PAC

When phase–amplitude coupling is observed between the HF and LF signals measured at the same location in a brain, this PAC is called the local PAC, and termed lPAC. On the other hand, when the PAC is observed between the HF and LF signals measured at different locations, this PAC is called the cross-location PAC, and here termed xPAC.

So far, most studies have investigated lPAC, but some recent studies have extended their scope to xPAC. For example, in a study using an electrocorticogram (ECoG) recording, the presence of xPACs during working memory tasks was reported [12]. The xPAC between cortical gamma oscillations and an LF oscillation from the

thalamus was reported in [13, 14]. We also show, in Sect. 9.5, that lPAC and xPAC can be detected from a healthy subject using hand-motor MEG data.

9.4 Quantification of Phase–Amplitude Coupling

9.4.1 Instantaneous Amplitude and Phase

In the PAC analysis, it is essential to calculate the instantaneous amplitude of the high-frequency signal and the instantaneous phase of the low-frequency signal. The procedure to obtain these quantities in the source space analysis is summarized as follows. Let us define the voxel time course as $x(t)$. The voxel time course is band-pass filtered into specific HF and LF bands of interest, resulting in the LF signal $x_L(t)$ and the HF signal $x_H(t)$.

The Hilbert transform is then applied to extract their instantaneous amplitudes and phases, such that

$$\mathcal{A}[x_L(t)] = A_L(t)\exp[i\theta_L(t)], \tag{9.1}$$

$$\mathcal{A}[x_H(t)] = A_H(t)\exp[i\theta_H(t)], \tag{9.2}$$

where $\mathcal{A}[\cdot]$ indicates an operator that converts a real-valued signal into its analytic signal. In the equations above, $A_L(t)$ and $A_H(t)$ are the instantaneous amplitudes of the LF and HF signals, and $\theta_L(t)$ and $\theta_H(t)$ are the instantaneous phases of the LF and HF signals. Once amplitudes and phases of the LF and HF signals are obtained, we proceed with the evaluation of the strength of the PAC using measures described in the following subsections.

9.4.2 Amplitude–Phase Diagram

To evaluate the strength of the PAC, it is necessary to derive an empirical distribution of the HF signal amplitude, $A_H(t)$, with respect to the phase of the LF signal, $\theta_L(t)$. To obtain such an empirical distribution, we observe $A_H(t)$ for a time window containing several cycles of the LF signal, and the amplitude values $A_H(t)$ are classified according to the phase of the LF signal. Let us divide the phase value into total q intervals such as $\Delta_1, \Delta_2, \ldots, \Delta_q$, where $\Delta_j = [2\pi\frac{j-1}{q}, 2\pi\frac{j}{q}]$. There are multiple (and usually many) values of $A_H(t)$ obtained when $\theta_L(t)$ has a value within Δ_j. The mean of such values of $A_H(t)$ is computed, and it is denoted Ψ_j, i.e.,

$$\Psi_j = \langle A_H(t)\rangle_{\theta_L(t)\in\Delta_j}, \tag{9.3}$$

where $\langle \cdot \rangle_{\theta_L(t) \in \Delta_j}$ indicates taking the mean of amplitude values obtained when $\theta_L(t)$ belongs to Δ_j. The mean amplitude Ψ_j is computed for all the phase bins $\Delta_1, \ldots, \Delta_q$, and the resultant set of $\Psi_1, \ldots, \Psi_q$ represents the distribution of the mean HF signal amplitude with respect to the phase of the LF signal.

The plot of these mean amplitudes Ψ_j with respect to the phase bins is called the amplitude–phase diagram, which directly expresses the dependence of the HF signal amplitude on the phase of the LF signal. Therefore, if no PAC occurs, $A_H(t)$ does not depend on $\theta_L(t)$. Thus, the value of Ψ_j is independent from the index j, resulting in a uniform amplitude–phase diagram. On the contrary, if a PAC occurs, the amplitude of the HF signal becomes stronger (or weaker) at specific phase values of the LF signal, and the amplitude–phase diagram deviates from the uniform distribution. We show several examples of the amplitude–phase diagram in Sect. 9.5.

9.4.3 Modulation Index (MI)

The modulation index [15] is a measure that quantizes the deviation of the amplitude–phase diagram from a uniform distribution. To compute the modulation index, we first normalize the mean amplitude values, $\Psi_1, \ldots, \Psi_q$, by their total sum. That is, the normalized mean amplitude value for the jth phase bin, expressed as $p(j)$, is obtained as

$$p(j) = \frac{\Psi_j}{\sum_{i=1}^{q} \Psi_i}. \tag{9.4}$$

This $p(j)$ is the normalized version of the amplitude–phase diagram, which can be interpreted as the empirically derived probability distribution of the occurrence of $A_H(t)$ obtained when $\theta_L(t)$ belongs to Δ_j. If no PAC occurs, there is no specific relationship between $A_H(t)$ and $\theta_L(t)$, resulting in $p(j)$ having a uniform value. Thus, the difference of the empirical distribution $p(j)$ from the uniform distribution can be a measure of the strength of PAC. The modulation index, $\mathcal{M}_I$, employs the Kullback–Leibler distance to assess this difference, and is defined as

$$\mathcal{M}_I = \mathcal{K}[p(j) \| u(j)] = \sum_{j=1}^{q} p(j) \log \left[\frac{p(j)}{u(j)} \right], \tag{9.5}$$

where $u(j)$ is the uniform distribution, which is $u(j) = 1/q$ $(j = 1, \ldots, q)$.

The time variation of the modulation index expresses temporal dynamics of the PAC, which represents the changes in the functional states of a brain. To compute the modulation index time variation, the source time course is divided into multiple time windows, and a value of the modulation index is obtained from each time window. Assuming that a total of K windows are used, we obtain a series of modulation index values, $\mathcal{M}_I(t_1), \mathcal{M}_I(t_2), \ldots, \mathcal{M}_I(t_K)$, which expresses the temporal change of the PAC strength.

9.4.4 Phase-Informed Time-Frequency Map

A "phase-informed" time-frequency map is the time-frequency map of the HF signal $x_H(t)$ displayed with the overlay of the sinusoidal time course $\cos(\theta_L(t))$. By displaying the time-frequency map of $x_H(t)$ in this manner, the relationship between the amplitude of the HF signal and the phase of the LF signal can be visualized in an intuitive manner. Particularly, the preference of the HF signal amplitude to the phase of the LF signal is clearly shown. This type of map was first introduced by Canolty et al. [11] and it is often called as the Canolty map. Examples of the phase-informed TF map obtained using hand-motor MEG data are shown in Sect. 9.5.

9.5 Source Space PAC Analysis: An Example Study Using Hand-Motor MEG Data

9.5.1 Experimental Design and Recordings

Here, we present results of an example study to demonstrate the noninvasive detection of PAC using the MEG source space analysis. We used a hand-motor task similar to the task used in [16]. A healthy, right-handed subject (30 years old male) participated in the MEG study.[2] The subject was asked to perform grasping movements (grasping a soft toy) with his left hand every 10 s. An auditory cue was provided to inform the grasp timing. The subject repeated grasping 64 times with his eyes open.

MEG data was acquired using a whole-head 160-channel MEG system (PQA160C, Yokogawa Electric Corporation, Japan) with a sample rate of 2,000 Hz. A band-pass filter with a bandwidth of 0.16–500 Hz was applied. Electromyogram was recorded from the subject's left forearm to monitor subject's hand movements. During the recordings, the subject lay in the supine position on a bed in a magnetically shielded room. The subject's T1–weighted MRI was obtained, which consists of 166 slices of 1.2 mm thickness. A single-slice image has 512×512 voxels in a field of view of 261×261 mm.

9.5.2 Data Analysis

9.5.2.1 Preprocessing

An ICA-based interference removal procedure was applied to remove eyeblink and MCG artifacts. An offline digital notch filter with the frequency of 60, 120, and 180 Hz

[2] Informed consent was obtained from the subject before the experiment. Handedness was determined using the Edinburgh Handedness Inventory [17].

was applied to remove the power line interference. Continuous MEG recordings were converted into multiple trials. A single trial had a time period between −6 and +6 s allowing an overlap of about 2 s to its adjacent trials. Here, the time origin was defined as the time of starting the initial rise in the electromyogram waveform. We excluded trials with the magnetic field strength exceeding 2.5 pT in any sensor channels.

9.5.2.2 Source Estimation

The watershed algorithm in FreeSurfer [18] was applied to the subject's MRI to generate triangulations on the inner skull surface. The dense triangulation on the folded cortical surface was divided into a grid of 15,000 voxels. The source space includes both hemispheres. The average distance between voxels is 2 mm. The sensor lead field was computed using the boundary element model provided by OpenMEEG [19]. The source estimation was performed using the depth-weighted minimum L2 norm method implemented in Brainstorm [20]. The anatomical labels were identified automatically by using the FreeSurfer software with the Desikan–Killiany atlas [21]. Since in our study, we focused only on the contralateral sensorimotor area, we defined the region of interest as one containing subareas such as precentral, postcentral, paracentral, caudal media frontal areas, and a part of the superior frontal area. The total number of voxels in the ROI resulted in 1,258. We computed a single time course at each voxel using the procedure described in Sect. 7.2 in Chap. 7. Since the estimated source orientation has a 180° ambiguity, we use either the estimated orientation or its opposite orientation that is closer to the direction of the cortical surface provided by Freesurfer.

9.5.2.3 Modulation Index Analysis

The LF and HF signals are extracted from voxel time courses by applying a finite impulse response band-pass filter. The frequency band of the LF signal ranges from 8.5 to 14.5 Hz, which was divided into 11 sub-bands with a bandwidth of 1 Hz; each sub-band half-overlapped with adjacent sub-bands. The frequency band of the HF signal ranges from 50 to 200 Hz; the band is divided into five half-overlapped sub-bands with the bandwidth of 50 Hz. We computed the modulation index described in Sect. 9.4.3 using all the combinations of the LF and HF sub-bands. The number of the phase bins, q, was set to 12.

When computing the time course of the modulation index, we divided a 12-second-long interval of trial data into 21 time windows in which the window width is 2 s with a 1.5 s overlap. The modulation index $\mathcal{M}_I$ was computed using all 21 sets of the time-windowed data to obtain the time course, $\mathcal{M}_I(t_1), \ldots, \mathcal{M}_I(t_{21})$, where $t_1, \ldots, t_{21}$ indicates the midpoints of the 21 time windows. We also computed the maximum of the modulation index, such that

$$\mathcal{M}_P = \max\left(\mathcal{M}_I(t_k)\right) \tag{9.6}$$

This $\mathcal{M}_P$ is used as a measure of the strength of the PAC. The surrogate-data method, similar to the one described in Sect. 7.8, is applied to assess the statistical significance of $\mathcal{M}_P$.

9.5.3 Results of Local PAC Analysis

9.5.3.1 Time Course Estimation and Spatial Mapping of the Modulation Index

We first analyzed local PAC (lPAC), and computed the modulation index time courses for all the sub-bands of the LF signal at all voxel locations. The bandwidth of the HF signal was fixed to 75 ± 25 Hz. The resultant time courses were classified using K-means clustering analysis, which thereby found three types of clusters. The first kind of the cluster, called type I, consists of the time courses that peaked before the onset of the hand movements. The time course averaged across all the type I time courses is shown in Fig. 9.2a. We computed $\mathcal{M}_P$ in Eq. (9.6) using the type I time courses, and assessed the statistical significance of $\mathcal{M}_P$ at all voxel locations. Voxels containing statistically significant lPAC activity were found with the LF signal of 12.5 ± 0.5 Hz near the primary motor area. These voxels are shown in Fig. 9.2b.

The second kind of the modulation index time courses, called type II, peaked immediately before or during the hand movements. The mean time course of this type is shown in Fig. 9.2c. Voxels containing statistically significant $\mathcal{M}_P$ (computed using the type II time courses) were found with the LF signal of 10.5 ± 0.5 Hz near the lateral part of the precentral area, as shown in Fig. 9.2d. The third kind of time courses, type III, peaked after the hand movements. The mean time course is shown in Fig. 9.2e. Voxels containing significant $\mathcal{M}_P$ (computed using the type III time courses) were found with the LF signal of 12.5 ± 0.5 Hz near the postcentral or central sulcus areas. These voxels are shown in Fig. 9.2f.

In summary, we found significant lPAC activities near the contralateral precentral area before the execution. However, following the execution, lPAC activities were found near the postcentral area and central sulcus. These findings may indicate that a certain relationship exists between these three types of lPAC activities, and the preparation and rebound of the motor and sensory systems.

9.5.3.2 Amplitude–Phase Diagram and Phase-Informed TF Map

An amplitude–phase diagram was derived using a voxel time course in the area labeled as Region A in Fig. 9.3a. Note that Region A shows significant lPAC activity showing the type I time course. In this derivation, the LF signal was set to 12.5 ± 0.5 Hz, and the HF signal to 75 ± 25 Hz. The data in the time window from -4.5 to -2.5 s (when the type-I lPAC was nearly maximum) was used. The resultant amplitude–phase diagram is shown in Fig. 9.3b in which each bar indicates the mean

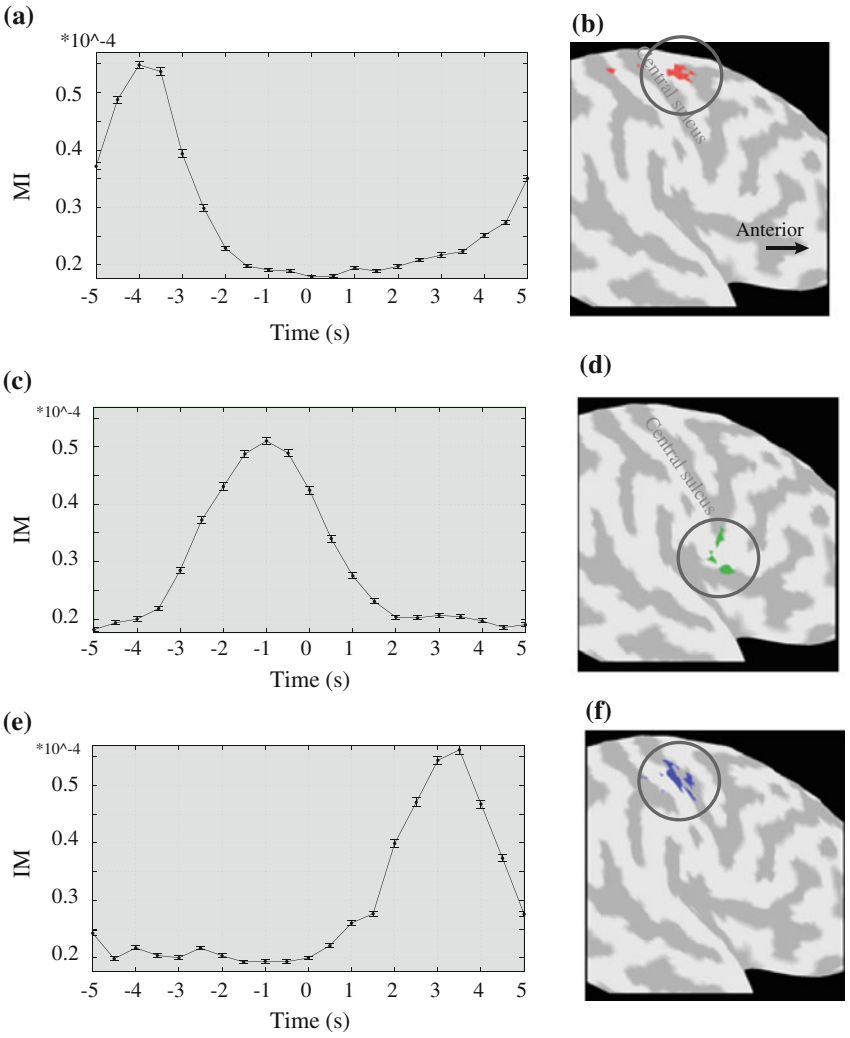

Fig. 9.2 **a** Modulation index time course of the type-I lPAC activity. **b** Voxels containing statistically significant lPAC activity having the type I time course. **c** Modulation index time course of the type II lPAC activity. **d** Voxels containing statistically significant lPAC activity having the type II time course. **e** Modulation index time course of the type III lPAC activity. **f** Voxels containing statistically significant lPAC activity having the type III time course. The significant activities in (**b**) and (**f**) were found when the LF signal of 12.5 ± 0.5 Hz and the HF signal of 75 ± 25 Hz were used. The significant activities in (**d**) were found when the LF signal of 10.5 ± 0.5 Hz and the HF signal of 75 ± 25 Hz were used. In (**a**), (**c**), and (**e**), the average time course is shown and the *error bar* indicates the range of ±standard deviation. Here, the time origin is equal to the onset of the hand movement. In (**b**), (**d**), and (**f**), the *blank circles* are used to indicate the locations of the voxels with significant activity

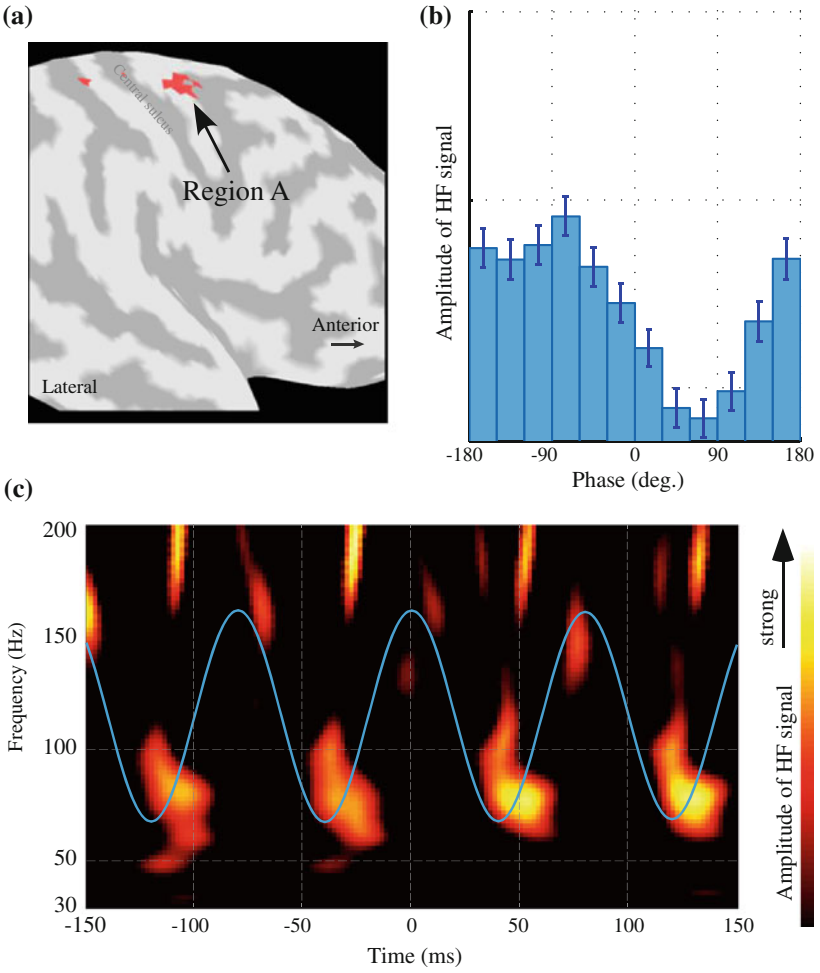

Fig. 9.3 Results of local PAC (lPAC) analysis. **a** Location of Region A where a significant type I lPAC activity was observed. The lPAC activity at Region A was analyzed. **b** Amplitude–phase diagram. Each *bar* indicates the mean amplitude value at that phase bin. The *error bar* indicates the ±standard deviation. In this analysis, the LF signal was set to 12.5 ± 0.5 Hz, and the HF signal was set to 75 ± 25 Hz. The data in the time window from -4.5 to -2.5 s was analyzed. **c** The phase-informed TF map. The amplitude of the HF signal is *color-coded* and displayed according to the *color bar*. The sinusoidal plot shows $\cos(\theta_L(t))$ where $\theta_L(t)$ is the instantaneous phase of the LF signal. Only the portion between ±150 ms from the LF peaks in the time window is displayed

amplitude value Ψ_j at that phase bin. It can be seen that the distribution of Ψ_j significantly deviates from the uniform distribution, indicating that a strong lPAC activity occurs. The preferred phase of the HF signal can be observed. That is, the amplitude of the HF signal increases significantly at a specific phase of the LF signal around $-150°$, and decreases around $+60°$.

The relationship between the HF signal amplitude and the LF signal phase can be seen more clearly in the phase-informed TF map in Fig. 9.3c. Here, the same voxel data from Region A has been used. The phase-informed TF map intuitively shows the preference of the HF signal amplitude to a specific phase of the LF signal. The map clearly indicates that the amplitude of the HF signal from 60 to 100 Hz significantly increases near $-150°$ of the phase of the LF signal.

9.5.3.3 LPAC Temporal Trend and Comparison to Event-Related Power Changes

Event-related power changes such as desynchronization (ERD) and synchronization (ERS) are representative aspects of brain activities. Comparison between PAC temporal trends and ERD/ERS should provide useful insights into brain information processing. Using the lPAC signal from Region A, we computed modulation index time courses with all 11 sub-bands of the LF signal. The HF signal was fixed to the frequency band of 75 ± 25 Hz. The results are shown in Fig. 9.4a. In this figure, the sizes of the circles represent the strength of the modulation index, and their colors represent the LF sub-bands.

The results in Fig. 9.4a show that LF signals at two sub-bands, 10 ± 0.5 and 12.5 ± 0.5 Hz, have significantly large modulation index values. The lPAC with LF signal of 12.5 ± 0.5 Hz reaches its maximum between -5 and -3 s but the lPAC with the LF signal of 10 ± 0.5 Hz reaches its maximum between -0.5 and $+0.5$ s. This observation suggests the presence of multiple brain rhythm scales underlying motor planning and execution.

A time-frequency representation of the broadband MEG signal from Region A is shown in Fig. 9.4b. In this figure, we can observe that the ERD whose center frequency is approximately equal to 12.5 Hz starts near -3 s. On the other hand, Fig. 9.4a shows that lPAC with LF of 12.5 ± 0.5 Hz starts decreasing around -3 s. Comparison between Fig. 9.4a, b shows that a strong 12.5 Hz ERS starts at $+3.5$ s at which the lPAC with the LF signal of 12.5 ± 0.5 Hz restarts. On the other hand, a stable lPAC activity with the LF band between 9 and 11 Hz (marked by the dotted circle in Fig. 9.4a) is observed at the hand-movement period when a significantly strong ERD is observed at the same frequency band near 10 Hz. Such coincidence between ERD/ERS activities and lPAC time courses suggests that lPAC and ERD/ERS activities are interrelated.

9.5.4 Results of Analyzing Cross-Location PACs

Finally, the cross-location PAC (xPAC) was analyzed using the same hand-motor MEG data. In this analysis, the LF signal is measured in voxels of the precentral area; these voxels are labeled Voxel B in Fig. 9.5a. The HF signal is measured at the voxel near the primary motor area; the voxel is labeled Voxel A in Fig. 9.5a. We first

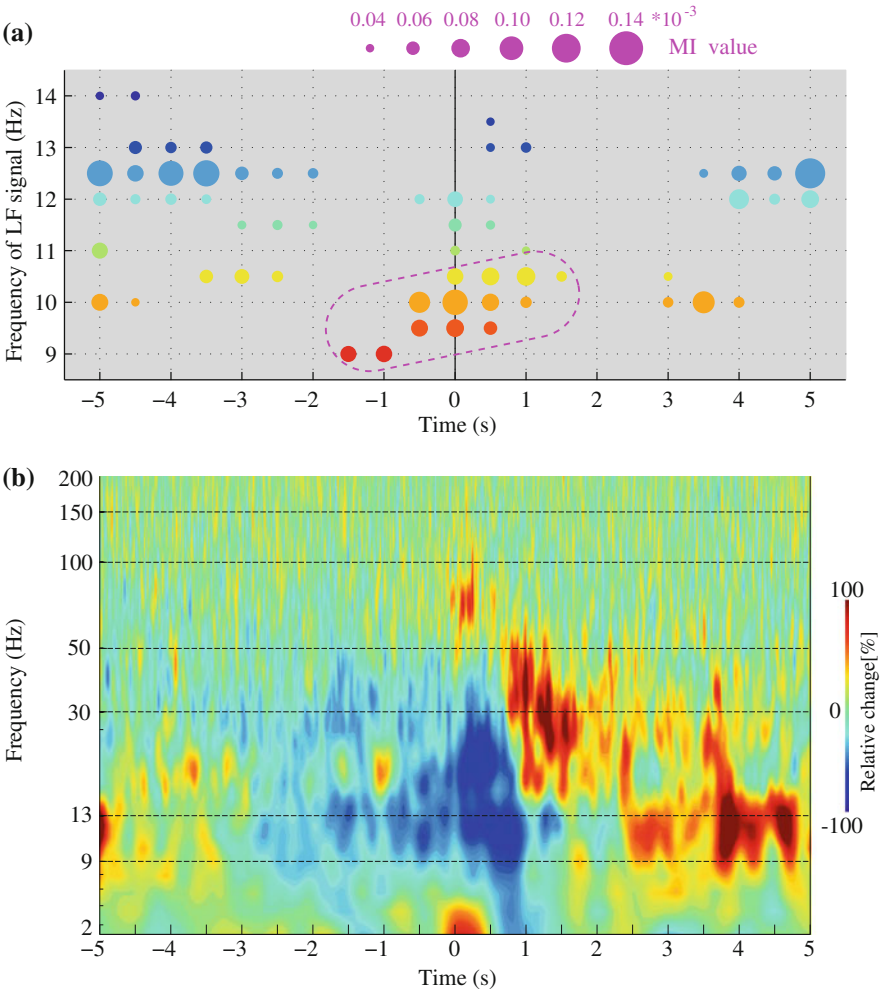

Fig. 9.4 **a** Modulation index (MI) time courses for the 11 sub-bands of the LF signal. The lPAC activity at Region A was analyzed, and the HF signal is fixed to 75 ± 25 Hz. *Sizes* of the *circles* represent the values of the modulation index, and their *colors* represent the LF sub-bands. A *dotted circle* indicates an lPAC activity with the LF band near 10 Hz observed when a significantly strong ERD is observed near 10 Hz. **b** The time-frequency representation of the (broadband) voxel time courses at Region A. The TF representation shows relative power change from the baseline, which is *color-coded* and displayed according the *color bar*. The time origin is defined as the onset of hand grasping

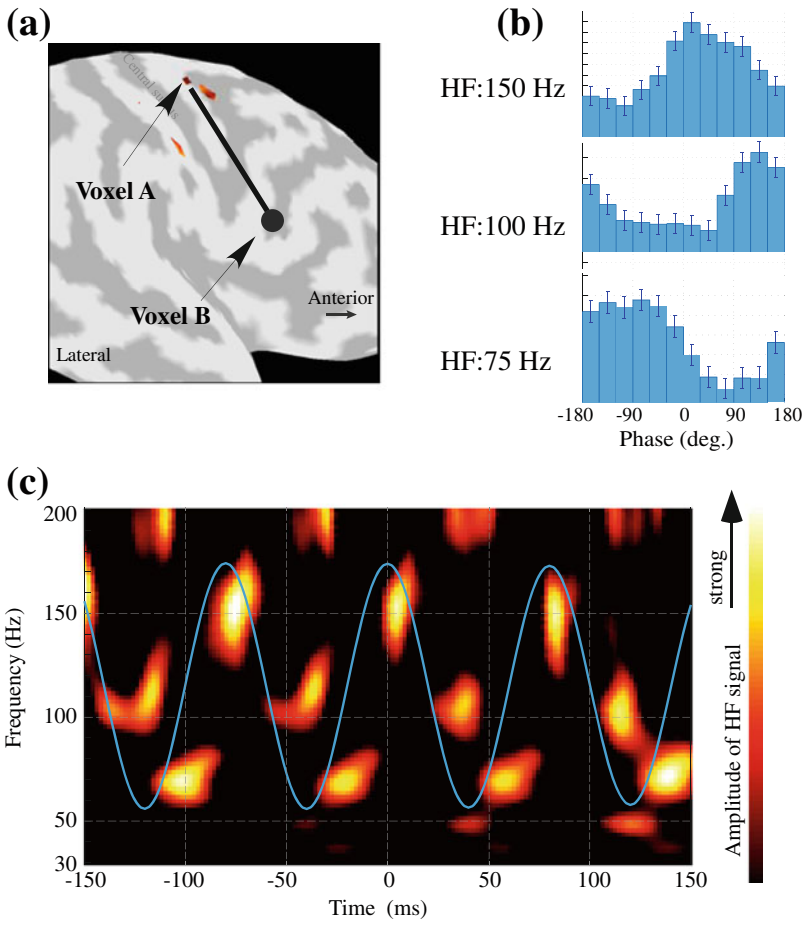

Fig. 9.5 Results of cross-location PAC (xPAC) analysis. **a** Voxel locations for measuring the LF and HF signals. The LF signal is measured at Voxel B and the HF signal is measured at Voxel A. **b** The amplitude–phase diagram for three HF sub-bands: 75 ± 25 Hz (*top panel*), 100 ± 25 Hz (*middle panel*), and 150 ± 25 Hz (*bottom panel*). Each *bar* indicates the mean amplitude value at that phase bin. The *error bar* indicates the ±standard deviation. The data in the time window from -4.5 to -2.5 s was analyzed and the frequency of the LF signal was fixed at 12.5 ± 0.5 Hz. **c** The phase-informed TF map. The amplitude of the HF signal is *color-coded* and displayed according to the *color bar*. The sinusoidal plot shows $\cos(\theta_L(t))$ where $\theta_L(t)$ is the instantaneous phase of the LF signal. Only the portion between ±150 ms from the LF peaks in the time window is displayed

computed the amplitude–phase diagrams using three sub-bands of the HF signal: 75 ± 25, 100 ± 25, and 150 ± 25 Hz. The data in the time window of -4 ± 1 s was used, and the frequency band of the LF signal was fixed at 12.5±0.5 Hz. The results are shown in Fig. 9.5b.

In this figure, significant increases in HF signal amplitudes were observed at specific phases of the LF signal. For the HF band of 75 ± 25 Hz, the amplitude of the HF signal increases significantly near $-105°$ of the LF phase. For the HF band of the 100 ± 25 Hz, the amplitude of the HF signal increases significantly near 135° of the LF phase. For the HF band of the 150 ± 25 Hz, the amplitude of the HF signal reaches its maximum near 15° of the LF phase.

Figure 9.5c shows the phase-informed TF map in which three HF components are observed near 75,100, and 150 Hz. These HF components have different preferred phases for the LF signal, indicating that the three HF components have different timing dependencies on the same LF signal. Such LF phase dependence of the HF signal amplitude across locations suggests the possibility of dynamic multiplexing neuronal codes that exploit timing differences across regions due to synaptic or propagation delays.

9.6 Summary

We have shown that the phase–amplitude coupling (PAC), both local PAC and cross-location PAC, can successfully be detected in MEG source space analysis. PAC is considered to reflect temporal coding of a brain. The measurement of PAC could therefore be a promising tool for monitoring information processing in the human brain. Specifically, PAC analysis can provide coupling-related information, including the frequency of brain rhythms, and the specific phases of LF components, which are preferred by the HF components. These preferred phases are considered to reflect the timing of information exchange. We believe that exploring xPAC is one of the most promising approaches to reveal the mechanisms of brain information processing. Our studies suggest that MEG source space analysis could be a powerful tool in xPAC studies, because of MEG's wide coverage of the brain and its noninvasiveness.

References

1. E. Privman, R. Malach, Y. Yeshurun, Modeling the electrical field created by mass neural activity. Neural Netw. **40**, 44–51 (2013)
2. A.K. Engel, P. Roelfsema, P. Fries, M. Brecht, W. Singer, Role of the temporal domain for response selection and perceptual binding. Cereb. Cortex **7**(6), 571–582 (1997)
3. A.K. Engel, P. Fries, P. König, M. Brecht, W. Singer, Temporal binding, binocular rivalry, and consciousness. Conscious. Cogn. **8**(2), 128–151 (1999)
4. P. Lakatos, G. Karmos, A.D. Mehta, I. Ulbert, C.E. Schroeder, Entrainment of neuronal oscillations as a mechanism of attentional selection. Science **320**(5872), 110–113 (2008)
5. S. Panzeri, N. Brunel, N.K. Logothetis, C. Kayser, Sensory neural codes using multiplexed temporal scales. Trends Neurosci. **33**(3), 111–120 (2010)
6. T. Akam, D.M. Kullmann, Oscillatory multiplexing of population codes for selective communication in the mammalian brain. Nat. Rev. Neurosci. **15**(2), 111–122 (2014)
7. G. Buzsaki, *Rhythms of the Brain* (Oxford University Press, New York, 2006)

8. O. Jensen, B. Gips, T.O. Bergmann, M. Bonnefond, Temporal coding organized by coupled alpha and gamma oscillations prioritize visual processing. Trends Neurosci. **37**(7), 357–369 (2014)
9. E.M. Zion Golumbic, N. Ding, S. Bickel, P. Lakatos, C.A. Schevon, G.M. McKhann, R.R. Goodman, R. Emerson, A.D. Mehta, J.Z. Simon et al., Mechanisms underlying selective neuronal tracking of attended speech at a cocktail party. Neuron **77**(5), 980–991 (2013)
10. C. Kayser, M.A. Montemurro, N.K. Logothetis, S. Panzeri, Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns. Neuron **61**(4), 597–608 (2009)
11. R.T. Canolty, E. Edwards, S.S. Dalal, M. Soltani, S.S. Nagarajan, H.E. Kirsch, M.S. Berger, N.M. Barbaro, R.T. Knight, High gamma power is phase-locked to theta oscillations in human neocortex. Science **313**(5793), 1626–1628 (2006)
12. R. van der Meij, M. Kahana, E. Maris, Phase-amplitude coupling in human electrocorticography is spatially distributed and phase diverse. J. Neurosci. **32**(1), 111–123 (2012)
13. T. Staudigl, T. Zaehle, J. Voges, S. Hanslmayr, C. Esslinger, H. Hinrichs, F.C. Schmitt, H.-J. Heinze, A. Richardson-Klavehn, Memory signals from the thalamus: early thalamocortical phase synchronization entrains gamma oscillations during long-term memory retrieval. Neuropsychologia **50**(14), 3519–3527 (2012)
14. T.H. FitzGerald, A. Valentin, R. Selway, M.P. Richardson, Cross-frequency coupling within and between the human thalamus and neocortex. Front. Hum. Neurosci. **7**, 84 (2013)
15. A.B. Tort, R. Komorowski, H. Eichenbaum, N. Kopell, Measuring phase-amplitude coupling between neuronal oscillations of different frequencies. J. Neurophysiol. **104**(2), 1195–1210 (2010)
16. T. Yanagisawa, O. Yamashita, M. Hirata, H. Kishima, Y. Saitoh, T. Goto, T. Yoshimine, Y. Kamitani, Regulation of motor representation by phase-amplitude coupling in the sensorimotor cortex. J. Neurosci. **32**(44), 15467–15475 (2012)
17. R.C. Oldfield, The assessment and analysis of handedness: the Edinburgh inventory. Neuropsychologia **9**(1), 97–113 (1971)
18. B. Fischl, Freesurfer. NeuroImage **62**(2), 774–781 (2012)
19. A. Gramfort, T. Papadopoulo, E. Olivi, M. Clerc et al., OpenMEEG: opensource software for quasistatic bioelectromagnetics. Biomed. Eng. Online **9**(1), 45 (2010)
20. F. Tadel, S. Baillet, J.C. Mosher, D. Pantazis, R.M. Leahy, Brainstorm: a user-friendly application for MEG/EEG analysis. Comput. Intell. Neurosci. **2011**, 8 (2011)
21. R.S. Desikan, F. Ségonne, B. Fischl, B.T. Quinn, B.C. Dickerson, D. Blacker, R.L. Buckner, A.M. Dale, R.P. Maguire, B.T. Hyman et al., An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. NeuroImage **31**(3), 968–980 (2006)

Appendix A
Bioelectromagnetic Forward Modeling

A.1 Neuronal Basis of MEG and EEG Signals

Neurons in the brain function electrically, as well as chemically, and have associated electric and magnetic fields which can be detected outside the head. We first discuss the anatomical and physiological properties of neurons in the mammalian brain, that contribute to the electric and magnetic fields detected by MEG and EEG. Neuronal currents are considered at a level sufficient for understanding the primary source of these fields.

A.1.1 Neurons and Synapses

Neurons are cells that are highly specialized for signal processing and conduction via electrochemical and electrical processes. The morphological structure of a neuron includes a cell body, called the soma, and elaborate branching structures called dendrites and axons that enable communication with sensory receptors, distant neurons, etc. Inputs to a neuron are collected in a continuous fashion by the dendrites, and represented as a spatially and temporally continuous variation of the transmembrane voltage. Multiple inputs are summed in the dendritic tree, and the net input is often represented as transmembrane voltage at the soma. When the somatic voltage reaches some threshold, a discrete voltage pulse is generated, called an action potential, which propagates down the axon as a discrete output. The end of the axon has elaborate branching to enable communication with target neurons.

Neuronal communication occurs for the most part via chemical transmission through synapses, although a small minority of neurons also communicate electrically through gap junctions. When an action potential reaches the presynaptic terminal of a synapse with another neuron, it causes the release of neurotransmitter into the synaptic cleft between two cells. The released neurotransmitter binds with some probability to sites of the postsynaptic terminal, stimulating a flow of current

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9

across the membrane of the postsynaptic neuron. Postsynaptic currents may flow inward or outward, depending upon the type of neurotransmitter involved (glutamate or GABA), and the type of ion that flows in response. Excitatory synapses have the effect of increasing the membrane potential of the postsynaptic cell, i.e., making it more positive, while inhibitory synaptic connections decrease it. Neurons are exclusively inhibitory or excitatory.

A.1.2 Cortical Anatomy

The mammalian cerebral cortex is the outer mantle of cells surrounding the central structures, e.g., brainstem and thalamus. It is unique to mammals, and is believed to be necessary for most higher-level brain functions. Topologically the cerebral cortex is comprised of two spherical shells, corresponding to the two hemispheres and connected by the corpus callosum. Cortical thickness varies between 2 and 3 mm in the human, and is folded around the subcortical structures so as to appear wrinkled. Its average surface area is about 3000 cm^2. It is estimated that there are roughly 10^{11} neurons in the human brain, and 10^{10} of these in the cortex.

The diversity of structures of neurons is extraordinary. Approximately 85 % are pyramidal cells whose dendritic trees have a distinctive, elongated geometry that makes possible the generation of extracellular fields at large distances. The remaining 15 % may be broadly classified as stellate cells, whose dendritic trees are approximately spherical, and make little or no contribution to distant field. Both cell types are interconnected and together form a single dynamical network, but it is believed that the fields at large distances are dominated by pyramidal cells because of their size and number.

Synaptic connections in the cortex are dense. Each cortical neuron receives 10^4–10^5 synaptic connections, with most inputs coming from distinct neurons. Pyramidal cells make excitatory connections to both cell types. They make intracortical connections over lengths ranging 0.5–3 mm, and cortico-cortical connections over lengths ranging 1–20 cm. Stellate cells make inhibitory connections to both cell types. They make intracortical connections over lengths ranging only 0.02–0.03 mm, much shorter than pyramidal cells. Thus connections in the cerebral cortex are said to exhibit long-range excitation and short-range inhibition.

An interesting distinction exists between intracortical and cortico-cortical connections. Intracortical connections are made locally between neighboring neurons, such that the probability of a connection between two neurons falls off smoothly as a function of distance. In contrast, corticocortical connections are made via subcortical white-matter fibers, and behave non-locally in the sense that connection to a distant neuron does not imply connections with intermediate neurons. Because of the diversity of scales of these synaptic connections, and the nonlocal nature of the cortico-cortical connections, the cerebral cortex exhibits rich spatio-temporal dynamics spanning a wide range of length and time scales.

A.1.3 Neuronal Currents

Electric currents in biological tissue are primarily due to ions, e.g., K+, Na+, Cl−, Ca2+, etc. These ions flow in response to the local electric field, according to Ohm's law, but also in response to their local concentration gradient, according to Fick's law [1]. In the resting state of the membrane, the concentration gradients and electric field are due to ion channel pumps, which use energy acquired from ATP to move ions across the membrane against their diffusion gradient.

To a good approximation, the concentration of each ion inside and outside the membrane may be assumed constant in time. The transmembrane voltage, however, changes radically in time, the strongest example being the action potential. Thus for the purposes of discussing neural activity, we take the transmembrane potential V_m as the primary dynamic state variable to be considered. By convention, V_m is defined as the potential inside relative to that outside, i.e., $V_m = V_i - V_o$. The current flowing across the membrane maybe be viewed as a function of V_m, and therefore as the basis of the extracellular fields detected by MEG and EEG.

In the resting state of the neuron, the balance between electric and diffusive flows determine the resting membrane potential of the neuron [2]. To consider this balance, we define two related quantities: the charge current density, $\boldsymbol{J}$ [C/m^2 s], and the ionic flux $\boldsymbol{j}$ [mol/m^2 s]. For a single ionic species, these are trivially related by: $\boldsymbol{J} = zF\boldsymbol{j}$, where $z = q/e$ is the signed integer number of charges carried by an ion, and F is the Faraday's constant (96,500 C/mol). The flux $\boldsymbol{j}$ has two contributions, arising from the local concentration gradient ∇C and the local electric field $\boldsymbol{E}$, where C indicates the ionic concentration for different ions defined as the number of ions (in mol) per unit volume. Fick's law states that ions diffuse down their concentration gradient according to the linear relation $\boldsymbol{j} = -D\nabla C$, where D is the diffusion coefficient.

Furthermore, an ion within an applied electric field $\boldsymbol{E}$ accelerates initially and reaches a terminal velocity $\boldsymbol{v}$ given by $\boldsymbol{v} = \mu(z/|z|)\boldsymbol{E}$ where the factor $(z/|z|)$ accounts for the fact that negative ions travel in the opposite direction of the electric field. The quantity μ is called the mobility and the diffusion coefficient, $D = \frac{RT}{|z|F}\mu$, where $R = 8.314$ J/(mol K) is the ideal gas constant. Counting flow of particles through a parallelepiped, the ionic flux is $\mu C\frac{z}{|z|}\boldsymbol{E}$. This equation is related to the more common way of expressing the flow of charges in an electric field via Ohm's law, which in a volume conductor is written as $\boldsymbol{J} = \sigma\boldsymbol{E}$. Therefore, the total ionic flux is the linear sum of its diffusive and electrical contributions:

$$\boldsymbol{j} = \boldsymbol{j}_S + \boldsymbol{j}_E = -\frac{\mu}{|z|}\left[\frac{RT}{F}\nabla C + zC\nabla\boldsymbol{\Phi}\right], \tag{A.1}$$

where the electric potential $\boldsymbol{\Phi}$ is defined as $\boldsymbol{E} = -\nabla\boldsymbol{\Phi}$. This equation can be used to derive the Nernst equation, and the Goldman-Hodgkin-Katz equation [3], for the resting potential of a neuron, and also form the basis for modeling synaptic activity.

A.1.4 Neuronal Populations

Neuronal activity gives rise to extracellular electric and magnetic fields, which are detected in MEG and EEG. The fields generated by a single neuron are much too small to be detected at the scalp, but the fields generated by synchronously active neurons, with advantageous geometric alignment, can be detected. Dendrites are more able than axons to generate fields which are detectable at large distances. Stellate cells have approximately spherical dendritic trees, so the resulting extracellular fields tend to add with all possible orientations, and effectively cancel at distance.

Pyramidal cells have similar dendritic trees, but the tree branches are connected to the cell body (soma) by a long trunk, called the apical dendrite. It is a fortuitous anatomical feature of the cortex that pyramidal cells have their apical dendrites aligned systematically along the local normal to the cortical surface. In this way, the fields of synchronously active pyramidal neurons superimpose geometrically to be measurable at the scalp. Consider an approximately 1 cm^3 region of cortex, containing an order of 10^7 aligned pyramidal cells. If only 1 % of these neurons were synchronously active, then the relative contribution of synchronous to asynchronous neurons would be $10^5/\sqrt{10^7} \sim 30$. Thus, scalp MEG and EEG are largely dominated by synchronous neural activity.

A.2 Electromagnetic Fields in Conductive Media

A.2.1 Maxwell's Equations

Outside the scalp, we measure the net fields produced by synchronously active neurons. For calculations at this scale, the brain and other head tissues can be considered bulk materials, characterized by properties such as electric conductivity σ. The challenge is to compute magnetic fields and scalp potentials as a function of neuronal source currents in a head-shaped conductive medium. We begin here with the fundamentals of electric and magnetic fields in matter, and later specialize to biological tissue.

The physics of electric and magnetic fields are summarized by Maxwell's equations. In matter the macroscopic fields obey

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon}, \tag{A.2}$$

$$\nabla \cdot \boldsymbol{B} = 0, \tag{A.3}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}, \tag{A.4}$$

$$\nabla \times \boldsymbol{B} = \mu \boldsymbol{J} + \mu\epsilon \frac{\partial \boldsymbol{E}}{\partial t}. \tag{A.5}$$

where $\boldsymbol{E}$ is the electric field, $\boldsymbol{B}$ the magnetic field, ϵ the dielectric permittivity, and μ the magnetic permeability. The Maxwell's equations above have source terms given by the charge density ρ and the current density $\boldsymbol{J}$. Additional contributions arise from the time derivatives of the fields.

According to Eq. (A.1), the electric current density $\boldsymbol{J}$ is expressed as

$$\boldsymbol{J} = \boldsymbol{J}_S + \boldsymbol{J}_E = \boldsymbol{J}_S + \sigma \boldsymbol{E}, \tag{A.6}$$

where $\boldsymbol{J}_E$ is the "ohmic" current which flows in response to the local electric field, and $\boldsymbol{J}_S$ caused by the ionic flux $\boldsymbol{j}_S$ is the "source" current (or often called the "impressed" current), which flows in response to transmembrane concentration gradients whose sum corresponds to the net currents summed over all ions. In Eq. (A.6), σ is the bulk conductivity of the material, which is the main parameter governing the spread of ionic currents through the volume. Variations in conductivity affect both EEG and MEG spatial structure, although MEG is less sensitive to this parameter than EEG.

Charge conservation law is derived from Maxwell's equations in the following manner. Taking the divergence of Eq. (A.5) and using Eq. (A.2), we obtain

$$\nabla \cdot \boldsymbol{J} + \epsilon \frac{\partial \nabla \cdot \boldsymbol{E}}{\partial t} = \nabla \cdot \boldsymbol{J} + \frac{\partial \rho}{\partial t} = 0, \tag{A.7}$$

where the relationship $\nabla \cdot (\nabla \times \boldsymbol{B}) = 0$ is used. Integrating over a closed volume V bounded by a surface S and using the Gauss theorem yield:

$$\oint_S \boldsymbol{J} \cdot \hat{\boldsymbol{n}}\, dS = -\frac{\partial}{\partial t} \int_V \rho\, dV, \tag{A.8}$$

where $\hat{\boldsymbol{n}}$ is the outward unit normal to S. The integral on the left is the total current flowing outward across the S, and the integral on the right is the total charge in volume, so this relation states that the current flowing outward across S is equal to the rate of change of the charge in the volume.

A.2.1.1 Potential Formulation

It is convenient to re-express Maxwell's equations in terms of potential functions, which are related to the fields by simple derivatives. Because $\nabla \cdot \boldsymbol{B} = 0$, it is possible to write $\boldsymbol{B}$ as a gradient of some other vector field $\boldsymbol{A}$, such that

$$\boldsymbol{B} = \nabla \times \boldsymbol{A}, \tag{A.9}$$

where $\boldsymbol{A}$ is called the magnetic vector potential. Substituting the equation above into Eq. (A.4), we obtain

$$\nabla \times \boldsymbol{E} + \frac{\partial}{\partial t} \nabla \times \boldsymbol{A} = \nabla \times \left(\boldsymbol{E} + \frac{\partial \boldsymbol{A}}{\partial t} \right) = 0. \tag{A.10}$$

Using a scalar potential $\boldsymbol{\Phi}$, we can write

$$\boldsymbol{E} + \frac{\partial \boldsymbol{A}}{\partial t} = -\nabla \boldsymbol{\Phi}, \tag{A.11}$$

because for any scalar field $\boldsymbol{\Phi}$, the identity $\nabla \times (\nabla \boldsymbol{\Phi}) = 0$ holds. Thus, we have

$$\boldsymbol{E} = -\nabla \boldsymbol{\Phi} - \frac{\partial \boldsymbol{A}}{\partial t}. \tag{A.12}$$

A.2.1.2 Gauge Transformations

Since electric and magnetic fields are defined as derivatives of the potentials, the potentials are not uniquely specified by their respective electromagnetic fields. This flexibility in defining $\boldsymbol{A}$ and $\boldsymbol{\Phi}$ is referred to as the choice of gauge. Considering again the relationship $\nabla \times \nabla \xi = 0$ for any scalar field ξ, the vector potential $\boldsymbol{A}$ is only determined up to the gradient of a scalar function, i.e., we can make the replacement $\boldsymbol{A} \rightarrow \boldsymbol{A} - \nabla \xi$ without changing the value of the magnetic field $\boldsymbol{B}$. This implies that,

$$\begin{aligned} \boldsymbol{E} &\rightarrow -\nabla \boldsymbol{\Phi} - \tfrac{\partial}{\partial t}(\boldsymbol{A} - \nabla \xi) \\ &= -\nabla(\boldsymbol{\Phi} - \tfrac{\partial \xi}{\partial t}) - \tfrac{\partial \boldsymbol{A}}{\partial t}. \end{aligned} \tag{A.13}$$

In other words, to keep $\boldsymbol{E}$ from being unchanged, we must also make the replacement of

$$\boldsymbol{\Phi} \rightarrow \boldsymbol{\Phi} + \frac{\partial \xi}{\partial t}. \tag{A.14}$$

This freedom to choose ξ allows us to impose an additional relationship between $\boldsymbol{A}$ and $\boldsymbol{\Phi}$. Common choices include the "Coulomb" gauge where $\nabla \cdot \boldsymbol{A} = 0$ or the "Lorentz" gauge, where $\nabla \cdot \boldsymbol{A} + \mu\epsilon(\partial \boldsymbol{\Phi}/\partial t) = 0$. In the following, we use the convenient "Gulrajani" gauge [4] for decoupling the differential equations for $\boldsymbol{A}$ and $\boldsymbol{\Phi}$, thereby reflecting the fundamental symmetry between electric potentials and magnetic fields. The Gulrajani gauge is expressed as

$$\nabla \cdot \boldsymbol{A} + \mu\epsilon \frac{\partial \boldsymbol{\Phi}}{\partial t} + \mu\sigma \boldsymbol{\Phi} = 0. \tag{A.15}$$

A.2.2 Magnetic Field and Electric Potential in an Unbounded Conducting Medium

A.2.2.1 Potential Equations

We have shown that using the vector and scalar potentials, the Maxwell's equations in Eqs. (A.3) and (A.4) are rewritten, respectively, as

$$\boldsymbol{B} = \nabla \times \boldsymbol{A}, \tag{A.16}$$

$$\boldsymbol{E} = -\nabla \boldsymbol{\Phi} - \frac{\partial \boldsymbol{A}}{\partial t}. \tag{A.17}$$

Let us express the other two Maxwell's equations using these potentials and the source current $\boldsymbol{J}_S$. Let us substitute the two equations above into Eq. (A.5), resulting in

$$\nabla \times (\nabla \times \boldsymbol{A}) = \mu \boldsymbol{J} - \mu\epsilon \nabla \frac{\partial \boldsymbol{\Phi}}{\partial t} - \mu\epsilon \frac{\partial^2 \boldsymbol{A}}{\partial t^2}. \tag{A.18}$$

The relationship in Eq. (A.6) is expressed using the potentials such that

$$\boldsymbol{J} = \boldsymbol{J}_S - \sigma \nabla \boldsymbol{\Phi} - \sigma \frac{\partial \boldsymbol{A}}{\partial t}. \tag{A.19}$$

Substituting Eq. (A.19) into (A.18), and using the identity $\nabla \times (\nabla \times \boldsymbol{A}) = \nabla(\nabla \cdot \boldsymbol{A}) - \nabla^2 \boldsymbol{A}$, we obtain

$$\nabla^2 \boldsymbol{A} - \nabla \cdot \left[\nabla \cdot \boldsymbol{A} + \mu\epsilon \frac{\partial \boldsymbol{\Phi}}{\partial t} + \mu\sigma \boldsymbol{\Phi} \right] - \mu\epsilon \frac{\partial^2 \boldsymbol{A}}{\partial t^2} - \mu\sigma \frac{\partial \boldsymbol{A}}{\partial t} = -\mu \boldsymbol{J}_S, \tag{A.20}$$

and finally

$$\nabla^2 \boldsymbol{A} - \mu\epsilon \frac{\partial^2 \boldsymbol{A}}{\partial t^2} - \mu\sigma \frac{\partial \boldsymbol{A}}{\partial t} = -\mu \boldsymbol{J}_S, \tag{A.21}$$

where the gauge relationship in Eq. (A.15) is used in Eq. (A.20). From Maxwell's equation in Eq. (A.2), a similar derivation leads to

$$\nabla^2 \boldsymbol{\Phi} - \mu\epsilon \frac{\partial^2 \boldsymbol{\Phi}}{\partial t^2} - \mu\sigma \frac{\partial \boldsymbol{\Phi}}{\partial t} = -\frac{\rho}{\epsilon}. \tag{A.22}$$

Finally, we use Eqs. (A.6), (A.7), and (A.11), to obtain the time derivative of the charge ρ, expressed as

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= -\nabla \cdot \boldsymbol{J} \\ &= -\nabla \cdot (\sigma \boldsymbol{E} + \boldsymbol{J}_S) \\ &= -\sigma \nabla \cdot \left(-\nabla \boldsymbol{\Phi} - \frac{\partial \boldsymbol{A}}{\partial t} \right) - \nabla \cdot \boldsymbol{J}_S \\ &= \sigma \nabla^2 \boldsymbol{\Phi} + \sigma \frac{\partial \nabla \cdot \boldsymbol{A}}{\partial t} - \nabla \cdot \boldsymbol{J}_S \\ &= \sigma \nabla^2 \boldsymbol{\Phi} - \sigma \frac{\partial}{\partial t} \left(\mu \epsilon \frac{\partial \boldsymbol{\Phi}}{\partial t} + \mu \sigma \boldsymbol{\Phi} \right) - \nabla \cdot \boldsymbol{J}_S \\ &= \sigma \nabla^2 \boldsymbol{\Phi} - \epsilon \mu \sigma \frac{\partial^2 \boldsymbol{\Phi}}{\partial t^2} - \mu \sigma^2 \frac{\partial \boldsymbol{\Phi}}{\partial t} - \nabla \cdot \boldsymbol{J}_S, \end{aligned} \tag{A.23}$$

where we again use the gauge relationship in Eq. (A.15).

A.2.2.2 Derivation of Potentials

To facilitate the solution of the equations for potentials, we can exploit linearity and also assume harmonic time dependence with angular frequency ω. This assumption does not require that the signals have perfect sinusoidal dependence, and is merely a mathematical convenience that helps subsequent discussion for determining the quasi-static regime. Namely, we express the solutions of the partial differential equations in Eqs. (A.21), (A.22), and (A.23), such that

$$\widetilde{\boldsymbol{A}}(t, \boldsymbol{r}) = \boldsymbol{A}(\boldsymbol{r}) e^{i\omega t}, \tag{A.24}$$

$$\widetilde{\Phi}(t, \boldsymbol{r}) = \boldsymbol{\Phi}(\boldsymbol{r}) e^{i\omega t}, \tag{A.25}$$

$$\widetilde{\rho}(t, \boldsymbol{r}) = \rho(\boldsymbol{r}) e^{i\omega t}, \tag{A.26}$$

$$\widetilde{\boldsymbol{J}}_S(t, \boldsymbol{r}) = \boldsymbol{J}_S(\boldsymbol{r}) e^{i\omega t}. \tag{A.27}$$

Substituting $\widetilde{\boldsymbol{A}}(t, \boldsymbol{r})$ and $\widetilde{\boldsymbol{J}}_S(t, \boldsymbol{r})$ into Eq. (A.21), we obtain the following partial differential equation of space alone,

$$\nabla^2 \boldsymbol{A}(\boldsymbol{r}) + k^2 \boldsymbol{A}(\boldsymbol{r}) = -\mu \boldsymbol{J}_S(\boldsymbol{r}), \tag{A.28}$$

where, $k^2 = -i\omega\mu(\sigma + i\omega\epsilon)$.

Substitution of $\widetilde{\Phi}(t, \boldsymbol{r})$ and $\widetilde{\rho}(t, \boldsymbol{r})$ into Eq. (A.22) also results in

$$\nabla^2 \boldsymbol{\Phi}(\boldsymbol{r}) + k^2 \boldsymbol{\Phi}(\boldsymbol{r}) = -\frac{\rho(\boldsymbol{r})}{\epsilon}. \tag{A.29}$$

Substitution of $\widetilde{\Phi}(t, \boldsymbol{r})$, $\widetilde{\rho}(t, \boldsymbol{r})$, and $\widetilde{\boldsymbol{J}}_S(t, \boldsymbol{r})$ into Eq. (A.23) gives the relationship:

$$i\omega\rho(\boldsymbol{r}) = \sigma \nabla^2 \boldsymbol{\Phi}(\boldsymbol{r}) - \epsilon\mu\sigma (i\omega)^2 \boldsymbol{\Phi}(\boldsymbol{r}) - \mu\sigma^2 (i\omega) \boldsymbol{\Phi}(\boldsymbol{r}) - \nabla \cdot \boldsymbol{J}_S(\boldsymbol{r}),$$

which is simplified to

$$\rho(\boldsymbol{r}) = \frac{1}{i\omega}\sigma\nabla^2\boldsymbol{\Phi}(\boldsymbol{r}) - \left(\epsilon\mu\sigma(i\omega) + \mu\sigma^2\right)\boldsymbol{\Phi}(\boldsymbol{r}) - \frac{1}{i\omega}\nabla\cdot\boldsymbol{J}_S(\boldsymbol{r}). \tag{A.30}$$

By substituting Eq. (A.30) into (A.29), we obtain

$$\begin{aligned}\nabla^2\boldsymbol{\Phi}(\boldsymbol{r}) - i\omega\mu(\sigma + i\omega\epsilon)\boldsymbol{\Phi}(\boldsymbol{r}) \\ = -\frac{1}{i\omega\epsilon}\sigma\nabla^2\boldsymbol{\Phi}(\boldsymbol{r}) + \left(\mu\sigma(i\omega) + \frac{\mu\sigma^2}{\epsilon}\right)\boldsymbol{\Phi}(\boldsymbol{r}) + \frac{1}{i\omega\epsilon}\nabla\cdot\boldsymbol{J}_S(\boldsymbol{r}).\end{aligned}$$

The equation above results in

$$\nabla^2\boldsymbol{\Phi}(\boldsymbol{r}) + k^2\boldsymbol{\Phi}(\boldsymbol{r}) = \frac{\nabla\cdot\boldsymbol{J}_S(\boldsymbol{r})}{\sigma + i\omega\epsilon}. \tag{A.31}$$

In the absence of boundary conditions, Eqs. (A.28) and (A.31) have the following well-known solutions.

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi}\int_V \frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}e^{-ik|\boldsymbol{r}-\boldsymbol{r}'|}d^3r', \tag{A.32}$$

and

$$\boldsymbol{\Phi}(\boldsymbol{r}) = -\frac{1}{4\pi(\sigma + i\omega\epsilon)}\int_V \frac{\nabla'\cdot\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}e^{-ik|\boldsymbol{r}-\boldsymbol{r}'|}d^3r'. \tag{A.33}$$

Note that the derivation of these solutions has not involved any assumptions about the frequency ω or the tissue parameters σ, μ, ϵ. The typical values of these parameters, together with assumptions about the frequency ω, allow simplifications of the above equations suitable for EEG and MEG. The phase shifts of the electric potentials due to capacitive effects are accounted by the denominator of Eq. (A.33):

$$(\sigma + i\omega\epsilon) = \sigma(1 + i\frac{\omega\epsilon}{\sigma}). \tag{A.34}$$

Thus capacitive effects may be ignored if $\omega\epsilon/\sigma \ll 1$. Although the very highest frequencies generated by the neural activity corresponding to action potentials may reach 1000 Hz, MEG and EEG signals are expected to be dominated by synaptic and dendritic activity that involves slower time scales less than 100 Hz ($\omega < 200\,\pi$ rad/s).

We make use of measured biological parameters to evaluate the dimensionless quantity $\omega\epsilon/\sigma$. To a best approximation, biological tissues are not magnetizable, thus $\mu = 4\pi \times 10^{-7}$ H/m, the value of vacuum. The dielectric permittivity ϵ is tissue and frequency dependent such that for brain and scalp, $\omega\epsilon/\sigma \simeq 0.01$ at 100 Hz, and $\omega\epsilon/\sigma \simeq 0.03$ at 1000 Hz. Thus, capacitive effects may be ignored to within 1 % errors.

We then apply the Taylor series expansion:

$$e^{-ik|\boldsymbol{r}-\boldsymbol{r}'|} \simeq 1 - ik|\boldsymbol{r}-\boldsymbol{r}'| + \ldots \tag{A.35}$$

For MEG and EEG applications, we assume that $|\boldsymbol{r}-\boldsymbol{r}'| < 0.1$ m, typical for head-radius and distance from cortical sources to the sensors. With $\omega\epsilon/\sigma \ll 1$, we have $|k| \simeq \sqrt{\omega\mu\epsilon}$. Head tissue conductivity is highly variable, but the nominal value is $\sigma \approx 0.1$ s/m. Using this value leads to $|k||\boldsymbol{r}-\boldsymbol{r}'| \approx 0.0013$. Thus, propagation delays may also be ignored to within a 1 % error. With these simplifications, we can obtain

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d^3r', \tag{A.36}$$

and

$$\boldsymbol{\Phi}(\boldsymbol{r}) = -\frac{1}{4\pi\sigma}\int_V \frac{\nabla'\cdot\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d^3r'. \tag{A.37}$$

These simplified solutions are termed the quasi-static solutions, because the dependence on ω has been eliminated. This rigorous derivations often ignored in MEG and EEG literature, are given here to make explicit assumptions about tissue parameters and source frequencies which underly the quasi-static formulations of EEG and MEG.

A.2.2.3 Computation of Magnetic Field and Electric Potential

From the magnetic vector potential, the magnetic field is computed using:

$$\boldsymbol{B}(\boldsymbol{r}) = \nabla\times\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\int_V \nabla\times\frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d^3r'. \tag{A.38}$$

We use the identity:

$$\begin{aligned}\nabla\times\frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} =& \nabla\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\times\boldsymbol{J}_S(\boldsymbol{r}') + \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\nabla\times\boldsymbol{J}_S(\boldsymbol{r}')\\ =& \frac{\boldsymbol{J}_S(\boldsymbol{r}')\times(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3}.\end{aligned} \tag{A.39}$$

Note that since ∇ is applied to $\boldsymbol{r}$, the relationships $\nabla|\boldsymbol{r}-\boldsymbol{r}'|^{-1} = -(\boldsymbol{r}-\boldsymbol{r}')/|\boldsymbol{r}-\boldsymbol{r}'|^{-3}$ and $\nabla\times\boldsymbol{J}_S(\boldsymbol{r}') = 0$ hold. Therefore, the magnetic field in an unbounded homogeneous medium is expressed as

$$\boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\int_V \frac{\boldsymbol{J}_S(\boldsymbol{r}')\times(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3}d^3r'. \tag{A.40}$$

This equation shows that the magnetic field in an unbounded homogeneous medium is expressed by the famous Biot-Savart law with replacing the total current $\boldsymbol{J}$ with the source (impressed) current $\boldsymbol{J}_S$. In other words, the ohmic current $\boldsymbol{J}_E$ does not contribute to the magnetic field in an unbounded homogeneous medium.

The expression for the scalar potential in Eq. (A.37) can be simplified in a following manner. Let us use the identity

$$\int_V \nabla' \cdot \frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r' = \int_V \frac{\nabla' \cdot \boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r' + \int_V \boldsymbol{J}_S(\boldsymbol{r}') \cdot \nabla' \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r'. \quad \text{(A.41)}$$

The Gauss theorem is expressed as

$$\int_V \nabla' \cdot \frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r' = \oint_S \frac{\boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} \cdot \hat{\boldsymbol{n}} dS',$$

where $\hat{\boldsymbol{n}}$ is again the outward unit normal to the surface S. If we assume that $\boldsymbol{J}_S(\boldsymbol{r}')$ becomes zero on S, the surface integral on the right-hand side of the equation above becomes zero, and we can then get the relationship:

$$\int_V \frac{\nabla' \cdot \boldsymbol{J}_S(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r' = -\int_V \boldsymbol{J}_S(\boldsymbol{r}') \cdot \nabla' \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|} d^3r'. \quad \text{(A.42)}$$

Therefore, using

$$\nabla' \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|} = \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}, \quad \text{(A.43)}$$

Eq. (A.37) is rewritten as

$$\boldsymbol{\Phi}(\boldsymbol{r}) = \frac{1}{4\pi\sigma} \int_V \boldsymbol{J}_S(\boldsymbol{r}') \cdot \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d^3r'. \quad \text{(A.44)}$$

The above equation is the expression to compute the electric potential in an unbounded homogeneous medium.

A.2.2.4 Dipoles in an Unbounded Homogeneous Medium

The transmembrane current density $\boldsymbol{J}_S$ arises due to concentration gradients, Source models for $\boldsymbol{J}_S$ fall into two categories: rigorous and phenomenological. Although a rigorous source model accounts reasonably accurately for each of the microscopic currents, it is difficult to derive such source models. A phenomenological source model is one which produces the same external fields, but is artificial in the sense that it does not actually reflect the microscopic details of the problem.

A representative phenomenological source model is the dipole model. The dipole is the simplest source for both $\boldsymbol{\Phi}$ and $\boldsymbol{B}$ and can be written as:

$$\boldsymbol{J}_S = \boldsymbol{Q}(\boldsymbol{r}_0) = \boldsymbol{Q}\delta(\boldsymbol{r} - \boldsymbol{r}_0). \tag{A.45}$$

Substituting this for the equations for electric potential in Eq. (A.44) and magnetic field in Eq. (A.40) in a homogenous infinite volume conductor, we get

$$\boldsymbol{\Phi}(\boldsymbol{r}) = \frac{1}{4\pi\sigma}\boldsymbol{Q} \cdot \frac{\boldsymbol{r} - \boldsymbol{r}_0}{|\boldsymbol{r} - \boldsymbol{r}_0|^3}, \tag{A.46}$$

and

$$\boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\boldsymbol{Q} \times \frac{\boldsymbol{r} - \boldsymbol{r}_0}{|\boldsymbol{r} - \boldsymbol{r}_0|^3}. \tag{A.47}$$

The equations above are the expressions for computing the electric potential and magnetic field produced by an current dipole at $\boldsymbol{r}_0$ embedded in an unbounded homogeneous conductive medium.

A.2.3 Magnetic Field from a Bounded Conductor with Piecewise-Constant Conductivity

We next consider the magnetic field generated by an inhomogeneous conductor. We assume that the region V can be divided into subregions $V_j, j = 1, \ldots, \varpi$, and the region V_j has conductivity σ_j. The surface of V_j is denoted S_j. We also assume that the conductivity, $\sigma(\boldsymbol{r})$, is zero outside V. We start the derivation from the Bio-Savart law:

$$\boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\int_V \boldsymbol{J}(\boldsymbol{r}') \times \frac{\boldsymbol{r} - \boldsymbol{r}'}{|\boldsymbol{r} - \boldsymbol{r}'|^3}\, d^3r' = \frac{\mu_0}{4\pi}\int_V \boldsymbol{J}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')\, d^3r', \tag{A.48}$$

where

$$\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') = \frac{\boldsymbol{r} - \boldsymbol{r}'}{|\boldsymbol{r} - \boldsymbol{r}'|^3}.$$

Substituting $\boldsymbol{J}(\boldsymbol{r}') = \boldsymbol{J}_S(\boldsymbol{r}') - \sigma(\boldsymbol{r}')\nabla\boldsymbol{\Phi}(\boldsymbol{r}')$ into Eq. (A.48), we can obtain

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{r}) &= \frac{\mu_0}{4\pi}\int_V \left[\boldsymbol{J}_S(\boldsymbol{r}') - \sigma(\boldsymbol{r}')\nabla\boldsymbol{\Phi}(\boldsymbol{r}')\right] \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')\, d^3r' \\ &= \frac{\mu_0}{4\pi}\int_V \boldsymbol{J}_S(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')\, d^3r' - \frac{\mu_0}{4\pi}\sum_{j=1}^{\varpi}\sigma_j \int_{V_j} \nabla\boldsymbol{\Phi}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')\, d^3r'.\end{aligned} \tag{A.49}$$

Using

$$\nabla\boldsymbol{\Phi}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') = \nabla \times [\boldsymbol{\Phi}(\boldsymbol{r}')\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')], \tag{A.50}$$

the second term on the right-hand side of Eq. (A.49) can be rewritten as

$$\begin{aligned}\int_{V_j} \nabla \boldsymbol{\Phi}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')\, d^3 r' &= \int_{V_j} \nabla \times [\boldsymbol{\Phi}(\boldsymbol{r}')\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')]\, d^3 r' \\ &= \int_{S_j} \widehat{\boldsymbol{n}}(\boldsymbol{r}') dS \times [\boldsymbol{\Phi}(\boldsymbol{r}')\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')] \\ &= \int_{S_j} \boldsymbol{\Phi}(\boldsymbol{r}')\widehat{\boldsymbol{n}}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') dS, \end{aligned} \tag{A.51}$$

where S_j indicates the surface of V_j, and $\widehat{\boldsymbol{n}}(\boldsymbol{r}')$ is the outward unit normal vector of a surface element on S_j. We use the Gauss theorem to derive the right-hand side of Eq. (A.51). Substituting Eq. (A.51) into (A.49), we can derive the following Geselowitz formula [5]:

$$\boldsymbol{B}(\boldsymbol{r}) = \boldsymbol{B}_0(\boldsymbol{r}) - \frac{\mu_0}{4\pi} \sum_{j=1}^{\varpi} (\sigma_j - \sigma_j') \int_{S_j} \boldsymbol{\Phi}(\boldsymbol{r}')\widehat{\boldsymbol{n}}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') dS, \tag{A.52}$$

where σ_j' is the conductivity just outside V_j. Also, in Eq. (A.52), $\boldsymbol{B}_0(\boldsymbol{r})$ is the magnetic field for the infinite homogeneous conductor in Eq. (A.40).

A.2.4 Magnetic Field from a Homogeneous Spherical Conductor

Here, we assume that V_j is spherically symmetric, and we set the coordinate origin at the center of V_j. We have

$$\boldsymbol{B}(\boldsymbol{r}) \cdot \boldsymbol{e}_r = \boldsymbol{B}_0(\boldsymbol{r}) \cdot \boldsymbol{e}_r - \frac{\mu_0}{4\pi} \sum_{j=1}^{\varpi} (\sigma_j - \sigma_j') \int_{S_j} \boldsymbol{\Phi}(\boldsymbol{r}')\widehat{\boldsymbol{n}}(\boldsymbol{r}') \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') \cdot \boldsymbol{e}_r dS, \tag{A.53}$$

where $\boldsymbol{e}_r$ is the unit vector in the radial direction, which is defined as $\boldsymbol{e}_r = \boldsymbol{r}/|\boldsymbol{r}|$. Since $\widehat{\boldsymbol{n}}(\boldsymbol{r}) = \boldsymbol{e}_r$ in the case of a spherically symmetric conductor, the second term is equal to zero, and we have the relationship

$$\boldsymbol{B}(\boldsymbol{r}) \cdot \boldsymbol{e}_r = \boldsymbol{B}_0(\boldsymbol{r}) \cdot \boldsymbol{e}_r. \tag{A.54}$$

This equation indicates that the radial component of the magnetic field is not affected by the volume current and that the radial component is determined solely by the primary current.

We next derive a closed-form formula for the magnetic field outside a spherically-symmetric homogeneous conductor. The derivation is according to Sarvas [6]. The

relationship $\nabla \times \boldsymbol{B}(\boldsymbol{r}) = 0$ holds outside the volume conductor, because there is no electric current. Thus, $\boldsymbol{B}(\boldsymbol{r})$ can be expressed in terms of the magnetic scalar potential $U(\boldsymbol{r})$, as

$$\boldsymbol{B}(\boldsymbol{r}) = -\mu_0 \nabla U(\boldsymbol{r}). \tag{A.55}$$

This potential function is derived from

$$U(\boldsymbol{r}) = \frac{1}{\mu_0}\int_0^\infty \boldsymbol{B}(\boldsymbol{r}+\tau\boldsymbol{e}_r)\cdot\boldsymbol{e}_r d\tau = \frac{1}{\mu_0}\int_0^\infty \boldsymbol{B}_0(\boldsymbol{r}+\tau\boldsymbol{e}_r)\cdot\boldsymbol{e}_r d\tau, \tag{A.56}$$

where we use the relationship in Eq. (A.54). Assuming that a dipole source exists at $\boldsymbol{r}_0$, by substituting Eq. (A.47) into Eq. (A.56) and performing the integral, we obtain

$$\begin{aligned} U(\boldsymbol{r}) &= \frac{1}{\mu_0}\int_0^\infty \boldsymbol{B}_0(\boldsymbol{r}+\tau\boldsymbol{e}_r)\cdot\boldsymbol{e}_r d\tau \\ &= \frac{1}{4\pi}\boldsymbol{Q}\times(\boldsymbol{r}-\boldsymbol{r}_0)\cdot\boldsymbol{e}_r \int_0^\infty \frac{d\tau}{|\boldsymbol{r}+\tau\boldsymbol{e}_r-\boldsymbol{r}_0|^3} \\ &= -\frac{1}{4\pi}\frac{(\boldsymbol{Q}\times\boldsymbol{r}_0)\cdot\boldsymbol{r}}{\Lambda}, \end{aligned} \tag{A.57}$$

where

$$\Lambda = |\boldsymbol{r}-\boldsymbol{r}_0|(|\boldsymbol{r}-\boldsymbol{r}_0|\boldsymbol{r}+|\boldsymbol{r}|^2-\boldsymbol{r}_0\cdot\boldsymbol{r}). \tag{A.58}$$

The well-known Sarvas formula [6] for $\boldsymbol{B}(\boldsymbol{r})$ is then obtained by substituting Eq. (A.57) into Eq. (A.55) and performing the gradient operation. The results are expressed as

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\nabla\frac{(\boldsymbol{Q}\times\boldsymbol{r}_0)\cdot\boldsymbol{r}}{\Lambda} &= \frac{\mu_0}{4\pi}\left[\frac{\boldsymbol{Q}\times\boldsymbol{r}_0}{\Lambda} - \frac{1}{\Lambda^2}(\boldsymbol{Q}\times\boldsymbol{r}_0)\cdot\boldsymbol{r}\nabla\Lambda\right] \\ &= \frac{\mu_0}{4\pi\Lambda^2}\left[\Lambda\boldsymbol{Q}\times\boldsymbol{r}_0 - [(\boldsymbol{Q}\times\boldsymbol{r}_0)\cdot\boldsymbol{r}]\nabla\Lambda\right], \end{aligned} \tag{A.59}$$

where

$$\begin{aligned} \nabla\Lambda = &\left[\frac{|\boldsymbol{r}-\boldsymbol{r}_0|^2}{|\boldsymbol{r}|} + \frac{(\boldsymbol{r}-\boldsymbol{r}_0)\cdot\boldsymbol{r}}{|\boldsymbol{r}-\boldsymbol{r}_0|} + 2|\boldsymbol{r}-\boldsymbol{r}_0| + 2|\boldsymbol{r}|\right]\boldsymbol{r} \\ &- \left[|\boldsymbol{r}-\boldsymbol{r}_0| + 2|\boldsymbol{r}| + \frac{(\boldsymbol{r}-\boldsymbol{r}_0)\cdot\boldsymbol{r}}{|\boldsymbol{r}-\boldsymbol{r}_0|}\right]\boldsymbol{r}_0. \end{aligned} \tag{A.60}$$

We can see that when $\boldsymbol{r}_0$ approaches the center of the sphere, $\boldsymbol{B}(\boldsymbol{r})$ becomes zero, and no magnetic field is generated outside the conductor from a source at the origin. Also, if the source vector $\boldsymbol{Q}$ and the location vector $\boldsymbol{r}_0$ are parallel, i.e., if the primary current source is oriented in the radial direction, no magnetic fields are generated outside the spherical conductor from such a radial source. This is because the two

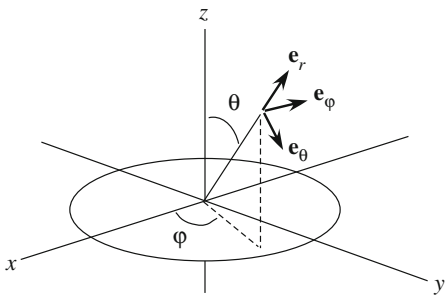

Fig. A.1 The three orthogonal directions $(\boldsymbol{e}_r, \boldsymbol{e}_\phi, \boldsymbol{e}_\theta)$ used to express the source vector when the spherically-symmetric conductor model is used for the forward calculation

terms on the left-hand side of Eq. (A.59) contain the vector product $\boldsymbol{Q} \times \boldsymbol{r}_0$, which is equal to zero when $\boldsymbol{Q}$ and $\boldsymbol{r}_0$ are parallel.

Therefore, when using the spherically-homogeneous conductor model, instead of the x, y, z directions, we usually use the three orthogonal directions $(\boldsymbol{e}_r, \boldsymbol{e}_\phi, \boldsymbol{e}_\theta)$ to express the source vector. These directions are illustrated in Fig. A.1. Because the $\boldsymbol{e}_r$ component of a source never creates a measurable magnetic field outside the spherical conductor, we can disregard this component and only deal with the $\boldsymbol{e}_\phi$ and $\boldsymbol{e}_\theta$ components of the source vector.

References

1. R. Plonsey, *Bioelectric phenomena*. Wiley Online Library, 1969.
2. H. Tuckwell, "Introduction to Theoretical Neurobiology vols. 1 and 2, 1988," *Cambridge University Press, Cambridge, Longtin A., Bulsara A., Moss F., Phys. Rev. Lett*, vol. 65, p. 656, 1991.
3. B. Hille, *Ion channels of excitable membranes*, vol. 507. Sinauer Sunderland, MA, 2001.
4. R. M. Gulrajani, *Bioelectricity and biomagnetism*. John Wiley and Sons, 1998.
5. D. B. Geselowitz, "On the magnetic field generated outside an inhomogeneous volume conductor by internal current sources," *IEEE Trans. Biomed. Eng.*, vol. 2, pp. 346–347, 1970.
6. J. Sarvas, "Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem," *Phys. Med. Biol.*, vol. 32, pp. 11–22, 1987.

Appendix B
Basics of Bayesian Inference

B.1 Linear Model and Bayesian Inference

This appendix explains the basics of Bayesian inference. Here, we consider the general problem of estimating the vector $\boldsymbol{x}$, containing N unknowns: $\boldsymbol{x} = [x_1, \ldots, x_N]^T$, from the observation (the sensor data) $\boldsymbol{y}$, containing M elements: $\boldsymbol{y} = [y_1, \ldots, y_M]^T$. We assume a linear model between the observation $\boldsymbol{y}$ and the unknown vector $\boldsymbol{x}$, such that

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{B.1}$$

where $\boldsymbol{H}$ is an $M \times N$ matrix that expresses the forward relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$, and $\boldsymbol{\varepsilon}$ is an additive noise overlapped to the observation $\boldsymbol{y}$.

To solve this estimation problem based on the Bayesian inference, we consider $\boldsymbol{x}$ a vector random variable, and use the following three kinds of probability distributions.

(1) $p(\boldsymbol{x})$: The probability distribution on the unknown $\boldsymbol{x}$. This is called the prior probability distribution. It represents our prior knowledge on the unknown $\boldsymbol{x}$.
(2) $p(\boldsymbol{y}|\boldsymbol{x})$: The conditional probability of $\boldsymbol{y}$ given $\boldsymbol{x}$. This conditional probability is equal to the likelihood. The maximum likelihood method, mentioned in Chap. 2, estimates the unknown $\boldsymbol{x}$ as the value of $\boldsymbol{x}$ that maximizes $p(\boldsymbol{y}|\boldsymbol{x})$.
(3) $p(\boldsymbol{x}|\boldsymbol{y})$: The probability of $\boldsymbol{x}$ given observation $\boldsymbol{y}$. This is called the posterior probability. The Bayesian inference estimates the unknown parameter $\boldsymbol{x}$ based on this posterior probability.

The posterior probability is obtained from the prior probability $p(\boldsymbol{x})$ and the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ using Bayes' rule,

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{\int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}}. \tag{B.2}$$

On the right-hand side of Eq. (B.2), the denominator is used only for the normalization $\int p(\boldsymbol{x}|\boldsymbol{y})\mathrm{d}\boldsymbol{x} = 1$, and often the denominator is not needed to estimate the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. Therefore, Bayes' rule can be expressed in a simpler form such as

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \tag{B.3}$$

where the notation $\propto$ means that both sides are equal, ignoring a multiplicative constant.

Bayesian inference uses Bayes' rule in Eq. (B.2) or (B.3) to estimate $\boldsymbol{x}$. A problem here is how to determine the prior probability distribution $p(\boldsymbol{x})$. A general strategy for this problem is to determine the prior probability by taking what we know about the unknown parameters $\boldsymbol{x}$ into account. However, if there is no prior knowledge on $\boldsymbol{x}$, we must use the uniform prior distribution, i.e.,

$$p(\boldsymbol{x}) = \text{constant}. \tag{B.4}$$

With Eq. (B.4), Bayes' rule in Eq. (B.3) becomes

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x}). \tag{B.5}$$

In this case, the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$ is equal to the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$, resulting in a situation that Bayesian and maximum likelihood methods give the same solution. The prior distribution in Eq. (B.4) is referred to as the non-informative prior.

Even when some prior information on the unknown parameter $\boldsymbol{x}$ is available, exact probability distributions are generally difficult to determine. Therefore, the probability distribution is usually determined according to the convenience in computing the posterior distribution. Some probability distributions have the same forms as the prior and posterior distributions for given $p(\boldsymbol{y}|\boldsymbol{x})$. One representative example of such distributions is the Gaussian distribution. That is, if the noise $\boldsymbol{\varepsilon}$ is Gaussian, a Gaussian prior distribution gives a Gaussian posterior distribution. The derivation of the posterior distribution in the Gaussian model is explained in Sect. B.3.

B.2 Point Estimate of Unknown $\boldsymbol{x}$

In Bayesian inference, the unknown parameter $\boldsymbol{x}$ is estimated based on the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$. Then, how can we obtain the optimum estimate $\widehat{\boldsymbol{x}}$ based on $p(\boldsymbol{x}|\boldsymbol{y})$? There are two ways to compute the estimate $\widehat{\boldsymbol{x}}$ based on a given posterior distribution. One way chooses the $\boldsymbol{x}$ that maximizes the posterior, i.e.,

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\boldsymbol{y}). \tag{B.6}$$

This $\widehat{\boldsymbol{x}}$ is called the maximum a posteriori (MAP) estimate.

The other way is to choose $\widehat{\boldsymbol{x}}$ that minimizes the squared error between the estimate $\widehat{\boldsymbol{x}}$ and the true value $\boldsymbol{x}$. The squared error is expressed as $E\left[(\widehat{\boldsymbol{x}}-\boldsymbol{x})^T(\widehat{\boldsymbol{x}}-\boldsymbol{x})\right]$, and the estimate $\widehat{\boldsymbol{x}}$ is obtained using

$$\widehat{\boldsymbol{x}} = \underset{\widehat{\boldsymbol{x}}}{\text{argmin}} E\left[(\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x})\right]. \tag{B.7}$$

This estimate is called the minimum mean squared error (MMSE) estimate. Here, taking the relationship,

$$\begin{aligned} E\left[(\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x})\right] &= \iint_{-\infty}^{\infty} (\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x}) p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} (\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\right] p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}, \end{aligned} \tag{B.8}$$

and the fact that $p(\boldsymbol{y}) \geq 0$ into consideration, the $\widehat{\boldsymbol{x}}$ that minimizes $E[(\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x})]$ is equal to

$$\widehat{\boldsymbol{x}} = \underset{\widehat{\boldsymbol{x}}}{\text{argmin}} \int_{-\infty}^{\infty} (\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x}. \tag{B.9}$$

Taking the derivative of the above integral with respect to $\widehat{\boldsymbol{x}}$, and setting it to zero, we have

$$\frac{\partial}{\partial \widehat{\boldsymbol{x}}} \int_{-\infty}^{\infty} (\widehat{\boldsymbol{x}} - \boldsymbol{x})^T(\widehat{\boldsymbol{x}} - \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x} = 2 \int_{-\infty}^{\infty} (\widehat{\boldsymbol{x}} - \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x} = 0. \tag{B.10}$$

Therefore, the MMSE estimate is equal to

$$\widehat{\boldsymbol{x}} = \int_{-\infty}^{\infty} \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x}. \tag{B.11}$$

That is, the MMSE estimate is equal to the mean of the posterior. Note that, when the posterior distribution is Gaussian, the MAP estimate and the MMSE estimate are equal, because the Gaussian distribution reaches its maximum at the mean.

B.3 Derivation of Posterior Distribution in the Gaussian Model

Let us consider the problem in which the time series of unknown parameters $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K$ are estimated using the time series of observation data $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$ where $\boldsymbol{y}(t_k)$ and $\boldsymbol{x}(t_k)$ are denoted $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$. The relationship in Eq. (B.1) holds between $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$, i.e.,

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{\varepsilon}. \tag{B.12}$$

In this chapter, the whole time series data $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$ is collectively denoted $\boldsymbol{y}$, and the whole time series data $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K$ are collectively denoted $\boldsymbol{x}$. We assume that the noise $\boldsymbol{\varepsilon}$ is Gaussian and is identically and independently distributed across time, i.e.,

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\varepsilon}|\boldsymbol{0}, \boldsymbol{\Lambda}^{-1}), \tag{B.13}$$

where we omit the notation of the time index k from $\boldsymbol{\varepsilon}$. In Eq. (B.13), $\boldsymbol{\Lambda}$ is a diagonal precision matrix of which the jth diagonal entry is equal to the noise precision for the jth observation data. Then, using Eqs. (B.13) and (C.3), the conditional probability $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is obtained as

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{H}\boldsymbol{x}_k, \boldsymbol{\Lambda}^{-1}). \tag{B.14}$$

The conditional probability of the whole time series of $\boldsymbol{y}_k$ given the whole time series of $\boldsymbol{x}_k$ is given by

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K) \\ &= \prod_{k=1}^{K} p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{H}\boldsymbol{x}_k, \boldsymbol{\Lambda}^{-1}). \end{aligned} \tag{B.15}$$

The prior distribution of $\boldsymbol{x}_k$ is assumed to be Gaussian and independent across time:

$$p(\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}). \tag{B.16}$$

The prior distribution for the whole time series of $\boldsymbol{x}_k$ is expressed as

$$p(\boldsymbol{x}) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K) = \prod_{k=1}^{K} p(\boldsymbol{x}_k) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{0}, \boldsymbol{\Phi}^{-1}). \tag{B.17}$$

In this case, the posterior probability is independent across time, and given by

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K) = \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{y}_k). \tag{B.18}$$

The posterior probability $p(\boldsymbol{x}_k|\boldsymbol{y}_k)$ can be derived by substituting Eqs. (B.16) and (B.14) into Bayes' rule:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k). \tag{B.19}$$

Actual computation of $p(\boldsymbol{x}_k|\boldsymbol{y}_k)$ is performed in the following manner. Since we know that the posterior distribution is also Gaussian, the posterior distribution is assumed to be

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{x}}_k, \boldsymbol{\Gamma}^{-1}), \tag{B.20}$$

where $\bar{\boldsymbol{x}}$ is the posterior mean, and $\boldsymbol{\Gamma}$ is the posterior precision matrix. The exponential part of the above Gaussian distribution is given by

$$-\frac{1}{2}(\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k)^T\boldsymbol{\Gamma}(\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k)=-\frac{1}{2}\boldsymbol{x}_k^T\boldsymbol{\Gamma}\boldsymbol{x}_k+\boldsymbol{x}_k^T\boldsymbol{\Gamma}\bar{\boldsymbol{x}}_k+\mathcal{C}, \tag{B.21}$$

where $\mathcal{C}$ represents terms that do not contain $\boldsymbol{x}_k$. The exponential part of the right-hand side of Eq. (B.19) is given by

$$-\frac{1}{2}\left[\boldsymbol{x}_k^T\boldsymbol{\Phi}\boldsymbol{x}_k+(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)\right]. \tag{B.22}$$

The above expression can be rewritten as

$$-\frac{1}{2}\boldsymbol{x}_k^T(\boldsymbol{\Phi}+\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H})\boldsymbol{x}_k+\boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{y}_k+\mathcal{C}', \tag{B.23}$$

where $\mathcal{C}'$ again represents terms that do not contain $\boldsymbol{x}_k$. Comparing the quadratic and linear terms of $\boldsymbol{x}_k$ between Eqs. (B.21) and (B.23) gives the relationships

$$\boldsymbol{\Gamma}=\boldsymbol{\Phi}+\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H}, \tag{B.24}$$

$$\bar{\boldsymbol{x}}_k=\boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{y}_k=(\boldsymbol{\Phi}+\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{y}_k. \tag{B.25}$$

The precision matrix and the mean of the posterior distribution are obtained in Eqs. (B.24) and (B.25), respectively. This $\bar{\boldsymbol{x}}_k$ is the MMSE (and also MAP) estimate of $\boldsymbol{x}_k$. Using the matrix inversion formula in Eq. (C.92), $\bar{\boldsymbol{x}}_k$ can also be expressed as

$$\bar{\boldsymbol{x}}_k=\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T+\boldsymbol{\Lambda}^{-1})^{-1}\boldsymbol{y}_k=\boldsymbol{\Upsilon}\boldsymbol{H}^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y}_k, \tag{B.26}$$

where $\boldsymbol{\Upsilon}$, which is equal to $\boldsymbol{\Phi}^{-1}$, is the covariance matrix of the prior distribution, and $\boldsymbol{\Sigma}_y$ is expressed in Eq. (B.30).

B.4 Derivation of the Marginal Distribution $p(\boldsymbol{y})$

The probability for the observation $\boldsymbol{y}$ is obtained using

$$p(\boldsymbol{y})=\prod_{k=1}^{K}p(\boldsymbol{y}_k)=\prod_{k=1}^{K}\int p(\boldsymbol{x}_k,\boldsymbol{y}_k)\mathrm{d}\boldsymbol{x}_k=\prod_{k=1}^{K}\int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k. \tag{B.27}$$

Substituting Eqs. (B.16) and (B.14) into the equation above, we get

$$p(\boldsymbol{y}) = \prod_{k=1}^{K} \int \mathcal{N}(\boldsymbol{y}_k | \boldsymbol{H}\boldsymbol{x}_k, \boldsymbol{\Lambda}^{-1}) \mathcal{N}(\boldsymbol{x}_k | \boldsymbol{0}, \boldsymbol{\Phi}^{-1}) \mathrm{d}\boldsymbol{x}_k. \tag{B.28}$$

This $p(\boldsymbol{y})$ is called the marginal likelihood. This integral is computed in Sect. 4.3. Using Eq. (4.30), we have

$$p(\boldsymbol{y}) \propto \prod_{k=1}^{K} \frac{1}{|\boldsymbol{\Sigma}_y|^{1/2}} \exp\left[-\frac{1}{2} \boldsymbol{y}_k^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{y}_k \right], \tag{B.29}$$

where $\boldsymbol{\Sigma}_y$ is expressed as

$$\boldsymbol{\Sigma}_y = \boldsymbol{\Lambda}^{-1} + \boldsymbol{H}\boldsymbol{\Phi}^{-1}\boldsymbol{H}^T. \tag{B.30}$$

This $\boldsymbol{\Sigma}_y$ is referred to as the model data covariance. Taking the normalization into account, we get the marginal distribution $p(\boldsymbol{y})$, such that

$$p(\boldsymbol{y}) = \prod_{k=1}^{K} p(\boldsymbol{y}_k) \quad \text{where} \quad p(\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{y}_k | \boldsymbol{0}, \boldsymbol{\Sigma}_y). \tag{B.31}$$

Thus, $\boldsymbol{\Sigma}_y$, the model data covariance, is the covariance matrix of the marginal distribution $p(\boldsymbol{y}_k)$.

B.5 Expectation Maximization (EM) Algorithm

B.5.1 Marginal Likelihood Maximization

The Bayesian estimate of the unknown $\boldsymbol{x}$, $\bar{\boldsymbol{x}}_k$, is computed using Eq. (B.25). However, when computing $\bar{\boldsymbol{x}}_k$, $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ are needed. The precision matrix of the prior distribution $\boldsymbol{\Phi}$ and that of the noise $\boldsymbol{\Lambda}$ are called the hyperparameters. Quite often, the hyperparameters are unknown, and should also be estimated from the observation $\boldsymbol{y}$. The hyperparameters may be estimated by maximizing $p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$, which is the likelihood with respect to $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$. This $p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ is called the marginal likelihood or the data evidence to discriminate it from the conventional likelihood $p(\boldsymbol{y}|\boldsymbol{x})$.

The marginal likelihood can be computed with $p(\boldsymbol{x}|\boldsymbol{\Phi})$ and $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Lambda})$ using[1]

$$p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \int_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Lambda}) p(\boldsymbol{x}|\boldsymbol{\Phi}) \mathrm{d}\boldsymbol{x}. \tag{B.32}$$

However, computation of $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ that maximize $p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ is generally quite troublesome, as is demonstrated in detail in Chap. 4. In the following, instead of directly

[1] The notation $\mathrm{d}\boldsymbol{x}$ indicates $\mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2 \cdots \mathrm{d}\boldsymbol{x}_K$.

maximizing the marginal likelihood, we describe an algorithm that maximizes a quantity called the average data likelihood to obtain estimates for the hyperparameters. This algorithm is called the expectation maximization (EM) algorithm, and is described in the following.

B.5.2 Average Data Likelihood

The EM algorithm computes the quantity called the average data likelihood. Computing the average data likelihood is much easier than computing the marginal likelihood. To define the average data likelihood, let us first define the complete data likelihood, such that

$$\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Lambda}) + \log p(\boldsymbol{x}|\boldsymbol{\Phi}). \tag{B.33}$$

If we observed not only $\boldsymbol{y}$ but also $\boldsymbol{x}$, we could have estimated $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ by maximizing $\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ with respect to these hyperparameters. However, since we do not observe $\boldsymbol{x}$, we must substitute for the unknown $\boldsymbol{x}$ in $\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ with some "reasonable" value.

Having observed $\boldsymbol{y}$, we actually know which values of $\boldsymbol{x}$ are reasonable, and our best knowledge on the unknown $\boldsymbol{x}$ is represented by the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. Thus, the "reasonable" value would be the one that maximizes the posterior probability, and one solution would be to use the MAP estimate of $\boldsymbol{x}$ in $\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$. A better solution would be to use all possible values of $\boldsymbol{x}$ in the complete data likelihood and average over it with the posterior probability. This results in the average data likelihood, $\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda})$:

$$\begin{aligned}\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}) &= \int p(\boldsymbol{x}|\boldsymbol{y}) \log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda}) \mathrm{d}\boldsymbol{x} \\ &= E\left[\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})\right] \\ &= E\left[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Lambda})\right] + E\left[\log p(\boldsymbol{x}|\boldsymbol{\Phi})\right],\end{aligned} \tag{B.34}$$

where the expectation $E[\,\cdot\,]$ is taken with respect to the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$. The estimates of the hyperparameters, $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ are obtained using

$$\widehat{\boldsymbol{\Lambda}} = \underset{\boldsymbol{\Lambda}}{\operatorname{argmax}}\, \Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}), \tag{B.35}$$

$$\widehat{\boldsymbol{\Phi}} = \underset{\boldsymbol{\Phi}}{\operatorname{argmax}}\, \Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}). \tag{B.36}$$

In the Gaussian model discussed in Sect. B.3, $p(\boldsymbol{x}|\boldsymbol{\Phi})$ and $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Lambda})$ are expressed in Eqs. (B.17) and (B.15), respectively. Substituting Eqs. (B.17) and (B.15) into (B.33), the complete data likelihood is expressed as[2]

[2] The constant terms containing 2π are ignored here.

$$\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \frac{K}{2}\log|\boldsymbol{\Phi}| - \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{x}_k^T\boldsymbol{\Phi}\boldsymbol{x}_k + \frac{K}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k). \tag{B.37}$$

Thus, the average data likelihood is obtained as

$$\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \frac{K}{2}\log|\boldsymbol{\Phi}| - \frac{1}{2}E\left[\sum_{k=1}^{K}\boldsymbol{x}_k^T\boldsymbol{\Phi}\boldsymbol{x}_k\right] + \frac{K}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}E\left[\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T\boldsymbol{\Lambda}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)\right]. \tag{B.38}$$

B.5.3 Update Equation for Prior Precision $\boldsymbol{\Phi}$

Let us first derive the update equation for $\boldsymbol{\Phi}$. The updated value of $\boldsymbol{\Phi}$, $\widehat{\boldsymbol{\Phi}}$, is the one that maximizes the average data likelihood, $\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda})$. The derivative of $\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ with respect to $\boldsymbol{\Phi}$ is given by

$$\frac{\partial}{\partial\boldsymbol{\Phi}}\Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \frac{K}{2}\boldsymbol{\Phi}^{-1} - \frac{1}{2}E\left[\sum_{k=1}^{K}\boldsymbol{x}_k\boldsymbol{x}_k^T\right]. \tag{B.39}$$

Here we use Eqs. (C.89) and (C.90). Setting this derivative to zero, we have

$$\widehat{\boldsymbol{\Phi}}^{-1} = \frac{1}{K}E\left[\sum_{k=1}^{K}\boldsymbol{x}_k\boldsymbol{x}_k^T\right]. \tag{B.40}$$

Using the posterior precision $\boldsymbol{\Gamma}$, the relationship

$$E\left[\sum_{k=1}^{K}\boldsymbol{x}_k\boldsymbol{x}_k^T\right] = \sum_{k=1}^{K}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^T + K\boldsymbol{\Gamma}^{-1} \tag{B.41}$$

holds. Therefore, the update equation for $\boldsymbol{\Phi}$ is expressed as

$$\widehat{\boldsymbol{\Phi}}^{-1} = \frac{1}{K}\sum_{k=1}^{K}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^T + \boldsymbol{\Gamma}^{-1}. \tag{B.42}$$

B.5.4 Update Equation for Noise Precision $\boldsymbol{\Lambda}$

We next derive the update equation for $\boldsymbol{\Lambda}$. The derivative of the average data likelihood with respect to $\boldsymbol{\Lambda}$ is given by

$$\frac{\partial}{\partial \boldsymbol{\Lambda}} \Theta(\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = \frac{K}{2}\boldsymbol{\Lambda}^{-1} - \frac{1}{2}E\left[\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T\right]. \tag{B.43}$$

Setting this derivative to zero, we get

$$\widehat{\boldsymbol{\Lambda}}^{-1} = \frac{1}{K}E\left[\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T\right]. \tag{B.44}$$

Using the mean and the precision of the posterior distribution, the above equation is rewritten as

$$\widehat{\boldsymbol{\Lambda}}^{-1} = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\bar{\boldsymbol{x}}_k)(\boldsymbol{y}_k - \boldsymbol{H}\bar{\boldsymbol{x}}_k)^T + \boldsymbol{H}\boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T. \tag{B.45}$$

B.5.5 Summary of the EM Algorithm

The EM algorithm is a recursive algorithm. We first set appropriate initial values to the hyperparameters $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$, which are then used for computing the posterior distribution, i.e., for computing the mean and the precision of the posterior distribution using

$$\begin{aligned}
\boldsymbol{\Gamma} &= \boldsymbol{\Phi} + \boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H}, \\
\bar{\boldsymbol{x}}_k &= \boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{y}_k = (\boldsymbol{\Phi} + \boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{y}_k.
\end{aligned}$$

With these parameter values of the posterior distribution, the hyperparameters $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ are updated using

$$\begin{aligned}
\widehat{\boldsymbol{\Phi}}^{-1} &= \frac{1}{K}\sum_{k=1}^{K}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^T + \boldsymbol{\Gamma}^{-1}, \\
\widehat{\boldsymbol{\Lambda}}^{-1} &= \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{y}_k - \boldsymbol{H}\bar{\boldsymbol{x}}_k)(\boldsymbol{y}_k - \boldsymbol{H}\bar{\boldsymbol{x}}_k)^T + \boldsymbol{H}\boldsymbol{\Gamma}^{-1}\boldsymbol{H}^T.
\end{aligned}$$

The step that computes the posterior distribution is called the E step, and the step that estimates the hyperparameters is called the M step. The EM algorithm updates

the posterior distribution and the values of hyperparameters by repeating the E and M steps. As a result of this recursive procedure, the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\Lambda})$ is increased. A proof may be found, for example, in [1].

B.5.6 Hyperparameter Update Equations when $\boldsymbol{\Phi} = \alpha \boldsymbol{I}$ and $\boldsymbol{\Lambda} = \beta \boldsymbol{I}$

In Sect. 2.10, we show that the Bayesian estimate of $\boldsymbol{x}_k$, $\bar{\boldsymbol{x}}_k$, becomes equal to the L_2-norm regularized minimum-norm solution if we use $\boldsymbol{\Phi} = \alpha \boldsymbol{I}$ and $\boldsymbol{\Lambda} = \beta \boldsymbol{I}$ in the Gaussian model. Let us derive update equations for the scalar hyperparameters α and β in this case.

The average data likelihood in this case is given by

$$\Theta(\alpha, \beta) = \frac{NK}{2} \log \alpha - \frac{\alpha}{2} E\left[\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{x}_k\right] + \frac{MK}{2} \log \beta - \frac{\beta}{2} E\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)\right].$$

Therefore, using

$$\frac{\partial}{\partial \alpha} \Theta(\alpha, \beta) = \frac{NK}{2\alpha} - \frac{1}{2} E\left[\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{x}_k\right] = 0$$

and

$$\begin{aligned} E\left[\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{x}_k\right] &= \operatorname{tr}\left[\sum_{k=1}^{K} E(\boldsymbol{x}_k \boldsymbol{x}_k^T)\right] \\ &= \operatorname{tr}\left[\sum_{k=1}^{K} \bar{\boldsymbol{x}}_k \bar{\boldsymbol{x}}_k^T + K \boldsymbol{\Gamma}^{-1}\right] \\ &= \sum_{k=1}^{K} \bar{\boldsymbol{x}}_k^T \bar{\boldsymbol{x}}_k + K \operatorname{tr}(\boldsymbol{\Gamma}^{-1}), \end{aligned}$$

we get

$$\widehat{\alpha}^{-1} = \frac{1}{NK} E\left[\sum_{k=1}^{K} \boldsymbol{x}_k^T \boldsymbol{x}_k\right] = \frac{1}{N}\left[\frac{1}{K} \sum_{k=1}^{K} \|\bar{\boldsymbol{x}}_k\|^2 + \operatorname{tr}(\boldsymbol{\Gamma}^{-1})\right] \tag{B.46}$$

for the update equation of α.

Regarding the hyperparameter β, using a similar derivation, we have

$$\widehat{\beta}^{-1} = \frac{1}{MK} E\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)\right]. \tag{B.47}$$

Here, considering the relationships

$$\begin{aligned} E\left[\sum_{k=1}^{K} (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)^T (\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{x}_k)\right] &= E\left[\sum_{k=1}^{K} \left(\boldsymbol{y}_k^T\boldsymbol{y}_k - \boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{y}_k - \boldsymbol{y}_k^T\boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_k\right)\right] \\ &= \sum_{k=1}^{K} \left(\boldsymbol{y}_k^T\boldsymbol{y}_k - E[\boldsymbol{x}_k^T]\boldsymbol{H}^T\boldsymbol{y}_k - \boldsymbol{y}_k^T\boldsymbol{H}E[\boldsymbol{x}_k] + E[\boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_k]\right) \\ &= \sum_{k=1}^{K} \left(\boldsymbol{y}_k^T\boldsymbol{y}_k - \bar{\boldsymbol{x}}_k^T\boldsymbol{H}^T\boldsymbol{y}_k - \boldsymbol{y}_k^T\boldsymbol{H}\bar{\boldsymbol{x}}_k + E[\boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_k]\right) \end{aligned} \tag{B.48}$$

and

$$\begin{aligned} E[\boldsymbol{x}_k^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_k] &= E[\mathrm{tr}(\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_k\boldsymbol{x}_k^T)] \\ &= \mathrm{tr}\left[\boldsymbol{H}^T\boldsymbol{H}E\left(\boldsymbol{x}_k\boldsymbol{x}_k^T\right)\right] \\ &= \mathrm{tr}\left[\boldsymbol{H}^T\boldsymbol{H}\left(\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^T + \boldsymbol{\Gamma}^{-1}\right)\right] \\ &= \bar{\boldsymbol{x}}_k^T\boldsymbol{H}^T\boldsymbol{H}\bar{\boldsymbol{x}}_k + \mathrm{tr}\left(\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\Gamma}^{-1}\right), \end{aligned} \tag{B.49}$$

we finally obtain

$$\widehat{\beta}^{-1} = \frac{1}{M}\left[\frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{y}_k - \boldsymbol{H}\bar{\boldsymbol{x}}_k\|^2 + \mathrm{tr}\left(\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\Gamma}^{-1}\right)\right]. \tag{B.50}$$

B.6 Variational Bayesian Inference

In the preceding sections, the unknown $\boldsymbol{x}$ is estimated based on the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$, and the hyperparameters are estimated by using the EM algorithm. In this section, we introduce a method called variational Bayesian inference, which makes it possible to derive approximate posterior distributions for hyperparameters [2].

B.6.1 Derivation of the EM Algorithm Using the Free Energy

B.6.1.1 Derivation of Posterior Distribution (E-step)

As a preparation for introducing the variational technique, we derive the EM algorithm in a different manner based on an optimization of a functional called the free energy. In this section, the hyperparameters are collectively expressed as $\boldsymbol{\theta}$. We define a functional such that,

$$\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}] = \int \mathrm{d}\boldsymbol{x} q(\boldsymbol{x})[\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log q(\boldsymbol{x})]. \tag{B.51}$$

This $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}]$ is a function of hyperparameters $\boldsymbol{\theta}$ and an arbitrary probability distribution $q(\boldsymbol{x})$. This $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}]$ is called the free energy using a terminology in statistical physics. We show, in the following, that maximizing the free energy $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}]$ with respect to $q(\boldsymbol{x})$ results in the E step, and maximizing it with respect to the hyperparameters results in the M step of the EM algorithm.

When maximizing $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}]$ with respect to $q(\boldsymbol{x})$, since $q(\boldsymbol{x})$ is a probability distribution, the constraint $\int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1$ must be imposed. Therefore, this maximization problem can be formulated such that,

$$\widehat{q}(\boldsymbol{x}) = \underset{q(\boldsymbol{x})}{\operatorname{argmax}}\ \mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}], \quad \text{subject to} \quad \int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1. \tag{B.52}$$

Such a constrained optimization problem can be solved by using the method of Lagrange multipliers, in which defining the Lagrange multiplier as γ, the Lagrangian is defined as

$$\begin{aligned}\mathbb{L}[q, \gamma] &= \mathcal{F}[q, \boldsymbol{\theta}] + \gamma \left[\int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 1\right] \\ &= \int_{-\infty}^{\infty} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x})[\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log q(\boldsymbol{x})] + \gamma \left[\int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 1\right].\end{aligned} \tag{B.53}$$

The constrained optimization problem in Eq. (B.52) is now rewritten as the unconstrained optimization problem in Eq. (B.53). The probability distribution $q(\boldsymbol{x})$ that maximizes the Lagrangian $\mathbb{L}[q, \gamma]$ is the solution of the constrained optimization problem in Eq. (B.52).

Differentiating $\mathbb{L}[q, \gamma]$ with respect to $q(\boldsymbol{x})$, and setting the derivative to zero, we have

$$\frac{\delta \mathbb{L}[q(\boldsymbol{x}), \gamma]}{\delta q(\boldsymbol{x})} = \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log q(\boldsymbol{x}) - 1 + \gamma = 0. \tag{B.54}$$

A brief explanation on the differentiation of a functional, as well as the derivation of Eq. (B.54), is presented in Sect. C.5 in the Appendix. Differentiating $\mathbb{L}[q, \gamma]$ with respect to γ gives

$$\frac{\partial \mathbb{L}[q(\boldsymbol{x}), \gamma]}{\partial \gamma} = \int_{-\infty}^{\infty} q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 1 = 0. \tag{B.55}$$

Thus, using Eq. (B.54), we have

$$\widehat{q}(\boldsymbol{x}) = e^{\gamma - 1} p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}), \tag{B.56}$$

and using Eq. (B.55), we also have

$$\int_{-\infty}^{\infty} q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = e^{\gamma - 1} \int_{-\infty}^{\infty} p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} = e^{\gamma - 1} p(\boldsymbol{y}|\boldsymbol{\theta}) = 1. \tag{B.57}$$

We thereby get

$$e^{\gamma - 1} = \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta})}. \tag{B.58}$$

Substitution of Eq. (B.58) into (B.56) results in the relationship

$$\widehat{q}(\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{\theta})} = p(\boldsymbol{x}|\boldsymbol{y}). \tag{B.59}$$

The above equation shows that the probability distribution that maximizes the free energy $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}]$ is the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$.

Note that to derive the posterior distribution, we do not explicitly use Bayes' rule. Instead, we use the optimization of the functional called the free energy. This idea is further extended in variational Bayesian inference in the following sections to derive an approximate posterior distribution in a more complicated situation.

B.6.1.2 Derivation of M-step

We next maximize the free energy with respect to the hyperparameter $\boldsymbol{\theta}$. Once $\mathcal{F}[q, \boldsymbol{\theta}]$ is maximized with respect to $q(\boldsymbol{x})$, the free energy is written as

$$\begin{aligned} \mathcal{F}[p(\boldsymbol{x}|\boldsymbol{y}), \boldsymbol{\theta}] &= \int \mathrm{d}\boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) [\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log p(\boldsymbol{x}|\boldsymbol{y})] \\ &= \mathcal{Theta}(\boldsymbol{\theta}) + \mathcal{H}[p(\boldsymbol{x}|\boldsymbol{y})], \end{aligned} \tag{B.60}$$

where

$$\mathcal{Theta}(\boldsymbol{\theta}) = \int \mathrm{d}\boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}). \tag{B.61}$$

This $\mathcal{Theta}(\boldsymbol{\theta})$ is equal to the average data likelihood, and

$$\mathcal{H}[p(\boldsymbol{x}|\boldsymbol{y})] = -\int \mathrm{d}\boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) \log p(\boldsymbol{x}|\boldsymbol{y}) \tag{B.62}$$

is the entropy of the posterior distribution. Since the entropy does not depend on $\boldsymbol{\theta}$, maximizing $\mathcal{F}[p(\boldsymbol{x}|\boldsymbol{y}), \boldsymbol{\theta}]$ with respect to $\boldsymbol{\theta}$ is equivalent to maximizing $\Theta(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Namely, the maximization of the free energy with respect to the hyperparameters results in the M step of the EM algorithm.

Note also that the free energy after the maximization with respect to $q(\boldsymbol{x})$ is equal to the marginal likelihood, $\log p(\boldsymbol{y}|\boldsymbol{\theta})$. This can be seen by rewriting Eq. (B.60) such that

$$\begin{aligned}\mathcal{F}[p(\boldsymbol{x}|\boldsymbol{y}), \boldsymbol{\theta}] &= \int \mathrm{d}\boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) \left[\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log p(\boldsymbol{x}|\boldsymbol{y})\right] \\ &= \int \mathrm{d}\boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) \log p(\boldsymbol{y}|\boldsymbol{\theta}) = \log p(\boldsymbol{y}|\boldsymbol{\theta}). \end{aligned} \tag{B.63}$$

This relationship is used to derive the expressions of the marginal likelihood for the Bayesian factor analysis in Chap. 5.

For an arbitrary probability distribution $q(\boldsymbol{x})$, the relationship between the free energy and the marginal likelihood is expressed as

$$\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}] = \log p(\boldsymbol{y}|\boldsymbol{\theta}) - \mathcal{K}_L \left[q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y})\right], \tag{B.64}$$

where $\mathcal{K}_L \left[q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y})\right]$ is the Kullback-Leibler (KL) distance defined in

$$\mathcal{K}_L \left[q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y})\right] = \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})} \mathrm{d}\boldsymbol{x}. \tag{B.65}$$

The KL distance represents a distance between the true posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ and the arbitrary probability distribution $q(\boldsymbol{x})$. It always has a nonnegative value, and is equal to zero when the two distributions are identical. Hence, for an arbitrary $q(\boldsymbol{x})$, the inequality $\mathcal{F}[q(\boldsymbol{x}), \boldsymbol{\theta}] \leq \log p(\boldsymbol{y}|\boldsymbol{\theta})$ holds, and the free energy forms a lower-bound of the marginal likelihood.

B.6.2 Variational Bayesian EM Algorithm

In Bayesian inference, to estimate the unknown parameter $\boldsymbol{x}$, we must first derive the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$, assuming the existence of an appropriate prior distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$. We then obtain an optimum estimate of the unknown $\boldsymbol{x}$ based on the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. When the hyperparameter $\boldsymbol{\theta}$ is unknown, a truely Bayesian approach is to first derive the joint posterior distribution $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$, and to estimate $\boldsymbol{x}$ and $\boldsymbol{\theta}$ simultaneously based on this joint posterior distribution. To derive the joint posterior $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$, we can use Bayes' rule, and obtain,

$$p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{B.66}$$

In order to compute the posterior distribution $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$, we must compute the integral

$$Z = \iint_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{x}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} d\boldsymbol{\theta}.$$

However, in general, this integral Z does not have a closed-form solution, and it is hard to compute the posterior distribution $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$ using Bayes' rule, i.e., Eq. (B.66).

In Sect. B.6.1, the posterior distribution is obtained using the optimization of the factional called the free energy, and not using Bayes' rule. We here use this variational technique to obtain the posterior distribution. The free energy is expressed as

$$\mathcal{F}[q(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{\theta}] = \int d\boldsymbol{\theta} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x}, \boldsymbol{\theta}) [\log p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) - \log q(\boldsymbol{x}, \boldsymbol{\theta})]. \tag{B.67}$$

We use an approximation that the joint posterior distribution is factorized, i.e.,

$$q(\boldsymbol{x}, \boldsymbol{\theta}) = q(\boldsymbol{x}) q(\boldsymbol{\theta}), \tag{B.68}$$

which is called the variational approximation. Using this approximation, the free energy is given by

$$\mathcal{F}[q(\boldsymbol{x}), q(\boldsymbol{\theta}), \boldsymbol{\theta}] = \int d\boldsymbol{\theta} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x}) q(\boldsymbol{\theta}) [\log p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) - \log q(\boldsymbol{x}) - \log q(\boldsymbol{\theta})]. \tag{B.69}$$

The best estimate of the posterior distributions $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{\theta}|\boldsymbol{y})$ are derived by jointly maximizing the free energy in Eq. (B.69). That is, the estimated posterior distributions, $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ and $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$, are given by

$$\begin{aligned} \widehat{p}(\boldsymbol{x}|\boldsymbol{y}), \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y}) &= \underset{q(\boldsymbol{x}), q(\boldsymbol{\theta})}{\operatorname{argmax}} \mathcal{F}[q(\boldsymbol{x}), q(\boldsymbol{\theta}), \boldsymbol{\theta}], \\ \text{subject to} \quad \int_{-\infty}^{\infty} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x}) &= 1, \text{ and } \quad \int_{-\infty}^{\infty} d\boldsymbol{\theta} q(\boldsymbol{\theta}) = 1. \end{aligned} \tag{B.70}$$

Therefore, according to Eq. (B.64), the estimated posterior distributions $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ and $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ satisfy the relationship:

$$\begin{aligned} \widehat{p}(\boldsymbol{x}|\boldsymbol{y}), \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y}) &= \underset{q(\boldsymbol{x}), q(\boldsymbol{\theta})}{\operatorname{argmin}} \mathcal{K}_L \left[p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) || q(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \right], \\ \text{where} \quad q(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) &= q(\boldsymbol{x}|\boldsymbol{y}) q(\boldsymbol{\theta}|\boldsymbol{y}). \end{aligned} \tag{B.71}$$

Namely, the estimated posterior distributions $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ and $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ obtained by maximizing the free energy jointly minimizes the KL distance between the true and approximated posterior distributions. In other words, the estimated joint posterior $\widehat{p}(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) = \widehat{p}(\boldsymbol{x}|\boldsymbol{y}) \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ is the "best" estimate in the sense that it minimizes the KL distance from the true posterior distribution.

Let us first derive $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$. Using arguments similar to those in Sect. B.6.1, defining the Lagrangian as

$$\mathbb{L}[q(\boldsymbol{x})] = \int \mathrm{d}\boldsymbol{x}d\boldsymbol{\theta}q(\boldsymbol{x})q(\boldsymbol{\theta})[\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) - \log q(\boldsymbol{x}) - \log q(\boldsymbol{\theta})] + \gamma\left[\int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 1\right], \tag{B.72}$$

differentiating it with respect to $q(\boldsymbol{x})$, and setting the derivative to zero, we have

$$\int d\boldsymbol{\theta}q(\boldsymbol{\theta})\left[\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) - \log q(\boldsymbol{x})\right] + \mathcal{C} = 0, \tag{B.73}$$

where

$$\mathcal{C} = -\int d\boldsymbol{\theta}q(\boldsymbol{\theta})\log q(\boldsymbol{\theta}) - 1 + \gamma.$$

Neglecting $\mathcal{C}$, we obtain

$$\log \widehat{p}(\boldsymbol{x}|\boldsymbol{y}) = \int d\boldsymbol{\theta}q(\boldsymbol{\theta})\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\left[\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta})\right], \tag{B.74}$$

where $E_{\boldsymbol{\theta}}[\,\cdot\,]$ indicates computing the mean with respect to the posterior distribution $q(\boldsymbol{\theta})$. Using exactly the same derivation, we obtain the relationship,

$$\log \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y}) = \int \mathrm{d}\boldsymbol{x}q(\boldsymbol{x})\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = E_{\boldsymbol{x}}\left[\log p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta})\right], \tag{B.75}$$

where $E_{\boldsymbol{x}}[\,\cdot\,]$ indicates computing the mean with respect to the posterior distribution $q(\boldsymbol{x})$.

Equations (B.74) and (B.75) indicate that, to compute the posterior distribution $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$, we need $q(\boldsymbol{\theta})$, namely $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$, and to compute the posterior distribution $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$, we need $q(\boldsymbol{x})$, namely $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$. Therefore, in variational Bayesian inference, we need an EM-like recursive procedure. We update $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ and $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ by repeatedly applying Eqs. (B.74) and (B.75). When computing $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ using Eq. (B.74), we use $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ obtained in the preceding step. When computing $\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ using Eq. (B.75), we use $\widehat{p}(\boldsymbol{x}|\boldsymbol{y})$ obtained in the preceding step. This recursive algorithm is referred to as the variational Bayesian EM (VBEM) algorithm.

References

1. C. M. Bishop, *Pattern recognition and machine learning*. Springer, New York, 2006.
2. H. T. Attius, "A variational Bayesian framework for graphical models," in *Advances in Neural information processing*, pp. 209–215, MIT Press, 2000.

Appendix C
Supplementary Mathematical Arguments

C.1 Multi-dimensional Gaussian Distribution

Let us define a column vector of N random variables as $\boldsymbol{x}$. The multi-dimensional Gaussian distribution for $\boldsymbol{x}$ is expressed as

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \tag{C.1}$$

Here $\boldsymbol{\mu}$ is the mean of $\boldsymbol{x}$ defined as $\boldsymbol{\mu} = E(\boldsymbol{x})$ and $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{x}$ defined as $\boldsymbol{\Sigma} = E\left[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\right]$ where $E(\cdot)$ is the expectation operator. Also, $|\boldsymbol{\Sigma}|$ indicates the determinant of $\boldsymbol{\Sigma}$. The Gaussian distribution in Eq. (C.1) is often written as

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

When the random variable $\boldsymbol{x}$ follows a Gaussian distribution with mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$, the expression

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{C.2}$$

is often used.

Two vector random variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have the linear relationship,

$$\boldsymbol{x}_2 = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{c},$$

where $\boldsymbol{A}$ is a matrix of deterministic variables, and $\boldsymbol{c}$ is a vector of deterministic variables. Then, if we have

$$\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

we also have

$$\boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{x}_2|\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{c}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T). \tag{C.3}$$

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9

Namely, $\boldsymbol{x}_2$ obeys the Gaussian distribution with a mean equal to $\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{c}$ and a covariance matrix equal to $\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T$.

Bayesian inference quite often uses the precision, instead of the variance. Let us define the precision matrix $\boldsymbol{\Lambda}$ that corresponds to the covariance natrix $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. When the precision $\boldsymbol{\Lambda}$ is used, the notation in Eq. (C.2) is used such that

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}). \tag{C.4}$$

Here, since we maintain the notational convenience

$$\mathcal{N}(\text{random variable}|\text{mean, covariance matrix}),$$

we must use $\boldsymbol{\Lambda}^{-1}$ in this notation. Using the precision matrix, $\boldsymbol{\Lambda}$, the explicit form of the Gaussian distribution is given by

$$p(\boldsymbol{x}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{N/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \tag{C.5}$$

Let us compute the entropy when the probability distribution is Gaussian. The definition of the entropy for a continuous random variable is

$$\mathcal{H}[p(\boldsymbol{x})] = -\int p(\boldsymbol{x}) \log p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = -E\left[\log p(\boldsymbol{x})\right]. \tag{C.6}$$

Substituting the probability distribution in Eq. (C.1) into the equation above, we get

$$\begin{aligned}\mathcal{H}[p(\boldsymbol{x})] = &- E\left[\log\left[\frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}}\right]\right] \\ &+ \frac{1}{2}E\left[(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right],\end{aligned} \tag{C.7}$$

where constant terms are omitted. In the equation above, the first term on the right-hand side is equal to $\frac{1}{2}\log|\boldsymbol{\Sigma}|$. The second term is equal to

$$\begin{aligned}\frac{1}{2}E\left[(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] &= \frac{1}{2}E\left[\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\right]\right] \\ &= \frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}E\left[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\right]\right] \\ &= \frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\right] = \frac{N}{2}.\end{aligned} \tag{C.8}$$

Therefore, omitting constant terms, the entropy is expressed as

$$\mathcal{H}[p(\boldsymbol{x})] = \mathcal{H}(\boldsymbol{x}) = \frac{1}{2}\log|\boldsymbol{\Sigma}|. \tag{C.9}$$

Note that we use a simplified notation $\mathcal{H}(\boldsymbol{x})$ in this book to indicate the entropy of the probability distribution $p(\boldsymbol{x})$, which should formally be written as $\mathcal{H}[p(\boldsymbol{x})]$[3]

C.2 Complex Gaussian Distribution

We derive the probability distribution when the random variable is a complex-valued Gaussian. The arguments in this section follows those of Neeser and Massey [1]. Let us define a column vector of N complex random variables as $\boldsymbol{z}$, and define $\boldsymbol{z} = \boldsymbol{x} + i\boldsymbol{y}$ where $\boldsymbol{x} = \Re(\boldsymbol{z})$, $\boldsymbol{y} = \Im(\boldsymbol{z})$ and $\boldsymbol{x}$ and $\boldsymbol{y}$ are real-valued N-dimensional Gaussian random vectors. Here, we assume that $E(\boldsymbol{z}) = 0$, and as a result, $E(\boldsymbol{x}) = 0$ and $E(\boldsymbol{y}) = 0$. The covariance matrix of $\boldsymbol{z}$ is defined as $\boldsymbol{\Sigma}_{zz} = E\left(\boldsymbol{z}\boldsymbol{z}^H\right)$.

We next define a $(2N \times 1)$ vector $\boldsymbol{\phi}$ such that $\boldsymbol{\phi} = [\boldsymbol{x}^T, \boldsymbol{y}^T]^T$. The covariance matrix of $\boldsymbol{\phi}$, $\boldsymbol{\Sigma}_{\phi\phi}$, is given by

$$\boldsymbol{\Sigma}_{\phi\phi} = E\left[\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\left[\boldsymbol{x}^T, \boldsymbol{y}^T\right]\right] = \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{yx}^T\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}, \tag{C.10}$$

where $\boldsymbol{\Sigma}_{xx} = E(\boldsymbol{x}\boldsymbol{x}^T)$, $\boldsymbol{\Sigma}_{yx} = E(\boldsymbol{y}\boldsymbol{x}^T)$, and $\boldsymbol{\Sigma}_{yy} = E(\boldsymbol{y}\boldsymbol{y}^T)$. We assume that the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{\phi}) = \frac{1}{(2\pi)^N |\boldsymbol{\Sigma}_{\phi\phi}|^{1/2}} \exp\left[-\frac{1}{2}\boldsymbol{\phi}^T \boldsymbol{\Sigma}_{\phi\phi}^{-1}\boldsymbol{\phi}\right]. \tag{C.11}$$

In this section, we show that this joint distribution is equal to

$$p(\boldsymbol{z}) = \frac{1}{\pi^N |\boldsymbol{\Sigma}_{zz}|} \exp\left[-\boldsymbol{z}^H \boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{z}\right], \tag{C.12}$$

which is called the complex Gaussian distribution.

To show this equality, we assume a property, called "proper", on the complex random variable $\boldsymbol{z}$. The complex random variable $\boldsymbol{z}$ is proper if its pseudo covariance $\bar{\boldsymbol{\Sigma}}_{zz}$ is equal to zero. The pseudo covariance is defined such that $\bar{\boldsymbol{\Sigma}}_{zz} = E\left(\boldsymbol{z}\boldsymbol{z}^T\right)$, which is written as

$$\bar{\boldsymbol{\Sigma}}_{zz} = E\left[(\boldsymbol{x} + i\boldsymbol{y})(\boldsymbol{x} + i\boldsymbol{y})^T\right] = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{yy} + i\left(\boldsymbol{\Sigma}_{yx} + \boldsymbol{\Sigma}_{yx}^T\right).$$

Therefore, $\bar{\boldsymbol{\Sigma}}_{zz} = \boldsymbol{0}$ is equivalent to the relationships

$$\boldsymbol{\Sigma}_{xx} = \boldsymbol{\Sigma}_{yy} \quad \text{and} \quad -\boldsymbol{\Sigma}_{yx} = \boldsymbol{\Sigma}_{yx}^T. \tag{C.13}$$

[3] The notation $\mathcal{H}(\boldsymbol{x})$ may look as if the entropy $\mathcal{H}(\boldsymbol{x})$ is a function of $\boldsymbol{x}$, but the entropy is a functional of the probability distribution $p(\boldsymbol{x})$, not a function of $\boldsymbol{x}$.

Note that proper complex Gaussian random vectors with zero mean are called circular. In this section, we assume that the circularity holds for $\boldsymbol{z}$. Under the assumption that $\boldsymbol{z}$ is a proper complex random vector, we derive

$$\boldsymbol{\Sigma}_{zz} = E\left[(\boldsymbol{x}+i\boldsymbol{y})(\boldsymbol{x}-i\boldsymbol{y})^T\right] = 2\left(\boldsymbol{\Sigma}_{xx}+i\boldsymbol{\Sigma}_{yx}\right), \tag{C.14}$$

and $\boldsymbol{\Sigma}_{zz}^{-1}$ is found to be[4]

$$\boldsymbol{\Sigma}_{zz}^{-1} = \frac{1}{2}\boldsymbol{\Delta}^{-1}\left(\boldsymbol{I} - i\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\right), \tag{C.15}$$

where

$$\boldsymbol{\Delta} = \boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}. \tag{C.16}$$

We first derive the relationship between the determinants $|\boldsymbol{\Sigma}_{zz}|$ and $|\boldsymbol{\Sigma}_{\phi\phi}|$. Using Eq. (C.14), we have

$$\boldsymbol{\Sigma}_{zz}^T = 2\left(\boldsymbol{\Sigma}_{xx}+i\boldsymbol{\Sigma}_{yx}^T\right) = 2\left(\boldsymbol{\Sigma}_{xx}-i\boldsymbol{\Sigma}_{yx}\right),$$

and $\boldsymbol{\Sigma}_{zz}^{-1}$ is expressed as

$$\boldsymbol{\Sigma}_{zz}^{-1} = \frac{1}{2}\boldsymbol{\Delta}^{-1}\left(\boldsymbol{\Sigma}_{xx}-i\boldsymbol{\Sigma}_{yx}\right)\boldsymbol{\Sigma}_{xx}^{-1} = \frac{1}{4}\boldsymbol{\Delta}^{-1}\boldsymbol{\Sigma}_{zz}^T\boldsymbol{\Sigma}_{xx}^{-1}.$$

Taking the determinant of both sides of the equation above, we get

$$|\boldsymbol{\Sigma}_{zz}|^{-1} = \frac{1}{4^N}|\boldsymbol{\Delta}|^{-1}\,|\boldsymbol{\Sigma}_{zz}|\,|\boldsymbol{\Sigma}_{xx}|^{-1},$$

and finally,

$$|\boldsymbol{\Sigma}_{zz}| = 2^N\sqrt{|\boldsymbol{\Sigma}_{xx}||\boldsymbol{\Delta}|}. \tag{C.17}$$

On the other hand, using the determinant identity in Eq. (C.94), $|\boldsymbol{\Sigma}_{\phi\phi}|$ is expressed as

$$|\boldsymbol{\Sigma}_{\phi\phi}| = \begin{vmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{yx}^T\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{vmatrix} = |\boldsymbol{\Sigma}_{xx}||\boldsymbol{\Sigma}_{yy}-\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}^T| = |\boldsymbol{\Sigma}_{xx}||\boldsymbol{\Delta}|, \tag{C.18}$$

where the relationships in Eq. (C.13) are used to obtain the right-most expression. Comparing Eqs. (C.17) and (C.18), we finally obtain the relationship,

$$|\boldsymbol{\Sigma}_{zz}| = 2^N\sqrt{|\boldsymbol{\Sigma}_{\phi\phi}|}. \tag{C.19}$$

[4] It is easy to see that $\boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zz} = \boldsymbol{I}$ holds.

We next show the equality of the two quadratic forms, i.e.,

$$z^H \boldsymbol{\Sigma}_{zz}^{-1} z = \frac{1}{2} \boldsymbol{\phi}^T \boldsymbol{\Sigma}_{\phi\phi}^{-1} \boldsymbol{\phi}.$$

To show this relationship, we use the matrix inversion formula in Eq. (C.93), and rewrite $\boldsymbol{\Sigma}_{\phi\phi}^{-1}$ such that[5]

$$\boldsymbol{\Sigma}_{\phi\phi}^{-1} = \begin{bmatrix} \boldsymbol{\Delta}^{-1} & \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Delta}^{-1} \\ -\boldsymbol{\Delta}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} & \boldsymbol{\Delta}^{-1} \end{bmatrix}, \tag{C.20}$$

where Δ is defined in Eq. (C.16). Let us define:

$$\boldsymbol{M}_c = \boldsymbol{\Delta}^{-1}, \tag{C.21}$$

$$\boldsymbol{M}_s = -\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Delta}^{-1}. \tag{C.22}$$

Then, $\boldsymbol{M}_c$ is symmetric because $\boldsymbol{\Delta}$ is symmetric. Using

$$\boldsymbol{\Delta} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} = \left(\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \right) \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} = \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Delta},$$

we also get the relationship:

$$\begin{aligned} \boldsymbol{M}_s &= -\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Delta}^{-1} = -\boldsymbol{\Delta}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Delta} \boldsymbol{\Delta}^{-1} = -\boldsymbol{\Delta}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \\ &= -\left(\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx}^T \boldsymbol{\Delta}^{-1} \right)^T = \left(\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Delta}^{-1} \right)^T = -\boldsymbol{M}_s^T, \end{aligned} \tag{C.23}$$

where $\boldsymbol{\Sigma}_{yx}^T = -\boldsymbol{\Sigma}_{yx}$ is used. Thus, $\boldsymbol{M}_s$ is skew-symmetric.

We define a matrix $\boldsymbol{M}$ as

$$\boldsymbol{M} = (\boldsymbol{M}_c + i\boldsymbol{M}_s), \tag{C.24}$$

and compute the quadratic form $z^H \boldsymbol{M} z$. Since we have the relationship,

$$\begin{aligned} \left(z^H \boldsymbol{M} z \right)^H &= z^H \boldsymbol{M}^H z = z^H (\boldsymbol{M}_c + i\boldsymbol{M}_s)^H z \\ &= z^H \left(\boldsymbol{M}_c^T - i\boldsymbol{M}_s^T \right) z = z^H (\boldsymbol{M}_c + i\boldsymbol{M}_s) z = z^H \boldsymbol{M} z, \end{aligned} \tag{C.25}$$

[5] In Eq. (C.20), the (2, 2)th element is $\boldsymbol{\Delta}^{-1}$, which can be shown as follows. According to Eq. (C.93), this element is equal to

$$\boldsymbol{\Sigma}_{yy}^{-1} + \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx} \left(\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx} \right) \boldsymbol{\Sigma}_{yx}^T \boldsymbol{\Sigma}_{yy}^{-1}.$$

Using Eqs. (C.13) and (C.91), this is equal to $(\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{yx})^{-1} = \boldsymbol{\Delta}^{-1}$.

the quadratic form $z^H \boldsymbol{M} z$ is real-valued. Therefore, we have

$$\begin{aligned} z^H \boldsymbol{M} z &= \Re\left[z^H \boldsymbol{M} z\right] = \Re\left[(\boldsymbol{x}^T - i\boldsymbol{y}^T)(\boldsymbol{M}_c + i\boldsymbol{M}_s)(\boldsymbol{x} + i\boldsymbol{y})\right] \\ &= \boldsymbol{x}^T \boldsymbol{M}_c \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{M}_s \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{M}_s \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{M}_c \boldsymbol{y} \\ &= [\boldsymbol{x}^T, \boldsymbol{y}^T] \begin{bmatrix} \boldsymbol{M}_c & -\boldsymbol{M}_s \\ \boldsymbol{M}_s & \boldsymbol{M}_c \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{\phi}^T \boldsymbol{\Sigma}_{\phi\phi}^{-1} \boldsymbol{\phi}. \end{aligned} \tag{C.26}$$

We now compute $\boldsymbol{M}$, such that

$$\boldsymbol{M} = \boldsymbol{\Delta}^{-1} - i\boldsymbol{\Delta}^{-1} \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} = \boldsymbol{\Delta}^{-1} \left(\boldsymbol{I} - i\boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1}\right).$$

By comparing the above $\boldsymbol{M}$ with Eq. (C.15) we can see that $\boldsymbol{\Sigma}_{zz}^{-1} = \frac{1}{2}\boldsymbol{M}$, and therefore, from Eq. (C.26), we can prove the relationship

$$z^H \boldsymbol{\Sigma}_{zz}^{-1} z = \frac{1}{2} \boldsymbol{\phi}^T \boldsymbol{\Sigma}_{\phi\phi}^{-1} \boldsymbol{\phi}. \tag{C.27}$$

On the basis of Eqs. (C.17) and (C.27), it is now clear that the real-valued joint Gaussian distribution in Eq. (C.11) and the complex Gaussian distribution in Eq. (C.12) are equivalent. Using exactly the same derivation for Eq. (C.9) and ignoring the constants, the entropy for the complex-valued Gaussian is obtained as

$$\mathcal{H}(z) = \log |\boldsymbol{\Sigma}_{zz}|. \tag{C.28}$$

C.3 Canonical Correlation and Mutual Information

C.3.1 Canonical Correlation

This appendix provides a concise explanation on canonical correlation. Let us define real-valued random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ such that

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}, \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{bmatrix}, \tag{C.29}$$

and consider computing the correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$. One way is to compute correlation coefficients between all combinations of (x_i, y_j). However, this gives total $p \times q$ correlation coefficients and the interpretation of these results may not be easy. The canonical correlation method first projects column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ onto the

direction of $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $\widehat{x} = \boldsymbol{a}^T\boldsymbol{x}$ and $\widehat{y} = \boldsymbol{b}^T\boldsymbol{y}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are real-valued vectors. It then computes the correlation between $\widehat{x}$ and $\widehat{y}$. This correlation depends on the choices of $\boldsymbol{a}$ and $\boldsymbol{b}$. The maximum correlation between $\widehat{x}$ and $\widehat{y}$ is considered to represent the correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$, and this maximum correlation is called the canonical correlation.

The correlation between $\widehat{x}$ and $\widehat{y}$, ρ, is expressed as

$$\begin{aligned}\rho &= \frac{E(\widehat{x}\widehat{y})}{\sqrt{E(\widehat{x}^2)}\sqrt{E(\widehat{y}^2)}} \\ &= \frac{\boldsymbol{a}^T E(\boldsymbol{x}\boldsymbol{y}^T)\boldsymbol{b}}{\sqrt{\left[\boldsymbol{a}^T E(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{a}\right]\left[\boldsymbol{b}^T E(\boldsymbol{y}\boldsymbol{y}^T)\boldsymbol{b}\right]}} = \frac{\boldsymbol{a}^T\boldsymbol{\Sigma}_{xy}\boldsymbol{b}}{\sqrt{\left[\boldsymbol{a}^T\boldsymbol{\Sigma}_{xx}\boldsymbol{a}\right]\left[\boldsymbol{b}^T\boldsymbol{\Sigma}_{yy}\boldsymbol{b}\right]}}.\end{aligned} \tag{C.30}$$

where $\boldsymbol{\Sigma}_{xy} = E(\boldsymbol{x}\boldsymbol{y}^T)$, $\boldsymbol{\Sigma}_{xx} = E(\boldsymbol{x}\boldsymbol{x}^T)$, and $\boldsymbol{\Sigma}_{yy} = E(\boldsymbol{y}\boldsymbol{y}^T)$ and the notation $E(\cdot)$ indicates the expectation. Here, $\boldsymbol{\Sigma}_{xx}$ and $\boldsymbol{\Sigma}_{yy}$ are symmetric matrices, and the relationship $\boldsymbol{\Sigma}_{xy}^T = \boldsymbol{\Sigma}_{yx}$ holds. The canonical correlation ρ_c is obtained by solving the following maximization problem:

$$\rho_c = \max_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{a}^T\boldsymbol{\Sigma}_{xy}\boldsymbol{b} \quad \text{subject to} \quad \boldsymbol{a}^T\boldsymbol{\Sigma}_{xx}\boldsymbol{a} = 1 \quad \text{and} \quad \boldsymbol{b}^T\boldsymbol{\Sigma}_{yy}\boldsymbol{b} = 1. \tag{C.31}$$

To solve this maximization problem, we first compute whitened vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, such that

$$\boldsymbol{\alpha} = \boldsymbol{\Sigma}_{xx}^{1/2}\boldsymbol{a}, \tag{C.32}$$

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}_{yy}^{1/2}\boldsymbol{b}. \tag{C.33}$$

Using these vectors, the optimization problem in Eq. (C.31) is rewritten as

$$\rho_c = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\alpha}^T\boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1/2}\boldsymbol{\beta} \quad \text{subject to} \quad \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1 \quad \text{and} \quad \boldsymbol{\beta}^T\boldsymbol{\beta} = 1. \tag{C.34}$$

The constrained maximization problem in Eq. (C.34) can be solved using the Lagrange multiplier method. Denoting the Lagrange multipliers ν_1 and ν_2, the Lagrangian is defined as

$$\mathbb{L} = \boldsymbol{\alpha}^T\boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1/2}\boldsymbol{\beta} - \frac{\nu_1}{2}\left(\boldsymbol{\alpha}^T\boldsymbol{\alpha} - 1\right) - \frac{\nu_2}{2}\left(\boldsymbol{\beta}^T\boldsymbol{\beta} - 1\right). \tag{C.35}$$

The optimum $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are obtained by maximizing $\mathbb{L}$ with no constraints. The derivatives of $\mathbb{L}$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are expressed as

$$\frac{\partial \mathbb{L}}{\partial \boldsymbol{\alpha}} = \boldsymbol{\Pi}_{xy}\boldsymbol{\beta} - \nu_1 \boldsymbol{\alpha} = 0, \tag{C.36}$$

$$\frac{\partial \mathbb{L}}{\partial \boldsymbol{\beta}} = \boldsymbol{\Pi}_{xy}^T\boldsymbol{\alpha} - \nu_2 \boldsymbol{\beta} = 0, \tag{C.37}$$

where $\boldsymbol{\Pi}_{xy} = \boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1/2}$. Using the constraint $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$, and left multiplying $\boldsymbol{\alpha}^T$ to Eq. (C.36) gives

$$\boldsymbol{\alpha}^T\boldsymbol{\Pi}_{xy}\boldsymbol{\beta} = \nu_1.$$

Using the constraint $\boldsymbol{\beta}^T\boldsymbol{\beta} = 1$, and left multiplying $\boldsymbol{\beta}^T$ to Eq. (C.37) gives

$$\boldsymbol{\beta}^T\boldsymbol{\Pi}_{xy}^T\boldsymbol{\alpha} = \nu_2.$$

Since the left-hand sides of the two equations above are equal, we have $\nu_1 = \nu_2 = \rho_c$.

Also, using Eq. (C.37), we get

$$\boldsymbol{\beta} = \frac{1}{\rho_c}\boldsymbol{\Pi}_{xy}^T\boldsymbol{\alpha},$$

and substituting the equation above into Eq. (C.36), we get

$$\left(\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T\right)\boldsymbol{\alpha} = \rho_c^2\boldsymbol{\alpha}. \tag{C.38}$$

This equation indicates that the squared canonical correlation ρ_c^2 is obtained as the eigenvalues of a matrix $\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T$, and the corresponding eigenvector is the solution for the vector $\boldsymbol{\alpha}$. Note that $\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T$ is a real symmetric matrix, the eigenvectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are real-valued, and therefore $\boldsymbol{a}$ and $\boldsymbol{b}$ are real-valued because $\boldsymbol{\Sigma}_{xx}$ and $\boldsymbol{\Sigma}_{yy}$ are real-valued matrices.

Denoting the eigenvalues in Eq. (C.38) as μ_j where $j = 1, \ldots, d$ and $d = \min\{p, q\}$, the canonical correlation between the two sets of random variables $x_1, \ldots, x_p$ and $y_1, \ldots, y_q$ is obtained as the largest eigenvalue μ_1, which is the best overall measure of the association between $\boldsymbol{x}$ and $\boldsymbol{y}$. However, other eigenvalues may provide complementary information on the linear relationship between those two sets of random variables. The mutual information described in Sect. C.3.2 is a measure that can take all the eigenvalues into account.

Also, it is worth mentioning that the matrices,

$$\boldsymbol{\Pi}_{xy}\boldsymbol{\Pi}_{xy}^T = \boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T\boldsymbol{\Sigma}_{xx}^{-1/2}$$

and

$$\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T \tag{C.39}$$

have the same eigenvalues, according to Property No. 9 in Sect. C.8. Thus, the canonical squared correlation can be obtained as the largest eigenvalue of the matrix $\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{T}$.

C.3.2 Mutual Information

Next we introduce mutual information, which has a close relationship with the canonical correlation under the Gaussianity assumption. We also assume real-valued random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and their probability distribution as $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$. The entropy is defined for $\boldsymbol{x}$ and $\boldsymbol{y}$ such that,

$$\mathcal{H}(\boldsymbol{x}) = -\int p(\boldsymbol{x}) \log p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{C.40}$$

$$\mathcal{H}(\boldsymbol{y}) = -\int p(\boldsymbol{y}) \log p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{C.41}$$

The entropy is a measure of uncertainty; $\mathcal{H}(\boldsymbol{x})$ represents the uncertainty when $\boldsymbol{x}$ is unknown and $\mathcal{H}(\boldsymbol{y})$ represents the uncertainty when $\boldsymbol{y}$ is unknown. The joint entropy is defined as

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = -\iint p(\boldsymbol{x}, \boldsymbol{y}) \log p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}, \tag{C.42}$$

which represents the uncertainty when both $\boldsymbol{x}$ and $\boldsymbol{y}$ are unknown. When $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent, we have the relationship

$$\begin{aligned} \mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) &= -\iint p(\boldsymbol{x}, \boldsymbol{y}) \log p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= -\iint p(\boldsymbol{x}) p(\boldsymbol{y}) \left(\log p(\boldsymbol{x}) + \log p(\boldsymbol{y})\right) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} = \mathcal{H}(\boldsymbol{x}) + \mathcal{H}(\boldsymbol{y}). \end{aligned} \tag{C.43}$$

The conditional entropy is defined as

$$\mathcal{H}(\boldsymbol{x}|\boldsymbol{y}) = -\iint p(\boldsymbol{x}, \boldsymbol{y}) \log p(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}, \tag{C.44}$$

which represents the uncertainty when $\boldsymbol{x}$ is unknown, once $\boldsymbol{y}$ is given. We then have the relationship,

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{H}(\boldsymbol{x}|\boldsymbol{y}) + \mathcal{H}(\boldsymbol{y}). \tag{C.45}$$

The above indicates that the uncertainty when both $\boldsymbol{x}$ and $\boldsymbol{y}$ are unknown is equal to the uncertainty on $\boldsymbol{x}$ when $\boldsymbol{y}$ is given plus the uncertainty when $\boldsymbol{y}$ is unknown.

The mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{H}(\boldsymbol{x}) + \mathcal{H}(\boldsymbol{y}) - \mathcal{H}(\boldsymbol{x}, \boldsymbol{y}). \tag{C.46}$$

When $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent, we have $\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = 0$ according to Eq. (C.43). Let us define $\boldsymbol{z}$ as $\boldsymbol{z} = [\boldsymbol{x}^T, \boldsymbol{y}^T]^T$. Assuming the Gaussian processes for $\boldsymbol{x}$ and $\boldsymbol{y}$, the mutual information is expressed as

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{H}(\boldsymbol{x}) + \mathcal{H}(\boldsymbol{y}) - \mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) \\ &= \frac{1}{2}\log|\boldsymbol{\Sigma}_{xx}| + \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{yy}\right| - \frac{1}{2}\log|\boldsymbol{\Sigma}_{zz}|.\end{aligned} \tag{C.47}$$

Here, $|\boldsymbol{\Sigma}_{zz}|$ is rewritten as

$$|\boldsymbol{\Sigma}_{zz}| = \begin{vmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{xy}^T & \boldsymbol{\Sigma}_{yy} \end{vmatrix} = \left|\boldsymbol{\Sigma}_{yy}\right| \left|\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T\right|, \tag{C.48}$$

where the determinant identity in Eq. (C.94) is used. Substituting Eq. (C.48) into (C.47), we have

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_{xx}|}{\left|\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T\right|} \\ &= \frac{1}{2}\log\frac{1}{\left|\boldsymbol{I} - \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T\right|}.\end{aligned} \tag{C.49}$$

Let us define the eigenvalues of $\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T$ as γ_j where $j = 1, \ldots, d$ and $d = \min\{p, q\}$.[6] Using these eigenvalues, we can derive

$$\begin{aligned}\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{1}{2}\log\frac{1}{\left|\boldsymbol{I} - \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^T\right|} \\ &= \frac{1}{2}\log\frac{1}{\prod_{j=1}^{d}(1-\gamma_j)} = \frac{1}{2}\sum_{j=1}^{d}\log\frac{1}{1-\gamma_j}.\end{aligned} \tag{C.50}$$

Note that γ_1 is equal to the canonical squared correlation ρ_c^2, according to the arguments in Sect. C.3.1.

When the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are complex-valued, the mutual information for the complex random vectors is expressed as

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{H}(\boldsymbol{x}) + \mathcal{H}(\boldsymbol{y}) - \mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{\Sigma}_{xx}| + \log\left|\boldsymbol{\Sigma}_{yy}\right| - \log|\boldsymbol{\Sigma}_{zz}|. \tag{C.51}$$

[6] Here, remember that p and q are the sizes of the column vector $\boldsymbol{x}$ and $\boldsymbol{y}$

Here, $\boldsymbol{\Sigma}_{xx}$, $\boldsymbol{\Sigma}_{yy}$, and $\boldsymbol{\Sigma}_{zz}$ are covariance matrices of the complex Gaussian distribution. Using exactly the same derivation, we can obtain

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = \log \frac{1}{\left| \boldsymbol{I} - \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{xy}^{T} \right|} = \sum_{j=1}^{d} \log \frac{1}{1 - \gamma_j}, \tag{C.52}$$

where γ_j is the jth eigenvalue of $\boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{xy}^{T}$.

C.3.3 Covariance of Residual Signal and Conditional Entropy

Let us next consider the regression problem:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{C.53}$$

where the vector $\boldsymbol{e}$ represents the residual of this regression. The $p \times q$ coefficient matrix $\boldsymbol{A}$ can be estimated using the least-squares fit,

$$\widehat{\boldsymbol{A}} = \underset{\boldsymbol{A}}{\operatorname{argmin}} E\left[\|\boldsymbol{e}\|^2 \right] = \underset{\boldsymbol{A}}{\operatorname{argmin}} E\left[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \right]. \tag{C.54}$$

Here, $E\left[\|\boldsymbol{e}\|^2\right]$ is expressed as

$$E\left[\|\boldsymbol{e}\|^2 \right] = E\left[\boldsymbol{y}\boldsymbol{y}^T - \boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{x} \right].$$

Differentiating $E\left[\|\boldsymbol{e}\|^2\right]$ with respect to $\boldsymbol{A}$ gives

$$\frac{\partial}{\partial \boldsymbol{A}} E\left[\|\boldsymbol{e}\|^2 \right] = -2E(\boldsymbol{y}\boldsymbol{x}^T) + 2\boldsymbol{A}E(\boldsymbol{x}\boldsymbol{x}^T).$$

Setting this derivative to zero, we obtain

$$\widehat{\boldsymbol{A}} = E(\boldsymbol{y}\boldsymbol{x}^T) E(\boldsymbol{x}\boldsymbol{x}^T)^{-1} = \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1}, \tag{C.55}$$

where $\boldsymbol{\Sigma}_{yx} = E(\boldsymbol{y}\boldsymbol{x}^T)$ and $\boldsymbol{\Sigma}_{xx} = E(\boldsymbol{x}\boldsymbol{x}^T)$. The covariance matrix of the residual $\boldsymbol{e}$ is defined as

$$\begin{aligned}\boldsymbol{\Sigma}_{ee} &= E\left[\left(\boldsymbol{y}-\widehat{\boldsymbol{A}}\boldsymbol{x}\right)\left(\boldsymbol{y}-\widehat{\boldsymbol{A}}\boldsymbol{x}\right)^T\right] \\ &= E\left[\boldsymbol{y}\boldsymbol{y}^T - \boldsymbol{y}\boldsymbol{x}^T\widehat{\boldsymbol{A}}^T - \widehat{\boldsymbol{A}}\boldsymbol{x}\boldsymbol{y}^T + \widehat{\boldsymbol{A}}\boldsymbol{x}\boldsymbol{x}^T\widehat{\boldsymbol{A}}^T\right] \\ &= E(\boldsymbol{y}\boldsymbol{y}^T) - E(\boldsymbol{y}\boldsymbol{x}^T)\widehat{\boldsymbol{A}}^T - \widehat{\boldsymbol{A}}E(\boldsymbol{x}\boldsymbol{y}^T) + \widehat{\boldsymbol{A}}E(\boldsymbol{x}\boldsymbol{x}^T)\widehat{\boldsymbol{A}}^T \\ &= \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\widehat{\boldsymbol{A}}^T - \widehat{\boldsymbol{A}}\boldsymbol{\Sigma}_{xy} + \widehat{\boldsymbol{A}}\boldsymbol{\Sigma}_{xx}\widehat{\boldsymbol{A}}^T, \end{aligned} \tag{C.56}$$

where $\boldsymbol{\Sigma}_{yy} = E(\boldsymbol{y}\boldsymbol{y}^T)$ and $\boldsymbol{\Sigma}_{xy} = E(\boldsymbol{x}\boldsymbol{y}^T)$. Substituting Eq. (C.55) into the equation above gives,

$$\boldsymbol{\Sigma}_{ee} = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}^T. \tag{C.57}$$

Let us assume that the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ follow the Gaussian distribution. According to Sect. C.1, we have the relationship,

$$\mathcal{H}(\boldsymbol{x}) = \frac{1}{2}\log|\boldsymbol{\Sigma}_{xx}| \quad \text{and} \quad \mathcal{H}(\boldsymbol{y}) = \frac{1}{2}\log|\boldsymbol{\Sigma}_{yy}|, \tag{C.58}$$

where we ignore constants that are not related to the current arguments. We also have

$$\begin{aligned}\mathcal{H}(\boldsymbol{x},\boldsymbol{y}) &= \frac{1}{2}\log\left|E\left[\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}^T, \boldsymbol{y}^T\end{bmatrix}\right]\right| \\ &= \frac{1}{2}\log\begin{vmatrix}E(\boldsymbol{x}\boldsymbol{x}^T) & E(\boldsymbol{x}\boldsymbol{y}^T)\\ E(\boldsymbol{y}\boldsymbol{x}^T) & E(\boldsymbol{y}\boldsymbol{y}^T)\end{vmatrix} = \frac{1}{2}\log\begin{vmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{yx}^T\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{vmatrix}. \end{aligned} \tag{C.59}$$

Thus, the conditional entropy is expressed as

$$\mathcal{H}(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{H}(\boldsymbol{x},\boldsymbol{y}) - \mathcal{H}(\boldsymbol{x}) = \frac{1}{2}\log\begin{vmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{yx}^T\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{vmatrix} - \frac{1}{2}\log|\boldsymbol{\Sigma}_{xx}|. \tag{C.60}$$

Using the determinant identity in Eq. (C.94), we finally obtain the formula to compute the conditional entropy

$$\begin{aligned}\mathcal{H}(\boldsymbol{y}|\boldsymbol{x}) &= \frac{1}{2}\log\begin{vmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{yx}^T\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{vmatrix} - \frac{1}{2}\log|\boldsymbol{\Sigma}_{xx}| \\ &= \frac{1}{2}\log\left|\boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}^T\right| = \frac{1}{2}\log|\boldsymbol{\Sigma}_{ee}|. \end{aligned} \tag{C.61}$$

The conditional entropy is expressed by the covariance of the residual signal $\boldsymbol{e}$ obtained by regressing $\boldsymbol{y}$ with $\boldsymbol{x}$.

C.4 Definitions of Several Vector Norms

The pth order norm ($p \geq 0$) of an N-dimensional vector, $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ is defined such that

$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_N|^p\right)^{1/p}. \tag{C.62}$$

When $p = 2$, the norm is called the L_2-norm, which is equal to the conventional Euclidean norm,

$$\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_N^2}. \tag{C.63}$$

The L_2-norm is usually denoted $\|\boldsymbol{x}\|$. When $p = 1$, the norm is expressed as

$$\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_N|, \tag{C.64}$$

which is called the L_1-norm. When $p = 0$, we can define $\mathcal{T}(x)$, which is called the indicator function, such that

$$\mathcal{T}(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0, \end{cases} \tag{C.65}$$

with the norm expressed as

$$\|\boldsymbol{x}\|_0 = \sum_{i=1}^{N} \mathcal{T}(x_i). \tag{C.66}$$

Namely, $\|\boldsymbol{x}\|_0$ is equal to the number of non-zero components of $\boldsymbol{x}$. When $p = \infty$, the norm becomes

$$\|\boldsymbol{x}\|_\infty = \max\{x_1, x_2, \ldots, x_N\}. \tag{C.67}$$

Namely, the $\|\boldsymbol{x}\|_\infty$ is equal to the maximum component of $\boldsymbol{x}$.

C.5 Derivative of Functionals

This section provides a brief explanation of the derivative of functionals. A conventional function relates input variables to output variables. A functional relates input functions to output variables. A simple example of a functional is:

$$I = \int_\alpha^\beta f(x)dx, \tag{C.68}$$

where $f(x)$ is defined in $[\alpha, \beta]$. This I is a functional of the function $f(x)$. The functional is often denoted as $I[f(x)]$.

Let us consider a function $F(x)$ that contains another function $f(x)$. One example is $F(x) = f(x)^2 + 2x$ where $F(x)$ consists of $f(x)$ and x. The integral of $F(x)$,

$$I[f(x)] = \int_{\alpha}^{\beta} F(x)dx, \tag{C.69}$$

is a functional of $f(x)$. Let us define a small change in $f(x)$ as $\delta f(x)$, and a change in $I[f(x)]$ due to $\delta f(x)$ as $\delta I[f(x)]$. For simplicity, $\delta I[f(x)]$ is denoted δI in the following arguments. To derive the relationship between δI and $\delta f(x)$, the region $[\alpha, \beta]$ is divided into small regions Δx. A contribution from the jth small region onto δI, δI_j, is equal to

$$\delta I_j = A_j \delta f(x_j) \Delta x. \tag{C.70}$$

Here we assume that δI_j is proportional to $\delta f(x_j)$ and A_j is a proportional constant. Then, δI is derived as a sum of contributions from all small regions such that

$$\delta I = \sum_j A_j \delta f(x_j) \Delta x. \tag{C.71}$$

Accordingly, when $\Delta x \to 0$, the relationship between $\delta f(x)$ and δI becomes

$$\delta I = \int_{\alpha}^{\beta} A(x) \delta f(x) dx. \tag{C.72}$$

In Eq. (C.72), $A(x)$ is defined as the derivative of the functional $I[f(x)]$ with respect to $f(x)$, and it is denoted $\delta I / \delta f$.

Let us compute this derivative of a functional. When a functional is given in Eq. (C.68), i.e., $F(x) = f(x)$, the amount of change δI due to the change from $f(x)$ to $f(x) + \delta f(x)$ is expressed as

$$\delta I = \int_{\alpha}^{\beta} \left[f(x) + \delta f(x) \right] dx - \int_{\alpha}^{\beta} f(x) dx = \int_{\alpha}^{\beta} \delta f(x) dx. \tag{C.73}$$

Therefore, we obtain $A(x) = \delta I / \delta f = 1$. In the general case where $F(x)$ consists of $f(x)$, the relationship

$$\delta I = \int_{\alpha}^{\beta} \left[F(x) + \delta F(x) \right] dx - \int_{\alpha}^{\beta} F(x) dx = \int_{\alpha}^{\beta} \delta F(x) dx \tag{C.74}$$

holds. Using the relationship

$$\delta F(x) = \frac{\partial F}{\partial f} \delta f(x),$$

we have

$$\delta I = \int_{\alpha}^{\beta} \delta F(x)dx = \int_{\alpha}^{\beta} \frac{\partial F}{\partial f}\delta f(x)dx, \tag{C.75}$$

and thus,

$$\frac{\delta I}{\delta f} = \frac{\partial F}{\partial f}. \tag{C.76}$$

As an example of the use of Eq. (C.76), let us derive the derivative of the functional $\mathbb{L}[q(\boldsymbol{x})]$ in Eq. (B.53). Ignoring the last term, which does not contain $q(\boldsymbol{x})$, Eq. (B.53) is rewritten as

$$\begin{aligned}\mathbb{L}[q(\boldsymbol{x})] = &\int_{-\infty}^{\infty} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) \\ &- \int_{-\infty}^{\infty} \mathrm{d}\boldsymbol{x} q(\boldsymbol{x}) \log q(\boldsymbol{x}) + \gamma \int_{-\infty}^{\infty} q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{C.77}$$

In the first term, since $F(\boldsymbol{x}) = q(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$, we have

$$\frac{\partial F}{\partial q} = \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}). \tag{C.78}$$

In the second term, since $F(\boldsymbol{x}) = q(\boldsymbol{x}) \log q(\boldsymbol{x})$, we have

$$\frac{\partial F}{\partial q} = \frac{1}{\partial q(\boldsymbol{x})}\left[q(\boldsymbol{x}) \log q(\boldsymbol{x})\right] = \log q(\boldsymbol{x}) + q(x)\frac{1}{q(x)} = \log q(\boldsymbol{x}) + 1. \tag{C.79}$$

In the third term, since $F(\boldsymbol{x}) = \gamma q(\boldsymbol{x})$, we have

$$\frac{\partial F}{\partial q} = \gamma. \tag{C.80}$$

Therefore, we have

$$\frac{\partial \mathbb{L}[q(\boldsymbol{x})]}{\partial q(\boldsymbol{x})} = \log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \log q(\boldsymbol{x}) - 1 + \gamma, \tag{C.81}$$

which is equal to Eq. (B.54).

C.6 Vector and Matrix Derivatives

Differentiating a scalar F with a column vector $\boldsymbol{x}$ is defined as creating a column vector whose jth element is equal to $\partial F/\partial x_j$. Assuming that $\boldsymbol{a}$ is a column vector and $\boldsymbol{A}$ is a matrix, The following relationships hold,

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{a}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{a}^T \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}, \tag{C.82}$$

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{x}, \tag{C.83}$$

$$\frac{\partial \,\mathrm{tr}(\boldsymbol{x}\boldsymbol{a}^T)}{\partial \boldsymbol{x}} = \frac{\partial \,\mathrm{tr}(\boldsymbol{a}\boldsymbol{x}^T)}{\partial \boldsymbol{x}} = \boldsymbol{a}. \tag{C.84}$$

Let us denote the (i, j)th element of a matrix $\boldsymbol{A}$ as $A_{i,j}$, Differentiating a scalar F with a matrix $\boldsymbol{A}$ is defined as creating a matrix whose (i, j)th element is equal to $\partial F / \partial A_{i,j}$. Representative identities are the following, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are column vectors and $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices.

$$\frac{\partial \,\mathrm{tr}(\boldsymbol{A})}{\partial \boldsymbol{A}} = \boldsymbol{I}, \tag{C.85}$$

$$\frac{\partial \,\mathrm{tr}(\boldsymbol{A}\boldsymbol{B})}{\partial \boldsymbol{A}} = \boldsymbol{B}^T, \tag{C.86}$$

$$\frac{\partial \,\mathrm{tr}(\boldsymbol{A}^T\boldsymbol{B})}{\partial \boldsymbol{A}} = \boldsymbol{B}, \tag{C.87}$$

$$\frac{\partial \,\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^T)}{\partial \boldsymbol{A}} = \boldsymbol{A}(\boldsymbol{B} + \boldsymbol{B}^T), \tag{C.88}$$

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}}{\partial \boldsymbol{A}} = \boldsymbol{x}\boldsymbol{y}^T, \tag{C.89}$$

$$\frac{\partial \log |\boldsymbol{A}|}{\partial \boldsymbol{A}} = (\boldsymbol{A}^{-1})^T. \tag{C.90}$$

C.7 Several Formulae for Matrix Computations Used in this Book

The following are representative formulae for the matrix inversion.

$$(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}, \tag{C.91}$$

$$(\boldsymbol{A}^{-1} + \boldsymbol{B}^T\boldsymbol{C}^{-1}\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C}^{-1} = \boldsymbol{A}\boldsymbol{B}^T(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T + \boldsymbol{C})^{-1}. \tag{C.92}$$

Also, we have

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{M}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}\boldsymbol{B}\boldsymbol{D}^{-1} \end{bmatrix}, \tag{C.93}$$

where

$$\boldsymbol{M} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{-1}.$$

Regarding the matrix determinant, we have the following identity,

$$\begin{vmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{vmatrix} = |\boldsymbol{A}||\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}| = |\boldsymbol{D}||\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}|. \tag{C.94}$$

When a matrix $\boldsymbol{A}$ is an invertible $(n \times n)$ matrix, and $\boldsymbol{B}$ and $\boldsymbol{C}$ are $(n \times m)$ matrices, the following matrix determinant lemma holds:

$$|\boldsymbol{A}||\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{C}| = |\boldsymbol{A} + \boldsymbol{C}\boldsymbol{B}^T|. \tag{C.95}$$

C.8 Properties of Eigenvalues

Representative properties of eigenvalues that may be used in this book are listed below.

1. If $\boldsymbol{A}$ is a Hermitian matrix and positive definite, all eigenvalues are real and greater than 0.
2. If $\boldsymbol{A}$ is a Hermitian matrix and positive semidefinite, all eigenvalues are real and greater than or equal to 0.
3. If $\boldsymbol{A}$ is a real symmetric matrix and positive definite, all eigenvalues are real and greater than 0. Eigenvectors are also real.
4. If $\boldsymbol{A}$ is a real symmetric matrix and positive semidefinite, all eigenvalues are real and greater than or equal to 0. Eigenvectors are also real.
5. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is nonsingular, eigenvalues of $\boldsymbol{A}$ are also eigenvalues of $\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{B}$.
6. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is unitary, eigenvalues of $\boldsymbol{A}$ are also eigenvalues of $\boldsymbol{B}^H\boldsymbol{A}\boldsymbol{B}$.
7. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is orthogonal, eigenvalues of $\boldsymbol{A}$ are also eigenvalues of $\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B}$.
8. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is positive definite, eigenvalues of $\boldsymbol{B}\boldsymbol{A}$ are also eigenvalues of $\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2}$.
9. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is positive definite, eigenvalues of $\boldsymbol{B}^{-1}\boldsymbol{A}$ are also eigenvalues of $\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}$.
10. Let us assume that $\boldsymbol{A}$ is an $(m \times n)$ matrix, $\boldsymbol{B}$ is an $(n \times m)$ matrix, and $n \geq m$. If $\lambda_1, \ldots, \lambda_m$ are eigenvalues of $\boldsymbol{A}\boldsymbol{B}$, $\lambda_1, \ldots, \lambda_m, 0, \ldots, 0$ are eigenvalues of $\boldsymbol{B}\boldsymbol{A}$.

C.9 Rayleigh-Ritz Formula

This section provides a proof of the Rayleigh-Ritz formula, which is according to [2]. We define $\boldsymbol{A}$ and $\boldsymbol{B}$ as positive definite matrices of the same dimension. We introduce the following notations and use them throughout the book:

- The minimum and maximum eigenvalues of a matrix $\boldsymbol{A}$ are denoted $\mathcal{S}_{min}\{\boldsymbol{A}\}$ and $\mathcal{S}_{max}\{\boldsymbol{A}\}$.
- The eigenvectors corresponding to the minimum and maximum eigenvalues of a matrix $\boldsymbol{A}$ are denoted $\boldsymbol{\vartheta}_{min}\{\boldsymbol{A}\}$ and $\boldsymbol{\vartheta}_{max}\{\boldsymbol{A}\}$.
- The minimum and maximum generalized eigenvalues of a matrix $\boldsymbol{A}$ with a metric $\boldsymbol{B}$ are denoted $\mathcal{S}_{min}\{\boldsymbol{A}, \boldsymbol{B}\}$ and $\mathcal{S}_{max}\{\boldsymbol{A}, \boldsymbol{B}\}$, and the corresponding eigenvectors are denoted $\boldsymbol{\vartheta}_{min}\{\boldsymbol{A}, \boldsymbol{B}\}$ and $\boldsymbol{\vartheta}_{max}\{\boldsymbol{A}, \boldsymbol{B}\}$.

Here, if the matrix $\boldsymbol{B}$ is nonsingular, the following relationships hold:

$$\begin{aligned}
\mathcal{S}_{max}\{\boldsymbol{A}, \boldsymbol{B}\} &= \mathcal{S}_{max}\{\boldsymbol{B}^{-1}\boldsymbol{A}\}, \\
\boldsymbol{\vartheta}_{max}\{\boldsymbol{A}, \boldsymbol{B}\} &= \boldsymbol{\vartheta}_{max}\{\boldsymbol{B}^{-1}\boldsymbol{A}\}, \\
\mathcal{S}_{min}\{\boldsymbol{A}, \boldsymbol{B}\} &= \mathcal{S}_{min}\{\boldsymbol{B}^{-1}\boldsymbol{A}\}, \\
\boldsymbol{\vartheta}_{min}\{\boldsymbol{A}, \boldsymbol{B}\} &= \boldsymbol{\vartheta}_{min}\{\boldsymbol{B}^{-1}\boldsymbol{A}\}.
\end{aligned}$$

Using $\boldsymbol{x}$ to denote a column vector with its dimension commensurate with the size of the matrices, this appendix shows that

$$\max_{\boldsymbol{x}} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x}} = \mathcal{S}_{max}\{\boldsymbol{A}, \boldsymbol{B}\}, \tag{C.96}$$

and

$$\operatorname*{argmax}_{\boldsymbol{x}} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x}} = \boldsymbol{\vartheta}_{max}\{\boldsymbol{A}, \boldsymbol{B}\}. \tag{C.97}$$

Since the value of the ratio $(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x})/(\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x})$ is not affected by the norm of $\boldsymbol{x}$, we set the norm of $\boldsymbol{x}$ so as to satisfy the relationship $\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} = 1$. Then, the maximization problem in Eq. (C.96) is rewritten as

$$\max_{\boldsymbol{x}} \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} = 1. \tag{C.98}$$

We change this constrained maximization problem to an unconstrained maximization problem by introducing the Lagrange multiplier κ. We define the Lagrangian $\mathbb{L}(\boldsymbol{x}, \kappa)$ such that

$$\mathbb{L}(\boldsymbol{x}, \kappa) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \kappa(\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} - 1). \tag{C.99}$$

The maximization in Eq. (C.98) is equivalent to maximizing $\mathbb{L}(\boldsymbol{x}, \kappa)$ with no constraints.

To obtain the maximum of $\mathbb{L}(\boldsymbol{x}, \kappa)$, we calculate the derivatives

$$\frac{\partial \mathbb{L}(\boldsymbol{x}, \kappa)}{\partial \boldsymbol{x}} = 2(\boldsymbol{A}\boldsymbol{x} - \kappa\boldsymbol{B}\boldsymbol{x}), \tag{C.100}$$

$$\frac{\partial \mathbb{L}(\boldsymbol{x}, \kappa)}{\partial \kappa} = -(\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} - 1). \tag{C.101}$$

By setting these derivatives to zero, we can derive the relationships, $\boldsymbol{Ax} = \kappa \boldsymbol{Bx}$ and $\kappa = \boldsymbol{x}^T \boldsymbol{Ax}$. Therefore, the maximum value of $\boldsymbol{x}^T \boldsymbol{Ax}$ is equal to the maximum eigenvalue of $\boldsymbol{Ax} = \kappa \boldsymbol{Bx}$, and the $\boldsymbol{x}$ that attains this maximum value is equal to the eigenvector corresponding to this maximum eigenvalue. Namely, we have

$$\max_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{Ax}}{\boldsymbol{x}^T \boldsymbol{Bx}} = \mathcal{S}_{\max}\{\boldsymbol{A}, \boldsymbol{B}\} = \mathcal{S}_{\max}\{\boldsymbol{B}^{-1}\boldsymbol{A}\},$$

and

$$\operatorname*{argmax}_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{Ax}}{\boldsymbol{x}^T \boldsymbol{Bx}} = \boldsymbol{\vartheta}_{max}\{\boldsymbol{A}, \boldsymbol{B}\} = \boldsymbol{\vartheta}_{max}\{\boldsymbol{B}^{-1}\boldsymbol{A}\}.$$

Using exactly the same derivation, it is easy to show that

$$\min_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{Ax}}{\boldsymbol{x}^T \boldsymbol{Bx}} = \mathcal{S}_{\min}\{\boldsymbol{A}, \boldsymbol{B}\}, \tag{C.102}$$

and

$$\operatorname*{argmin}_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{Ax}}{\boldsymbol{x}^T \boldsymbol{Bx}} = \boldsymbol{\vartheta}_{min}\{\boldsymbol{A}, \boldsymbol{B}\}. \tag{C.103}$$

References

1. F. D. Neeser and J. L. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions on Information Theory*, vol. 39, pp. 1293–1302, 1993.
2. F. R. Gantmacher, *The Theory of Matrices*. New York, NY: Chelsea Publishing Company, 1960.

Index

© Springer International Publishing Switzerland 2015

K. Sekihara and S.S. Nagarajan, *Electromagnetic Brain Imaging*,
DOI 10.1007/978-3-319-14947-9