Auditory System Characterization

A Dissertation Presented

by

Hiroki Asari

Farish-Gerry Fellow

to

The Watson School of Biological Sciences

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

the Biological Sciences

at Cold Spring Harbor Laboratory

July 2007
Preface

The world is full of sounds—rustle of leaves, whistle of winds, and bustle of cities—and in our everyday lives, we enjoy music and pleasant conversations with friends and families. Such auditory communications play highly significant roles in social interactions in many animal species including humans, songbirds, crickets, and so on, but they would never be accomplished without the specialized module in the brain; the auditory system.

The three main functions of the auditory system are to separate, localize, and identify sound signals we receive at the cochlea. To perform these tasks, the auditory system became sophisticated during the course of evolution, and now reached such an amazing level where coincidence detectors in owls can identify interaural time differences with the precision of microseconds (Moiseff and Konishi, 1981; Konishi, 2003), and hair cells in the vertebrate ear can detect a tip movement of only a tenth of a nanometer (Sellick et al., 1982; Crawford and Fettiplace, 1985; Hudspeth, 1989, 1997). Moreover, a trained ear can establish a sense of “absolute pitch” (Zatorre, 2003; Levitin and Rogers, 2005), and even listen out and transcribe a single instrument from a musical piece played by an orchestra.

How can the brain perform such outstanding feats? Although the auditory system works so efficiently that one might almost forget about how difficult the auditory tasks are, it should be mentioned that the underlying computations are extremely complex and overwhelming. In fact, no artificial system can compete with the brain in solving any acoustic—and any sensory—signal processing problem. The brain generally works up to the noise limit of the sensors across their entire dynamic range in all ecological conditions, whereas an artificial system
requires a huge amount of tuning to solve a particular problem of interest but yet works only in a carefully controlled demo (as I show an example in Chapter 2) or only in a “platonic” world. As such, not surprisingly, the process beyond the auditory nerve (i.e., the very first step of the auditory processing in the brain) is not known very well despite many decades of psychological and physiological research (for the earliest electrophysiological studies on the auditory cortex, see e.g., Woolsey and Walzl, 1942; Tunturi, 1944; Bremer and Bonnet, 1949).

In this dissertation, I combined two complementary approaches to study neuronal dynamics in the auditory cortex and challenged how the brain processes acoustic signals.

1. As a top-down theoretical analysis of neural behaviors at the population level, we exploited the idea of sparse overcomplete linear representations and developed a model for monaural blind source separation (Asari et al., 2006, 2007).

2. As a bottom-up experimental analysis at the single-cell level, we used whole-cell recordings in vivo and examined how neural responses in the auditory cortex depend on stimulus history and its context in order to build plausible models that characterize the relationship between input sounds and output neural responses (Asari and Zador, in preparation).

This thesis therefore consists of two major parts. Following the introductory Chapter 1 overviewing the characteristics of the auditory system and the challenges in auditory systems research, Chapter 2 describes the theoretical part in detail. Inspired by a striking anatomical feature of many sensory processing problems that many more neurons appear to be engaged in the internal representations of the signal than in its transduction (Section 2.1), I demonstrate an example of how sparse overcomplete linear representations can directly solve difficult acoustic signal processing problems (Sections 2.2–2.3); i.e., monaural source separation using solely the cues provided by the differential filtering imposed on a source by its path from its origin to the cochlea (the head-related transfer function; Bregman, 1990). The model of sparse representations then makes several experimentally testable predictions, which in turn can be used to test
the model with experimental data (Section 2.4). This model framework can be generalized to exploit other monaural separation cues such as common onset time, and binaural information such as interaural time and level differences (Section 2.5).

The experimental part is then covered in Chapter 3. Because sensory signal processing in the brain depends on stimulus history and its context (Section 3.1), here we assessed the relevant time-scales and how past events influence the responses in the auditory cortex (Sections 3.2–3.3). We found that the context-dependence sometimes lasted as long as four seconds or longer in some neurons; and the changes in lower-order sound properties (e.g., intensity) had larger and longer effects on the following response dynamics than the changes in higher-order properties (e.g., amplitude-modulation; Section 3.5). The data were also analyzed from a viewpoint of model construction (Section 3.4), showing that the window length of at least several seconds was required to capture the stimulus-related predictable response power fully enough (Section 3.5). Although the linear model performance did not improve substantially by just extending the window length and even by incorporating static nonlinearities, these results suggest that complex bottom-up modulations on longer time-scales should contribute a lot to the nature of stimulus encoding in the auditory cortex and its functions (Section 3.6).

Finally, Chapter 4 briefly recapitulates the results from Chapters 2 and 3, and discusses future challenges. I will close with a general discussion on how we could possibly meet theory and biology together for better understanding the principles and the algorithms underlying computations and functions in the brain. This thesis is intended to be self-contained, and thus all mathematical details and conventional algorithms used for the data analysis and the simulations are explained in Appendix A.
Acknowledgments

This thesis would never have been done without generous supports and encouragements from many people. First and foremost, I would like to thank my advisor Anthony M. Zador for his attention, guidance, insight, and support throughout the entire period of this work. Tony taught me how to think clearly and logically to conduct science, and how to design and perform critical experiments. Our numerous scientific discussions and his many constructive comments have greatly influenced and improved this work.

I would also like to offer my sincerest gratitude to Barak A. Pearlmutter, who first introduced theoretical science to me. I owe a lot to him, especially on the theoretical side of this thesis in Chapter 2. In addition, Barak generously accepted to serve as my external examiner.

I am deeply grateful to Hysell Oviedo, Michael S. Wehr, and Christian K. Machens. Hysell and Mike showed me how to do whole-cell recordings \textit{in vivo}—a very powerful yet difficult technique in neurophysiology. Mike also provided me his recording data for the model analysis in Chapter 3, and many discussions with Christian as well as his analysis scripts were very helpful for completing the experimental side of this thesis.

I cannot describe in words how thankful I am to all my colleagues in the Zador Lab; Michael R. DeWeese, Tomáš Hromádka Santiago Jaramillo, Ed Kramer, Susana Lima-Mainen, Gonzalo Otazu, Shraddha Pai, Simon Rumpel, Lung-Hao Tai, and Yang Yang. I feel so privileged to work with them and learn how exciting science is. Many thanks go in particular to Tomáš and Susana for their valuable comments on this dissertation. I would also like to express
my special thanks to Barry Burbach, Anita Matthes, and Becky Dong for their technical and administrative supports.

Of course my committee members have always been invaluable to this thesis. Carlos D. Brody guided and kept me on track for the big picture of the work as a chair, and deep insights and critical advices from Alexei A. Koulakov and Eero P. Simoncelli from theoretical viewpoints were just indispensable. Z. Josh Huang taught and influenced me a lot on my biological perspectives, and deserves my deepest gratitude for being my academic mentor.

I also have to acknowledge generous supports from the Farish-Gerry Fellowship and the Watson School of Biological Sciences, under the leadership of Winship Herr and Lilian Clark.

Finally, I would like to thank my parents, Yukie and Ichiro, and my brother, Seiki, for all their understandings and encouragements. I could not have opened this exciting journey into science without their dedicated supports.
Contents

Preface i

Acknowledgments iii

List of Figures x

List of Tables xii

1 Introduction 1

1.1 Auditory System Organization 2

1.1.1 Hair Cells and Auditory Nerve Fibers 3

1.1.2 Cochlear Nucleus 4

1.1.3 Superior Olivary Nucleus 4

1.1.4 Inferior Colliculus 6

1.1.5 Medial Geniculate Nucleus 6

1.1.6 Primary Auditory Cortex 7

1.2 Characteristics of Auditory Signal Processing 8

2 Top-down characterization: population level analysis 12

2.1 Sparse Overcomplete Representations 14

2.1.1 Blind Source Separation: Overview 15
# 2 Source Separation Model: Problem Formulation

## 2.1 Head-Related Transfer Functions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.2 Head-Related Transfer Functions</td>
<td>18</td>
</tr>
</tbody>
</table>

## 2.2 Source Separation Model: Problem Formulation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1 Dictionary Method Approach</td>
<td>21</td>
</tr>
<tr>
<td>2.2.2 HRTF-Based Approach</td>
<td>24</td>
</tr>
<tr>
<td>2.2.3 Neural Representation for Source Separation</td>
<td>26</td>
</tr>
<tr>
<td>2.2.4 Dictionary Learning</td>
<td>30</td>
</tr>
</tbody>
</table>

## 2.3 Monaural Separation: Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1 Introduction</td>
<td>34</td>
</tr>
</tbody>
</table>

## 2.4 Predictions on Neural Behaviors

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1 Optimal Feature Estimation</td>
<td>41</td>
</tr>
<tr>
<td>2.4.2 Asymmetry of sparse representations</td>
<td>42</td>
</tr>
<tr>
<td>2.4.3 Context-Dependence of Receptive Field</td>
<td>49</td>
</tr>
<tr>
<td>2.4.4 Top-Down Receptive Field Modulation</td>
<td>52</td>
</tr>
<tr>
<td>2.4.5 Sparse Activities</td>
<td>54</td>
</tr>
</tbody>
</table>

## 2.5 Discussion

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.1 Model Implications</td>
<td>59</td>
</tr>
<tr>
<td>2.5.2 Model Extensions</td>
<td>62</td>
</tr>
</tbody>
</table>

## 2.A Appendix: Feature Estimation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.A.1 Learning Algorithm</td>
<td>66</td>
</tr>
<tr>
<td>2.A.2 Simulation Procedures</td>
<td>70</td>
</tr>
<tr>
<td>2.A.3 Results</td>
<td>71</td>
</tr>
<tr>
<td>2.A.4 Discussion</td>
<td>72</td>
</tr>
</tbody>
</table>

## 3 Bottom-up characterization: single-cell level analysis

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>79</td>
</tr>
<tr>
<td>3.1.1 Sensory Coding Models</td>
<td>80</td>
</tr>
</tbody>
</table>
3.1.2 Motivation and Background Studies ........................................ 82

3.2 Experimental Methods ......................................................... 84
   3.2.1 Surgery ............................................................... 84
   3.2.2 Whole-Cell Patch-Clamp Recordings ............................... 85
   3.2.3 Stimulus Design ..................................................... 87
   3.2.4 Stimuli ................................................................. 89

3.3 Data Analysis Methods ......................................................... 94
   3.3.1 Context-Dependence ............................................... 96
   3.3.2 Exponential Curve Fit ............................................... 100
   3.3.3 Response Predictability ............................................ 100

3.4 Encoding Models ............................................................... 101
   3.4.1 Linear-Nonlinear Cascade Models ................................. 101
   3.4.2 Nonlinear Models .................................................... 103
   3.4.3 Model Performance ................................................ 105

3.5 Results ............................................................................. 106
   3.5.1 Context-Dependence ............................................... 107
   3.5.2 Relation to Acoustic Properties .................................. 113
   3.5.3 Relation to Response Predictability ............................. 114
   3.5.4 Model Performance ................................................ 119

3.6 Discussion ...................................................................... 123
   3.6.1 Context-Dependence ............................................... 123
   3.6.2 Response Predictability ............................................. 125
   3.6.3 Possible Mechanisms ............................................... 126
   3.6.4 Plausible Encoding Model ........................................ 127
4 Coda

4.1 Conclusions and Future Challenges ........................................... 129
4.2 General Discussion ................................................................. 132

A Technical Notes

A.1 Matrix Decomposition .............................................................. 136
   A.1.1 Principal Component Analysis ............................................. 136
   A.1.2 Singular Value Decomposition .......................................... 139
   A.1.3 Independent Component Analysis ....................................... 141
   A.1.4 Non-negative Matrix Factorization ..................................... 145
   A.1.5 Characteristics of Each Factorization Method ....................... 146

A.2 Linear Regression ..................................................................... 153
   A.2.1 Regularization Methods .................................................. 154
   A.2.2 Probabilistic Interpretation ............................................... 156
   A.2.3 Dynamic Models .............................................................. 157

A.3 Nonlinear Regression .............................................................. 159
   A.3.1 Perceptron ....................................................................... 160
   A.3.2 Artificial Neural Network .................................................. 160
   A.3.3 Support Vector Regression ................................................ 164

A.4 Information Theory .............................................................. 166
   A.4.1 Entropy ........................................................................... 167
   A.4.2 Mutual Information .......................................................... 169
   A.4.3 File Compression Methods ............................................... 176

Bibliography .................................................................................. 178
List of Figures

2.1 Overcomplete representation in two dimensions .................................. 29
2.2 Non-negative matrix factorization (NMF) can be used to find the parts of sound ensembles ................................................................. 32
2.3 Separation of three musical sources ..................................................... 36
2.4 Performance of different separation approaches with three sources .......... 37
2.5 Separation performance for different source locations ......................... 38
2.6 Neural representations underlying source separation .......................... 39
2.7 Prediction 1: Stimulus optimization requires multi-neuron recording .... 43
2.8 Prediction 2: Asymmetry between linear decoding and linear encoding .. 47
2.9 Piecewise linear encoding ................................................................. 48
2.10 Prediction 3: Dependence of STRF on stimulus context ..................... 51
2.11 Prediction 4: Top-down spatial expectation modulates neuronal tuning ... 53
2.12 Sparseness of responses in the primary auditory cortex ...................... 56
2.13 Geometry of overcomplete representations in $\mathbb{R}^2$ ....................... 63
2.14 Population variability in sparse overcomplete representations ............. 64
2.15 Feature estimation in $\mathbb{R}^3$ from a simulated single-unit data set .......... 73
2.16 Feature estimation and Fisher information matrices .......................... 74
2.17 Feature estimation for up to 20 dimension ...................................... 75
3.1 Experimental design and auditory stimuli used in this study .......................... 88
3.2 Recording data statistics ................................................................. 95
3.3 Context-dependent effects on spontaneous dynamics .................................. 109
3.4 Context-dependence of neural responses lasted for several seconds ............ 110
3.5 Context-dependence can last for seconds ............................................ 111
3.6 No clear correlation between recording depth and longest-lasting context- dependence effects ................................................................. 112
3.7 Context-dependence caused by frequency changes .................................. 116
3.8 Relation between context-dependence and sound properties ..................... 117
3.9 Population analysis summary ............................................................ 118
3.10 Poor prediction performance of classical linear-nonlinear cascade models ... 120
3.11 Context-dependence of response predictability ....................................... 121
A.1 NMF found “meaningful” parts in Chinese characters while ICA failed .......... 148
A.2 The original non-overlapping basis images and those obtained by NMF ...... 150
A.3 Original overlapping basis images $A_0$ and those obtained by NMF, PCA, and ICA ................................................................. 152
A.4 Entropy estimation by file compression ................................................ 179
List of Tables

3.1 Summary of recording data ...................................................... 86
3.2 Performance of generic nonlinear encoding models ...................... 122
Chapter 1

Introduction

*Je pense, donc je suis*—René Descartes.

The brain is indeed the most complex computational device we have ever had. The brain not only controls the internal states and coordinated movements of our bodies, but also processes sensory signals for appropriately responding to the outside world, and above all, it is the brain that makes us human (or at least, makes us scientists).

This amazing organ, occupying only $\sim 2\%$ of body mass but yet consuming $\sim 20\%$ of total energy, consists of several hundred billion of neurons and even more glial cells in humans (Abeles, 1991; Kandel et al., 2000; Smith, 2000). The former works as a computational unit that sends and receives electro-chemical signals—or “spikes” in essentially a binary manner—between each other, whereas the latter provides physical, functional, and nutritional support for nearby neurons. Therefore neurons are the main players of the brain from the computational viewpoint (“neuron doctrine;” McCulloch and Pitts, 1943; Barlow, 1972). Aside from the plastic properties and developmental aspects, however, the basic performance of neurons themselves is not necessarily superior to that of the state-of-the-art transistors that work as computational units in very large-scale integration (VLSI) systems. In fact, (1) the switching speed of a neuron—or the firing rate—is limited up to $\sim 10^3$ Hz, whereas that of transistors
can be several orders of magnitude faster (e.g., \(\sim 10^{15}\) Hz for the leading-edge supercomputers); and (2) the conduction velocity of action potentials is at most \(\sim 10^2\) m/sec in myelinated axons, whereas that of electronic circuits can be \(\sim 10^8\) m/sec. Considering that the number of constituent units is comparable between the two systems (\(\sim 10^{10}\)), there is then no reason to imagine that we cannot build an “artificial brain.” But the truth is that currently available computers can outperform the nervous system only in some “tedious routines,” and in most computations, the brain is second to none at this moment.

Among many problems we confront in our everyday lives, some require huge efforts for the brain to solve, whereas many others do not. (Ironically, the former tasks include the “tedious routines” that an artificial system is good at handling.) It should however be emphasized that the fact that the brain can do effortlessly does not necessarily mean that the underlying computations are readily tractable. One such type of the problems is sensory inference. Taking acoustic signal processing problems for example, the brain can easily separate, localize, and identify the sound sources from a mixture of sounds coming from multiple sources, but no artificial system can do these tasks in general settings.

Section 1.1 will review the organization of mammalian auditory systems, specifically focusing on what computations would take place at each level. Then in Section 1.2, by comparing to other sensory modalities (especially to the visual systems), I will describe some characteristics and challenges of auditory signal processing problems, as well as the motivations of this study.

### 1.1 Auditory System Organization

The goal of the auditory system is to “feel” rapid changes of local air pressure in a certain range (e.g., the dynamic range of human hearing is 20–20,000 Hz; Bregman, 1990; Kandel et al., 2000; Smith, 2000), and make sense of the acoustic environment by extracting behaviorally meaningful information such as communication calls. For this purpose, nature has developed
(1) a special mechanical device to receive and transform sounds into an appropriate signal format for the nervous system (i.e., a series of action potentials); (2) faithful ways to transmit such acoustic information to downstream subcortical and cortical stations for further (serial and parallel) processing; and (3) neural circuits to represent and perceive sounds, eventually leading to appropriate motor actions. Below I will briefly overview the ascending auditory pathway—from hair cells at the cochlea up to the (primary) auditory cortex—focusing on the computations and the current working models at each station (for more comprehensive reviews on the anatomical and physiological characters, see e.g., Schreiner et al., 2000; Read et al., 2002; Malmierca, 2003; Malmierca and Irvine, 2005; Winer et al., 2005).

### 1.1.1 Hair Cells and Auditory Nerve Fibers

The first—and a fundamental—auditory processing is the decomposition of acoustic signals into frequency components (Fletcher, 1940; Hudspeth, 1989, 1997). This transformation from the time domain to appropriate time-frequency representations depends on the mechanical properties and the anatomical organization of the cochlea, where each frequency element leads to the vibration of the basilar membrane only at specific locations, due to resonance effects. Therefore each mechanosensory hair cell on the membrane responds only to a limited range of frequencies—or the frequency bandwidth—and such frequency analysis leads to the tonotopical organization of the cochlea as manifested in the tuning curve properties of the ascending auditory nerve fibers (Kiang et al., 1967; Kiang and Moxon, 1974).

From a computational viewpoint, the cochlea can thus be considered as a bank of bandpass filters. Conventional choices of time-frequency analysis are the short-term Fourier transform (STFT or spectrogram; see Eq.(3.25) on page 102; Cohen, 1995) or a Gammatone filter bank (Irino and Patterson, 2001), but more sophisticated cochlear models have been developed by incorporating psychoacoustic knowledge and physiological hair cell dynamics such as adaptations (Patterson, 1974; Lyon, 1982; Meddis, 1986; Seneff, 1988; Patterson and Holdsworth, 1996). Note that certain acoustic signals will become less overlapped in the frequency domain,
which in turn makes it easier to separate the sources (Bregman, 1990). Also note that, as in many other sensory systems, sound perception generally follows the Weber-Fechner law (Weber, 1846; Fechner, 1860), and thus the perceptual scale of frequencies (or pitches) is often chosen in the logarithmic scales (e.g., in units of octave as in “piano keys” or in units of Mel as measured in psychophysical studies; Stevens et al., 1937); likewise, the sound pressure level (SPL) is also often expressed in the logarithmic scale where 0 dB SPL approximately corresponds to the threshold of human hearing (the sound pressure of $\sim 20 \mu Pa$ or $\sim 10^{-16}$ watt/cm$^2$; Smith, 2000), and approximately related to the perceptual “loudness” by a power law with a coefficient around 0.6 (Stevens, 1956).

1.1.2 Cochlear Nucleus

Cochlear nucleus is the first “relay station” that receives inputs mainly from the ipsilateral auditory nerve fibers and sends outputs to both ipsilateral and contralateral superior olivary nuclei (Kandel et al., 2000; Smith, 2000). The tonotopical organization is preserved in the cochlear nucleus, and neurons in the cochlear nucleus can be classified into several types based on their morphological features and their “response maps,” showing areas of excitation and inhibition plotted on the sound-level vs. frequency coordinates (Young, 1984). In addition, spiking patterns can also be classified into several categories on the basis of the shape of peri-stimulus time histograms (PSTHs; Kiang, 1975). These observations suggest that spectro-temporal, sound levels, and some other forms of coding schemes are already employed at this level (Oertel, 1991), but it remains to be addressed what kind of computations the cochlear nucleus performs.

1.1.3 Superior Olivary Nucleus

Binaural inputs first meet each other at superior olivary nucleus, a group of nuclei in the pons, which receives inputs from (anterior ventral) cochlear nuclei bilaterally and sends outputs to
in inferior colliculus through lateral lemniscus (Kandel et al., 2000; Konishi, 2003). It thus plays a very important role in sound localization by exploiting binaural cues, and is in fact one of the best characterized auditory stations from functional viewpoints.

The lateral superior olive exploits the interaural level differences (ILD)—a major cue in localizing high frequency sounds—where some cells are excited by ipsilateral sounds and inhibited by contralateral sounds (E-I cells), leading to faithfully encoding the intensity differences as small as 10 dB SPL. Other cells are responsive to similar variations in the sound intensities at both ears (E-E cells; Caird and Klinke, 1983).

The medial superior olive is involved in detecting the interaural time differences (ITD), which is particularly useful for locating low frequency sounds. It is believed to be the site of the coincidence detection originally proposed by Jeffress (1948), where a spatial array of cells receives inputs from both ipsilateral and contralateral sides but with different “wire length” from the two sides. This causes a certain internal delay for signals to reach, and thus the convergence of the signals from the two ears coincides only when the difference in these latencies matches exactly to the arrival time difference of the sounds between the two ears (see also Joris et al., 1998; Palmer, 2004).

Note that sound sources can be localized using monaural cues, although it is more difficult—and less efficient—than using binaural cues (Bregman, 1990). One such monaural cue is the spectral cue, where the detailed shape of the pinnae, head, and torso acts as a differential filter imposed on a source in a location-dependent manner (head-related transfer function; see also Section 2.1.2). This spectral cue would help determine the sources in vertical as well as horizontal axes, whereas the interaural time or intensity differences would mainly help localize the signals in the azimuthal plane. Although the use of spectral cues has been studied well in (human) psychophysics (Wightman and Kistler, 1989; Bregman, 1990; Hofman and van Opstal, 2002), it is not clear yet where and how they are encoded in the neural circuits and exploited—binaurally or monaurally—for auditory signal processing including sound localization (Knudsen and Konishi, 1979; Wenzel et al., 1993; Carlile et al., 2005).
1.1.4 Inferior Colliculus

Inferior colliculus is located in the midbrain where a tonotopical map is also preserved, and many cells show strong binaural responses and preferences to rather complex stimuli (such as amplitude- or frequency-modulations) but not to steady tones (Ryan and Miller, 1978; Kandel et al., 2000; Smith, 2000). Major ascending auditory pathway converges to this principal midbrain station of the auditory pathway, and thus it would play important roles in auditory scene analysis (Caird and Klinke, 1987; Davis, 2005). For example, inferior colliculus is responsive to interaural delay (Skottun et al., 2001) and may form a topographic as well as tonotopic map of the acoustic environment (FitzPatrick, 1975; Langner et al., 2002). In addition, some neurons in the inferior colliculus show sensitivities to spectral changes, which would help recognize specific phonemes and intonations in speech (Fitch et al., 1997). But it is unknown yet what exactly the inferior colliculus does and how it contributes to auditory scene analysis.

1.1.5 Medial Geniculate Nucleus

Medial geniculate nucleus is the final subcortical station before the auditory signals reach the (primary) auditory cortex (Clarey et al., 1992; de Ribaupierre, 1997; Suga et al., 2002; Jones, 2003; Suga and Ma, 2003). This auditory thalamus consists of many sub-nucleus types, distinguished on the basis of anatomical features and coding properties (Calford and Webster, 1981; Calford, 1983; Winer, 1985; Winer et al., 1999; He and Hu, 2002; Read et al., 2004). A tonotopical map is organized as in the inferior colliculus (Aitkin and Webster, 1971), and some cells show binaural responses whereas others show monaural responses, mainly to inputs from the contralateral side (Altman et al., 1970a,b; Cetas et al., 2002). Cells in the medial geniculate nucleus show a wide variety of response types; both broadly and narrowly tuned cells are observed (Aitkin et al., 1966; He and Hu, 2002), and some cells respond only to complex stimuli whereas others even show multi-modal responses, receiving visual and somatosensory inputs as well as auditory signals (Bordi and LeDoux, 1994a,b; Komura et al., 2005). Note
however that virtually nothing is known about computational aspects of this auditory signal processing station, even though extensive physiological research has been conducted over the past decades (for the earliest electrophysiological studies, see e.g., Rose and Galambos, 1952; Galambos et al., 1952; Galambos, 1952).

1.1.6 Primary Auditory Cortex

Primary auditory cortex is the first cortical station that receives inputs from the auditory thalamus and further processes the acoustic signals to make sense of them (for review, see e.g., Ehret, 1997; Schreiner et al., 2000; Read et al., 2002; Winer et al., 2005; King and Schnupp, 2007; for review on auditory processing in higher cortices, see e.g., Kaas et al., 1999; Griffiths and Warren, 2004; Griffiths et al., 2004). Representations of the signals would be most likely in the time-frequency domain, as suggested by the tonotopic organization (Merzenich et al., 1973, 1975), but auditory cortical neurons have a wide variety in receptive field sizes and show highly heterogeneous response patterns (Hromádka, 2007), and it is currently unclear what acoustic features these neurons respond to, or what stimulus would be optimal for exciting auditory neurons (but see approaches in deCharms et al., 1998; Machens et al., 2005; O’Connor et al., 2005; and also Sections 2.4.1 and 2.A).

Note however that there is no consensus on the definition of the “auditory cortex” because many functional subregions seem to exist. Among many areas—more than 10 divisions in cats as well as in monkeys—seemingly involved in the auditory signal processing, several “primary” areas have a tonotopic map (the primary auditory cortex, the anterior auditory field, and the posterior, ventral, and ventroposterior auditory areas; Phillips and Irvine, 1981; Schreiner and Mendelson, 1990; Schreiner et al., 1992; Mendelson et al., 1993, 1997; Eggermont, 1998), suggesting their roles in spectro-temporal—and binaural—analysis based on the information directly inherited from subcortical areas (Miller et al., 2001). In contrast, some non-tonotopic areas (the secondary auditory cortex and the suprasylvian fringe) typically have broader tuning curves and longer durations, and are thought to be involved in process-
ing communication signals and non-spectral stimuli (Schreiner and Cynader, 1984; He et al., 1997; Rauschecker and Tian, 2000), whereas others (the limbic-related fields and the posterior ectosylvian fields) receive not only auditory but also visceral and visual inputs, and would process multi-modal information (Kelly, 1973, 1974; Colavita et al., 1974; Bowman and Olson, 1988a,b; Kayser et al., 2005). Note that the rodent auditory cortex seems to be organized in a slightly simpler manner (Kelly and Sally, 1988; Sally and Kelly, 1988; Doron et al., 2002; Polley et al., 2007), and rats were used as experimental animals in this thesis (Chapter 3).

With respect to computations, the primary auditory cortex has traditionally been considered as stimulus feature—or, spectro-temporal “edge” (Fishbach et al., 2001, 2003)—detectors (Nelken et al., 2003), and linear models have been widely used to characterize the response patterns in the auditory cortex (e.g., Eggermont et al., 1983; Kowalski et al., 1996; Klein et al., 2000; Depireux et al., 2001; Theunissen et al., 2001; Escabí and Schreiner, 2002; Linden et al., 2003; Machens et al., 2004). However, such oversimplified models do not fully capture the computational properties of auditory cortical processing in general (see also Chapter 3).

1.2 Characteristics of Auditory Signal Processing

Computational problems faced by different sensory modalities share many characteristics; e.g., signals are processed by neural circuits in units of spikes (McCulloch and Pitts, 1943; Barlow, 1972), be they visual, tactile/somatosensory, or auditory signals. In addition, the fact that the auditory cortex is able to process and “see” visual information after rewiring visual projections from the retina to the ascending auditory pathway (Sharma et al., 2000) suggests that the principles underlying the computations would most likely be the same; i.e., complex input signals are transformed into rather simple context-invariant features, which in turn collectively form biologically meaningful representations and interpretable objects (for the auditory system, see e.g., Nelken et al., 2003; Nelken, 2004; for the visual system, see e.g., Carandini et al., 2005; Rust and Movshon, 2005). This common framework then gives a justification of
the conventional systems neuroscience approaches—even though we often treat the brain as a “black box” that transforms input signals into output behavioral responses, the signals must be transformed in such a way that the representations become somehow more easily interpreted (or decoded) by us experimenters (or by the “homunculus;” Penfield and Rasmussen, 1950) as the signals are transmitted to “higher” processing stages (e.g., Bialek et al., 1991; Rieke et al., 1997; deCharms and Zador, 2000; Dayan and Abbott, 2001; Shamma, 2001; Pouget et al., 2000, 2003).

The difficulties in such sensory signal processing problems reside in; (1) animals are bombarded with continuously changing high-dimensional inputs, and thus must deal with the “curse of dimensionality” (Bellman, 1961) and the “frame problems” (McCarthy and Hayes, 1969); and (2) associated computations to extract meaningful information are often ill-posed, and thus appropriate constraints must be imposed to guarantee the compatibility, uniqueness, and continuity of the solutions (Marr, 1982). To decipher how the brain performs sensory inferences, as Marr (1982) pointed out in his seminal work, (at least) three levels of the understandings would be required;

(1) computational theory to identify the logics and goals of computations;

(2) representation and algorithm to achieve the computations and appropriate transformations of input signals; and

(3) hardware and implementation to physically realize the algorithm in neural circuits.

In this thesis, on the one hand, the theoretical approach works on the middle level and Chapter 2 demonstrates how auditory streams can be segregated by exploiting the idea of sparse overcomplete representations as a working principle. On the other hand, the experimental approach works on the bottom level, trying to characterize the properties of neural dynamics in the auditory cortex for building a plausible encoding model at the single-cell level (Chapter 3).

Since Marr (1982) formulated the basic framework of theoretical analysis on sensory signal processing problems, theoretical and computational research on the visual systems has
been expanded and leading ahead of the analysis on any other modality, including the auditory systems. This would be because (1) we humans are visually-oriented animals; (2) the visual systems research has attracted more scientists; and (3) we have accumulated more physiological and psychological evidences on the visual systems for historical reasons (Palmer, 1999; Kandel et al., 2000; Smith, 2000). Consequently, our current knowledge on the sensory signal processing in the brain comes mainly from the extensive research on the visual system, which potentially results in a biased view on the underlying mechanisms. We thus have to be aware that each sensory system has its own specific problems, even though the strategy or the general principles could be the same among all modalities in a broad sense (see also Chapter 4).

The auditory signal processing problems are distinct at least in the following three respects. First, an acoustic environment we encounter often consists of sounds from a rich combination of sounds, and behaviorally relevant information can be masked by the many irrelevant sounds—or background noise—that may even constitute the majority of the acoustic energy received. Therefore, the auditory system must segregate such overlapped signals in time into individual streams, as opposed to the visual system that must infer occluded parts of superimposed objects in space.

Second, sensory processing would be facilitated by transforming signals into an appropriate feature domain—e.g., time-frequency representations for acoustic signals as is most likely achieved at the cochlea (Section 1.1.1)—but distinct neural circuit structures at the subcortical level suggest that each sensory modality has its own ways for pre-processing the signal to achieve a seemingly common goal; i.e., based on the information from the receptor surface activity, subcortical stations compute several aspects of the received signals that are (1) in some sense optimized for representing and analyzing the environment, and (2) readily used for subsequent processing in the cortex to make sense of the external world. For example, the fact that the auditory system has many stations at the brainstem would reflect the computational complexity of the pre-processing and a large size of feature dimensions required to extract before reaching the auditory cortex—e.g., interaural time, level, and spectral differences are exploited.
at the subcortical level for sound source localization (Joris et al., 1998; Konishi, 2003; Carlile et al., 2005; see also Section 1.1.3). In contrast, the many receptor types in the somatosensory system (Kandel et al., 2000; Smith, 2000) and the complex local retinal processing schemes in vision (Masland, 2001) seem to accomplish the task of generating sufficient feature dimensions without equally extensive processing at the brainstem itself.

The third feature of auditory signal processing is that sound pressure waves evolve in time and thus there is (almost) no information on “instantaneous” signals, suggesting that the temporal integrations of the received signals are required. What is more, auditory signals inevitably disappear shortly due to their physical properties, and thus precise and quick processing would be needed to extract meaningful information out of such transient signals. In contrast, visual objects are often robust in time, and thus spatial processing would be more critical than temporal processing for the visual systems, even though temporal information is also important to perform some visual tasks (e.g., motion detection).

These characteristics of auditory signal processing in fact underlie the motivations of this thesis. The theoretical work in Chapter 2 is motivated foremost by the computational demands of source separation—or, how to extract features from sound ensembles and use them for computations—and the experimental approach in Chapter 3 aims at deciphering the temporal dynamics of auditory cortical neurons especially because psychophysical studies show that stimulus integration over time is critical for acoustic signal processing (Bregman, 1990); for details of the specific motivations, see the introductory sections in each chapter. Although in this thesis I have concentrated on a restricted form of the challenges in the auditory signal processing problems and the two approaches hardly met each other yet, I believe that such complementary approaches are required for better understanding computations in the brain; see Chapter 4 for more general discussions on this topic and future challenges.
Chapter 2

Top-down characterization: population level analysis

This chapter explores top-down theoretical characterizations of neural behaviors at the population level: specifically, the idea of sparse overcomplete representations is exploited to develop a model for monaural blind source separation (Asari et al., 2006, 2007). The main goals here are to explore a model of computation with sparse representations, and to derive new experimentally testable predictions from this model, which in turn can be used to verify or falsify the model with experimental data. To make the model concrete, we consider a specific computation—a special case of the monaural cocktail party problem in which the head-related transfer function (HRTF) provides the critical cue for disentangling sources (Bregman, 1990). We focus on this special case not because it is of central importance from a psychophysical perspective—in a general setting, the HRTF is typically just one of many cues, and often not the most important—but rather because this problem provides a convenient way to illustrate the key predictions. The same sparse framework can be generalized to exploit other cues for source separation, and applied to other sensory processing problems (e.g., vision) as well.

In order to extract behaviorally relevant information embedded in natural acoustic environments, animals must be able to separate auditory streams originating from distinct acoustic
sources (“cocktail party problem;” Cherry, 1953). The auditory cortex has orders of magnitude more neurons than the cochlea (Kandel et al., 2000), and here we show how this anatomical feature can contribute to computation by selecting the sparsest representation, instead of merely to overcome neuronal noise as is often assumed (Pouget et al., 2000, 2003). The model makes testable predictions about the dynamic nature of representations in the auditory cortex, and the successful source separation supports the hypothesis that sparse representations directly subserve computations of interest in the brain.

This chapter is organized as follows. First, Section 2.1 explains a rationale for using sparse overcomplete representations as a model of neuronal sensory processing. I also give a brief overview and research history on blind source separation (BSS; for more comprehensive review, see e.g., Choi et al., 2005; Divenyi, 2005; O’Grady et al., 2005). In Section 2.2, we formulate a general framework for the monaural BSS problem. Section 2.2.1 then introduces the “dictionary method” approach, which is extended in Section 2.2.2 to exploit additional segregation cues; e.g., the HRTF in this study. In contrast to much previous work (e.g., Knudsen and Konishi, 1979; Wightman and Kistler, 1989; Wenzel et al., 1993), the HRTF is used here to separate auditory streams rather than to localize them in space; the model assumes that the locations of the sources have already been determined by other mechanisms. As described in Section 2.2.3, sparse representations of given signals in an overcomplete basis are achieved by $L_1$-norm minimization, and non-negative matrix factorization (NMF) is used in Section 2.2.4 for finding such dictionary elements suitable for sparse representations of given acoustic sources (see also Appendix Section A.1.4; for other learning algorithms, see Lewicki and Sejnowski, 2000; Smith and Lewicki, 2005, 2006; Pearlmutter and Olsson, 2006)—these algorithms are employed just for convenience, and it is beyond the scope of this study how the brain can actually achieve sparse overcomplete representations. Section 2.3 shows the separation results, where the HRTF-based method is applied to digital mixtures of three sources positioned at three distinct locations in space. Section 2.4 then describes several testable predictions drawn from the model, one of which was examined and found compatible with ex-
perimental data (Section 2.4.2) under the assumption that multiple single-unit recordings from multiple animals can be considered as equivalent to multi-unit recordings from individual animals. Finally, I discuss perspectives and plausibility of sparse overcomplete representations as a generic model for signal processing in Section 2.5.

2.1 Sparse Overcomplete Representations

A striking feature of many sensory processing problems is that there appear to be many more neurons engaged in the internal representations of the signal than in its transduction. For example, humans have only about 30,000 cochlear neurons, but at least a thousand times as many neurons in the auditory cortex (Kandel et al., 2000). Assuming that the “representational fidelity”—the amount of information that can be represented by a single spike—of neurons in cortex is comparable to that at the periphery, the “representational capacity” of cortex is far in excess of what is needed to form merely a complete representation. In other words, the cortical representation of sensory stimulus is overcomplete in the sense that many more neurons are available than are needed to represent the stimulus with high fidelity (Olshausen and Field, 1997; Lee et al., 1999; Lewicki and Sejnowski, 2000; Zibulevsky and Pearlmutter, 2001). Such apparently redundant internal representations have sometimes been proposed as necessary to overcome neuronal noise (Pouget et al., 2000, 2003). Here we instead posited that they would contribute to computations in the brain.

But how does the brain choose a unique representation if many different patterns of cortical activity could all faithfully represent any given pattern of neural activity at the periphery? We propose that the cortex exploits this excess “representational bandwidth” (DeWeese et al., 2005)—or, the excess degrees of freedom—by selecting the sparsest representation within an overcomplete basis set. Sparseness provides a powerful and useful constraint on neural activities. It is biologically appealing because such representations are metabolically efficient (Levy and Baxter, 1996; Laughlin and Sejnowski, 2003; Lennie, 2003), and the principle of sparse
(or “efficient;” Barlow, 1961, 2001) coding has been used to predict receptive field properties of both auditory and visual neurons (Olshausen and Field, 1996, 1997; Bell and Sejnowski, 1997; Simoncelli and Olshausen, 2001; Lewicki, 2002; Klein et al., 2003; Smith and Lewicki, 2005, 2006). In addition, there is growing evidence that natural auditory (DeWeese et al., 2003; Machens et al., 2004) and visual (Baddeley et al., 1997; Vinje and Gallant, 2000) stimuli activate only a relatively small number of neurons (Olshausen and Field, 2004). Thus the representational sparseness should be viewed as at least provisionally consistent with the current experimental evidence about cortical representations.

The motivation for sparseness in this thesis, however, originates not just from the coding (or metabolic) efficiency per se (Laughlin, 2001), but rather from the computational demands of source separation. Section 2.2 shows that constraining neural activity to be sparse selects one of the possible representations for a given stimulus, and Section 2.3 demonstrates that the resulting pattern of neural activity in fact solves the source separation problem, even when multiple sources are audible to only a single ear. This supports the idea that sparse representations may underlie efficient computations in the auditory cortex. Furthermore, the framework is quite general and can serve as a starting point for understanding how cortical circuits might exploit other sensory cues as well.

2.1.1 Blind Source Separation: Overview

Animals in nature confront an acoustic environment consisting of sounds from a rich, indeed often bewildering, combination of sources. Survival depends on responding appropriately to potential threats, food sources, and mates (e.g., at a cocktail party), while at the same time ignoring all the irrelevant sound sources that may constitute the majority of the acoustic energy received. Source separation, or “stream segregation,” is therefore one of the central problems in acoustic processing that organisms must solve. Animals must face many of the same challenges in solving this and other sensory processing problems as do artificial systems, and the insights gained from the one can be applied to the other. However, little is currently known about how
animals solve this problem (but see Fishman et al., 2004; Micheyl et al., 2005), and no artificial system can solve it in a general setting.

Animals exploit a variety of binaural and monaural cues to separate acoustic sources (Bregman, 1990). For example, two tones occurring simultaneously are more likely to be grouped together perceptually—i.e., perceived as arising from the same source—than the same notes occurring sequentially. Such grouping makes sense under the assumption that the auditory system is trying to discover the statistically independent causes of the acoustic signals received at the ears (Bell and Sejnowski, 1995, 1997; Lewicki and Sejnowski, 2000; Simoncelli and Olshausen, 2001); simultaneous onset of two tones is unlikely to arise purely by chance, and thus it is more parsimonious to assume that the tones were caused by a single source (e.g., as harmonics of a single fundamental frequency). Many of the spectral, temporal and spatial cues used for stream segregation can be interpreted in this context.

The earliest approach to the blind source separation (BSS) problem in artificial systems research dates back to 1986, where Hérault and Jutten used a recurrent neural network to separate instantaneous linear mixtures of non-Gaussian independent sources received at multiple sensors (and further developed in Jutten and Hérault, 1991). The key assumption made in this seminal work was independence—the concept of independent component analysis (ICA) was most clearly stated in Comon (1994). But additional constraints are needed in general on the probability distribution of the sources; i.e., non-Gaussianity.¹ The idea of non-Gaussianity was later exploited by Hyvärinen and Oja (1997) to develop the “FastICA” algorithm (see also Appendix Section A.1.3).

In parallel to BSS approaches, Linsker (1989) proposed unsupervised learning rules based on information theory, where the goal was to maximize the mutual information between the inputs and outputs of a neural network. Mutual information is in fact a natural measure of independence, and Comon (1994) showed that minimizing the mutual information between the sources is equivalent to maximizing their non-Gaussianity. Bell and Sejnowski (1995) then put

¹The BSS problems has no general solution if we assume Gaussian sources (e.g., Hyvärinen and Oja, 2000).
the BSS problem into the framework of information theory, and demonstrated the separation and deconvolution of mixed sources using stochastic natural gradient learning rules, originally proposed by Amari et al. (1996). BSS research in the early ages focused on even- or over-determined problems where the number of sensors/microphones is equal to or larger than the number of sources, respectively. In such cases, it is generally sufficient to simply assume that the (non-Gaussian) sources are statistically independent, and thus most approaches focus on recovering an “unmixing matrix” which inverts the “mixing matrix” governing the weighting of each source at each sensor (see also Comon et al., 1991; Belouchrani et al., 1997; Amari and Cichocki, 1998).

In under-determined cases, however, the problem is degenerated and such approaches fail: thus assumptions stronger than simple independence are required (Cauwenberghs, 1999; Jang and Lee, 2003). The first approach was proposed by Belouchrani and Cardoso (1994) where maximum a posteriori (MAP) estimation method was used for separating discrete sources. Particularly difficult is the monaural case, and in a broad sense, the approaches for monaural separation can be classified into the following three categories.

(1) In model-based techniques (Roweis, 2000; Reyes-Gomez et al., 2004; Benaroya et al., 2006; Ellis, 2006; Radfar et al., 2006), patterns of the sources are first obtained in the training phase. Then those patterns whose combinations well constitute the signals are selected, and used to estimate the underlying sources directly or build filters for separation. The model-based approaches are similar to speech enhancement methods where the goal is to recover the target signal interfered with noise (Ephraim and Cohen, 2006).

(2) In dictionary methods (Lee et al., 1999; Lewicki and Sejnowski, 2000; Girolami, 2001; Hochreiter and Mozer, 2001; Zibulevsky and Pearlmutter, 2001; Benaroya et al., 2003; Li et al., 2004; Smaragdis, 2004; Pearlmutter and Olsson, 2006; Schmidt and Olsson, 2006), a set of (overcomplete) basis functions—or dictionary elements—is first obtained in the training phase by using factorization techniques such as ICA, NMF, and/or sparse
decomposition. Then the sources are projected onto the basis set under appropriate constraints on the coefficients—such as the sparseness prior—and the underlying sources are estimated, e.g., by maximum likelihood based on the coefficient distributions given the basis dictionaries. Note that the performance depends on the “personalized” dictionaries that exclusively encode one source signal but not the others.

(3) Computational auditory scene analysis (CASA) exploits the knowledge on the acoustic signal processing in humans and aims to replicate it (Bregman, 1990; Cooke and Brown, 1993; Brown and Cooke, 1994; Nakatani and Okuno, 1999; Hu and Wang, 2004; Li et al., 2006). That is, the signals are first transformed into an appropriate time-frequency representation, and then sound elements are grouped into the underlying sources based on spectral, temporal and spatial cues; e.g., common onset/offset time, comodulation of stimulus power, fundamental frequency and harmonics, and so on (“regularities” in auditory streams; Bregman, 1993). Note however that it remains to be addressed how the auditory system actually performs this grouping process (see also Section 1.1), and thus it is often (over)simplified in current approaches.

The approach in this study can be considered as a combination of the second and the third approaches; i.e., to adopt a practical computational framework for the cocktail party problem based on the dictionary methods, and exploit one particular sort of monaural segregation cues that animals use, specifically, the spectral cues introduced by the differential filtering imposed by the HRTF (see below). Section 2.2 describes this model in detail, and Section 2.3 provides examples of applications for segregating auditory streams perceived monaurally.

2.1.2 Head-Related Transfer Functions

The auditory system uses a wide variety of psychophysical cues to segregate auditory streams (Bregman, 1990), including both binaural and monaural cues. Many monaural cues have been identified—such as common onset time or comodulation of stimulus power in different parts of
the spectrum—but for simplicity, here we focus on just one set of cues; those provided by the differential filtering imposed on a source by its path from its origin in space to the cochlea. This filtering—or “spectral coloring”—is caused both by the head and the detailed shape of the ear (the head-related transfer function; HRTF), and by the environment on sources at different positions in space (e.g., by room reverberations). The HRTF depends on the spatial position—both the relative azimuth and elevation—of the source. At some frequencies, the HRTF can attenuate sound from one location by as much as 40 dB more than from another, and such HRTF cues help in source separation when present (Yost et al., 1996). Although every individual has his or her own HRTF, the basic characteristics of HRTFs are similar across individuals. Here we used a representative left human pinna HRTF downloaded from http://www.itakura.nuee.nagoya-u.ac.jp/HRTF/ (Nishino et al., 2001; note that many other HRTF databases are available elsewhere, e.g., Gardner and Martin, 1994; Algazi et al., 2001).

The HRTF is also important for generating a three-dimensional experience of sound, so that acoustic sources that bypass the HRTF (e.g., those presented with headphones) are typically perceived unnaturally, as though arising inside the head (Wightman and Kistler, 1989; Kulkarni and Colburn, 1998). Note however that the HRTF is used here to separate auditory streams rather than to localize them in space, in contrast to much previous work on the role of the HRTF in sound localization (Knudsen and Konishi, 1979; Wightman and Kistler, 1989; Wenzel et al., 1993; Hofman and van Opstal, 2002). Also note that spectral cues are not strictly required for sound localization; binaural cues can provide robust cues even in the absence of spectral cues (see Section 1.1.3). Conversely, source separation can proceed when spectral cues are weak—or indeed, even when spatial cues are completely absent, as for example when picking out a violin from within a concerto played over a single speaker. This illustrates a general principle: no single cue is essential to source separation, and the auditory system will promiscuously exploit any available cues.

Nevertheless, it is often reasonable to assume that sound arriving from different locations should be treated as arising from distinct sources. In this work, we thus assume that all
sounds from a given position are defined to belong to the same source, and any sounds from a
different position are defined to belong to different sources. We emphasize that although sound
localization (the process by which an animal determines where in space a source is located) is
related to source separation (the process by which an animal extracts different auditory streams
from a single waveform), the two computations are distinct; neither is necessary nor sufficient
for the other. Here we focus only on the separation problem, and assume that source localiza-
tion occurs by other mechanisms.

2.2 Source Separation Model: Problem Formulation

The acoustic signals we hear are in most cases a mixture of sounds coming from multiple
sources. Thus it is rare that we can listen to an acoustic source without interference from
other sources, but our auditory system filters the “noise” out of our conscious perception so
effectively that we are often almost unaware of the interference. This apparent effortlessness is
however deceptive; no artificial system can yet solve the source separation problem in a general
setting even in this prosperous age of information technology.

Suppose there are \( P \) acoustic sources located at known distinct positions in space, with
\( x_i(t) \) being the time course, \( t \), of the stimulus sound pressure of the \( i \)-th source at its point of
origin. Associated with each position is a known filter given by \( h_i(t) \). In what follows \( h_i(t) \) will
be considered as the HRTF, but in general \( h_i(t) \) will include not just the filtering of the head
and external ear, but also the filter function of the acoustic environment such as reverberation.
The signal \( y(t) \) at the ear is then the sum of the filtered signals:

\[
y(t) = \sum_{i=1}^{P} h_i(t) * x_i(t) = \sum_{i=1}^{P} \tilde{x}_i(t),
\]  

(2.1)
where \( * \) indicates convolution and \( \tilde{x}_i(t) = h_i(t) * x_i(t) \) is the \( i \)-th source in isolation following filtering; i.e., \( x_i(t) \) is the \( i \)-th source measured in source space whereas \( \tilde{x}_i(t) \) is the same source measured in sensor space.

The organism’s goal in source separation is to recover the underlying sources \( x_i(t) \) from the signal \( y(t) \), using knowledge of the directional filters \( h_i(t) \). Note that the actual spatial locations of the sources—and thus corresponding filters \( h_i(t) \)—are not computed during the separation but assumed to be identified beforehand by other mechanisms. This particular monaural version of the separation problem is a special—more difficult—case of the binaural problem, or the multiple microphone case in artificial systems (see above Section 2.1.1). While the problem cannot be solved in general for all classes of sounds, we should be able to obtain solutions for certain types of source distributions because humans by and large have the ability to isolate what is being said even by a single ear in a cocktail party situation (Helmholtz, 1863; Bregman, 1990).

### 2.2.1 Dictionary Method Approach

As in the auditory system, the observed signal is first transformed into a time-frequency representation:

\[
2 \quad Y = \text{TF}\{y(t)\},
\]

e.g., by the short-time Fourier transform (STFT or spectrogram; see Eq. (3.25) on page 102) or the Gammatone filter bank (Bregman, 1990; Lewicki, 2002). For notational and computational convenience, here we discretize time and frequency, and restrict \( \text{TF} \) such that \( Y \) is a real-valued matrix with column vectors, \( y \in \mathbb{R}^N \), denoting the short segments of the spectrogram in time—specifically, to constrain \( Y \) to be non-negative, here we work on the power of the STFT unless otherwise indicated.

Certain types of sources will become less overlapped in the transformed domain, which in turn facilitates the separation of the signals. More generally, if the sources can be assumed sparsely distributed in the frequency domain, additivity is approximately preserved in the transform.

\[\text{In this dissertation, \textbf{boldface} is used to indicate vectors and matrices (in lower- and upper-case letters, respectively).}\]
formed mixture:

\[ y = \sum_{i=1}^{P} h_i \bullet x_i = \sum_{i=1}^{P} \tilde{x}_i, \quad (2.2) \]

where \( \bullet \) indicates elementwise multiplication, \( h_i \) is the HRTF in the frequency domain, and \( x_i \) and \( \tilde{x}_i \) are the transformed \( i \)-th source signals in source and sensor spaces, respectively. If instead the sources are densely distributed, it is less likely that the additivity assumption in Eq.(2.2) holds because sound waves can be interfered with each other. The separation performance will then be limited in this approach because the model here employs linear representations (see below).

Let us assume that each short segment (e.g., 5 msec) of each acoustic source in the time-frequency domain \( \tilde{x}_i \) (as it sounds at the cochlea) is represented by the sparsely distributed activities \( c_{ij} (> 0) \) of a population of neurons indexed both by components \( j = 1, \ldots, R_i \) and source positions \( i = 1, \ldots, P \):

\[ \tilde{x}_i = \sum_{j=1}^{R_i} \tilde{d}_{ij} c_{ij} = \tilde{D}_i c_i, \quad (2.3) \]

where the \( j \)-th column of \( \tilde{D}_i \) consists of neural feature \( \tilde{d}_{ij} \) for the \( \{ij\} \)-th neuron, and the \( j \)-th element of \( c_i \) holds the corresponding neural activity \( c_{ij} \). The “dictionary” \( \tilde{D}_i \) forms a (not necessarily orthogonal, and possibly overcomplete) linear basis, and \( c_{ij} \) is interpreted here as the spike rate during each time frame. From Eqs.(2.2) and (2.3), the signal \( y \) is then given by:

\[ y = \sum_{i=1}^{P} \tilde{x}_i = \sum_{i=1}^{P} \sum_{j=1}^{R_i} \tilde{d}_{ij} c_{ij} = \begin{bmatrix} \tilde{D}_1 & \cdots & \tilde{D}_P \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_P \end{bmatrix} = \tilde{D} c. \quad (2.4) \]
Here we allow the number of dictionary elements to be larger than the dimensionality of the signal ($\sum_i R_i \geq N$), and thus $\tilde{D}$ is potentially overcomplete; i.e., many possible decompositions/representations exist.

This “dictionary method” assumes a linear generative model based on the factorizations of each source $\tilde{x}_i$ in terms of dictionary sets $\tilde{D}_i$ and the corresponding coefficients $c_{ij}$, which suggests a linear relationship between an auditory stimulus and its neural representation in terms of the neural features and activities. This assumption of linearity is common in both visual and auditory physiology (e.g., Olshausen and Field, 1996, 1997; Lewicki, 2002). For example, a population of neurons in the primary visual cortex is often assumed to represent a visual scene in terms of a collection of oriented edges (Hubel and Wiesel, 1959); in this case the scene and the features in Eqs.(2.3) and (2.4) would be rewritten as functions of spatial rather than temporal coordinates, but the formulation would be otherwise identical. Similarly, in auditory physiology, stimuli are sometimes represented as a weighted sum of basis functions such as moving ripples (Kowalski et al., 1996; Klein et al., 2003); in the context of Eqs.(2.3) and (2.4), this implies assuming a one-to-one correspondence between a basis function (derived from the ripple basis) $\tilde{d}_{ij}$ and the firing rate $c_{ij}$ of a corresponding neuron.

The successful separation of the sources $\tilde{x}_i$ from the mixture $y$ using the dictionary methods (Eqs.(2.3) and (2.4)) requires:

1. a learning algorithm to obtain dictionary sets $\tilde{D}_i$ that almost exclusively encode $\tilde{x}_i$ but not the others $\tilde{x}_j$ for all $j \neq i$, and

2. a method to compute such coefficients $c$ (given $y$ and $\tilde{D}$) in Eq.(2.4) that allows the reconstruction of $\tilde{x}_i$ according to Eq.(2.3).

Finding good (overcomplete) dictionaries from training data is a subject of ongoing research (e.g., Lewicki and Sejnowski, 2000; Kreutz-Delgad et al., 2003). In particular, non-negativity constraints have been used to learn signal dictionaries for single-channel separation of audio.

---

3The underlying source $x_i$ can then be obtained by the deconvolution, if necessary, for $\tilde{x}_i = h_i \ast x_i$. 23
signals (Benaroya et al., 2003), and incorporating a sparseness prior (Pearlmutter and Olsson, 2006) or a convolutive extension of NMF (Smaragdis, 2004, 2007) have been particularly effective in this regard. However, this dictionary method approach does not work well when source signals are from identical distributions, or from different distributions with the same statistics (Jang and Lee, 2003). To resolve this problem, in the next Section 2.2.2, we will introduce additional assumption to exploit the HRTF as a separation cue that “tags” the dictionary elements so they can be assigned to the appropriate sources in the same framework. As for the second condition on the computation of appropriate coefficients in Eq.(2.4), Section 2.2.3 will show that linear programming can be employed to find the sparse decompositions for the source separation.

### 2.2.2 HRTF-Based Approach

In order to exploit the HRTF as a separation cue in the dictionary method framework, we further assume that each source $x_i$ can be expressed as a linear combination of (not necessarily orthogonal) basis elements $d_j$:

$$x_i = \sum_{j=1}^{\hat{R}} d_j c_{ij} = Dc_i,$$

(2.5)

where the basis elements $d_j$ in source space are related to the neural features $\tilde{d}_{ij}$ in sensor space by convolution with each filter $h_i$ (in the time-frequency domain):

$$\tilde{d}_{ij} = h_i \bullet d_j.$$

(2.6)

The basis elements $d_j$ reflect statistical correlations within sources; each source typically consists of several such elements. These basis elements can then be thought of as an internal model of the components of acoustic sources, in the same way that edges might be thought of as components of visual sources (objects). Because the neural representation involves pre-filtering with the HRTF (Eq.(2.6)), the coefficient $c_{ij}$ associated with feature $\tilde{d}_{ij}$ is then better thought
of as representing the hypothesis that an element \(d_j\) is present at position \(i\). In the same way, neurons in the (primary) visual cortex can be thought of as representing the hypothesis \(\tilde{d}_{ij}\) that an oriented edge \(d_j\) is present at a particular position \(i\) in the visual field. In other words, the elements \(d_j\) reflect the basic properties of neurons, which in turn reflects the statistical structure of stimulus, whereas the features \(\tilde{d}_{ij}\) arise from the top-down modulation on the elements \(d_j\) according to the position information \(h_i\) determined by other mechanisms (see also Section 2.4.4).

As before, the signal \(y \in \mathbb{R}^N\) received at the ear can then be expressed as a linear combination of the dictionary elements in sensor space \(\tilde{d}_{ij}\):

\[
y = \sum_{i=1}^{P} h_i \cdot x_i \quad \text{from (2.2)}
\]

\[
= \sum_{i=1}^{P} h_i \cdot \left( \sum_{j=1}^{R} d_j c_{ij} \right) \quad \text{from (2.5)}
\]

\[
= \sum_{i=1}^{P} \sum_{j=1}^{R} \tilde{d}_{ij} c_{ij} \quad \text{from (2.6)}
\]

\[
= \tilde{D}c. \quad (2.7)
\]

In the BSS model in Eqs.(2.5)–(2.7), the neural representation of the signal \(y\) is directly related to the underlying sources \(x_i\), and the separation procedures consist of the following two steps.

1. A set of dictionaries in source space \(D\) is learned from a training set of “unmixed” signals \(x_i\).

2. Given a convolutional mixture \(y\) and position-dependent filters \(h_i\), appropriate coefficients \(c_{ij}\) are obtained for Eq.(2.7) under a sparseness prior (see below Section 2.2.3). A given source \(i\) can then be reconstructed by summing over all basis elements \(d_j\) associated with position \(i\) using Eq.(2.5).
Note that we no longer have to use “personalized” dictionaries $\tilde{D}_i$ for each source $\tilde{x}_i$ as in Eqs.(2.3) and (2.4) but could use any dictionary set $D$ that captures the spectral correlations in the sources $x_i$ as in Eq.(2.5) and permits sparse representations where only a small number of coefficients $c_{ij}$ are significantly non-zero (i.e., only a small fraction of neurons are active) in Eq.(2.7). Also note that separation and deconvolution are simultaneously achieved here by estimating the coefficients using a post-HRTF (sensor space) dictionary $\tilde{D}$ but reconstructing the signals using a pre-HRTF (source space) dictionary $D$ (Figure 2.3); the reconstruction is therefore invariant to changes in stimulus position.

2.2.3 Neural Representation for Source Separation

A population of neural activities satisfying Eqs.(2.5)–(2.7) will effectively solve the source separation problem, since a given source $i$ can be reconstructed merely by summing over all neurons associated with position $i$. This formulation therefore recasts source separation into the following problem, “How could the brain find the appropriate neural activities?” In this study, the neural representation is assumed to be overcomplete (Olshausen and Field, 1997; Riesenhuber and Poggio, 2000), where many different neural activity patterns $c$ could represent the stimulus $y$ equally well (Figure 2.1A). Thus the model should impose additional constraints (regularizers) to choose a unique well-defined representation. Note however that the goal is not merely to represent the stimulus $y$, but to find a representation in which the underlying sources $x_i$ are apparent and from which they can be readily recovered.

Sparse Neural Representation

A biologically appealing constraint on the neural representation is a sparseness prior (Olshausen and Field, 1996, 1997; Bell and Sejnowski, 1997; Chen et al., 1998; Lee et al., 1999; Lewicki and Sejnowski, 2000; Vinje and Gallant, 2000; Simoncelli and Olshausen, 2001; Zibulevsky and Pearlmutter, 2001; Hahnloser et al., 2002; Olshausen and O’Connor, 2002),
leading to an energy-efficient representation (Levy and Baxter, 1996; Laughlin and Sejnowski, 2003; Lennie, 2003). Sparseness provides a mathematical instantiation of Occam’s Razor, stating that the simplest explanation (in some sense) is preferred, and is compatible with the “efficient coding” hypothesis (Barlow, 1961, 2001), according to which the goal of sensory processing is to construct an efficient representation of the sensory environment (see Section 2.1).

One interpretation of this sparseness assumption is to represent the acoustic stimulus \( y \) using the minimum number of spikes (Figure 2.1C); formally, \( c \) should be optimized to minimize its \( L_1 \)-norm: \( \|c\|_1 = \sum_{ij} |c_{ij}| \). Considering the balance between the sparseness prior and robustness to noise (or the accuracy of the fit), the neural representation \( c \) can be computed as:

\[
\arg\min_c \|c\|_1 \text{ subject to } \|\tilde{D}c - y\|_p \leq \beta \tag{2.8}
\]

where \( \beta \) is proportional to the noise level and with \( p = 1, 2, \) or \( \infty \). Letting \( \beta \to 0 \) is equivalent to assuming that the noise is very small, and the solution converges to the one with no noise. The Gaussian noise case, \( p = 2 \), can be solved by semidefinite programming methods (Fletcher, 1985), and both \( p = 1 \) and \( p = \infty \) can be solved by linear programming (e.g., the \text{linprog} routine in the MATLAB Optimization Toolbox). The solutions presented in Section 2.3 all used \( p = 1 \). For this case, noise vectors \( e^+ \) and \( e^- \) are introduced and Eq.(2.8) can then be rewritten in the standard form:

\[
e^+, c^-, e^+, e^- \geq 0
\]

\[
\tilde{D}c^+ - \tilde{D}c^- + e^+ - e^- = y
\]

\[
1^T e^+ + 1^T e^- \leq \beta
\]

where \( 1 \) is a column vector whose elements are all one. In the simulations, four different noise levels were examined (\( \log_{10}\|y\|_1/\beta = 1, 2, 3, 4 \)), and the one with the best separation performance on average was selected as the result (Figures 2.4 and 2.5).

\(^4\textit{Pluralitas non est ponenda sine necessitate} \text{ (William of Ockham; Thorburn, 1918).}\)
Minimizing sparseness in the $L_1$-norm sense is not the only possible choice. One natural alternative is the $L_0$-norm: $\|c\|_0 = \sum_{ij} c_{ij}^0$ if we define $0^0 \overset{\text{def}}{=} 0$, which minimizes the total number of active neurons rather than the total number of spikes. Although this constraint also seems biologically sensible, it leads to a computationally intractable (NP-complete) combinatorial problem (Donoho and Elad, 2003); moreover, in many cases it leads to the same solution as the minimum $L_1$-norm solution (Li et al., 2004), particularly in the presence of a noise model. Therefore, only the “$L_1$ solution” is considered here and the “$L_0$ solution” is not pursued.

**Dense Neural Representation**

An alternative regularizer is that implicit in the pseudoinverse (Strang, 1988)—a dense prior—corresponding to the least-squares solution (see Appendix Section A.2). Bounding the total amount of noise as in Eq.(2.8), we have:

$$\arg\min_c \|c\|_2^2 \text{ subject to } \|\tilde{D}c - y\|_p \leq \beta.$$  (2.9)

The pseudoinverse $\tilde{D}^\dagger$ finds the solution $c$ (for $\beta = 0$ and $p = 2$) that minimizes the $L_2$-norm, i.e., the squared neural activity: $\|c\|_2^2 = \sum_{ij} c_{ij}^2$ (Figure 2.1B). However, it is not obvious why it would be useful for the brain to minimize this quantity, which has units of “spikes-squared,” rather than some other quantity such as “spikes.” Moreover, we show in Section 2.3 that it fails in practice to separate the sources successfully.

Pseudoinverses ($L_2$-norm minimization) can be computed with the `pinv` routine in MATLAB, which uses an algorithm based on singular value decomposition (SVD; see also Appendix Section A.1.2). In the simulations, a non-negativity constraint was not imposed on the coefficients; as a result, the dense solution consists of negative coefficients as well as positive ones, whereas all the substantially non-zero elements are positive for the sparse solution. Figure 2.6 on page 39 then shows the absolute values of the dense solution coefficients.
Figure 2.1: Overcomplete representation in two dimensions. (A) Three non-orthogonal basis vectors (neural features) $\tilde{d}_{ij} \in \mathbb{R}^2$ constitute an overcomplete representation, offering many possible ways to represent a data point $y$ with no error. (B) The conventional solution is given by the pseudoinverse (Eq.(2.9)), yielding a dense neural representation where the squared sum of the coefficients (neural activities): $\|c\|^2 = \sum_{ij} c_{ij}^2$, is minimized. This representation invokes all neural features about evenly. (C) The sparse solution (Eq.(2.8)) invokes at most two neural features because it minimizes: $\|c\|_1 = \sum_{ij} |c_{ij}|$. (D) Comparison of neural activity for the two cases. For the dense representation, all three neurons participate about equally, whereas for the sparse representation activity is concentrated in neuron 2. From Asari et al. (2006), with permission.
**Probabilistic Interpretation of Regularizers**

Interpreted probabilistically, the regularizers on neural representations correspond to maximum likelihood estimates using different *a priori* assumptions about the processes generating the stimuli, whose estimates are represented as the neural activities participating in a representation (Figure 2.1D; see also Appendix Section A.2.2). The pseudoinverse (Eq.(2.9)) assumes that the underlying causes represented by the activities \( c_{ij} \) were drawn from a Gaussian distribution: 

\[
p(c_{ij}) \propto e^{-c_{ij}^2},
\]

while the sparseness regularizer (Eq.(2.8)) assumes a Laplacian distribution:

\[
p(c_{ij}) \propto e^{-|c_{ij}|}.
\]

Because a Laplacian distribution has more elements very close to—and very far from—zero than does a Gaussian with the same variance, it corresponds to a sparser description in terms of \( c_{ij} \). Without the noise term, the maximum likelihood estimates using any prior yields perfect (zero reconstruction error) representations of the stimulus; the prior here is on the distribution of the underlying causes represented by the coefficients \( c_{ij} \), rather than on the distribution of reconstruction errors (as for example in robust fitting methods). Only when a noise term is added do the neuronal activities \( c_{ij} \) cease to represent the stimulus perfectly.

### 2.2.4 Dictionary Learning

Successful source separation in the framework of Eqs.(2.5)–(2.7) requires that two conditions be satisfied. First, the sources must be sparsely representable, as is the case with natural auditory stimuli (Attias and Schreiner, 1997; Lewicki, 2002; Klein et al., 2003; Smith and Lewicki, 2005, 2006). Second, the sources must have spectral correlations matched to the HRTF. Therefore, it is critical to obtain appropriate dictionary sets (from training data samples) that are (1) suitable for sparse representations, and (2) sufficiently discriminative by themselves, or become distinct by the filtering process (Eq.(2.6)).

Non-negative matrix factorization (NMF) is used here to generate a set of basis elements from spectrograms obtained from samples of various audio sources (Figure 2.2A; solo
instrumental music, natural sounds and speech). NMF is an algorithm for factorizing a data matrix under non-negativity constraints (Lee and Seung, 1999; see also Appendix Section A.1.4). In contrast to some other decomposition approaches, such as principal component analysis (PCA), NMF often yields representations in which the elements are fairly local even without explicitly imposing a sparseness prior on the coefficients, which can be interpreted as “parts” (see also Appendix Section A.1.5).

When applied to music, NMF typically yielded elements suggestive of musical notes, each with a strong fundamental frequency and weaker harmonics at higher frequencies. In many cases, listeners could easily use timbre to identify the instrument from which a particular element was derived. When applied to sounds from other ensembles (natural sounds and speech), NMF yielded elements that had rich harmonic structure, but it was not in general easy to interpret the elements (e.g., as vowels). Nonetheless, these elements captured aspects of the statistical structure of the underlying ensemble of sounds, and led to sparse representations of the ensembles (Figure 2.2B).

The choice of NMF in this context was merely a matter of convenience; any basis could be used as long as it captures the spectral correlations in the sources and permits a sparse representation. For simplicity, here we did not explicitly impose the $L_1$-sparseness prior on the learning rules (Appendix Section A.1.4) and thus NMF is not necessarily the optimal algorithm for our model, or the “algorithm” by which features are established in real neural circuits—such features must surely arise through a complex interaction of genetic and environmental cues. Then we need not expect to find a precise correspondence between the features obtained by NMF and those observed in the auditory cortex. In this respect, the results in this study complement previous work on finding the features underlying auditory or visual scenes (Bell and Sejnowski, 1997; Olshausen and Field, 1996, 1997; Lewicki, 2002; Schwartz and Simoncelli, 2001); the emphasis here is not on the elements themselves, but rather on how they work together to form a representation that separates sources.
Figure 2.2: **Non-negative matrix factorization (NMF) can be used to find the parts of sound ensembles.** (A) NMF basis elements for three sound classes (music, natural sounds, and speech) were aligned in columns by the peak frequency. Power is concentrated in the fundamental frequency, but higher harmonics are clearly visible. Note that each column, which reflects statistical correlations present in the sources, is an example of $d_j$ in Eq.(2.5); it is the filtered versions $\tilde{d}_{ij}$ in Eq.(2.6) that form the neural representation as in Eq.(2.8). (B) The ability of the NMF bases in (A) to represent sounds is quantified in terms of the “sparseness index,” defined as: $\|c_i\|_0 / \text{dim } x_i$, in the presence of a single (unmixed) source $x_i$. Here this index is unity for a dense representation ($\text{dim } c_i / \text{dim } x_i = R/N = 1$; see also Eq.(2.5)), and approaches zero as the representation becomes sparser. The distribution of the index was $0.61 \pm 0.27$, $0.64 \pm 0.17$, and $0.49 \pm 0.13$ (median ± interquartile range) for music, natural sounds, and speech, respectively, over 10,000 test samples. From Asari et al. (2006), with permission.
Simulation Procedures

In the simulations, the spectrograms in the data matrix $X$ (as in Eq.(A.1) on page 136; for details, see Appendix Section A.1) were obtained from music sounds, natural sounds, or speech sounds; commercial audio CDs (instrumental solos; classical and jazz, one each on cello, clarinet, trumpet, harp, and harpsichord), the audio CDs *The Diversity of Animal Sounds* and *Sounds of Neotropical Rainforest Mammals* (Cornell Laboratory of Ornithology, Ithaca, NY, USA), and spoken poetry (Dylan Thomas, T. S. Eliot, Frank O’Hara and William Butler Yeats on the commercial audio CD *Poetry speaks; Hear great poets read their work from Tennyson to Plath*, Sourcebooks Inc., 2001, ISBN 1570717206), respectively. Samples of 100–150 sec were taken, stereo channels averaged, and the signal down-sampled from the original 44.1 kHz to 8 kHz. Log-scaled spectrograms were generated (using a custom MATLAB routine) with a bin size of 5 msec and 75 frequency bands ranging from 55–3,951 Hz in steps of 1/12 octave. Each column of $X$ held a strip of spectrogram, yielding a dimensionality of $N = 75$, and $M = 5,000$ samples were used for the training (for the training algorithm, see Eqs.(A.47) and (A.48) on page 146). The training samples were distinct from those used for the testing; specifically, we used 10,000 samples to assess the representational sparseness achieved by the NMF basis (Figure 2.2), and 20,000 random combinations of three sources (out of the 10,000 samples in Figure 2.2) to assess separation performance (Figure 2.4).

Each NMF run (with the “factorization rank”—or, the number of basis elements that one expects underlie a given data set—being $R = 15$; for details, see Appendix Section A.1) consisted of 500 iterations with 10 restarts from random initial conditions, with the restart that yielded the minimum total error chosen. The five basis matrices $A$ obtained by NMF for each individual source were concatenated to form a (complete) source-space basis matrix of $\bar{R} = 5R = 75$ basis elements: $D = [ A_1 | A_2 | \cdots ]$. Each column of $D$ was then filtered through each of three different HRTFs ($P = 3$), resulting in a feature matrix with $P\bar{R} = 225$ columns: $\tilde{D} = [ h_{1}^{\top} \cdot D | \cdots | h_{P}^{\top} \cdot D ]$. The source locations were randomly chosen but 90° apart from each other in the simulations (e.g., in Figure 2.3 the three sources were located
on your left, center, and right, corresponding to the HRTFs for azimuth $-90^\circ$, $0^\circ$, and $90^\circ$, respectively). The analyses on the natural sound and speech sound were performed in a similar manner, with 5,000 training samples for each data matrix $X$.

The ability of the NMF dictionaries to represent sounds can be quantified in terms of the “sparseness index” (Figure 2.2), defined as: $\|c_i\|_0 / \dim x_i \in (0, R/N = 1]$, in the presence of a single (unmixed) source $x_i$ (see also Eq.(2.5); in practice, $\|c_i\|_0$ was computed with some tolerance, so the number of elements larger than $1 \times 10^{-5}$). The noise level was $\log_{10}\|y\|_1/\beta = 1$ in Figure 2.2B, resulting in the reconstruction signal-to-noise ratio (SNR) of $18.3\pm3.8$, $16.0\pm3.0$, and $18.0\pm3.6$ (median ± interquartile range in dB) for music, natural sound, and speech ensembles, respectively. The sparseness index approaches zero as the representation becomes sparser, and the distribution of the index was $0.61\pm0.27$, $0.64\pm0.17$, and $0.49\pm0.13$ (mean ± interquartile range), respectively, over 10,000 test samples. This suggests that the NMF dictionaries generally led to sparse representations of the ensembles.

### 2.3 Monaural Separation: Results

This section demonstrates the model’s ability to separate sources using the dictionary elements obtained by NMF (Section 2.2.4). Specifically, the performance was examined with the digital mixtures of three sources located at three distinct positions in space (Figure 2.3). On the left column are the spectrograms of the sources at their origin. Here, two of the sources (a harp playing the note “D”, center and bottom) were chosen to be identical; therefore, this example is particularly challenging because the only cue for separating the sources is the filtering imposed by the HRTF.

Separation was nevertheless quite successful (compare left and right columns). These results were typical; whenever the underlying assumptions about the sparseness of the stimulus were satisfied, sources consisting of mixtures of music, natural sounds, or speech were all separated well (Figure 2.4). Separation worked particularly well for mixtures of sparsely rep-
resentable sources (i.e., smaller sparseness index values), whereas it did not work for sources that were not sparsely represented (i.e., larger sparseness index values). Figure 2.5 shows that separation without differential pre-filtering by the HRTF was unsuccessful, as was separation using the Gaussian prior (dense representation) instead of the sparseness prior. For a measure of separation performance, the SNRs were computed as: \[ \langle \| x_i \|^2 / \| x_i - \hat{x}_i \|^2 \rangle_i \], where \( \hat{x}_i \) is the estimate of \( x_i \), and \( \langle \cdot \rangle_i \) indicates the average across sources.

The neural representations underlying separation provide insight into these results. Figure 2.6A shows the representations of each of the three sources (the same as in Figure 2.3) presented in isolation. In each panel, the activity in a population of 225 neurons (corresponding to the 225 features: \( \tilde{d}_{ij} = h_i \cdot d_j \)) is indicated by the intensity of points on a 15×15 grid. Since the sources occupy three positions (\( i = 1, 2, 3 \)), there are three copies of the basis \( d_j \) in each panel (corresponding to the three filters \( h_i \)). The activity patterns are sparse; only a relatively small number of units are active in each representation. In this example, because the middle and the right sources (source 2 and source 3, respectively) were chosen to be identical, the middle and right neural representations differ mostly by a shift.

The procedure for recovering a source from such a representation is straightforward; the estimate of the left source (source 1) is simply the summed activity of the left third of the neurons—those representing features pre-filtered by the HRTF corresponding to the leftmost position in space; and likewise for the middle and right thirds. Thus the HRTF works as a “tag” for grouping together elements from a single source. This suggests that the procedure for source separation in our model conceptually consists of two distinct steps (although in practice the two steps occur simultaneously). In the first step, the stimuli are decomposed into the appropriate features. In the second step, the features are tagged and bundled together with other features from the same source. It is for this bundling or “tagging” step that the HRTF along with the prior knowledge of source locations is essential.
Figure 2.3: **Separation of three musical sources.** Three musical instruments $x_i$ at three distinct spatial locations were filtered (by $h_1, \ldots, h_3$, corresponding to the HRTFs for azimuth $-90^\circ, 0^\circ,$ and $90^\circ$ with zero elevation, respectively) and summed to produce the input:

$$y = \sum_i \tilde{x}_i = \sum_i h_i \cdot x_i,$$

and then separated using a sparse overcomplete representation to produce the output:

$$\hat{x}_i = \sum_j \hat{c}_{ij} d_j.$$ 

Two of the sources (a harp playing the note “D,” center and bottom) here were chosen to be identical; this example is thus particularly challenging, since the only cue for separating the sources is the filtering imposed by the HRTF. Nevertheless, separation was good as seen by comparing the left (Original) and right (Output) columns. From Asari et al. (2006), with permission.
Figure 2.4: **Performance of different separation approaches with three sources.** The separation performance (SNR across sources) is shown as a function of the sum of the “sparseness index” of the three sources ($\sum_{i=1}^{P} \|c_i\|_0 / \dim x_i$ for $P = 3$, averaged over 20,000 sample sets). Sparse prior (black) always outperforms dense prior (gray), and excellent separation was achieved especially when the sources are sparsely representable. As demonstrated by the good performance of the “combined” example in which a concatenated basis was taken from all the ensembles, the model does not depend strongly on choosing the basis carefully. Because $L_1$-norm minimization (Eq.(2.8)) gives at most $N$ non-zero coefficients, high separation performance can be achieved if $\sum_{i} \|c_i\|_0 / \dim x_i \leq 1$, i.e., each source can be represented by equal to or less than $N/P$ basis on average. From Asari et al. (2006), with permission.
Figure 2.5: Separation performance for different source locations. Using a typical example of three novel stimuli (trumpet and two same harp; Figure 2.3), separation performance (vertical axis) was examined with all the possible combinations of the three sources (from 0 to 120 degrees apart in steps of 5 degrees; horizontal axis). The average performance is shown here under either sparse (black) or dense (gray) prior. Separation was unsuccessful at angle zero since differential filtering cannot be exploited, whereas the performance gets better as the sources get further apart. From Asari et al. (2006), with permission.
Figure 2.6: **Neural representations underlying source separation.** Each panel shows the activity of a population of 225 neurons, corresponding to the 225 features: $\tilde{d}_{ij} = h_i \cdot d_j$. The intensity of each dot in the $15 \times 15$ grid is proportional to the log of the firing rate of each neuron. Since the sources occupy three positions $i$, there are three copies of the basis $d_j$ in each panel (corresponding to the three filters $h_i$). The copies are arranged from left to right for convenience ($\hat{c}_{1j}$, $\hat{c}_{2j}$, and $\hat{c}_{3j}$ for $j = 1, \ldots, 75$), and separated by vertical lines. However, the arrangement is for purposes of illustration only; we do not mean to imply any spatial organization of sources within the cortex. The sources are the same as in the previous figure. (A) Sparse representations of the three sources (corresponding to the filtered spectrograms in Figure 2.3) presented in isolation. Only a relatively small number of units are active in each panel. (B,C) Sparse representation of the mixed sources (input spectrogram in Figure 2.3). Activity is approximately the sum of the activities of the isolated sources in (A) in the presence of HRTF cues (B) but not in the absence of the separation cues (C). (D) Dense representation of the mixed sources. Note that most units are active. From Asari et al. (2006), with permission.
The failure of the dense representation to separate sources (Figure 2.4) results from a failure of the first step. Instead of decomposing the sources into a small number of features, the dense representation (Figure 2.6D) assumes that each instrument contributed about equally to the received signal, and so finds a representation in which a large fraction of neurons are active. That is, instead of “explaining” the sources in terms of two harps and a trumpet, the dense representations also find some clarinet, some cello, etc., at all positions. This is intrinsic to the dense solution, since it seeks the “minimum power” solution in which neural activity is spread among the population (Figure 2.1B).

The failure of even the sparse approach when the spectral cues induced by the HRTF are absent (Figure 2.5; leftmost point, showing 0-degree separation) results from a failure at the second step. That is, the sparse approach finds a useful decomposition at the first step even without the HRTF, but in the absence of HRTF cues the active features are not tagged, and so the features cannot be assigned appropriately to distinct sources (Figure 2.6C). Other psychophysical cues relevant for source separation—such as common onset time—might provide alternative or additional tags in this same framework. A more general formulation of source separation might allow tagging on longer time scales, so that a feature active at one moment might be more (or less) likely to be active the next, reflecting the fact that sources tend to persist. But this approach was not pursued any further here.

2.4 Predictions on Neural Behaviors

The sparse representation model makes experimentally testable predictions about the nature of the neural representation underlying source separation. Such predictions can be used to validate/falsify the model, which is critical to make a bridge between theory and experiments, and to explore what exactly nature chose among many possible mechanisms that could all work well for achieving organisms’ goals.
2.4.1 Optimal Feature Estimation

In this model, the firing rate of a given neuron \( c_{ij} \) is maximized when the stimulus perfectly matches with the neuron’s feature, i.e., when \( y = \tilde{d}_{ij} \). Since the feature \( \tilde{d}_{ij} \) is used in the linear reconstruction of the stimulus from the neural activities (Eq.(2.7) on page 25), one might imagine that the optimal stimulus (i.e., the stimulus that maximizes the firing rate) can be obtained by estimating the optimal linear decoder of the target neuron considered alone. Experiments based on this idea have shown that the optimal linear decoder can sometimes drive neurons in the auditory cortex to fire vigorously (deCharms et al., 1998; O’Connor et al., 2005).

Surprisingly, this model predicts that the linear estimate of the decoder obtained in this way is not the optimal stimulus, although the optimal decoder itself is linear (Eq.(2.7)). Instead, finding the optimal stimulus by the linear methods requires recording from all the neurons involved in the representation (but see Section 2.A). This follows from the assumption that the features are not orthogonal. Note that in this model, optimal decoding need not take neural correlations into account, even when they are present.

This first prediction is illustrated by a simulation in Figure 2.7 where a \( 1,168 \times 3,600 \) feature matrix \( \tilde{D} \) was used, each column of which held a neural feature spanning over 96 msec (16 bins with a bin size of 6 msec) and ranging between 55–3,520 Hz (73 frequency bands in steps of 1/12 octave). As the original feature of a target neuron, the one obtained from cello ensembles was chosen, and thus cello sounds were used as input stimuli in the simulation. The optimal linear decoder was then estimated by least squares (see Section A.2; for simplicity, here we did not use regularization methods) using the activities of a target and a variable number of other simulated neurons (over 200 random combinations). The vertical axis in Figure 2.7 then shows the firing rate of the target neuron (normalized to its maximum firing rate) in response to the stimulus that matches the estimated optimal linear decoder. When the optimal decoder is estimated from only the target neuron, the firing rate is sub-maximal. As the number of neurons used for the estimation is increased (horizontal axis), the response of the target neuron converges to unity on average, indicating that the optimal decoder has converged to the target
neuron’s feature. Stimulus optimization for the target neuron has thus improved by recording from other neurons involved in the representation.

Figure 2.7 represents a novel and testable prediction of the model: jointly estimating the optimal linear decoder from a population of neurons should yield a stimulus that is closer to optimal. Moreover, it also leads to a novel experimental approach for finding the optimal stimulus. Although in principle the activity of all neurons involved in the representation must be recorded, in practice the activity of even a few can be useful. With modern techniques (e.g., tetrodes) for isolating the activity of several nearby neurons, this approach might be practical.

2.4.2 Asymmetry of sparse representations

A second testable prediction is that there should be an asymmetry between encoding and decoding: the optimal encoding function is nonlinear but the optimal decoding function is linear. Here decoding refers to the process of “reading out” a neural representation (e.g., by forming an estimate or reconstruction of the stimulus), whereas encoding refers to the process by which the nervous system constructs a pattern of neural activities from a stimulus. Surprisingly, however, this asymmetry emerges only for populations of neurons; the optimal linear encoder and decoder of an isolated neuron perform about equally with a poor performance (Figure 2.8).

The fact that optimal decoding of a neuronal population is linear—i.e., that the optimal linear decoder of the neuronal population response provides perfect reconstruction of the stimulus under the model, so no nonlinear model can do better—is a direct consequence of our fundamental assumption that the neural representation is a linear combination of features (Eq.(2.7)). Note however that the linear reconstruction works only when we know the firing rates of all the active neurons given a stimulus, i.e., all the non-zero coefficients \( c_{ij} > 0 \) in Eq.(2.7). Otherwise, we cannot avoid the error in the linear decoding model due to the unknowns. Thus the reconstruction performance by the linear decoding should depend on the number of active neurons we could record from in multi-unit experiments.
Figure 2.7: **Prediction 1: Stimulus optimization requires multi-neuron recording.** The vertical axis shows the simulated firing rate of a target neuron (normalized to its maximum firing rate) in response to the presentation of a stimulus corresponding to the optimal linear decoder constructed by recording the activity of a target neuron and a variable number of other neurons. When the optimal linear decoder is estimated from only the target neuron, the firing rate is sub-maximal. As the number of neurons used in this simulation to estimate the optimal linear decoder is increased (horizontal axis), the response of the target neuron converges to unity, indicating that the optimal decoder has converged to the target neuron’s feature. From Asari et al. (2006), with permission.
The linearity of population decoding does not necessarily imply that the neural encoding function—the inverse transformation from the stimulus to the response—needs to be linear; and in general it is not. Sparseness induces a nonlinear (more precisely, piecewise linear) encoding (Figure 2.9). Formally, the encoding function of a neuron is obtained as the corresponding row of $\bar{D}^\dagger$, where $\bar{D}$ is a “packed matrix” whose columns are the subset of features involved in the sparse representation of a given stimulus $y \in \mathbb{R}^N$, i.e., only the columns corresponding to the nonzero elements of $c$. This matrix satisfies the decoding relation:

$$y = \bar{D}c,$$

where $\bar{c}$ consists of $c$ without zero elements. Because of the sparseness prior ($L_1$-norm minimization), however, the matrix $\bar{D}$ is at most full-rank and constructed from only at most $N$ features. Hence it is not overcomplete, and the encoding can be specified by the pseudoinverse:

$$\bar{c} = \bar{D}^\dagger y.$$

Note that constructing the matrix $\bar{D}$ requires knowledge of the solution $c$, so that Eq.(2.11) does not actually constitute an algorithm for finding $c$ under the sparseness prior (Eq.(2.8)). Piecewise linearity then arises because the encoding function $\bar{D}^\dagger$ is linear for all stimuli that activate the same subset of features $\bar{D}$, but changes for different subsets.

The point in the prediction here is not in the piecewise linearity itself, but in the asymmetry between the linearity of the decoding function and the nonlinearity of the encoding function. In fact, any saturating nonlinearities must be introduced as a preprocessing in the model to make it more plausible; unlike physiology, doubling the stimulus $y$ necessarily doubles the neural representation $c$ in the current model; i.e., $y = \tilde{D}c$ in Eq.(2.7) implies $ky = \tilde{D}(kc)$ for any scalar $k \in \mathbb{R}$ (see also Section 2.5.2).

The prediction that there is an asymmetry between the linearity of the decoding function and the nonlinearity of the encoding function can be tested experimentally (Figure 2.8). Given a set of stimulus-response pairs (i.e., the neural responses to an ensemble of sounds)
obtained from a population of neurons, the model predicts that a linear stimulus reconstruction approach (i.e., a decoding model) will outperform a linear “forward” (i.e., encoding) model, but only if the optimal linear reconstructors are estimated from a population of neurons. The idea that a linear approximation is better suited for the neural decoding than encoding function was first exploited to estimate the information rate of fly visual neurons (Bialek et al., 1991). By contrast, our model predicts that, if the neural representation is sparse and overcomplete, then the asymmetry should emerge only in multi-neuron recordings; i.e., linear decoding does not provide an advantage over linear encoding for single neuron experiments, whereas the former outperforms the latter for multi-neuron experiments.

**Simulation Procedures**

To illustrate the asymmetry of linear encoding and decoding in the framework of sparse overcomplete representations, simulations were conducted in 25 dimensions with 75 neurons. In the simulations, the three-fold overcomplete features (a $25 \times 75$ feature matrix $\tilde{D}$) were first generated randomly on the unit hypersphere. Neural activities $c$ for sample stimuli drawn from a Gaussian distribution were then determined by Eq.(2.8) without noise ($\beta = 0$). The optimal linear decoding and encoding filters were then estimated by least squares (see Section A.2), where a fraction of the elements in $c$ was used for the linear filter estimation. Figure 2.8 showed the average results over 200 random samplings at each level.

For simulated single unit data (Figure 2.8B), we computed the mutual information between the simulated neural responses $c$ and stimulus $y$ using the equation:

$$ I(c, y) = H(c) - H(c | y) = H(c), $$

where $H(c)$ is the response entropy, and the conditional of the response given the stimulus satisfies: $H(c | y) = 0$ because the relation between stimuli and responses was deterministic. Thus the mutual information between the single neuron and the stimulus was just equal
to the response entropy, which was estimated by direct binning from the histogram of neural responses (see also Section A.4.2). This total information was compared to either; the mutual information between the optimal linear estimate of the response $\hat{c}$ given the stimulus and the actual stimulus (encoding; $I(\hat{c}, y)$); or between the optimal linear estimate of the stimulus $\hat{y}$ given the response and the actual response (decoding; $I(c, \hat{y})$). For these information estimations, the Gaussian approximation was used to bound the entropy of the reconstruction error (Bialek et al., 1991; Cover and Thomas, 1991; Rieke et al., 1997).

For multi-unit data (Figure 2.8D), the computation of the full mutual information (rather than the linear approximation) was computationally intractable. Therefore, the following simpler measure of the reconstruction quality of the models was computed:

$$1 - \left< \frac{\|\text{reconstruction error}\|_2}{\|\text{response or signal}\|_2} \right> = 1 - \left< \frac{1}{\text{SNR}} \right>, \quad (2.13)$$

where $\|\cdot\|_2$ denotes the $L_2$-norm and $\langle \cdot \rangle$ the mean over data. This measure is based on the relative power (standard deviation) of the model errors, and it gives zero for pure noise and one for perfect reconstruction.

**Experimental Data Analysis**

To compare the results of the simulation to those of physiology, we analyzed whole-cell recording data for auditory cortical neurons in the anesthetized rats in response to natural sounds. For the single-unit analysis (Figure 2.8A), subsets of data in the previous work were used (Machens et al., 2004). In a first data set (7 cells), a fixed set of natural sounds were repeatedly presented up to 20 times, and these data were used to estimate total information using the direct method (Borst and Theunissen, 1999; see also Appendix Section A.4.2). In a second data set (8 cells), as many natural sounds as possible were presented, each once or twice. Both the first and the second sets of data were used to examine the performance of linear encoding and decoding models, in a similar manner as the analysis of simulated single-unit data.
Figure 2.8: **Prediction 2: Asymmetry between linear decoding and linear encoding.** (A, B) The mutual information between the stimulus and the responses of single neurons (Total; Eq.(2.12); mean ± standard deviation) was compared with the mutual information captured by linear encoding (Enc) or decoding (Dec) models. For details, see *Simulation Procedures and Experimental Data Analysis* in Section 2.4.2, as well as Appendix Section A.4.2. Both encoding and decoding models capture only a fraction of the total information at the single-cell level in both physiology (A) and simulations (B). (C, D) The reconstruction quality (Eq.(2.13); mean ± standard deviation) is plotted for the optimal linear decoder (black) and the optimal linear encoder (gray). In simulations (D), encoding and decoding perform comparably when only a few neurons are recorded, but the reconstruction quality of decoding grows faster as the number of recorded neurons increases. In physiology (C), the performance was equally low for both encoding and decoding up to 75 cells (for experimental details, see Chapter 3). Note however that the performance increases in a faster rate for the linear decoding. Panels (B) and (D) from Asari et al. (2006), with permission.
Figure 2.9: **Piecewise linear encoding.** (A) Three features (*blue, red, and black arrows*) in two dimensions constitute an overcomplete basis. A sample signal is indicated with a *gray arrow*. (B) Tuning curves for the three features are piecewise linear. The firing rate of each of the three units in (A) is given along the “path” of a signal moving between the features (*dashed gray arrows* in (A); from feature 1 to feature 3). Because the sample space is two-dimensional, any given point is represented by at most two active neurons. Decoding is linear; the data point (e.g., the one indicated at “∗” corresponding to the *data point* in (A)) is recovered by a weighted sum of the features, with the corresponding neural activities constituting the weights. Encoding, however, is nonlinear; the slope of neurons’ activation functions can change at the boundaries, whenever any neuron becomes active or inactive. The basic intuition shown here generalizes to the other examples, in which the dimensionality of the space (given by the number of elements in the spectrogram) is much higher. Adapted from Asari et al. (2006), with permission.
For the multi-unit analysis (Figure 2.8C), we performed whole-cell recordings where the same set of natural sound ensembles was presented (75 cells; see also Chapter 3), and used 20 random combinations of the cells to measure the reconstruction quality of the linear encoding and decoding at each level of multiplicities (Eq. (2.13)). Here we assume that multiple single-unit data could be considered as equivalent to multi-unit data.

Figure 2.8 shows that neither linear encoding nor linear decoding worked well at the single-cell level, consistent with the prediction of the model. At the population level, the performance of linear decoding increased in a faster rate as more neurons were used, but the linear decoding did not significantly outperform the linear encoding in contrast to the simulation. Considering the fact that the auditory cortex consists of by far more neurons than 75 cells, however, the discrepancy between the simulation and physiology would be most likely because we do not have enough number of neurons. This is also supported by the result that the reconstruction qualities were far smaller than one, i.e., the linear models did not work well for either direction with up to the 75 cells. By extrapolation, the linear decoding would then be expected to substantially outperform the linear encoding at some point if we recorded from more neurons—probably around several hundred cells. It was in fact reported that natural scenes could be reconstructed fairly well from ensemble responses of 177 cells in the lateral geniculate nucleus (Stanley et al., 1999).

2.4.3 Context-Dependence of Receptive Field

A third prediction follows from the piecewise linearity of the encoding (Figure 2.9): the linear component of receptive fields should depend on the acoustic context. Following conventional usage in auditory physiology, we use the term spectro-temporal receptive field (STRF$^5$) to refer only to the linear component of the encoding function, even though the function itself may be highly nonlinear (Kowalski et al., 1996; Theunissen et al., 2000, 2001). Here we define the

\footnote{In visual physiology, “STRF” is used to refer to the “spatial temporal receptive field,” but the quantities are analogous.}
acoustic context of a neural feature $\tilde{d}_{ij}$ with respect to a stimulus $y$ as the collection of other features activated simultaneously by that stimulus; in music for example, the features tend to resemble musical notes, and the acoustic context can be thought of as the set of notes (e.g., in a chord) that accompany a given note.

The STRF can be viewed as the “slope” of a neuron’s tuning curve in a high-dimensional input space (Figure 2.9; see also Eqs.(2.10) and (2.11)). In simulations, we used a $1,168 \times 3,600$ feature matrix $D$ (the same one as in Figure 2.7), and used two different sets of 300 active features (i.e., $1,168 \times 300$ packed matrices $\tilde{D}$) to estimate the STRFs for the two different contexts (Figure 2.10). Some neural features were then active only in either contexts, whereas others in both contexts, including the one shown in Figure 2.10A; the gross structures of the STRF (e.g., the excitatory band around 880 Hz) are preserved in both contexts, but the secondary character (e.g., relative strength of sidebands) is context-sensitive. Changes in the STRF for different stimuli can be larger or smaller than in this example, but this stimulus-dependent “bottom-up” modulation on the neural encoding is suggestive of the non-classical receptive field modulation observed in visual and auditory cortices (Bar-Yosef et al., 2002; David et al., 2004; Valentine and Eggermont, 2004).

Context-dependence as defined here is stronger than simple nonlinearity. Specifically, the prediction is that there should exist extended subregions of stimulus space where the encoding function of a given target neuron is one linear function, and across some boundary in stimulus space switch to a second linear function. These boundaries are demarcated by the activation of another (non-target) neuron in the population and the de-activation of a second (non-target) neuron (Figure 2.9). This prediction could be tested using a multi-neuron recording technique such as two-photon calcium imaging in vivo (Svoboda et al., 1997; Stosiek et al., 2003; Ohki et al., 2005, 2006). For example, we could vary stimulus properties smoothly enough between distinct types of stimuli (e.g., from male to female voices), and ask if the ensembles of evoked neurons change on a one-neuron-by-one-neuron basis as is predicted by the model, or a totally new pattern of active neurons suddenly emerges at some point.
Figure 2.10: **Prediction 3: Dependence of STRF on stimulus context.** (A) Spectrogram of a trumpet feature, showing a strong fundamental around 880 Hz and some higher harmonics. (B, C) The STRFs corresponding to the feature in (A) when that feature is activated in two different contexts (clarinet or flute played simultaneously), derived under the assumption of a sparse neural representation. The STRF provides the *encoding* from the stimulus to neural activity. The color at any point of the STRF indicates the value (in spikes/second) of the kernel which is convolved with the spectrogram of the stimulus to generate a neural response. Under the sparse assumption, the encoding is piecewise linear (Figure 2.9), and the STRFs shown are two out of the many possible pieces. The STRF is obtained from the appropriate row of the matrix $\overline{D}^\dagger$ (see Eq.(2.11)). (D) The difference between the two STRFs. The STRFs show the same basic harmonic structure, but differ in details such as the relative contributions of the excitatory and inhibitory sidebands. The differences can be as large as the STRFs themselves. From Asari et al. (2006), with permission.
The locally linear encoding induced by sparseness may help reconcile some of the apparent contradictions in the auditory literature. STRFs obtained using a “moving ripple” basis can predict responses to linear combinations of basis elements (Kowalski et al., 1996) fairly well. However, linear encoding (STRF) models generally fail to predict neural responses when the stimulus domain is extended to include a wide selection of complex sounds (Linden et al., 2003; Machens et al., 2004; see also Chapter 3), consistent with the idea that ripples represent a subspace within which encoding is linear. Context sensitivity may also provide an explanation for a proposed neural correlate of comodulation masking release in which the addition of a pure tone can suppress the response to temporally-modulated noise (Nelken et al., 1999); this form of contextual modulation cannot be explained by any purely linear encoding model.

2.4.4 Top-Down Receptive Field Modulation

The model also predicts a form of top-down modulation of receptive fields and neuronal tuning by spatial expectation. The modulation arises from Eq.(2.6), where the dictionaries $\tilde{d}_{ij}$ can be viewed as hypotheses about what a basis element $d_j$ would sound like if it arose from a source at a particular position $i$ in space (see also Section 2.2.2).

Even in an overcomplete representation, however, it would be hard to imagine that every possible position is represented simultaneously. Rather, we have assumed the instantiation of only those features corresponding to those positions at which a sound source is present (Eqs.(2.6) and (2.7)). In the simulations in Section 2.3, for example, sources at three positions lead to a three-fold overcomplete representation. If instead we had assumed features for every possible position in space up to a spatial precision of, e.g., 5° of azimuth, then the representation would be $360°/5° = 72$-fold overcomplete—or higher, if sources at different elevations are also considered. Such a high degree of overcomplete representation would be unwieldy and computationally intractable for both artificial and biological systems.
Figure 2.11: **Prediction 4: Top-down spatial expectation modulates neuronal tuning.** The *blue curve* shows the normalized activity of a model neuron tuned to a cello basis element $d$ in response to the cello sound coming from $205^\circ$ azimuth (“*”). When the expectation ($h_{205^\circ} \cdot d$; spectrogram on the right) matches the actual sound, the response is maximal. When there is a mismatch between the actual and the expected sound ($h_{180^\circ} \cdot d$; spectrogram on the left), however, the response is diminished (*arrowhead* at $180^\circ$) as much as two-fold.
Limiting the features to only those positions at which sources are present requires some form of “top-down” knowledge about source position. The current model describes source separation only on short time scales (e.g., 5 msec in Section 2.3), thus any computation that integrates information over longer time scales might provide the necessary top-down information. Likely candidates for such information include binaural or monaural spatial cues, or visual cues (e.g., the ventriloquist effect).

Because the features available in a representation are established dynamically in response to spatial knowledge—e.g., possibly as in “shifer circuits” (Anderson and Van Essen, 1987) or “dynamic routing circuits” (Olshausen et al., 1993, 1995)—there may be a transient mismatch between the actual and the expected position. Such a mismatch will reduce the neuronal response (Figure 2.11). Thus our model predicts that if a listener can be “misled” into expecting a source to arise from a position different from its actual position, the reduced activity should be detected experimentally.

The model thus predicts two ways in which receptive fields should be dynamic; i.e., receptive fields should depend on (1) stimulus context as “bottom-up” modulation, and (2) spatial expectation as a form of “top-down” modulation that could be extended to include modulation by attention, reward, or other high-level task constraints, e.g., in the form of explicit Bayesian priors on the stimuli. Note that the focus here is not on the receptive field properties themselves, but rather on how the resulting sparse representation can subserve a computation. Thus the predictions are not about the detailed structure of receptive fields, but rather about how they interact.

2.4.5 Sparse Activities

The last prediction—or, rather the premise—of the model is the sparse encoding, which implies that most stimuli should elicit only modest firing in most neurons, as has been observed experimentally for both simple and complex visual and auditory stimuli (Figure 2.12; Vinje and Gallant, 2000; DeWeese et al., 2003; Machens et al., 2004; Hromádka, 2007). Note that
sparseness implies not that responses must be weak for all stimuli, but merely that stimuli elicit only a small number of spikes across the neuronal population. Also note that sparseness in this model is a constraint on the activity of the population of neurons involved in a representation, rather than on the activity of any single neuron. The model is thus fully consistent with experiments indicating that it is sometimes possible to optimize stimuli online to obtain high firing rates (deCharms et al., 1998; Barbour and Wang, 2003; O’Connor et al., 2005); such a stimulus can be considered as the “feature” associated with the neuron in the model framework (see also Section 2.A).

The particular interpretation of the sparseness here (Eq.(2.8)) suggests that the neural representation $c$ for a given stimulus $y$ is optimized to minimize its $L_1$-norm: $\|c(y)\|_1$, or the total spike count. The triangle inequality gives: $\|c(\sum_i y_i)\|_1 \leq \sum_i \|c(y_i)\|_1$, i.e., the total spike counts to represent a mixture of stimuli should be equal to or less than the sum of the total spike counts to represent each stimulus in isolation. The linear generative model (Eq.(2.7)) also implies: $c(ky) = kc(y)$ for all $k \in \mathbb{R}$, i.e., the neural responses should be in proportion to the stimulus intensity. But this is often not the case in physiology, and thus a more plausible model should incorporate some appropriate saturating processes (see also Section 2.5.2).

Directly assessing the sparseness of a neuronal representation experimentally is difficult. The key issue is how many neurons (or spikes) participate in the representation of a typical stimulus. Ideally this would be measured by recording all spikes from all neurons simultaneously, but this is not possible using the experimental techniques currently available. There is nonetheless growing evidence that natural stimuli activate only a relatively small population of neurons in the cortex (Figure 2.12; Baddeley et al., 1997; Vinje and Gallant, 2000; DeWeese et al., 2003; Machens et al., 2004; Olshausen and Field, 2004; Hromádka, 2007).
Figure 2.12: **Sparseness of responses in the primary auditory cortex.** The spontaneous firing rate was $0.25 \pm 0.65$ Hz (median ± interquartile range; 194 cells recorded by *in vivo* whole-cell patch-clamp techniques; see Section 3.2 in Chapter 3 for experimental details), whereas the firing rate evoked by natural sounds was $0.33 \pm 0.69$ Hz, and they are not significantly different (Wilcoxon matched-pairs signed-rank test). This suggests that most stimuli elicited only modest firing in most neurons in the primary auditory cortex, supporting the idea of sparse representations (see also Hromádka, 2007).
2.5 Discussion

Sparse overcomplete representations can separate a monaural mixture of sound sources into constituent auditory streams (Section 2.3). Source separation is a complex computation, and we could no more expect to solve the whole problem in its entirety here than one could expect to solve completely its visual analog—scene segmentation—or any of the many other challenging problems in computational vision. Thus we have instead concentrated on a restricted form of the problem involving only the spatial cues introduced by the HRTF (Section 2.2.2), with the expectation that the framework can be generalized to understand how some other cues might be used in a similar manner (see below).

In the framework of the model, it is critical to use an appropriate overcomplete basis in order to achieve acceptable separation performance (Figure 2.4). NMF was used here for finding a set of basis elements with which auditory stimuli could be represented sparsely (Section 2.2.4). Although we did not test basis sets obtained from other approaches, we do not expect that the results would be sensitive to the particular method used to find the basis; any sparse basis would likely have worked.

Recent advances in ICA have emphasized the utility of sparse overcomplete representations for source separation problems in acoustic, visual and other domains (Farid and Adelson, 1999; Lee et al., 1999; Lewicki and Sejnowski, 2000; Rickard and Dietrich, 2000; Bofill and Zibulevsky, 2001; Zibulevsky and Pearlmutter, 2001; Levin and Weiss, 2004; Li et al., 2004). The formulation here for source separation has been built on these ideas, extending the framework to exploit (monaural) separation cues that animals use: specifically, the HRTF to “tag” dictionary elements so they can be assigned to the appropriate sources (Eqs.(2.5)–(2.7) in Section 2.2.2). Other psychophysical cues could be used in a similar manner; information from two (or more) sensors, such as interaural time and level differences, can be readily incorporated by simply replacing the single-input-single-output HRTF filters by single-input-two-output fil-
ters, doubling the size (column length) of the dictionary elements and leaving the algorithm otherwise unchanged.

A good model or theory in biological sciences should not only explain well a phenomenon of interest, but also give some insights on the underlying mechanisms and make experimentally testable predictions (“falsifiability;” Popper, 1934). We have thus identified several clear experimental predictions on the characteristics of neural representations in the auditory cortex (Section 2.4), summarized as follows.

- The optimal linear decoder estimated from an experiment in which the activity of multiple neurons are recorded should maximize a target neuron’s firing rate if we present a stimulus corresponding to the decoder.

- There should be an asymmetry between the performance of the optimal linear encoder and decoder, but this asymmetry should become evident only in the context of multi-neuron recording experiments; the model predicts that the optimal linear encoder and decoder in a single neuron experiment both underperform the “true” optimal (i.e., non-linear) decoder.

- The STRF should be dynamically influenced by acoustic context as well as by spatial expectation on the source locations.

- Neurons should show sparse activities, and subadditivity of spike counts should be observed in response to a mixture of sounds.

These predictions can be used to validate—or falsify—the model of sparse overcomplete linear representations.

In the following Section 2.5.1, I will discuss more implications of employing sparse overcomplete linear representations as a model of sensory processing in the brain. Section 2.5.2 will then close this chapter with possible extensions and perspectives of the model.
2.5.1 Model Implications

Although there is nothing in the model that explicitly ties it to one or another brain area, we would imagine that, at least in mammals, the operations described above most likely occur in the cortex, rather than at subcortical stations. First, receptive fields in auditory cortex are heterogeneous, and often have broad and complex spectro-temporal structures (Sutter, 2000; see also Section 1.1.6) required to exploit the HRTF. Second, and more significantly, auditory cortex has the characteristics expected to form an overcomplete representation. This model then provides a possible explanation for an important question about cortical organization, “Why are there so many more neurons in the auditory (or visual) cortex than in the cochlea (or retina)?” The answer provided here, motivated by the ability of sparse overcomplete representations to separate sources, is potentially quite general, and may be applicable to other brain regions and/or computations as well.

The anatomical organizations of the sensory systems suggest that the representation in the cortex would be highly overcomplete, which has pros and cons in the framework of sparse representations. One advantage is that a bigger dictionary generally leads to a sparser representation, which in turn gives less “coding cost” on average, i.e., smaller $L_1$-norm values: $|c|_1$. For illustrative purposes, let us consider a situation in $\mathbb{R}^2$ where basis functions $\tilde{d}_m$ for $m = 1, \ldots, M$ are on the unit circle at angles $\alpha_m$, and a data point $y$ is located at angle $\theta \in [\alpha_m, \alpha_{m+1}]$, i.e.,

$$\alpha_1 < \cdots < \alpha_m \leq \theta \leq \alpha_{m+1} < \cdots < \alpha_M \quad (\alpha_1 + 2\pi),$$

(2.14)

$$\|y\|_2 = \|\tilde{d}_m\|_2 = 1, \quad (\text{for all } m)$$

(2.15)
and we define: $\alpha_{M+1} \overset{\text{def}}{=} \alpha_1 + 2\pi$ for convenience. The $L_1$-norm minimization (Eq.(2.8) with no noise; $\beta = 0$) then yields: $y = c_m \tilde{d}_m + c_{m+1} \tilde{d}_{m+1}$, with the coding cost of (Figure 2.13):

$$C_m(\theta) = \sum_{m=1}^{M} |c_m| = c_m + c_{m+1}$$

$$= \frac{\sin(\alpha_{m+1} - \theta)}{\sin(\alpha_{m+1} - \alpha_m)} + \frac{\sin(\theta - \alpha_m)}{\sin(\alpha_{m+1} - \alpha_m)}$$

$$= \cos \left( \theta - \frac{\alpha_m + \alpha_{m+1}}{2} \right) \sec \left( \frac{\alpha_{m+1} - \alpha_m}{2} \right).$$

(2.16)

Let $P_m(\theta)$ be the probability density of $y$ for $\theta \in [\alpha_m, \alpha_{m+1}]$. In the case of the uniform distribution: $P_m(\theta) = 1/2\pi$ for all $m$, the coding cost can then be given on average as:

$$\left\langle \sum_{m=1}^{M} \int_{\alpha_m}^{\alpha_{m+1}} P_m(\theta) C_m(\theta) d\theta \right\rangle = \left\langle \sum_{m=1}^{M} \frac{1}{\pi} \tan \left( \frac{\alpha_{m+1} - \alpha_m}{2} \right) \right\rangle$$

$$\approx \frac{M}{\pi} \tan \left( \frac{\pi}{M} \right)$$

$$\rightarrow 1 \quad \text{as } M \rightarrow \infty$$

(2.17)

where $\langle \cdot \rangle$ denotes the mean, and the approximation holds because $\langle \alpha_{m+1} - \alpha_m \rangle \approx 2\pi/M$ for large $M$. This suggests that the coding cost becomes smaller on average, approaching towards the sparsest case: $\|c\|_1 = \|y\|_2 = 1$, as the dictionary size $M$ increases.

From a practical viewpoint, it would thus be preferable to have more basis functions for achieving better “shrinkage” in the sense of coding cost or data compression. In addition, it would lead to a higher computational power in the sparse method framework, since it would potentially allow to sparsely represent a broader class of source distributions (Olshausen and Field, 2004). Hence we think that sparse representations can be a generic model for signal processing even in control theory or statistics as well as in neuroscience, and further advances
in optimization and learning algorithms will find out its practical usages in many aspects, including the cocktail party problem in more general settings.

Several disadvantages also follow from having a larger number of neurons. First, it requires more maintenance costs, and second, it leads to less robust representations to a noisy input; i.e., the neural response becomes more susceptible to input noise. For illustration, let us consider the same situation in $\mathbb{R}^2$ as before (Eqs. (2.14) and (2.15)). Then the susceptibility of an active neuron $|dc_m|$ to the input noise level $\|dy\|$ is given on average as (Figure 2.13A):

$$\langle \frac{|dc_m|}{\|dy\|} \rangle = \langle \csc (\alpha_{m+1} - \alpha_m) \rangle \approx \frac{M}{2\pi},$$

(2.18)

where the approximation holds for large $M$. Eq. (2.18) then suggests that the response variability at the single-cell level ($|dc_m|$) can be high in an overcomplete representation, even though the population noise (or the reconstruction error $\|dy\|$) is small. It should also be mentioned that a given neuron can be coactive only with a limited set of neurons in this scenario of sparse representations (Figure 2.14A), but the number of such potentially coactive neurons and that of all possible combinations of the coactive neurons with the neuron of interest grow exponentially faster than the dimensionality of stimulus/feature space (blue and red line in Figure 2.14A, respectively). In a high-dimensional highly-overcomplete system, a noisy input could then elicit so many different patterns of neural population activity (Figure 2.14B) if the noise level is large enough to go across the “boundaries” of the piecewise linear encoding functions (Figure 2.9). Nevertheless, we could say in turn that such a system can represent an input signal faithfully enough as a population even if the individual neural noise—and the variability of neural ensembles that encode a given signal—would be high, suggesting that the fact that neural activities are often notoriously noisy in experiments could simply be a natural outcome of the representation strategy—instead of the “nuisances”—in the brain. From an evolitional viewpoint, the anatomical organization of the brain also implies that the computational power

---

6From a practical viewpoint, this noise susceptibility problem can be alleviated to some extent by choosing an appropriate noise level ($\beta$) in the noise model (Eq. (2.8) on page 27).
and other advantages associated with employing an overcomplete system have outweighed the disadvantages in natural selection, especially for humans.

2.5.2 Model Extensions

Although the HRTF-based sparse separation model was inspired by the salient cortical organization and worked well under appropriate conditions, there are some discrepancies between the current model and physiological data. One potential problem is the particular algorithm for achieving sparse representations in the model. Linear programming was used here to solve the $L_1$-norm minimization problem (Eq. (2.8) on page 27), but this strategy is unlikely to be employed in the brain because such an iterative algorithm takes too much time to reach an optimal solution. Although it is not yet known how sparse cortical representations are achieved, it seems likely that the underlying circuitry involves lateral interactions, including some parallel algorithms instead of serial processing alone. In fact, “sparsification” is similar to divisive normalization approaches that are motivated by both computational and circuit considerations (Schwartz and Simoncelli, 2001), and explicit “biologically plausible” circuit dynamics and connection weights can be obtained using gradient descent to minimize the total neural activity (Olshausen, 2002; Perrinet, 2004; Fischer et al., 2007).

It should also be mentioned that the model described in Section 2.2 works frame by frame and thus the temporal information in longer time-scales is not incorporated in the model framework, although it implicitly exploits such integrated information as a top-down modulation of neural features (see Section 2.4.4). Of course the model can be reformulated into a convolutive form in the time domain (or in the time-frequency domain): $y(t) = \sum_{ij} c_{ij}(t) \ast \tilde{d}_{ij}(t)$, or even into a continuous form: $y(t) = \int c(u, t) \circ \tilde{d}(u, t) \, du$, where “$\circ$” indicates some linear operator. But the critical question from a biological viewpoint is, again, how the cortex achieves sparse representations if indeed sparseness underlies source separation (and other computations) in the auditory cortex. Explicit circuit dynamics should then be considered as well to build a more plausible model (see also Chapter 4).
Figure 2.13: **Geometry of overcomplete representations in $\mathbb{R}^2$.** (A) Three features ($\tilde{d}_i$ for $i = m$, $m + 1$, and $m + 2$) in $\mathbb{R}^2$ constitute an overcomplete basis. A sample signal $y$ is located between the $m$-th and $(m + 1)$-th features; i.e., at angle $\theta \in [\alpha_m, \alpha_{m+1}]$ in polar coordinate. The sensitivity of coefficients (double-arrows; $|dc_m|$) to perturbations on the signal (black circle; $\|dy\|$) depends on the angle between the active features: $\alpha_{m+1} - \alpha_m$. (B) Coding cost: $C_m(\theta) = \|c\|_1$ for data point $y \in \mathbb{R}^2$ as a function of the angle $\theta$; Eq.(2.16). Note the normalization: $\|y\|_2 = 1$, in contrast to Figure 2.9.
Figure 2.14: Population variability in sparse overcomplete representations. (A) The number of potentially coactive features and that of facets around a feature. Left: In $\mathbb{R}^2$ for example, the number of potentially coactive features (or “neurons;” blue arrows) with a target feature (black arrow) is two, and the number of all possible combinations of coactive features with the target (i.e., “facets” around the target; red dotted lines) is also two. Note that some features (gray arrows) are never coactive with the target. Right: For the computation in higher dimensions, we randomly generated (20- to 40-fold) overcomplete basis sets and data (of length $1\times10^3$ to $8\times10^5$) on the unit hypersphere, and used $L_1$-norm minimization to determine the coefficients, from which the number of potentially coactive features and that of facets around each feature were computed. The blue and red lines show the mean values, respectively, and the error bars show the standard deviation. Note that, because of the property of $L_1$-norm minimization, the number of potentially coactive features in $\mathbb{R}^N$ would be closely related to the “kissing number” in $\mathbb{R}^{N-1}$, i.e., the maximum number of unit hyperspheres that can simultaneously contact one unit hypersphere (Zong, 1999). (B) Population variability in sparse representations. By simulations, we computed the probability that a given set of neurons is activated by a noisy input in $\mathbb{R}^4$ and $\mathbb{R}^8$ (left and right, respectively) with 20-, 50-, and 100-fold overcomplete features (randomly distributed on the unit hypersphere; cyan, magenta, and green, respectively). For each case, we tested 100 random stimuli $y$, each examined over 10,000 times with additive Gaussian noise: $\mathcal{N}[0, \|y\|/100]$, and counted how many times a given ensemble of neurons is activated—i.e., the pattern of non-zero coefficients in $L_1$ solutions. The “population variability” gets larger in higher dimensional space and with the larger number of features—e.g., in the case of $\mathbb{R}^8$ with 100-fold overcomplete features, the primary pattern (population activity pattern #1) is active only for $\sim60\%$ of the trials.
Another problem of our formulation results from the fundamental assumption of using the linear generative model (Eq. (2.7)): \( ky = \tilde{D}(kc) \) for any scalar \( k \in \mathbb{R} \). The objective in Eq. (2.8) then satisfies: \( \|kc\|_1 = |k|\|c\|_1 \), i.e., the neural activity can be arbitrarily large, in proportion to the intensity of input stimuli. But there is a physical limit on the firing rate (up to \( \sim 10^3 \) Hz; Kandel et al., 2000) and thus this is not the case in physiology. One way to address this problem is to preprocess (or scale) input signals \( y \) to limit its length—e.g., by sigmoidal functions—and then solve the \( L_1 \)-norm minimization (Eq. (2.8)). Such saturation nonlinearities could happen at subcortical stations before the signal reaches the cortex, because of a limited maximum firing rate of neurons.

An alternative solution is to transform coefficients \( c \) with some nonlinear functions after solving Eq. (2.8). In this case, Eq. (2.8) can be considered as an assignment step for finding active features under the sparseness prior, i.e., just a handy alternative for finding the “\( L_0 \) solutions” (Donoho and Elad, 2003; see also Section 2.2.3). The interpretation of the sparseness prior then becomes slightly different, and the transformed coefficients may not necessarily result in the representation with the minimum total spike counts. Here we could even apply different saturation functions to each neuron—e.g., depending on cell types—but such procedures would not affect the overall computational framework proposed here as long as the “inverse” transform exists for faithfully reconstructing the input signals.

The problem could also be addressed to some extent when multiple cells share a single dictionary (or basis) element. The \( L_1 \) solutions in Eq. (2.8) can then be considered as representing the group activities, each distributed over a population of similar cells. Additional constraints are required in this scenario to choose a unique representation within the group of cells, but this idea of feature sharing would be interesting in many senses, because such constraints could contribute to further computations in the brain. In fact, even though the auditory cortex has by far more neurons than the periphery does, neural features \( \tilde{d}_{ij} \) for all possible source locations in space would be unlikely to exist in the cortex, forming a highly-overcomplete dictionary for sound representations. We would instead imagine that ensembles
of cells share common basis elements \( (d_j) \), and they are differentiated by top-down modula-
tions \( (h_i) \) imposed by attention and/or the cortical state (Eq.(2.6); see also Section 2.4.4). Note
that the susceptibility to input noise (Section 2.5.1) could also be somewhat alleviated because
the effective number of features in Eq.(2.8) becomes smaller than the actual number of neurons.

2.A Appendix: Feature Estimation

Section 2.4.1 described a prediction that the optimal linear feature estimation in an overcom-
plete basis requires the recordings from all active neurons. It is however technically impossible
to perform such experiments at this moment, and from practical viewpoints, it would be more
useful to obtain better feature estimates even from single-unit recordings.

Here I thus explore a learning algorithm for the feature estimation (Section 2.A.1),
simply by exploiting gradient descent to minimize the disparity between the actual and esti-
mated activity of a target neuron. Note that neural features are the subspace in stimulus space
that characterizes the response properties of neurons, and thus identifying the optimal features
would help address the neural coding problem (deCharms et al., 1998; Barbour and Wang,
2003; Machens et al., 2005; O’Connor et al., 2005). Section 2.A.2 then describes simulation
methods, and the application examples (in relatively lower dimensional cases) are shown Sec-
tion 2.A.3. Finally, pros and cons of the algorithm will be discussed in Section 2.A.4.

2.A.1 Learning Algorithm

Suppose we have stimulus-response pairs for a single neuron \( (y^{(t)}, c_1^{(t)}) \), where superscripts
\( t = 1, \ldots, T \) indicate stimulus indices, and here we assign the subscript \( m = 1 \) to the target
neuron without loss of generality. An estimated target feature \( \hat{d}_{m=1} \) can then be given by:

\[
\text{minimize } E_{m=1} = \sum_t \mathcal{D} \left( \hat{c}_{m=1}^{(t)}, c_1^{(t)} \right),
\]  

(2.19)
where

\[ \hat{c}(t) = \arg \min_c \|c\|_1 \text{ subject to } y(t) = \hat{D}c. \]  

(2.20)

Note that \( \hat{D}(\cdot, \cdot) \) is a distance measure, and \( \hat{c}_1(t) \) is the estimated activity of the target neuron in response to \( y(t) \in \mathbb{R}^N \), given by the \( L_1 \)-norm minimization (Eq.(2.20); for a noise model, see Eq.(2.8)) using an estimated dictionary matrix: \( \hat{D} = \left( \hat{d}_1 \cdots \hat{d}_{\hat{M}} \right) \) with the estimated dictionary size of \( \hat{M} (> N) \). Also note the normalization: \( \| \hat{d}_m \|_2 = 1 \) for all \( m \), and \( \| y(t) \|_2 = 1 \) for all \( t \).

**Least Squares**

One natural measure of a distance \( D \) is the Euclidean distance (\( L_2 \)-norm), in which case the objective in Eq.(2.19) becomes:

\[ E_{ls} = \frac{1}{2} \sum_t \left( \hat{c}_1(t) - c_1(t) \right)^2, \quad \text{and} \quad \nabla_{d_m} E_{ls} = \sum_t \left( \hat{c}_1(t) - c_1(t) \right) \nabla_{d_m} \hat{c}_1(t). \]  

(2.21)

Using the “packed matrix” \( \bar{D}(t) \) for the \( t \)-th input, we have: \( y(t) = \bar{D}(t)c \) as in Eqs.(2.10) and (2.11) and thus the derivative \( \nabla_{d_m} \hat{c}_1(t) \) in Eq.(2.21) can be approximated as:

\[ \Delta y(t) = \bar{D}(t) \Delta c(t) + (\Delta \bar{D}(t)) \hat{c}(t), \quad \text{and} \quad \nabla_{d_m} \hat{c}_1(t) = \frac{\partial \hat{c}_1(t)}{\partial d_m} \approx -\hat{c}_m(t) d_1(t), \]  

(2.22)

where we assume: \( \Delta y(t) = 0 \), and \( (d_m(t))^\top \) is the \( m \)-th row of the pseudoinverse: \( (\bar{D}(t))^\dagger \).

**Symmetrized Kullback-Leibler divergence**

In least squares (Eqs.(2.21) and (2.22)), such signals have little effect on the gradient \( \nabla_{d_m} E_{ls} \) when \( \hat{c}_m = 0 \), but some of them could in fact be \( c_1(t) > 0 \) and thus supposedly informative. To compensate this “inefficiency,” additional objectives should be introduced, e.g., the one
based on the symmetrized Kullback-Leibler (KL) divergence:

$$E_{KL} = \left( \sum_t c_1^{(t)} \log \frac{c_1^{(t)}}{\hat{c}_1^{(t)}} \right) + \left( \sum_t \hat{c}_1^{(t)} \log \frac{\hat{c}_1^{(t)}}{c_1^{(t)}} \right), \quad \text{and} \quad (2.23)$$

$$\nabla_{\hat{d}_m} E_{KL} = \sum_t \left( 1 - \frac{c_1^{(t)}}{\hat{c}_1^{(t)}} + \log \frac{\hat{c}_1^{(t)}}{c_1^{(t)}} \right) \nabla_{\hat{d}_m} \hat{c}_1^{(t)}, \quad (2.24)$$

where the gradient $\nabla_{\hat{d}_m} \hat{c}_1^{(t)}$ is given by Eq.(2.22). Note that the $L_1$ solutions generally satisfy $\hat{c}_m \in [0, 1]$ because of the normalization on $d_m$ and $y$, and thus they can be considered as the probability of neural responses: $p_{\text{target}}(\text{spike} \mid y, D) = c_1$ and $\hat{p}_1(\text{spike} \mid y, \hat{D}) = \hat{c}_1$.

**Optimization Algorithm**

The objective in Eq.(2.19) can then be written as the weighted sum of Eqs.(2.21) and (2.23):

$$E_1 = E_{ls} + \gamma E_{KL}, \quad (2.25)$$

where $\gamma (\geq 0)$ is a parameter for the ratio of the effects between the square error ($E_{ls}$ in Eq.(2.21)) and the symmetrized KL divergence ($E_{KL}$ in Eq.(2.23)), and the derivative of $E_1$ can be computed from Eqs.(2.21), (2.22) and (2.24). Then, starting from (random) initial estimates $\hat{d}_m$ for $m = 1, \ldots, \hat{M}$, we can write the iterative learning algorithm based on the gradient descent method as follows.

1. **step.1** Compute the $L_1$ solution $\hat{c}_1^{(t)}$ as in Eq.(2.20) for each input $y^{(t)}$ using (overcomplete) dictionaries $\hat{d}_m$.

2. **step.2** Compute the objective $E_m$ for all $m$ as in Eq.(2.25), and find and select such $m$ as the target that has the minimum $E_m$. (Without loss of generality, assign $m = 1$ for the estimated target feature.)
**Step 3** Compute the gradient $\nabla_{d_m} E_1$ by using Eqs.(2.21), (2.22) and (2.24), and update the estimates: $\hat{d}_m \leftarrow \text{normal} \left( d_m - \lambda \nabla_{d_m} E_1 \right)$.

Note that $\text{normal}(\cdot)$ is the operator for column-wise normalization to have a unit length, and the step-size $\lambda \in \mathbb{R}$ should be positive and small.

**Estimate Evaluation**

The performance of the algorithm can be evaluated by Fisher information matrix for each estimated feature, which can be approximated as the Hessian matrix of the objective:

$$H_m = \nabla_{d_m}^2 \left( E_{ls} + \gamma E_{KL} \right),$$

(2.26)

where we have, from Eqs.(2.21) and (2.23),

$$\nabla_{d_m}^2 E_{ls} = \sum_t \begin{bmatrix} \left( \nabla_{d_m} \hat{c}_1(t) \right) & \left( \nabla_{d_m} \hat{c}_1(t) \right)^\top & \left( \hat{c}_1(t) - c_1(t) \right) & \nabla_{d_m}^2 \hat{c}_1(t) \end{bmatrix},$$

(2.27)

$$\nabla_{d_m}^2 E_{KL} = \sum_t \begin{bmatrix} \frac{c_1(t) + \hat{c}_1(t)}{(\hat{c}_1(t))^2} \left( \nabla_{d_m} \hat{c}_1(t) \right) & \left( \nabla_{d_m} \hat{c}_1(t) \right)^\top & \left( 1 - \frac{c_1(t)}{\hat{c}_1(t)} + \log \frac{\hat{c}_1(t)}{c_1(t)} \right) \nabla_{d_m}^2 \hat{c}_1(t) \end{bmatrix}. \quad (2.28)$$

From Eq.(2.22), we also have:

$$\nabla_{d_m}^2 \hat{c}_1 \approx - \left( \nabla_{d_m} \hat{c}_1 \right) d_1^\top - \hat{c}_m \nabla_{d_m} d_1^\top \approx \hat{c}_m \left( d_m d_1^\top + d_1 d_m^\top \right),$$

(2.29)

where the superscript $(t)$ is omitted for brevity. Note that the variance of the estimates is given as the inverse of the Hessian matrix $H_m$ (Eq.(2.26)).

The estimation performance was also evaluated by the Pearson product-moment correlation coefficients between the original and estimated target features: $d_1^\top \hat{d}_1$. Note the normalization: $\|d_1\|_2 = \|\hat{d}_1\|_2 = 1$. 69
2.A.2 Simulation Procedures

Original Feature Set

Here we tested the algorithm only in rather low dimensional examples ($N \leq 20$). Under the assumption that all the features and data points are on the unit hypersphere, a set of original features $d_m$ was first generated using random uniform distribution, forming 5- to 20-fold over-complete representations ($M \in [5N, 20N]$). Sample data were also distributed randomly on the unit hypersphere (data length: $T \in [2 \times 10^3, 1 \times 10^4]$). Then $L_1$ solutions $c^{(t)}$ were identified for each $t$, and one feature was arbitrarily chosen as a target ($m = 1$) for simulations.

Initial Condition

An ideal starting condition would be the one that has a single feature $\hat{d}_1$ in the area where the target feature was active ($c_1^{(t)} > 0$), and the rest features $\hat{d}_{m\neq 1}$ surrounding the area in stimulus space. For simplicity, however, a set of features was randomly generated as an initial guess with $\hat{M} = M$ in the simulations, even though there is no clue to estimate $M$ in reality. Depending on the initial conditions, the algorithm sometimes failed to find the feature of interest, but it worked well in most cases (see Sections 2.A.3 and 2.A.4).

Parameters

It is critical to choose appropriate parameters to reach the target feature. Starting with relatively larger parameter values, here we gradually decreased them as the estimated feature got closer to the target.\textsuperscript{7} Taking $N = 3$, $\hat{M} = M = 20$, and $T = 1 \times 10^3$ for example, $(\gamma, \lambda) = (0.1, 5 \times 10^{-3})$ was used for the first 20 iterations, $(0.1, 2 \times 10^{-3})$ for the next 30 iterations, then $(0.1, 1 \times 10^{-3})$ for the following 50 iterations, and $(0.02, 5 \times 10^{-4})$ for the rest 100 iterations. In this case (200 iteration times in total; Figures 2.15 and 2.16), the computation took about 45 minutes using MATLAB with 1.5 GHz processor and 1.5 GB RAM.

\textsuperscript{7}Because of the piecewise linearity (see Figure 2.9 on page 48), care should be taken about crossing the discontinuities, which results in an abrupt change of the objective value (Figure 2.15B).
Evaluation

To see how well the estimated features were determined by the algorithm, the Fisher information matrix was computed for each estimate using the approximation of Eq.(2.26). In Figure 2.16, the Hessian matrix was first calculated for each estimated feature, which transforms the unit sphere in $\mathbb{R}^3$ into an ellipsoid. The axes of the ellipsoid are given as the eigenvectors of the Hessian matrix, and the size of the ellipsoids is in proportion to the eigenvalues (but arbitrarily scaled in Figure 2.16 for presentation purposes). The ellipsoid was then projected on the surface of the unit sphere, specifically on the plane perpendicular to the corresponding estimated feature, which gave an ellipse in spherical coordinates.

2.A.3 Results

Although the success depends on the parameter values and initial conditions, the algorithm worked well in most cases to find the target feature by iterative updates (e.g., 200 iterations for Figures 2.15 and 2.16; $d_1^\top \hat{d}_1 = 0.98 \pm 0.03$ over 15 simulations, mean $\pm$ standard deviation). The trajectory of the estimation was not always the shortest path between the target feature and the one from initial guess, but the estimated feature got closer to the target as we updated the estimates with appropriate parameter values. The objective value could sometimes show an abrupt increase for the first tens of iterations (Figure 2.15B); such increases would mostly result from crossing the discontinuities due to the piecewise linear properties of the $L_1$ solutions (Figure 2.9), but the objective would eventually converge to zero (or its local minimum).

Figure 2.16 shows a typical example of estimated features with the ellipses of the Fisher information matrices (Eq.(2.26)). The ellipse for the estimated target feature was generally large, meaning that the estimate was reasonably well-determined, whereas the coactive features (or the features surrounding the target feature) had rather thin ellipses whose major axes were directed towards the estimated target feature. This suggests that the algorithm could estimate fairly well the boundary of the active region for the target feature, but it could say less about
where on the border the coactive features were located. Note that those features that were not coactive with the target had no information, i.e., there is no way to estimate these features and we cannot say anything about them.

Figure 2.17 shows estimated features for 5, 10, and 20 dimensions (200 iterations; two examples, each). The higher the dimension, the worse the performance (from low to high dimensions, $d^T_i \hat{d}_i = 0.99 \pm 0.01, 0.91 \pm 0.06, \text{ and } 0.84 \pm 0.08$ over 10, 6, and 12 simulations, respectively; mean ± standard deviation), probably due to a finite data length. Note that the number of potentially coactive features with the target increases much faster than the dimensionality, and the number of the facets around the target—i.e., the number of all possible combinations of coactive features with the target—increases even faster (Figure 2.14A). That is, if we uniformly sample stimulus space, we need a huge amount of data to determine the active region (facets) for the target in higher dimensional space (“curse of dimensionality;” Bellman, 1961), which could be a problem for estimating the target feature with this algorithm.

2.A.4 Discussion

Here we introduced a new—yet straightforward—algorithm to estimate an optimal neural feature in the context of sparse overcomplete representations. The algorithm worked well at least for rather low dimensional cases ($N \leq 20$; Figure 2.17). However, some modifications would be needed in practice to apply the algorithm to physiological data, because the current algorithm works not for online but only for offline estimation (see in contrast deCharms et al., 1998; Barbour and Wang, 2003; Machens et al., 2005; O’Connor et al., 2005), and because input signals (and features) are assumed to have a unit length. It would be difficult to appropriately normalize sensory stimuli especially for sounds without losing their characteristic structures, but some synthetic or “naturalistic” stimuli might be applicable—such as temporally orthogonal ripple combinations (TORCs) for auditory stimuli—since their total power is typically normalized (Klein et al., 2000; see also Section 3.2.4 in Chapter 3).
Figure 2.15: **Feature estimation in \( \mathbb{R}^3 \) from a simulated single-unit data set.** (A) Original features and estimated features in spherical coordinates \((N = 3, \hat{M} = M = 20, T = 1 \times 10^3)\). Blue dots and small circles show data \((c_1 = 0\) and \(c_1 > 0\), respectively), while small red open diamonds are the data showing \(\hat{c}_1 > 0\). Original features are shown in large green circles; starting from the initial guess (black squares), estimates were updated for 200 iterations in the simulations (large red diamonds). The filled symbols show the target feature. (B) The objectives as a function of iteration times. The mean square error \((MSE; E_{ls} \text{ in Eq.}(2.21))\) and the symmetrized KL divergence \((KL; E_{KL} \text{ in Eq.}(2.23))\) are shown in blue and green, respectively, and the total cost \((Total; E_{ls} + \gamma E_{KL} \text{ in Eq.}(2.25))\) is shown in red. In the simulations, the parameter values were changed at after 20, 50, and 100 iterations (for details, see Section 2.A.2). (C) Original and estimated coefficients for the target feature (blue and red lines, respectively), sorted by the coefficient values in the descending order. Because of the sparseness prior, only a subset of data (302 out of 1000) activates the target feature. The fact that the estimated coefficients are almost overlapped with the original ones indicates a successful estimation of the target feature.
Figure 2.16: **Feature estimation and Fisher information matrices.** Another simulation result in $\mathbb{R}^3$ ($N = 3$, $\tilde{M} = M = 20$, $T = 1 \times 10^3$) was displayed in the same format as in Figure 2.15A. The red ellipses around estimated features (red diamonds) represent the Fisher information matrices, approximated as in Eqs.(2.26)–(2.29). The size of the ellipses was arbitrarily rescaled for illustration purpose (for details, see Section 2.A.2).
Figure 2.17: **Feature estimation for up to 20 dimension.** The *left, middle, and right* columns show the results for 5, 10, and 20 dimensional cases (two examples each), respectively, starting from several different initial conditions (5, 3, and 3 restarts, respectively). The *blue thick lines* and *red thin lines* show the coordinate values of the original and the estimated target feature, respectively. The parameters are \((N, \hat{M}, M, T) = (5, 50, 50, 5 \times 10^3), (10, 100, 100, 1 \times 10^4), (20, 200, 200, 1 \times 10^4)\) from left to right columns, respectively, and the iteration times were 200 for all the examples.
The algorithm works best with the initial condition in which one estimated feature \( \hat{d}_1 \) is located in the area where the target is active, and the rest \( \hat{d}_{m \neq 1} \) surrounding the area. Note that the estimated features \( \hat{d}_{m \neq 1} \) far outside the area—i.e., those that are not coactive with \( \hat{d}_1 \)—do not contribute to the objective (Eq. (2.25)), and thus they cannot be estimated from the single-unit data on the target neuron (Figure 2.16). This suggests a way to improve the algorithm; i.e., to redistribute such nuisance estimates around the active region for the target, and to remove nuisance data \( (c_1 = 0) \) located far away from the region for saving computation time. We might instead specify just the boundary around the active region and the target feature for simplifying the algorithm.

Another way to improve the algorithm would be to combine the objective (Eq. (2.25)) with some other objectives for learning overcomplete dictionaries (e.g., Lewicki and Sejnowski, 2000; Kreutz-Delgad et al., 2003), such as those to maximize sparseness or to have a good separation performance. In this way, we could have a set of features that globally shows efficient coding properties, as well as a good fit to a given dataset for particular features. It would also be interesting to rewrite the algorithm in the framework of probability theory (see e.g., MacKay, 2003), using a Laplacian distribution as a prior for the coefficients. Such modifications would enable us to apply the algorithm (or its variants) to experimental data to test the sparse representation models.

One advantage of the algorithm would be its ability to find the active region for a target feature in addition to the feature itself, although the location of coactive features on the boundary cannot be estimated precisely (Figure 2.16). This would then enable us to estimate how many features are needed on average to tile the space—or, the expected degree of overcompleteness for the nervous system—by exploiting the relation: 

\[
NS_{N-1} = \sum_m A_m = M \langle A_m \rangle_m,
\]

where \( S_{N-1} = N\pi^{N/2}/\Gamma(1 + N/2) \) is the surface area of the unit hypersphere in \( \mathbb{R}^N \) with the Gamma function: \( \Gamma(z) = \int_0^\infty t^{z-1} e^{-t} \, dt \), and \( A_m \) for \( m = 1, \ldots, M \) is the surface area of the active region for the \( m \)-th feature with \( \langle A_m \rangle_m \) being its average.

\[\text{The Gamma function satisfies } \Gamma(z) = z\Gamma(z - 1) \text{ in general and is an extension of the factorial function: } \Gamma(N+1) = N! \text{ for all natural numbers } N. \text{ Also note that } \Gamma(1) = 1 \text{ and } \Gamma(1/2) = \sqrt{\pi}.\]
On the other hand, a disadvantage would be a long computational time required for each iteration. This is because a better estimate generally requires more data, and because the linear programming problem needs to be solved for all the data points for every iteration. In particular, the fact that the number of potentially coactive features and surrounding facets for a target feature increases much faster than the dimensionality of the features (Figure 2.14A) suggests that a huge amount of data (that activate the target) would be required for the feature estimation in higher dimensional cases (“curse of dimensionality;” Bellman, 1961); this could be a big problem for developing an online estimation algorithm in this framework.

Another problem is that the algorithm sometimes fails to find the target feature, presumably due to convergence to a local minimum. It would thus be safe to start with several different conditions, and see if the obtained target feature converges or not. A more sophisticated method could also be used to find the global minimum, such as simulated annealing methods (Kirkpatrick et al., 1983; Press et al., 1992). Note also that a sparse and/or biased sampling could result in several local minima, and thus a dense and uniform distribution would be preferable for the data samples for this algorithm.
Chapter 3

Bottom-up characterization: single-cell level analysis

How does the brain work? It is in fact more of a philosophy what kind of answer one would consider satisfactory; it could be full descriptions of gene expression profiles and the interactions, it could be entire wiring diagrams of the brain, or it could be complete mathematical formulations that govern spiking patterns of neurons. An answer I pursue in this chapter belongs to the latter class; i.e., the simplest model that can account for neural responses to sensory stimuli—specifically, the one describing how neurons in the auditory cortex behave in the presence of acoustic stimuli. Because our knowledge on the response dynamics is limited, however, this chapter explores bottom-up experimental characterizations of neural behaviors at the single-cell level to obtain necessary information for building a plausible model well-tailored to the observed properties. We used in vivo whole-cell patch-clamp recordings and assessed how neural responses in the (anesthetized rat) auditory cortex depend on stimulus history and its context (Asari and Zador, in preparation).

This chapter consists of six sections. Section 3.1 first gives a brief overview on previous approaches for modeling neural responses (Section 3.1.1), and describes motivations as well as background studies on context-dependence in the auditory systems (Section 3.1.2).
A novel experimental approach is then described in Section 3.2 for estimating the time course of context-dependence of auditory neurons. Details of data analysis methods are described in Section 3.3, and some neural encoding models employed here—such as spectro-temporal receptive field (STRF) models—in Section 3.4 (see also Appendix Sections A.2–A.3). Section 3.5 shows our main results; i.e., the context-dependent effects generally decayed over seconds, and could last as long as four seconds or longer in some neurons (Section 3.5.1). Such effects were shown to be more susceptible to the changes in lower-order sound properties such as overall stimulus intensities than to the changes in higher-order properties such as amplitude-modulation rates (Section 3.5.2). We also estimated the upper bound for the performance of an encoding model when only a finite length of stimulus history is available (Section 3.5.3), and identified that a window length of at least several seconds was required to capture the stimulus-related predictable response power fully enough. As shown in Section 3.5.4, however, the linear model performance did not improve substantially by just extending the window length and even by incorporating static nonlinearities. Finally, in Section 3.6, we will discuss relations to previous studies (Sections 3.6.1–3.6.2) as well as plausible mechanisms and roles of context-dependence in auditory signal processing (Section 3.6.3)—e.g., in stream segregation and others that require integration over seconds—and speculate possible directions to improve encoding models for the auditory cortical neurons (Section 3.6.4).

### 3.1 Introduction

Scientists have typically taken two complementary approaches to elucidate the laws of nature; *observation* and *theory*. To understand the sensory coding mechanisms in the brain for example, we first look for “neural correlates” of the signal processing (often those of psychophysical counterparts), and then rationalize the observations under some assumptions or hypotheses, often in mathematical terms to build up a theory. Such a theory might in turn be supported or falsified by further experimental observations, and these processes of “trials and errors” are the
key basic steps we could take towards a full understanding of a complex system such as the nervous system.

Any statement that describes a system is a model, be it qualitative or quantitative; we make some assumptions and give a most reasonable description and interpretation for empirical or theoretical understandings. Here I focus on quantitative ones—we cannot do without equations, I believe, to completely clarify a system—and Section 3.1.1 briefly overviews existing sensory coding models on single-neuron dynamics. Such quantitative models in neuroscience could range over various levels—in one extreme we can build models on the basis of detailed biophysical properties of ion channels and cell membranes (Hodgkin and Huxley, 1952; FitzHugh, 1961; Nagumo et al., 1964; Gerstner and Kistler, 2002; Agüera y Arcas and Fairhall, 2003; Agüera y Arcas et al., 2003), and in the other extreme we could treat the system as a “black-box” (Herz et al., 2006)—but those in sensory systems neuroscience can be typically classified into the latter type in a broad sense.

Section 3.1.2 then concisely describes the “observations” reported in previous work, and describes what has been missing for building a plausible encoding model—here we focus on the relevant time-scales of response dynamics in the auditory cortex to identify the minimum necessary window length for a neural encoding model.

### 3.1.1 Sensory Coding Models

One goal of neuroscience is to understand how the brain processes sensory information. As a first step, computational and systems neuroscience often targets to characterize the relationship between inputs (sensory stimuli) and outputs (neural responses), and a plenty of models and methods on this topic have been proposed over the past decades (for review, see e.g., Meister and Berry, 1999; Simoncelli et al., 2004; Herz et al., 2006; Wu et al., 2006). Such sensory coding models typically describe either of two complementary aspects of the sensory signal processing in the brain. The first—and best studied—is the encoding process by which a stimulus is converted by the nervous system into neural activity. Less studied is the decoding process,
where experimenters attempt to use neural activity to reconstruct the stimulus that evoked it. Because decoding models typically require considering the activity of populations of neurons (e.g., linear generative models for sparse or efficient codings; Olshausen and Field, 1996; van Hateren and Ruderman, 1998; Lewicki, 2002; Smith and Lewicki, 2005, 2006; see also Chapter 2), they are less experimentally accessible (but see e.g., Bialek et al., 1991; Warland et al., 1997; Stanley et al., 1999) and thus here we focus on encoding directions on single-neuron dynamics.

The central ideas of both classical and recent—more sophisticated—encoding models are; (1) to extract features as linear combinations of the inputs; and (2) to model the outputs as a nonlinear function of these features. These two steps can be considered as a “unit” process in artificial neural networks, and many encoding models—often consisting of cascades of linear-nonlinear transformations—can then be thought of as multi-layer neural networks with fewer “hidden” units to achieve sufficient dimensionality reduction for interpretation purposes (see also Appendix Section A.3.2). In this view, many learning methods developed separately in different fields including neuroscience can be considered as identical. In fact, a goal is essentially the same between the fields; “pattern” or feature identification in the data of interest (Victor, 2005). What is critical for sensory coding models is then not the (re)discovery of the methods but rather the interpretation from biological viewpoints. I would put forward that, with biologically reasonable interpretations, we could apply state-of-the-art techniques in different fields (e.g., statistics or machine learning; Tarca et al., 2007) for our own purposes, such as to decipher the neural code. Appropriate interpretations would then provide some insights on the fundamental questions of how we can go further from the characterizations to the understandings of the mechanisms underlying neural behaviors, or how we can go further from “neural correlates” to their functions (see also Chapter 4).

A good realistic model should then have both high expressive power—i.e., good fitness to data—and biologically reasonable interpretations. However, there is often a trade-off between them. On the one hand, artificial neural networks have in general high expressive power
(Hornik et al., 1989; Barron, 1993) but they are not so popular in neuroscience because the relation between model parameters and experimentally measured quantities is obscure (but see Lehky et al., 1992; Lau et al., 2002; Prenger et al., 2004). On the other hand, linear models have been widely used in the auditory—and visual—physiology due to their simplicity and interpretability (Eggermont et al., 1983; Klein et al., 2000; Theunissen et al., 2001; Escabí and Schreiner, 2002; Linden et al., 2003; Machens et al., 2004). Linear spectro-temporal receptive field (STRF) models have been fairly successful in describing the input-output function of some stimulus ensembles in auditory cortex (Kowalski et al., 1996; Depireux et al., 2001), but have generally yielded only poor results for other ensembles, including those consisting of natural stimuli (Machens et al., 2004). In this thesis, I began with the simplest model—i.e., the linear STRF model—because the concept of “receptive field” (Barlow, 1953; Kuffler, 1953; Ringach, 2004) is so simple and easy to interpret, and aimed at (1) experimentally collecting relevant information on neural dynamics and (2) searching directions to improve the model performance based on the experimental results.

3.1.2 Motivation and Background Studies

Why has the classical STRF-based approach failed to provide a general model? A straightforward answer is that the actual input-output function is nonlinear; e.g., it might include multiplicative interactions between different frequency bands. However, the space of nonlinear functions is huge, and it is not feasible to fit general high-order models (but see Marmarelis and Naka, 1972; Eggermont, 1993; Rotman et al., 2001); e.g., if the input spectrogram is discretized with a (rather coarse) frequency resolution of 1/4 octave over 5 octaves and a (rather coarse) temporal resolution of 10 msec over 200 msec, then the number of parameters for a linear model is \( N = 5/4 \times 200/10 = 400 \), while \( O(N^2) \) for a second-order Wiener model, and in general \( O(N^n) \) for an \( n \)-th order Wiener model (Volterra, 1930; Wiener, 1958). Thus the success of general “black-box” models is quickly limited by the “curse of dimensionality” (Bellman, 1961), i.e., the fact that the amount of data required to fit a general model increases
exponentially with the order of the model; even general black-box models—such as multi-layer artificial neural networks—that are guaranteed to succeed in principle are often data-limited in practice. Although in some cases these difficulties can be circumvented by the judicious choice of nonlinearities (e.g., Chichilnisky, 2001; Schwartz and Simoncelli, 2001; Fishbach et al., 2001, 2003; Nykamp and Ringach, 2002; Paninski, 2003a; Sharpee et al., 2004, 2006; Rust et al., 2005), it is difficult to know \textit{a priori} what form the nonlinearities should take.

One way to reduce the number of model parameters is to tailor the model to the observed properties of auditory cortex neurons. The parameter count above illustrates that the system’s “memory”—i.e., the dependence of the neuron’s input-output behavior on stimulus history or context—is one of the primary determinants of model complexity; doubling the length of the memory (e.g., from 200 to 400 msec in the example above) doubles the number of input variables (from $N$ to $2N$ for fixed temporal resolution). Thus it would be useful to characterize the length of the system’s memory.

Sensory signal processing in the brain indeed depends on stimulus history and its context. The visual system is by far the best characterized sensory system, and the effects of stimulus history have been examined over a wide range of time-scales, from tens of milliseconds to minutes (for a review, see Kohn, 2007). For acoustic signal processing, psychophysical studies have demonstrated that stimulus context strongly affects sound perception and auditory scene analysis in humans (Massaro, 1972; Bregman, 1990; Oxenham, 2001), and physiological studies on the sensitivities to stimulus context (e.g., forward masking) suggest that the responses of neurons in the primary auditory cortex (area A1) are highly dependent on stimulus history (e.g., Abeles and Goldstein, 1972; Hocherman and Gilat, 1981; Phillips, 1985; Calford and Semple, 1995; Brosch and Schreiner, 1997, 2000; Wehr and Zador, 2005). Most auditory physiological studies so far, however, have looked only at shorter time-scales (i.e., tens or hundreds of milliseconds) except for some recent works (Condon and Weinberger, 1991; Malone et al., 2002; Ulanovsky et al., 2003, 2004; Bartlett and Wang, 2005), and it remains to be addressed how long and how large past events would influence the activity of neurons in A1.
We have therefore developed a novel experimental approach for estimating the time course of context-dependence of neurons in the primary auditory cortex of anesthetized rats and compared it to several encoding models (Sections 3.2–3.4). Whole-cell patch-clamp recordings in vivo were used to examine subthreshold responses to various sequences of natural and synthetic stimulus ensembles with rich temporal and spectral structures. We found that context-dependence could last for a surprisingly long time—sometimes as long as four seconds or longer (Section 3.5). However, extending the memory of linear models did not improve their performance, indicating that these long-lasting effects of context were nonlinear. The slow stimulus adaptation may play a role in stream segregation and other forms of auditory processing that require integration over seconds (Section 3.6).

3.2 Experimental Methods

3.2.1 Surgery

Long-Evans rats (20–27 days old) were anesthetized (30 mg/kg ketamine and 0.24 mg/kg medetomidine) in strict accordance with the National Institutes of Health guidelines as approved by the Cold Spring Harbor Laboratory Animal Care and Use Committee. After the animal was deeply anesthetized, it was placed in a custom naso-orbital restraint, which left the ears free and clear. A cisternal drain was made, and a small craniotomy and durotomy were performed above the left primary auditory cortex (area A1). The cortex was covered with physiological buffer (in mM: NaCl, 127; Na₂CO₃, 25; NaH₂PO₄, 1.25; KCl, 2.5; MgCl₂, 1; and glucose, 25) mixed with 1.5% agarose. Temperature was monitored rectally and maintained at 37 °C using a feedback controlled blanket. Depth of anesthesia was monitored throughout the experiment, and supplemental anesthesia was provided when required.
3.2.2 Whole-Cell Patch-Clamp Recordings

Whole-cell recordings were obtained in vivo using standard blind patch-clamp methods (Pei et al., 1991; Margrie et al., 2002; Machens et al., 2004). Electrodes were pulled from filamented, thin-walled, borosilicate glass (outer diameter, 1.5 mm; inner diameter, 1.17 mm; World Precision Instruments, Sarasota, FL) on a vertical two-stage puller (Narishige, East Meadow, NY). Internal solution contained (in mM): KCl, 20; K-Gluconate, 100; HEPES, 10; MgCl$_2$, 2; CaCl$_2$, 0.05; Mg-ATP, 4; Na$_2$GTP, 0.3; Na$_2$-Phosphocreatine, 10; and ~2.5% micro-emerald or micro-ruby (dextran-conjugated fluorescent dye; Invitrogen, catalogue number: D-7156 and D-7162, respectively); pH 7.3; diluted to 275 mOsm. Resistance to bath was 3.5–5.0 MΩ before seal formation. We used a custom data acquisition system written in MATLAB (Mathworks, Natick, MA), and sampled membrane potential at 10 kHz using an Axopatch 200B (Molecular Devices, Palo Alto, CA) in current-clamp mode with no online series resistance compensation. Mean series resistance was 68.8±16.7 MΩ (mean ± standard deviation; 189 cells). Note that whole-cell recordings select for neurons solely on the basis of the experimenter’s ability to form a “GΩ seal,” as opposed to conventional extracellular methods (e.g., using tungsten electrodes; Hubel, 1957) that have a selection bias towards high-firing neurons.

Recordings were made from A1 as determined by the tonotopic gradient and by the frequency-amplitude tuning properties of cells and local field potentials. We recorded from almost all subpial depths (range: 85–847 µm, as determined from micro-manipulator travel; see also Figure 3.6). Thirteen cells were recovered histologically, which were verified to be pyramidal cells (e.g., Figure 3.4C). All together, we recorded from 194 cells in 139 animals, out of which 123 cells met our criterion for the analysis (see below Section 3.3). Of these, 39 cells were examined with natural sounds, 14 cells with temporally-orthogonal ripple combinations, 6 cells with dynamic moving ripples, 39 cells with modulated harmonic tones, and 27 cells with modulated pink noise (see below Section 3.2.4, and also Table 3.1 and Figure 3.2).
Sound properties varied among conditioning stimuli

<table>
<thead>
<tr>
<th>Probe</th>
<th>All</th>
<th>Amplitude</th>
<th>Frequency</th>
<th>AM</th>
<th>FM</th>
<th>Higher-order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Sound</td>
<td>305 (39)</td>
<td>23 (23)</td>
<td>25 (25)</td>
<td>25 (25)</td>
<td>25 (25)</td>
<td>63 (27)</td>
</tr>
<tr>
<td>TORC</td>
<td>39 (9)</td>
<td>40 (9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMR</td>
<td>8 (2)</td>
<td>20 (5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MHT</td>
<td>25 (20)</td>
<td>25 (21)</td>
<td>71 (27)</td>
<td>57 (25)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>74 (27)</td>
</tr>
</tbody>
</table>

Table 3.1: **Summary of recording data.** Shown is the number of probe stimuli tested in at least two different contexts repeated over at least four trials. Each row represents the probe stimulus type whereas each column indicates the stimulus properties varied among the conditioning stimuli. The corresponding number of recorded cells in A1 is shown in parentheses. A given cell could be tested with more than one probe stimulus, each of which could in turn be tested with more than one type of conditioning stimulus ensemble (see also Sections 3.2.3–3.2.4 and Figure 3.1). *Natural sounds* would differ in *all* possible sound properties, allowing us to examine overall context-dependence effects (Figures 3.4 and 3.5), whereas synthetic sounds (*TORC*, *DMR*, *MHT*, and *MPN*) were used to examine the effects caused by the changes in each of the following sound properties among conditioning stimuli (Figure 3.8): *amplitude* over the maximum range of 60 dB attenuation, *frequency* with the maximum shift of 4 octaves (e.g., Figure 3.7), *AM* and *FM* with the maximum difference of 20 Hz (rate) and 3-fold (depth), and *higher-order* properties by comparing natural sounds and corresponding modulated pink noise. Acronyms: TORC, temporally-orthogonal ripple combination; DMR, dynamic moving ripple; MHT, modulated harmonic tone; MPN, modulated pink noise; AM, amplitude-modulation; FM, frequency-modulation; A1, primary auditory cortex.
3.2.3 Stimulus Design

During the recordings, sequences of various stimulus combinations were presented in a randomly interleaved manner. To maximize the yield in finite recording length (typically \( \sim 20 \) min), \( N \) sequences were generated for \( N \) stimulus fragments \((S_i \text{ for } i = 1, \ldots, N)\) that allow us to examine the responses to all stimulus pairs \((S_i S_j; \text{conditioning stimulus } S_i, \text{probe stimulus } S_j)\) and to each stimulus following a “silent” period, i.e., an inter-sequence interval. More formally, we presented given \( N \) stimulus fragments and a “silent” period as in a cyclic code over a finite field \( \mathbb{F}_{N+1} \) of block length 2 (van Lint, 1992). For \( N = 4 \), for example, we would randomly present the following four sequences: \( S_1 S_3 S_2 S_4 \), \( S_3 S_4 S_1 S_2 \), \( S_4 S_2 S_3 S_5 \), and \( S_2 S_2 S_1 S_4 S_3 \) (Figure 3.1A). Inter-stimulus intervals and inter-sequence intervals were 0 and \( \sim 6 \) sec, respectively.

Natural sound ensembles were used to examine the overall context-dependent effects (Section 3.5.1), whereas synthetic sound ensembles (with or without one additional natural sound fragment) were used to examine the effects of the changes in each of the following acoustic properties (Section 3.5.2); amplitude, frequency, amplitude-modulation (AM), frequency-modulation (FM), and higher-order spectro-temporal acoustic features (for details of the stimulus ensembles, see Section 3.2.4 and also Figure 3.1B). Although here we used the same stimulus design in these two cases, care should be taken for the analysis. Any stimulus fragment can be a probe stimulus in both cases, and in the former analysis, any stimulus (including “silence”) can be a conditioning stimulus as well (see also Figures 3.1A and 3.4). In the latter case, however, only an appropriate set of stimulus fragments can be used as conditioning stimuli; e.g., to assess the frequency change effects with three synthetic frequency variants \((S_i \text{ for } i = 1, \ldots, 3)\) and one natural sound fragment \((S_4)\), the stimulus design in Figure 3.1A for instance can be used for the recordings, but the analysis should be conducted by using only those frequency variants \((S_i \text{ for } i = 1, \ldots, 3)\) as conditioning stimuli for each probe fragment \((S_i \text{ for } i = 1, \ldots, 4; \text{see also Figure 3.7})\). For details of the analysis, see Section 3.3.
Figure 3.1: **Experimental design and auditory stimuli used in this study.** (A) Experimental design for analyzing context-dependence. During the recording, we presented well-designed \( N \) sequences of a given set of \( N \) sound fragments with no inter-stimulus interval in a randomly interleaved manner (shown is an example for \( N = 4 \); inter-sequence interval was \( \sim 6 \) sec in this study). For the analysis, we aligned the recording data to examine the response variability to a given sound fragment (probe; \( S_1 \) in this example) due to the presence of different preceding stimuli (context; “silence,” \( S_1 \), \( S_2 \), \( S_3 \), and \( S_4 \) in this example). The choice of conditioning stimuli depends on the goal of the analysis (see Sections 3.2.3–3.2.4). Note that the response power (light gray line; \( \mathcal{P}[r_{ij}(t)] \) from Eq.(3.12)) to a probe stimulus at time \( t \) from the onset of the probe can be divided into noise power (\( \mathcal{P}[\varepsilon_{ij}(t)] \)) and stimulus-related power (black line; \( \mathcal{P}[\mu(t) + \nu_i(t)] \)) that can be further decomposed into context-independent fraction (\( \mathcal{P}[\mu(t)] \) in Eq.(3.19)) and context-dependent fraction (dark gray line; \( \mathcal{P}[\nu_i(t)] \) in Eq.(3.20)).

(B) Natural sounds and synthetic sounds. Natural sound fragments (\( S_{NS1} \) and \( S_{NS2} \); 4.11 sec long; sound pressure waveforms, spectrograms, and temporal and spectral marginal distributions are shown.) differ a lot, which causes a large and long context-dependence when they are used as conditioning stimuli (Figures 3.4 and 3.5). On the other hand, the temporal and spectral patterns in the marginal distributions between modulated pink noise \( S_{MPN1} \) and the corresponding natural sound \( S_{NS1} \) are nearly identical, resulting in a small and short context-dependence (Figures 3.8 and 3.9). Synthetic sounds such as modulated harmonic tones (Eq.(3.4); sound pressure waveform and corresponding spectrograms; 1 sec long) can be used to assess the effects of the changes in sound properties in more detail. Compared to \( S_{MHT} \) for example, \( S_{\Delta AMP} \) has 30 dB less power, \( S_{\Delta FREQ} \) has the frequency components up-shifted by 1.5 octaves, \( S_{\Delta AM} \) has slower amplitude-modulation rates by 4 Hz on average, and \( S_{\Delta FM} \) has half the standard deviation for the frequency-modulation depth. We also used temporally-orthogonal ripple combinations and dynamic moving ripples (Eqs.(3.1)–(3.3); examples are not shown here, but see elsewhere, e.g., Klein et al., 2000; Escabí and Schreiner, 2002) but only for examining the effects of amplitude and frequency changes in conditioning stimuli. For details, see Sections 3.2–3.3.
3.2.4 Stimuli

All stimuli were delivered at 97.656 kHz using a TDT System 3 with an ED1 electrostatic speaker (Tucker-Davis Technologies, Alachua, FL) in free-field configuration (speaker located \(\sim 8\) cm lateral to, and facing, the contralateral ear) in a double-walled sound booth (Industrial Acoustics Company, Bronx, NY). The speaker had a maximum intensity (at 10 V command voltage) of 92 dB sound pressure level (SPL), and its frequency response was flat from 1 to 22 kHz to within standard deviation of 3.7 dB. Sound levels were measured with a type 7012 one-half inch ACO Pacific microphone (ACO Pacific, Belmont, CA) positioned where the contralateral ear would be (but with the absence of animal).

Natural Sounds

All natural sound fragments were taken from commercially available audio compact discs (CDs), originally sampled at 44.1 kHz and resampled at 97.656 kHz for stimulus presentation: *The Diversity of Animal Sounds* and *Sounds of Neotropical Rainforest Mammals* (Cornell Laboratory of Ornithology, Ithaca, NY). The majority of the sound sections lasted for 3.5–6.5 sec, but some were shorter (2–3 sec) to examine as many stimulus combinations as possible (see above the cyclic stimulus design in Section 3.2.3 and Figure 3.1). The sound segments were chosen from original sound tracks to have minimum “silent” periods (especially at the onset and termination), and a 5 msec cosine-squared ramp was applied at the onset and termination to make sure a smooth connection between the segments even with no inter-stimulus interval.\(^1\) The peak amplitude of each segment was normalized to the \(\pm 10\) V range of the speaker driver. The natural sound stimuli consisted of 46 different sounds in total, covered almost all frequencies from 0 to 22 kHz, and ranged from narrow- to broad-band stimuli. Note however that only a subset of stimuli (typically \(N \sim 7\) fragments) was tested on any particular cell.

\(^1\)No “click” sound (due to discontinuity) was detected at the connection point at least to my ears.
Synthetic Sounds

All synthetic sounds were sampled at 97.656 kHz and lasted for 4–5.5 sec. We used temporally-orthogonal ripple combinations (TORCs; for details, see e.g., Klein et al., 2000) and dynamic moving ripples (DMRs; for details, see e.g., Escabí and Schreiner, 2002) to examine how the changes in amplitudes and frequencies in conditioning stimuli would affect the responses to following probe stimuli. Modulated harmonic tones were used to assess the effects of AM and FM changes as well as the changes in amplitudes and frequencies; and modulated pink noise was used for testing the effects of the changes in higher-order sound properties, such as complex interactions between spectro-temporal constituents (Figure 3.1B and Table 3.1). In this study, no cell was tested with all the sound properties due to a limited recording length (typically, ∼20 min). Strictly speaking, we cannot directly compare the context-dependence effects caused by the changes in different acoustic properties; however, the comparisons we made (Figure 3.9) would not be unfair because the acoustic properties were varied across almost entire range that A1 neurons can follow faithfully (e.g., sound trains or temporal modulations up to tens of Hz; Creutzfeldt et al., 1980; Joris et al., 2004).

Temporally-Orthogonal Ripple Combinations  The following equation was used to generate ripples and their combinations:

\[ y(t) = \sum_i y_E(t, x_i) \times y_C(t, f_i), \]  \hspace{1cm} (3.1)

where the envelope \( y_E(t, x_i) \) and carriers \( y_C(t, f_i) \) are respectively given by:

\[ 20 \log_{10}[y_E(t, x_i)] = a_0 + \sum_{j,k} \frac{a_{jk}}{2} \cos[2\pi(\Psi_j(t) + \Omega_k(t) x_i) + \Phi_{jk}], \]  \hspace{1cm} (3.2)

\[ y_C(t, f_i) = \sin[2\pi f_i t + \varphi_i]. \]  \hspace{1cm} (3.3)
Note that $a_{kj}$ ($> 0$) is a sinusoidal modulation depth around the mean $a_0$ in dB, $\omega_j(t) = \partial \Psi_j / \partial t$ in Hz and $\Omega_k(t)$ in cycles/octave are temporal and spectral ripple modulations, respectively, $x_i = \log_2[f_i/f_0]$ in octaves is a logarithmic frequency axis relative to $f_0$ in Hz, and $\Phi_{jk}$ and $\varphi_i$ are random initial phases.

For generating TORC-based stimuli, envelopes of seven “short” TORCs were first generated, each consisting of six ripples with temporal modulation: $\omega_j = 4j$ Hz (for $j = 1, \ldots, 6$), and each having a fixed ($k = 1$) spectral modulation: $\Omega = -1.5, -0.9, -0.3, 0, 0.6, 1.2,$ and $1.8$ cycles/octaves, respectively. All TORCs had rise and fall times of $5$ msec, modulation depth of $\leq 30$ dB ($a_{jk} = 30/6$), and lasted for $250$ msec. Such short TORC envelopes were then randomly adjoined to generate a “default envelope” that lasted for $4–5.5$ sec.

To examine the effects of the changes in sound intensities in conditioning stimuli, we applied the default envelope to the carrier frequencies over a bandwidth of $5$ octaves (0.88 to 28.16 kHz in steps of $1/128$ octaves), scaled the peak amplitude to the speaker driver range, and then varied the amplitudes over the maximum range of $40$ dB attenuation. To examine the effects of frequency changes, we generated a default envelope over the bandwidth of $2$ or $4$ octaves, and chose carrier sinusoidals within the range of $6$ octaves (0.625 to 40 kHz in steps of $1/128$ octaves, random phase at time zero) so that the signals had the same envelope with shifted bandwidth (e.g., 0.625–10 kHz, 2.5–40 kHz, and so on). We then normalized the peak amplitude of sound fragments with respect to their total signal powers ($\int |y(t)|^2 dt$) and uniformly scaled to fit them all within the speaker driver range.

**Dynamic Moving Ripples** To generate the DMR envelopes (Eq.(3.2); $j = k = 1$), spectral modulations $\Omega(t)$ were sampled at $6$ Hz from a uniform distribution in interval $\pm 1.5$ cycles/octave, and temporal modulations $\omega(t)$ were sampled at $3$ Hz from a uniform distribution ranging between $\pm 25$ Hz, both of which were then up-sampled to $97.656$ kHz using a cubic interpolation procedure (interp1 function with the “cubic” option in MATLAB).
Carrier frequencies were chosen as in the TORC stimuli, and applied to the envelope as in Eqs.(3.1)–(3.3). All DMR signals had rise and fall times of 5 msec and modulation depth of \( \leq 30 \) dB. The peak amplitude was scaled in the same way as the TORC fragments (see above).

**Modulated Harmonic Tones**  The following equation was used to generate modulated harmonic tones:

\[
y_{\text{MHT}}(t) = A(t) \sum_{i=0}^{M-1} \cos \left[ 2^{i/m} \phi(t) + \varphi_i \right],
\]

where \( A(t) \) is the envelope, and \( \varphi_i \) and \( \phi(t) \) are the initial and time-varying phases, respectively.

In this study, \( M = 5 \) tones were combined with the density: \( m = 0.5 \) or 1 tone/octave. The derivative of the phase \( \phi(t) \) with respect to time gives the instantaneous frequency \( f(t) \):

\[
\frac{\partial \phi(t)}{\partial t} = 2\pi f(t).
\]

Normal distributions sampled at 48 Hz were used to generate the envelope \( A(t) \) and the instantaneous frequency \( f(t) \). The mean and standard deviation of \( A(t) \) were chosen to be [40–65] and [15–5] dB, respectively, whereas the mean \( f(t) \) ranged over 3 octaves (0.375–3 kHz) and the standard deviation from 1 to 1/3 octaves. We then up-sampled \( A(t) \) and \( f(t) \) to 97.656 kHz using a cubic interpolation procedure, and used Eqs.(3.4) and (3.5) to generate the signal \( y_{\text{MHT}}(t) \) with random initial phase \( \varphi_i \).

For examining the effects of amplitude changes, we generated a signal for given \( A(t) \) and \( f(t) \), normalized its peak amplitude within the speaker driver range, and varied the amplitudes over the maximum range of 60 dB attenuations. For examining the effects of frequency changes, we generated signals with a fixed \( A(t) \) but shifted \( f(t) \) by up to ±2 octaves, normalized their total signal powers, and uniformly scaled the signals to fit them all within the speaker driver range. For examining the effects of the changes in AM or FM, we used fixed mean \( A(t) \) and \( f(t) \). Before the up-sampling procedures, however, \( A(t) \) or \( f(t) \) were scaled to vary the modulation depth by up to 3-fold and/or limited the modulation rates to have the bandwidth of
4 Hz; i.e., the fastest and slowest modulation rates were [20–24] and [0–4] Hz, respectively. The synthetic signals were normalized with respect to their total power, and uniformly scaled to fit them all within the speaker driver range.

**Modulated Pink Noise** Starting from white noise \( x_0(t) \), we used the following iterative procedures to produce modulated pink noise \( y_{\text{MPN}}(t) \) that has asymptotically the same temporal and spectral modulation patterns as a target natural sound \( y_{\text{NS}}(t) \). First, we computed the analytic signal of \( y_{\text{NS}}(t) \) by using the Hilbert transform \( \mathcal{H}[\cdot] \), and decomposed it into the envelope \( A_{\text{NS}}(t) \) and the phase \( \phi_{\text{NS}}(t) \):

\[
y_{\text{NS}}(t) + \mathcal{H}[y_{\text{NS}}(t)] = A_{\text{NS}}(t) \exp[j\phi_{\text{NS}}(t)],
\]

(3.6)

where \( j^2 = -1 \). Second, using the Fourier transform \( \mathcal{F}[\cdot] \), we filtered the signal from the \((i - 1)\)-th iteration \( x_{i-1}(t) \) to have the same power spectrum as \( y_{\text{NS}}(t) \):

\[
\tilde{z}_i(\omega_k) = \tilde{x}_{i-1}(\omega_k) \cdot \left| \frac{\tilde{y}_{\text{NS}}(\omega_k)}{\tilde{x}_{i-1}(\omega_k)} \right|,
\]

(3.7)

where \( \tilde{y}(\omega) = \mathcal{F}[y(t)] \) denotes the signal \( y \) in the Fourier domain. Third, we computed the analytic signal of \( z_i(t) = \mathcal{F}^{-1}[\tilde{z}_i(\omega)] \) as in Eq.(3.6):

\[
z_i(t) + \mathcal{H}[z_i(t)] = B_i(t) \exp[j\psi_i(t)],
\]

(3.8)

where \( B_i(t) \) and \( \psi_i(t) \) are the envelope and the phase, respectively. Finally, we generated the signal for the \( i \)-th update as:

\[
x_i(t) = A_{\text{NS}}(t) \cos[\psi_i(t)].
\]

(3.9)

That is, \( x_i(t) \) is a pink noise with the envelope of the target \( y_{\text{NS}}(t) \). In this study, we updated the synthetic signal 1000 times to generate modulated pink noise: \( y_{\text{MPN}}(t) = x_{1000}(t) \). The signals
$y_{NS}(t)$ and $y_{MPN}(t)$ were normalized with respect to their total power, and then uniformly scaled to fit them all within the speaker driver range.

### 3.3 Data Analysis Methods

All data analysis was performed in MATLAB. As a preprocessing, we applied a median filter to clip spikes from the recording data (10 msec window), and centered the subthreshold responses to have zero mean (i.e., $r(t) - \langle r(t) \rangle_t$, instead of subtracting the resting potential; $\langle \cdot \rangle_t$ indicates the average over time $t$; see also Figure 3.2).

The noise correlation was computed as the auto-correlation of additive noise $\varepsilon_{ij}(t)$ (defined as in Eq.(3.12) and estimated as in Eq.(3.15); see also Eq.(A.85) on page 168):

$$\rho_{ij}(t) = \frac{\varepsilon_{ij}(-t) \ast \varepsilon_{ij}(t)}{\sigma_{\text{noise}}^2}, \quad (3.10)$$

where $\ast$ indicates convolution. The correlation function $\rho_{ij}(t)$ represents a similarity in the noise signal over time, and thus a long correlation time suggests the existence of slow internal (stimulus-independent) dynamics in neural behaviors. However, our data showed that $\rho_{ij}(t)$ had a sharp peak with a rapid decay (within a few hundred milliseconds; Figure 3.2D) and thus a slow-time scale identified in neural responses in A1 (see Section 3.5 and Figure 3.11) would be governed by stimulus-dependent mechanisms. Together with the fact that the noise distribution agrees fairly well to the Gaussian distribution with the same standard deviation (Figure 3.2C), our data support the additive independent and identically-distributed (i.i.d.) Gaussian noise assumption in our model in Eq.(3.12).
Figure 3.2: **Recording data statistics.** (A) Distribution of recorded membrane potential (around its mean value). The mean membrane potential over time was first subtracted from entire recording data for each cell (i.e., \( r(t) - \langle r(t) \rangle_t \), where \( \langle r(t) \rangle_t = -60.5 \pm 6.4 \) mV; mean ± standard deviation; 123 cells), and histogram was made in step of 0.1 mV; shown is then the mean probability distribution over the populations. The peak value (mode; −4.0 mV) would be a good estimate of the resting potential, i.e., −64.5 mV. (B) Distribution of mean correlation coefficients over trials (gray arrow head indicates the mean). All pairwise Pearson product-moment correlation coefficients were first computed between response traces to each stimulus fragment over trials, and the average was taken over pairs and stimuli for each cell; shown is the distribution of this average value over the population (0.27±0.09; mean ± standard deviation; 123 cells). (C) Distribution of additive noise components (estimated as in Eq.(3.15)). Although the noise distribution (black) has a slightly long tail in the positive direction (mode; −1.0 mV), it agrees fairly well to the Gaussian distribution (gray) with the same standard deviation (\( \hat{\sigma}_{\text{noise}} = 5.1 \) mV). (D) Noise correlation (Eq.(3.10)). Note the sharp peak at time zero with a rapid decay within a few hundred milliseconds. Together with (C), the results suggest that the additive i.i.d. Gaussian noise assumption (\( \varepsilon_{ij}(t) \sim \mathcal{N}[0, \sigma^2_{\text{noise}}] \)) more or less holds.
3.3.1 Context-Dependence

For those recordings that we could test at least four repeats on any given combinations of a probe stimulus and at least two conditioning stimuli, context-dependence—i.e., the response variability to a probe stimulus due to the presence of different conditioning stimuli—was examined in two ways; (1) significance measure in the statistics sense, and (2) fractional power measure in the response dynamics.

Significance Measure

For each sampled time point $t$ on a probe stimulus (with $t = 0$ indicating the transition from conditioning to probe stimuli), we performed a one-way non-parametric analysis of variance (Kruskal-Wallis test; Kruskal and Wallis, 1952) for equal medians among the responses $r_{ij}(t)$ over trials $j = 1, \ldots, m$ in all the contexts $i = 1, \ldots, n$ we examined; see also Eq.(3.12). Briefly, we first ranked the data $r_{ij}(t)$ into integers $r_{ij}(t)$ from 1 to $mn$ for each time point $t$. The test statistic $KW(t)$ is then given as:

$$KW = (mn - 1) \sum_i m(\langle r_{ij} \rangle_j - \langle \langle r_{ij} \rangle_j \rangle_i)^2 \sum_i \sum_j (r_{ij} - \langle \langle r_{ij} \rangle_j \rangle_i)^2 = \frac{3\langle (2\langle r_{ij} \rangle_j - mn - 1)^2 \rangle_i}{mn + 1},$$

(3.11)

where $\langle \cdot \rangle$ indicates the average over trials with subscript $j$ and over contexts with subscript $i$, and the probability distribution of $KW$ can be approximated as a chi-square distribution with $n - 1$ degrees of freedom. (The time $(t)$ is ignored here for brevity.)

Our criterion for the significance level was $p < 0.01$ for $\geq 5$ msec (i.e., at least 50 consecutive time points to avoid false positives due to multiple comparisons over time). Here we did not perform any post hoc analysis, partly because of data limitation, and partly because the goal of this analysis was to detect whether or not neurons showed context-dependence but not to identify what causes the difference in the response patterns—stimulus space is huge in general and thus we asked the latter question by taking a special stimulus design with synthetic sounds (see Sections 3.2.3–3.2.4); i.e., we fixed all the stimulus properties among condition-
ing stimuli but one so as to examine the effects of changes in a particular sound property of interest. The significance measure was then used in the population data analysis to compute the proportion—or, the probability—of observing significant context-dependence at a given moment. The noise floor—or, the level of false positive—was determined by resampling methods, where the trials were randomly shuffled to lose the information on the contexts, followed by the same significance test described above. For each probe, we repeated this procedure 1000 times, and took the average over the population to identify the chance level of declaring “significance” in this analysis.

**Fractional Power Measure**

A second measure was introduced to examine the contribution of context-dependence to response dynamics; i.e., a quantity meaningful from a modeling (instead of just statistical) perspective, based on the fraction of the response power that depends on stimulus history and its context (see also Section 3.3.3). We first assumed the following additive model for the response to a probe stimulus over time $t$:

$$ r_{ij}(t) = \mu(t) + \nu_i(t) + \varepsilon_{ij}(t). \quad (3.12) $$

That is, the observed response $r_{ij}(t)$ in the $i$-th context for the $j$-th trial consists of i.i.d. Gaussian noise: $\varepsilon_{ij}(t) \sim \mathcal{N}[0, \sigma_{\text{noise}}^2]$—with zero mean and the variance of $\sigma_{\text{noise}}^2$—and stimulus-related (predictable) parts, which can be further decomposed into context-dependent and independent fractions: $\nu_i(t)$ and $\mu(t)$, respectively. Here, each component can be estimated as:

$$ \tilde{\mu}(t) = \left\langle \left\langle r_{ij}(t) \right\rangle_j \right\rangle_i \quad (3.13) $$

$$ \tilde{\nu}_i(t) = \left\langle r_{ij}(t) \right\rangle_j - \left\langle \left\langle r_{ij}(t) \right\rangle_j \right\rangle_i \quad (3.14) $$

$$ \tilde{\varepsilon}_{ij}(t) = r_{ij}(t) - \left\langle r_{ij}(t) \right\rangle_j \quad (3.15) $$
where \(\langle \cdot \rangle\) indicates the average over probes (with no subscript).

Then the measure was defined as the context-dependent fractional power: 
\[
P[\nu_i(t)] \overset{\text{def}}{=} \langle \langle \nu_i^2(t) \rangle_i \rangle,
\]
normalized by the predictable response power: 
\[
P[\mu(t) + \nu_i(t)].
\]
In the infinite data limit (i.e., every possible probe examined with all contexts), the predictable response power would become time-invariant: 
\[
P[\mu(t) + \nu_i(t)] \to \sigma^2 \overset{\text{def}}{=} \langle P[\mu(t) + \nu_i(t)] \rangle_t,
\]
where the symbol “\(\overset{\text{def}}{=}\)” means “equal in expectation.” In practice, however, the population average of the response power was nonstationary over time—typically, with large fluctuation soon after the transition from conditioning to probe stimuli (Figure 3.5B) because of a finite recording length and stimulus design—and thus the predictable response power (the denominator of Eqs.(3.21) and (3.23) shown below) was smoothed by taking the running average over \([0, 2t] \) at time \(t\). Note that the “power” is usually computed as the average over time, but we here assumed ergodicity and thus took the average over the populations.

The fractional response powers can be estimated by considering the “average power” and the “power of the average”, under the assumption that the additive components in Eq.(3.12) are all uncorrelated between each other at any given moment (see also Sahani and Linden, 2003). Considering the average over trials (for \(j = 1, \ldots, m\)), we have:

\[
\langle \langle r_{ij}^2(t) \rangle_j \rangle_i \overset{\text{def}}{=} P[\mu(t) + \nu_i(t)] + \langle \langle \varepsilon_{ij}^2(t) \rangle_j \rangle_i,
\]

\[
\langle \langle r_{ij}(t) \rangle_j \rangle^2_i \overset{\text{def}}{=} P[\mu(t) + \nu_i(t)] + \langle \langle \varepsilon_{ij}(t) \rangle_j \rangle_i^2.
\]

From central limit theorem, we have: 
\[
\langle \langle \varepsilon_{ij}(t) \rangle_j \rangle_i^2 \overset{\text{def}}{=} \langle \langle \varepsilon_{ij}^2(t) \rangle_j \rangle_i / m,
\]
and thus:

\[
\hat{P}[\mu(t) + \nu_i(t)] = \left\langle \left\langle \frac{m \langle r_{ij}(t) \rangle_j^2 - \langle r_{ij}^2(t) \rangle_j}{m - 1} \right\rangle_i \right\rangle,
\]

where \(\langle \cdot \rangle\) (without subscripts) indicates the average over all the tested probe stimuli in the population data. Similarly, considering the average of the trial average (\(\langle r_{ij}(t) \rangle_j \)) over contexts
(i = 1, \ldots, n), we have:

\[
\hat{P}_i[\mu(t)] = \langle n\langle r_{ij}(t)\rangle_j^2 - \langle r_{ij}(t)\rangle_j^2 \rangle_i / (n - 1).
\]  

(3.19)

Therefore, from Eqs.(3.18) and (3.19), the context-dependent fractional power can be given in expectation as:

\[
\hat{P}[\nu_i(t)] = \hat{P}[\mu(t) + \nu_i(t)] - \hat{P}[\mu(t)],
\]  

(3.20)

and we could use the following quantity as a measure of the contribution of context-dependence to response dynamics:

\[
\frac{\hat{P}[\nu_i(t)]}{\hat{P}[\mu(t) + \nu_i(t)]}.
\]  

(3.21)

The denominator was smoothed by taking the moving average over \([0, 2t]\) at time \(t\) before computing Eq.(3.21) (and Eq.(3.23) in Section 3.3.3).

This fractional power measure differs from the significance measure in two ways. First, the former is continuous over time, whereas the latter involves arbitrary thresholding procedure to determine the significance level and is consequently binary. Second, the fractional power measure involves the normalization (by the stimulus-related power; \(P[\mu(t) + \nu_i(t)]\)) and thus depends on the “relative” difference to the overall fluctuation between the response patterns caused by stimulus history, whereas the significance measure (and the numerator of the fractional power measure; i.e., context-dependent fraction; \(P[\nu_i(t)]\)) depends on the “absolute” differences. Therefore the fractional power measure can be said to be more faithful in these respects (see also the relation to the response predictability in Section 3.3.3).
3.3.2 Exponential Curve Fit

To measure the relevant time-scales of the context-dependence, we fit a (sum of) exponential processes to the population data (Figures 3.5, 3.8, and 3.11):

\[ \alpha + \sum_k \alpha_k \exp \left( -\frac{t}{\tau_k} \right), \]  

(3.22)

where \( \alpha_k \) and \( \tau_k \) indicate the decay size and constant, respectively. We used the \texttt{lsqcurvefit} function in the MATLAB Optimization Toolbox for the curve fitting, and used the following criterion for choosing the number of exponential processes: \( |\alpha_k| > \sum_k |\alpha_k| / 10 \) for all \( k \), that is, the contribution of an exponential process must be at least one-tenth of the total. Confidence intervals of the parameters were computed by resampling methods (200 repeats with randomly selected 1000 samples; Figure 3.9 on page 118).

3.3.3 Response Predictability

To analyze how response predictability in A1 depends on stimulus history and its context over time, we computed the time-course of the ratio between context-independent fractional power, \( \mathcal{P}[\mu(t)] \), and the stimulus-related response power, \( \mathcal{P}[\mu(t) + \nu_i(t)] \). From a modeling viewpoint, the former is the best we could reach in the response estimation exploiting the stimulus history for a limited duration of \( t \), whereas the latter is the very best level that no model could outperform under the additive noise assumption. (Note that the context-dependent response power \( \mathcal{P}[\nu_i(t)] \) is the fraction that is \textit{not} accessible when only a finite stimulus history is available, and that the noise power \( \sigma^2_{\text{noise}} = \langle \mathcal{P}[\epsilon_{ij}(t)] \rangle_t \) is the fraction that is \textit{never} accessible under the additive noise assumption.) Therefore, the following ratio indicates the context-dependence of the response predictability, which constitutes an upper-bound estimate of the response prediction
performance given a window length $t$:

$$\frac{\hat{P}[\mu(t)]}{\hat{P}[\mu(t) + \nu_i(t)]}.$$  \hspace{1cm} (3.23)

This Eq.(3.23) was computed from Eqs.(3.18) and (3.19).

### 3.4 Encoding Models

Several neural encoding models were examined in this study to compare their performance to the upper-bound estimate (Eq.(3.23)) for further analyzing the context-dependence of the response predictability. Specifically, we used STRF-based models (Section 3.4.1) as well as general high-order models such as artificial neural networks and support vector regression (Section 3.4.2). Such generic models do have some problems in their interpretations from biological viewpoints, but we exploited these models simply for examining the data fit performance. Here we (re)analyzed the recording data from the previous work (20 cells; Machens et al., 2004) because the recording data we collected for examining context-dependence were not tested with enough varieties of stimuli (Sections 3.2.3–3.2.4), which could cause a bias in the parameter estimation (Paninski, 2003a; Simoncelli et al., 2004; also see Figure 3.10); the total length of distinct stimulus fragments of this and the previous works was typically $\sim40$ sec and $>5$ min, respectively.

#### 3.4.1 Linear-Nonlinear Cascade Models

Spectro-temporal receptive field (STRF; Klein et al., 2000) models can be formulated in general as:

$$\hat{r}(t) = \iiint \text{STRF}(\tau, \omega) \cdot S(t - \tau, \omega) d\tau d\omega,$$  \hspace{1cm} (3.24)
where \( \hat{r}(t) \) is the estimated response, and \( S(t, \omega) \) is the spectrogram (short-time Fourier transform; Cohen, 1995) of the sound pressure waveform \( s(t) \):

\[
S(t, \omega) = 20 \log_{10} \left| \frac{1}{\sqrt{2\pi}} \int e^{-j\omega \tau} s(\tau) h(\tau - t) d\tau \right|. \tag{3.25}
\]

The energy of a window function \( h(t) \) is typically taken to be unity (\( \int |h(t)|^2 dt = 1 \)) to ensure that the energy of the spectrogram \( S(t, \omega) \) is equal to that of the signal \( s(t) \).

For the parameter estimation, we exploited linear regression techniques (for details, see Appendix Section A.2). Specifically, we first discretized time \( t \) and frequency \( \omega \), and re-ordered the indices to simplify Eq.(3.24) into the following form:

\[
\hat{r} = S\beta, \tag{3.26}
\]

where \( \hat{r} \) and \( \beta \) are column vectors of the estimated response and STRF, respectively, and the \( i \)-th row of the matrix \( S \) consists of the \( i \)-th stimulus vector. Ridge regression was then used to estimate the STRF (parameters): \( \beta \) (for details, see Appendix Section A.2.1).

In this study we performed 10-fold cross-validation, i.e., split the data set into training (90%) and validation (10%) data sets, used the training data set to estimate STRF with various power constraint values (\( \lambda \) in Eq.(A.53) on page 154), and chose the one (\( \hat{\beta}_{\text{ridge}} \)) that gave the best model performance on the validation data set (see below Eq.(3.29) in Section 3.4.3). The resulting model performance on the training and validation data set can then be considered as the upper and lower estimates, respectively.

To vary the window length while fixing the model complexity—i.e., the number of free parameters in a model—for a fair comparison of the model performance (Figure 3.11), the time bin sizes were varied in a pseudo-logarithmic scale: \( \Delta_k t = 2^k \) msec for \( k = 2, \ldots, 10 \) from near to distant past. In this study we set the number of bins for \( \Delta_k t \) (45 in total) as \([45, 0, \ldots, 0], [9, 36, \ldots, 0], \ldots, [9, 8, \ldots, 1] \), resulting in models with window lengths of 180, 324,
548, 884, 1364, 2004, 2772, 3540, and 4052 msec, respectively. Frequency discretization was \( \Delta x = 3 \) bins/octave, ranging between 0.4–22 kHz (18 frequency bins).

**Static Nonlinearity**

Static nonlinearities can be given as a nonlinear transformation \( g_{sn} \) that acts on the output of the linear model (Eq.(3.24)) to form a new (better) estimate (see e.g., Simoncelli et al., 2004; Machens et al., 2004):

\[
g_{sn} : \hat{r}(t) \mapsto \tilde{r}_{sn}(t).
\]  

(3.27)

Here we used a scatter plot smoothing technique to identify the best transformation \( \tilde{g}_{sn} \) on the training data set (for details, see Appendix Section A.3.1)—resulting in the upper estimate of the model performance—and then applied it to the validation data set for computing the lower estimate.

**3.4.2 Nonlinear Models**

In addition to classical STRF-based models, the performance of two generic nonlinear models was also examined; artificial neural networks (multi-layer perceptrons) and support vector regression (for details, see Appendix Section A.3). These models typically have high expressive power—in fact, they can provide asymptotically optimal solutions to an arbitrary fitting problem under appropriate conditions (Hecht-Nielsen, 1989; Hertz et al., 1991)—and thus we asked if the model performance was comparable to the upper bound estimate of the response predictability with a given window length. As before, here we used 10-fold cross-validation to find appropriate parameters, although it was not very successful for support vector regression (Table 3.2). Because of a limited data length and the huge computational power (and memory resources) required for the training, we only examined models with window lengths of up to several hundred milliseconds.
Multi-Layer Neural Networks

Here we used classical three-layer (input, hidden, and output) feedforward networks, with log-sigmoid and linear transfer functions for hidden and output units, respectively (see Appendix Section A.3.2). The time discretization was $\Delta t = 4$ msec, and we tested the model with window lengths of 40, 200, and 500 msec for spectrogram-based models with a frequency discretization of $\Delta x = 3$ bins/octave ranging from 0.4 to 22 kHz (18 frequency bins). We also assessed cochleagram-based models (window length: 200 msec) with 100 frequency bins (ranging between 0.4–22 kHz) in the equivalent rectangular bandwidth (ERB) space (Moore and Glasberg, 1983). Here we used a Gammatone filter bank to generate the cochleagrams of input sounds (Auditory Toolbox for MATLAB; Slaney, 1993).

As a preprocessing, we performed principal component analysis (see Appendix Section A.1.1) to whiten the data, and chose those principal components that capture at least 0.1% of total variance (or at least 100 principal components in total) for further processing.

For training the neural networks, we started with 12 hidden units and a single output unit, and trained the network with a conjugate-gradient backpropagation algorithm (MATLAB Neural Network Toolbox; see also Appendix Section A.3.2). Then the least contributing unit was pruned one by one unless the performance on the validation data set got worse, typically ending up with nine or ten hidden units. The training procedures were restarted with 10 random initial conditions, and the one with the best performance (see Section 3.4.3) was chosen as a final result.

Support Vector Regression

As an alternative to the multi-layer perceptrons, we examined the performance of support vector regression ($\nu$-SVR; Schölkopf et al., 2000) with gaussian kernels, which in fact is equivalent to radial basis function networks (see also Appendix Section A.3.3). An open-source library was used for the training (LIBSVM; Chang and Lin, 2001).
As before, the time and frequency discretizations were $\Delta t = 4$ msec and $\Delta x = 3$ bins/octave (ranging from 0.4 to 22 kHz), and the model had a window length of 200 msec. Before training the model, the data were scaled to have zero mean and unit variance.

To find appropriate parameters, a coarse grid search ($1/\lambda$ and $\gamma$ for $10^{-4}, 10^{-2}, 10^0,$ and $10^2$ (four steps in $[-4, 2]$ in log-space); see also Appendix Section A.3.3) was first performed with a random subset of 5000 data for each cell, and then a fine grid search (three steps in $[-1, 1]$ in log-space, centered around the best parameter value from the coarse grid search) with another random subset of 5000 data. Such a two-step grid search was performed for $\nu \in [0.1, 0.5]$ in steps of 0.1, and the best parameter set obtained ($\lambda, \gamma$ and $\nu$) was used for training a $\nu$-SVR with the entire training data set. However, support vector regression tends to overfit a training data set as far as we tested (see Table 3.2) partly because of a limited data length and partly because of inappropriate choices of the parameters (and kernels/models).

3.4.3 Model Performance

The performance of the neural encoding models described above was quantified as the ratio between the estimated response power captured by a model, $\sigma^2_{\text{model}}$, and the stimulus-related (predictable) response power, $\sigma^2$ (Sahani and Linden, 2003; Machens et al., 2004). Note the similarity to the analysis of response predictability in Section 3.3.3.

Assuming additive i.i.d. Gaussian noise: $\varepsilon_j(t) \sim \mathcal{N}[0, \sigma^2_{\text{noise}}]$ over trials (for $j = 1, \ldots, m$) and time $t$, we can express the observed response for the $j$-th trial as: $r_j(t) = \rho(t) + \varepsilon_j(t)$, with the stimulus-related components $\rho(t)$ (equivalent to $\mu(t) + \nu_i(t)$ in Eq.(3.12)), and the total power in the observed response as: $\sigma^2_{\text{total}} = \sigma^2 + \sigma^2_{\text{noise}}$, with the stimulus-related power: $\sigma^2 \equiv \langle \rho^2(t) \rangle_t$ in the limit of large $t$. (As before, we use $\langle \cdot \rangle_t$ to indicate the average over time with subscript $t$ and over trials with subscript $j$. Note that $\langle \rho(t) \rangle_t = 0$, because of the preprocessing of $r_j(t)$ to have zero mean.)
From central limit theorem, the power of the average response over trials can be written as $\langle \langle r_j(t) \rangle_j \rangle_t^2 = \sigma^2 + \sigma^2_{\text{noise}}/m$. Hence, the predictable response power $\sigma^2$ can be estimated as:

$$\hat{\sigma}^2 = \left\langle \frac{m \langle r_j(t) \rangle_j^2 - \langle r_j^2(t) \rangle_j}{m - 1} \right\rangle_t,$$

(3.28)

where we use $\hat{\sigma}^2_{\text{total}} = \langle \langle r_j^2(t) \rangle_j \rangle_t$ and note the similarity to Eqs.(3.18) and (3.19).

The model performance $\sigma^2_{\text{model}}/\sigma^2$ is then given as:

$$\frac{\hat{\sigma}^2_{\text{total}} - \hat{\sigma}^2_{\text{error}}}{\hat{\sigma}^2},$$

(3.29)

where $\hat{\sigma}^2_{\text{error}} = \langle \langle (r_i(t) - \hat{r}_i(t))^2 \rangle_j \rangle_t$ is the model error power. In Eq.(3.23) the average was taken over the population but not over the time, but the quantities in Eqs.(3.23) and (3.29) are equivalent under the assumption of ergodicity.

### 3.5 Results

To characterize the time course and magnitude of context-dependent effects on neural responses in rat primary auditory cortex (area A1), neurons were probed with a variety of spectro-temporally rich stimuli (e.g., animal vocalizations) in sequence (see Section 3.2 and also Figure 3.1). The use of such complex stimuli allowed us to probe a larger fraction of stimulus space than conventional protocols using tones and others (e.g., Theunissen et al., 2000, 2001; Sen et al., 2001; Bar-Yosef et al., 2002; Machens et al., 2004; Garcia-Lazaro et al., 2006).

Our analysis consisted of the following three parts. First, we assessed the overall context-dependence of neurons in A1 using natural sound ensembles (Section 3.5.1). Second, we used synthetic sounds to characterize how context-dependence depended on stimulus properties such as stimulus intensity and modulation rates (Section 3.5.2). Finally, we further quantified the context-dependence from the viewpoint of model construction, i.e., measured the response predictability given all the past stimulus information within an arbitrary window...
length (Section 3.5.3). Note that here we only estimated the upper-bound of the prediction performance, and do not have an actual encoding model yet that can perfectly predict neural responses in A1 from acoustic signals (but see approaches in Section 3.5.4).

### 3.5.1 Context-Dependence

Firing rates in A1 were typically low under our experimental conditions (Figure 2.12 on page 56; spontaneous, 0.25±0.65 Hz; evoked, 0.33±0.69 Hz; median ± interquartile range for 194 cells; see also Wehr and Zador, 2005). We therefore examined subthreshold responses rather than firing rates—because subthreshold responses consist of a continuous variable in time (membrane potential) rather than a sparse binary time series (a train of action potentials), we could obtain good estimates of activity even in the complete absence of spiking activity. Note however that we did not analyze in detail the membrane potential dynamics at “silence” periods because spontaneous activity prevented us from detecting especially the long-lasting events saliently, even though a subset of cells showed significant context-dependent fluctuations soon after the onset of inter-sequence intervals (Figure 3.3).

Figure 3.4 shows a typical example of subthreshold responses to a six-second natural sound stimulus in two different natural sound contexts, i.e., preceded by two different six-second conditioning stimuli. Consistent with previous work (Machens et al., 2004), this neuron showed high trial-to-trial reliability (Figure 3.4A) within each set of trials for which the conditioning stimuli were held fixed—the correlation coefficient of the response traces across trials in a given context was 0.61±0.07 (mean ± standard deviation) for the seven natural sound fragments tested in this cell. The reliability varied within a given neuron as a function of the stimuli tested, and across neurons; the mean correlation coefficient was 0.31±0.09 (mean ± standard deviation; 39 cells) over the population examined with natural sound ensembles.

Changing the conditioning stimulus—i.e., the stimulus context—caused a dramatic change in the response to the probe stimulus (Figure 3.4B, blue vs. red). In this example, the effects of the context on the response lasted more than four seconds. Interestingly, the
context-induced differences could sometimes be intermittent; in this example, the two average response traces showed no difference in the interval two to four seconds after the onset of the probe, but diverged again at around four seconds.

Here we used two measures to quantify the differences in the probe stimulus induced by context. The first (Figure 3.4D) assessed whether the differences in the traces were statistically significant ($p < 0.01$ for at least 5 msec; see Section 3.3.1), whereas the second method (Figure 3.4E) assessed the component of the response power (variance at a given time) dependent on stimulus history (Eqs.(3.18)–(3.21) in Section 3.3.1). The two measures generally agreed quite well, as can be confirmed by noting that when the two traces were significantly different (vertical gray strips), the power was typically high.

Although context-dependent effects could manifest intermittently in a given cell (as in Figure 3.4), across the population these effects showed an orderly monotonic decay (Figure 3.5). Of 305 natural sound probe stimuli tested with different—typically, around five to eight—natural sound contexts in 39 cells, significant effects were observed in 204 cases (66.9%; Figure 3.5A), and about a third (30.3%) of the events occurred longer than one second after the onset of a probe stimulus (Figure 3.5A). This fraction represents a lower bound on the maximum duration of the possible effect in a given cell, since the number of conditioning-probe combinations tested per neuron was quite small, and was not tailored to the properties of the cell. For both measures there was a long decay constant of about one second ($\tau = 0.90$ and 1.04 sec, respectively; see Figure 3.5 for details). Since this decay represents the dynamics of the average response—we did not have enough data per neuron to reliably estimate the time constant for each neuron individually—it could include some responses with longer constants (e.g., Figure 3.4) and some with shorter. We found no systematic relationship between the duration or strength of context-dependence and the recording depth or cortical layers (Figure 3.6; see also Ulanovsky et al., 2004). In conclusion, a neuron’s “memory” of the stimulus history it has experienced can persist for at least one or several seconds.
Figure 3.3: **Context-dependent effects on spontaneous dynamics.** A subset of cells in A1—typically those with low spontaneous activities as in this example—showed context-dependent fluctuations soon after the onset of inter-sequence intervals (time zero indicates the termination of **stimuli** or the onset of **silence** periods). Spikes were clipped by a median filter (window length: 10 msec). In this example, the spontaneous dynamics—examined at a silence period preceded by three different stimuli (**blue**, **red**, and **green**; median over 12 trials each)—diverged several hundred milliseconds after the onset of the silence (**black** line, **p**-value for Kruskal-Wallis test; **gray** vertical strip indicates time points where **p** < 0.01 for ≥ 5 msec). For details, see Section 3.3.1 and also Figure 3.4.
Figure 3.4: **Context-dependence of neural responses lasted for several seconds.** (A) Typical subthreshold responses of a rat A1 neuron to part of a natural sound sequence (spectrogram; time zero indicates the transition from conditioning to probe stimuli) over six repeats (red lines). Spikes were clipped by a median filter (window length: 10 msec). The neuron showed high trial-to-trial reliability (correlation coefficients across trials to this particular probe stimulus, 0.74±0.08; and across trials across all natural sound stimuli examined in this cell, 0.61±0.07; mean ± standard deviation). (B) The median responses to the probe stimulus in two different contexts; red line for the one shown in (A), and blue line for the one in response to the same probe stimulus but preceded by a “silence” period in this example. Note however that silence is not always the “exception” that causes a difference in the following response dynamics. Significant dependence on the stimulus history was observed for longer than four seconds (gray bands; see (D) for details), whereas the responses between two to four seconds after the onset of the probe stimulus were not so much dependent on the conditioning stimuli we tested. (C) The recorded cell in this example was histologically identified as a layer II-III pyramidal neuron (scale-bar; 100 µm). (D) We performed a pointwise statistical (Kruskal-Wallis) test for equal medians between the responses to the probe stimulus in all different contexts (black line; p-values), and the gray bands show the time points where the context-dependence was statistically significant under the criterion: $p < 0.01$ for $\geq 5$ msec. (E) The response power that depends on the context (black line; $\langle \hat{\nu}^2(t) \rangle_i$ from Eqs.(3.18)–(3.20) without population average; see Section 3.3 for details) well represents the magnitude of the context-dependence. The population average of this quantity with and without normalization by the average predictable power is shown in Figure 3.5B.
Figure 3.5: **Context-dependence can last for seconds.** (A) **Significance measure.** *Top:* Each raster shows periods during which the significance measure exceeds threshold (*p* < 0.01 for ≥ 5 msec; see Section 3.3) for a particular probe-stimulus combination in a given neuron (an example is shown as gray bands in Figure 3.4). The rasters are sorted according to the longest-lasting effect, so successive rasters may correspond to different neurons. Significant context-dependence was observed in about two-thirds of probe stimuli (204 out of 305 sets in 39 neurons). *Bottom:* The black curve shows the proportion—or the probability—of observing the significant context dependence, and the orange curve shows the noise floor computed by resampling methods. The former is well fit by the sum of two exponentials (gray; *α₁* = 0.17, *τ₁* = 0.20 sec, *α₂* = 0.09, and *τ₂* = 0.90 sec for Eq.(3.22) with the mean noise floor over time *α* = 2.8×10⁻³). Around 30% of the context-dependent events occurred at ≥ 1 sec from the onset of probe stimuli (brown; cumulative probability). (B) **Fractional power measure.** *Top:* From the same population data, we computed the stimulus-related response power (black, \(\tilde{P}[\mu(t) + \nu_i(t)]\) in Eq.(3.18); gray, its moving average over the data in [0, 2t] at time *t*) and the fraction that depends on stimulus history and its context (thick; \(\tilde{P}[\nu_i(t)]\) in Eq.(3.20)). Note that the latter corresponds well to the significance measure as shown in the bottom panel of (A). See also Figure 3.1A. *Bottom:* The ratio of the context-dependent power to the stimulus-related power represents well the contribution of stimulus history to the response dynamics (black; Eq.(3.21)). The decay size and constant are: *α₁* = 0.49, *τ₁* = 1.04 sec, with *α* = 0 (gray). The absence of a fast time-scale suggests that many of the fast components in (A) resulted from the large fluctuation in the response dynamics induced by the transition from conditioning to probe stimuli, while slow components were by and large left intact.
Figure 3.6: **No clear correlation between recording depth and longest-lasting context-dependence effects.** Each dot represents the longest-lasting effect examined with natural sound conditioning-probe combinations for each recorded cell, as a function of recording depth as determined from micro-manipulator travel (39 cells), and the histograms are shown in *blue* lines. There was no clear relation between the two quantities, consistent with previous work (see also Ulanovsky et al., 2004)
3.5.2 Relation to Acoustic Properties

We initially chose natural sounds because of their spectro-temporally rich melange of acoustic structures, as we expected that the changes in such a full variety of sound properties would allow us to observe as large and long-lasting effects as possible. However, a problem is that we could not tell by comparing two or more conditioning natural sound fragments what exactly made neurons in A1 behave differently in response to the same probe stimulus.

To study what stimulus properties contribute to the context-dependence, we then used synthetic stimuli for which we can control the stimulus parameters well. To examine the frequency change effects among conditioning stimuli for example (Figure 3.7), we first generated a fragment of synthetic sounds—e.g., dynamic moving ripples (Eqs.(3.1)–(3.3) on page 90)—and then up- or down-shifted its spectral components over the maximum range of 4 octaves to make (conditioning) stimulus ensembles where only a particular sound property of interest—frequency in this example (Figure 3.7A)—is different but all the other parameters—e.g., stimulus intensities and modulation patterns—are the same. In this and similar ways, here we examined the effects of the changes in each of the following acoustic properties: amplitude (over a maximum range of 60 dB attenuation), frequency (with a maximum shift of 4 octaves), amplitude-modulation (AM; with a maximum difference in modulation rate and depth of 20 Hz and 3-fold, respectively), frequency-modulation (FM; with the same maximum difference range as for AM), and higher-order spectro-temporal structures (by comparing natural sounds and corresponding modulated pink noise; Eqs.(3.6)–(3.9) on page 93); for details, see Sections 3.2.3–3.2.4, Table 3.1, and Figure 3.1B.

Figures 3.8 and 3.9 show the population results and the summary of the analysis, respectively. When we varied either intensities or frequencies (i.e., lower-order sound properties) in conditioning stimuli, the context-dependent effects were observed in 77 out of 95 cases (81.1%; 31 cells) and in 73/110 (64.6%; 35 cells), respectively (Figure 3.8A). The effects were almost as large and long as those caused by the difference between natural sound contexts (Figure 3.9B)—i.e., by the changes in any possible acoustic feature. However, the slow
components in the significance measure were less apparent in the context-dependence caused by the changes in the lower-order sound properties (Figure 3.9A). Single exponentials with the time-scale of several hundred milliseconds explained the effects in these two cases well (Figure 3.8A).

We then examined the effects of AM and FM changes in conditioning stimuli using modulated harmonic tones (Eqs.(3.4) and (3.5) on page 92), and also the changes in even higher-order acoustic properties such as complex interactions between spectro-temporal components by comparing the differences between modulated pink noise and its corresponding natural sounds (see Methods Section 3.2.4 and Figure 3.1B). The context-dependent effects were observed in 62 out of 96 cases (64.6%; 27 cells), in 47/82 (57.3%; 25 cells), and in 59/137 (43.1%; 27 cells), respectively, but the effects were substantially smaller and shorter than the effects caused by the changes in natural sound contexts in both measures (Figure 3.9); i.e., higher-order sound properties contributed to the context-dependence mainly on shorter time-scales (on the order of 100 msec). We thus conclude from Figures 3.8 and 3.9 that neural behaviors in A1 are more susceptible to changes in lower-order sound properties such as overall intensities and frequencies than to changes in higher-order properties including AM and FM.

### 3.5.3 Relation to Response Predictability

Population data showed that a time-scale relevant for A1 neurons was as slow as on the order of seconds (Figure 3.5). But, how important would past events be for predicting neural responses? To address this question, we estimated the best possible performance for the response prediction when only a finite length of stimulus history is available.

We first assumed that the observed responses to a given probe stimulus consisted of additive noise and stimulus-related components that can be further decomposed into context-dependent and independent fractions (Eq.(3.12) on page 97). We could then estimate the stimulus-related—or the *predictable*—fraction as the trial average in each context, whereas
the context-independent fraction as the mean response over all contexts (for details, see Section 3.3.3). The latter gives the best estimate of the responses to a probe stimulus without any knowledge on the conditioning stimuli, whereas the former corresponds to the very best response estimate that no model could outperform under the additive noise assumption. Because the prediction performance can be assessed by examining how much of the predictable response power a given model can capture (Linden et al., 2003; Sahani and Linden, 2003; Machens et al., 2004), we could then estimate the upper-bound of the prediction performance (given a finite window length) as the ratio of the response powers between the context-independent and the predictable fractions (Eq.(3.23) on page 101). Note the relation to the fractional response power measure for context-dependence, where we computed the ratio of the response powers between the context-dependent and stimulus-related fractions (Figures 3.5B and 3.8B; Eq.(3.21) on page 99).

The estimated upper-bound (blue curve in Figure 3.11) then indicates that no model could capture more than a half \(1 - |\alpha_1| = 0.51\) of the response power given the window length of \(<100\) msec, and that the stimulus history for at least several seconds should be considered to predict neural responses in A1 fully enough (\(\tau = 1.04\) sec). Such long-lasting effects would not be simply explained—or “expected”—by some “trivial” internal processes of neural dynamics because the noise correlation (Eq.(3.10) on page 94) had a sharp peak with a rapid decay (within a few hundred milliseconds; Figure 3.2D). Stimulus-dependent bottom-up mechanisms would thus be involved in the modulation of neural responses in A1 over seconds (see also Section 3.6.3).
Figure 3.7: **Context-dependence caused by frequency changes.** (A) Typical median sub-threshold responses of a rat A1 neuron to part of a sound sequence, where a natural sound stimulus was probed with two different synthetic conditioning stimuli with different bandwidths (*blue* line for a context with 0.625–2.5 kHz (*top spectrogram*), and *red* line for another context with 10–40 kHz (*bottom spectrogram*); time zero indicates the transition from conditioning to probe stimuli). Spikes were clipped by a median filter (window length: 10 msec). Although the responses were less reliable in this example (correlation coefficients across trials to this particular probe stimulus, 0.28±0.06; and across trials across all stimuli examined in this cell, 0.21±0.11; mean ± standard deviation) than those in Figure 3.4, significant context-dependence was observed for longer than four seconds as well (*gray* bands; see (B) for details). (B) *Black* line represents *p*-values and *gray* bands show the time points where the context-dependence was statistically significant under the criterion: *p* < 0.01 for ≥ 5 msec (in the same format as Figure 3.4D). (C) The response power that depends on the context (*black* line; in the same format as Figure 3.4E). The population average of this quantity normalized by the average predictable power is shown in Figure 3.8B (*frequency*).
Figure 3.8: **Relation between context-dependence and sound properties.** Using synthetic sounds, we examined the effects of the changes in a particular sound property of interest on the responses to following probe stimuli. From top to bottom panels, shown are the population analysis for the changes in amplitude (95 sets in 31 cells; *red*), frequency (110 sets in 35 cells; *blue*), AM rates and depths (96 sets in 27 cells; *magenta*), FM rates and depths (82 sets in 25 cells; *cyan*), and higher-order properties (137 sets in 27 cells; *green*). The *black* curves show the exponential curve fit as in Eq.(3.22) with the constant $\alpha$ being the mean noise floor over time and zero for (A) and (B), respectively. (A) **Significance measure.** Shown is the population analysis based on the probability of observing context-dependence, in the same format as Figure 3.5A but overlaid. The *orange* curves show the noise floor, and from top to bottom panels, the parameters for the exponential model (*black*) were $[\alpha_1, \tau_1, \alpha] = [0.23, 0.25, 2.8 \times 10^{-3}]$, $[0.19, 0.27, 2.3 \times 10^{-3}]$, $[0.13, 0.17, 2.3 \times 10^{-3}]$, $[0.07, 0.15, 2.1 \times 10^{-3}]$, and $[0.04, 0.13, 3.6 \times 10^{-3}]$, respectively. (B) **Fractional power measure.** Shown is the population analysis, indicating the contribution of context-dependence to the response dynamics, in the same format as Figure 3.5B. From top to bottom panels, the parameters for the exponential curves (*black*; all with $\alpha = 0$) were $[\alpha_1, \tau_1] = [0.57, 0.66]$, $[0.54, 0.97]$, $[0.86, 0.11]$, $[0.64, 0.15]$, and $[0.52, 0.20]$, respectively.
Figure 3.9: **Population analysis summary.** Parameters for the exponential curves (Eq.(3.22); top, $\sum_k \alpha_k$; bottom, $\tau_k$) in Figures 3.5 and 3.8 are shown with 95% confidence intervals computed by resampling methods (see Methods Section 3.3.1). (A) **Significance measure.** From left to right, the intervals are; $\sum_k \alpha_k \in [0.23, 0.28], [0.21, 0.25], [0.17, 0.22], [0.11, 0.14], [0.06, 0.09]$, and $[0.03, 0.06]$, respectively; and $\tau_k \in \{[0.17, 0.24] and [0.74, 1.02]\}, [0.23, 0.28], [0.24, 0.30], [0.16, 0.18], [0.13, 0.16]$, and $[0.09, 0.23]$, respectively. (B) **Fractional power measure.** From left to right, the intervals are; $\sum_k \alpha_k \in [0.46, 0.52], [0.52, 0.63], [0.51, 0.57], [0.82, 0.90], [0.58, 0.70]$, and $[0.45, 0.61]$, respectively; and $\tau_k \in [0.91, 1.18], [0.54, 0.81], [0.85, 1.09], [0.10, 0.12], [0.12, 0.18]$, and $[0.16, 0.25]$, respectively. Abbreviations; Nat, Natural Sounds; Amp, Amplitude; Freq, Frequency; AM, amplitude-modulation; FM, frequency-modulation; High, higher-order property.
3.5.4 Model Performance

This long time-scale explains in part why classical linear encoding (spectro-temporal receptive field; STRF) models with a limited window length, typically a few hundred milliseconds, have failed. The performance of STRF-based models was in general unsatisfactory, consistent with previous work (~20%; Sahani and Linden, 2003; Machens et al., 2004). The performance did not improve significantly, however, just by extending the window length (up to ~4 sec; red bands in Figure 3.11 for mean lower and upper bound estimates) even with static nonlinearities (green lines), and we could not identify distinct structures or “features” in the STRFs at longer than several hundred milliseconds. This suggests a role of A1 neurons in more than detecting stimulus features (Nelken et al., 2003; Nelken, 2004), but it remains to be addressed how neurons in A1 exploit stimulus history and its context on such a long time-scale.

We also examined the performance of nonlinear generic models—i.e., artificial neural networks and support vector regression—and asked if they could reach the upper bound estimated from our experimental data. Because of a limited data length, we examined models with a rather short window length (up to 500 msec; for details, see Section 3.4.2), but even a generic nonlinear model did not reach the upper bound (black lines in Figure 3.11; see also Table 3.2 for models with 200 msec window). One reason for this failure would be an inappropriate choice of initial transformation from time domain to time-frequency domain (and preprocessing). Inspired by the auditory system, here we examined two types of initial transforms—log-frequency spectrograms and cochleagrams in dB scale (Slaney, 1993)—but neither worked well as inputs to artificial neural networks (see also Gill et al., 2006). Another possible mode of the failure would be a rather coarse discretization resolution over time and frequencies (4 msec bin and 3 bins/octaves, respectively), and thus relevant features for neurons might have been lost. Finer discretization however requires a larger amount of data because of an increase in the number of free parameters. Then it would be needed to either employ state-of-the-art methods or devise some judicious nonlinear transformations to extract “relevant features,” and interpret them from biological viewpoints (see also Section 3.6.4).
Figure 3.10: **Poor prediction performance of classical linear-nonlinear cascade models.**

(A) An example of the STRF (200 msec window with $\Delta t = 4$ msec; 18 frequency bins ranging from 0.4 to 22 kHz with $\Delta x = 3$ bins/octave) for an auditory cortical neuron (for details on the estimation, see Section 3.4.1). We can clearly see that the STRF has a broadband excitatory region following a relatively narrow-band inhibitory region. (B) Response estimation example. Estimated responses on a validation data set (STRF model as shown in (A), red; Linear-Nonlinear (LN) model, green) capture fairly well the occurrence of depolarization and/or hyperpolarization, but the size is far smaller than the original response (blue). The mean lower and upper model performances in this example were 0.07 and 0.17 for the STRF model, respectively, and 0.10 and 0.22 for the LN model, respectively. (C) Relation between model performance and data length. We randomly chose a subset of data to limit the data length (15, 30, 60, and 120 sec; 20 cells from Machens et al., 2004), and assessed the performance of the STRF-based model (200 msec window with the bin size of 4 msec, frequency range of 0.4–22 kHz with discretization resolution 3 bins/octave; STRF, red band for Eq.(3.24); LN, green lines for Eq.(3.27)) computed as in Eq.(3.29). Crosses and open circles respectively show the average model performance on the validation and training data sets, corresponding to the lower and upper estimates of the performance. Model performance does not depend much on the data length, suggesting that the regularization (ridge regression; see Appendix Section A.2.1) worked well, but the optimal regularization strength (the optimal value of $\lambda$ in Eq.(A.53) on page 154) depends on data length in general. The symbols “[ ]” in red and “]” in green at $\sim 40$ sec respectively represent the mean performance of STRF and LN models on the data recorded in this study (lower and upper “hems” representing lower and upper estimates, respectively; 23 cells examined with natural sound ensembles), showing that the two data sets from this work and the previous one are comparable. This result justifies the comparison between the upper bound performance (estimated from our data; blue curve) and the model performance (data from Machens et al., 2004; red, green, and black curves) in Figure 3.11.
Figure 3.11: **Context-dependence of response predictability.** Using responses to natural sound probe stimuli in different natural sound contexts (305 probe stimuli in 39 cells), we computed the ratio between the context-independent fraction of the response power and the stimulus-related response power in A1 neurons (blue; Eq.(3.23) in Section 3.3.3). The latter is given by the mean response over trials in each context—or the best response estimate under additive noise assumption—and the former is given by the mean over all contexts—or the best estimate of the responses to a probe stimulus without any knowledge on the conditioning stimulus. Therefore, the ratio represents the upper-bound of the response prediction performance for a given window length, which asymptotically approached the true upper limit (black dotted line; model performance = 1) by extending the window length—or available stimulus history—on the time-scale of seconds (cyan; $\alpha_1 = -0.49$ and $\tau_1 = 1.04$ sec for exponential curve fit as in Eq.(3.22) with $\alpha = 1$). In contrast, the performance (Eq.(3.29); 20 cells from Machens et al., 2004) of linear encoding models (STRF, red; Eq.(3.24)) was low for any window length up to $\sim$4 sec even with static nonlinearities (LN, green; Eq.(3.27)), and no model could reach the estimated upper bound (ANN, black; see also Section 3.4.2) as far as we tested. Crosses and open circles respectively show the average model performance on the validation and training data sets, corresponding to the lower and upper estimates of the performance. Here we varied the bin sizes for changing the window length of the STRF models while fixing the model complexity (for details, see Section 3.4.1). See also Figure 3.10.
<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear (STRF)</td>
<td>0.17±0.13</td>
<td>0.11±0.09</td>
</tr>
<tr>
<td>Linear-Nonlinear</td>
<td>0.23±0.15</td>
<td>0.14±0.14</td>
</tr>
<tr>
<td>ANN (spectrogram)</td>
<td>0.29±0.14</td>
<td>0.25±0.12</td>
</tr>
<tr>
<td>ANN (cochleagram)</td>
<td>0.27±0.09</td>
<td>0.15±0.10</td>
</tr>
<tr>
<td>ν-SVR</td>
<td>0.32±0.16</td>
<td>0.07±0.12</td>
</tr>
</tbody>
</table>

Table 3.2: **Performance of generic nonlinear encoding models.** We examined classical STRF-based models as well as generic nonlinear models having the window length of 200 msec. The inputs to the models were log-frequency spectrograms (except for one case; *ANN with cochleagram*) with rather coarse time and frequency discretizations (for details, see Section 3.4), and the outputs were subthreshold dynamics of auditory cortical neurons from previous works (20 cells; Machens et al., 2004). The model performance (Eq.(3.29)) on the *training* and *validation* data sets represents the upper and lower estimates, respectively (see also Figure 3.11). No models reached the upper-bound estimate (≈0.6; Figure 3.11). Acronyms: STRF, spectro-temporal receptive fields; ANN, artificial neural networks; SVR, support vector regression.
3.6 Discussion

We have examined how neural responses in the rat primary auditory cortex (area A1) and their predictability depend on stimulus history and its context. Whole-cell recordings performed in vivo in the anesthetized rat showed that the context-dependence in A1 sometimes lasted as long as four seconds in some neurons (Figures 3.4 and 3.5); and that the dependence was more susceptible to the changes in lower-order conditioning sound properties—such as overall intensities and frequencies—than to the changes in higher-order properties—such as AM and FM rates and depths—even though the latter had as distinct effects as did the former (Figures 3.8 and 3.9). We also identified that the faithful prediction of response dynamics in A1 requires nonlinear integrations of stimulus history and its context over seconds (Figure 3.11). These results suggest complex mechanisms for stimulus-dependent bottom-up modulation of neural responses in A1, with roles not only in stimulus feature detections but also in stream segregation and other auditory tasks where longer temporal processings would presumably be involved.

3.6.1 Context-Dependence

Our main goal in this study is to characterize the “memory” length of auditory cortical neurons for developing a simple—yet plausible—encoding model well-tailored to the observed properties of response dynamics. Here we used two measures for analyzing context-dependence because we wanted to examine not just whether the effects of past events were statistically significant—i.e., our ability to induce and detect context-dependence reliably—but also fractional response power depending on stimulus history—i.e., the contribution of the context-dependence effects to response dynamics in A1—as prior knowledge for model construction (for details, see Section 3.3). Both measures identified a slow time-scale in the population data (on the order of seconds; Figure 3.5), while the former also had faster components (on the order of several hundred milliseconds). This is because the neural responses fluctuated a lot...
soon after the transition from conditioning to probe stimuli (Figure 3.5B, top panel), presumably as a manifestation of context-dependence induced by our stimuli. The absence of the fast components in the fractional power measure then suggests that the slow time-scale is most relevant from a modeling perspective (see also Figure 3.11), and in fact the time-scale agrees well with that of “echoic memory” (typically between 0.5 and 2 seconds; Neisser, 1967; Glucksberg and Cowen, 1970; Darwin et al., 1972; Triesman and Rostron, 1972; Rostron, 1974; Kubovy and Howard, 1976). Interestingly, in contrast to the monotonic decays in the population data (Figure 3.5), a subset of cells showed nonmonotonic effects that cannot be explained simply by exponential processes (e.g., Figure 3.4). This suggests that the mode of context-dependence could differ from cell to cell. However, we did not have enough data to characterize the relevant time-scales of individual cells, and we conclude that the memory of A1 neurons typically lasts for at least one or several seconds.

Our experimental design differs from previous work in the following three respects. First, most studies on context-dependence thus far used rather simple stimuli, such as pure or AM/FM tones and clicks (e.g., Abeles and Goldstein, 1972; Hocherman and Gilat, 1981; Phillips, 1985; Calford and Semple, 1995; Brosch and Schreiner, 1997, 2000; Ulanovsky et al., 2003, 2004; Bartlett and Wang, 2005; Wehr and Zador, 2005), whereas we used rather complex natural and synthetic stimuli because A1 neurons are extremely sensitive to small changes in stimuli (Bar-Yosef et al., 2002). Second, we sequentially presented such complex stimuli in a randomly interleaved manner instead of just changing inter-stimulus intervals as in conventional two-tone stimulus paradigms. Finally, we examined the context-dependence effects at the subthreshold level to achieve a high temporal resolution despite low firing rates in A1 (Wehr and Zador, 2005; Hromádka, 2007). This novel approach allowed us to detect long-lasting effects that have not been observed in previous studies, and the relation between the time-scales and the stimulus properties we identified (Figures 3.8 and 3.9) could be used as a hint to build a model characterizing the relationship between input acoustic signals and output neural responses. From this modeling viewpoint, it is also advantageous to work on the sub-
threshold levels; such encoding models target the inputs to the cells and thus can be simpler to some extent because we do not have to consider nonlinear spike generating mechanisms of the target cell itself (see also Section 3.6.4 below).

3.6.2 Response Predictability

Context-dependence is highly related to response predictability. In fact, the performance of neural encoding models was bounded by the window length, and asymptotically approached the upper limit on the time-scale of seconds ($\tau = 1.04$ sec; Figure 3.11). This long time-scale then explains in part why classical linear encoding models have failed (Sahani and Linden, 2003; Machens et al., 2004). Such models typically have a window length of several hundred milliseconds, but not more than around two-thirds of the stimulus-related predictable response power can be captured on such a short time-scale.

In this study we extended the window length for the linear models (while fixing the model complexity) and incorporated static nonlinearities as well, but the performance did not improve much (Figure 3.11) and we failed to identify stimulus features in STRFs longer than several hundred milliseconds. This casts a doubt on considering neurons in A1 simply as stimulus feature detectors (or, spectro-temporal “edge” detectors; Fishbach et al., 2001, 2003). Many acoustic signal processing problems such as stream segregation in fact require dynamic/nonlinear integration of stimulus history and its context over seconds (Bregman, 1990). The context-dependence reported in this study might then be a neural substrate of such processings, and thus not only system identification but also functional viewpoints might be needed to understand and model the response dynamics in A1.

The findings on the relation between the context-dependence and sound properties (Figures 3.8 and 3.9) imply that such integrations would be somewhat less complex for longer time-scales—i.e., incorporating mainly the lower-order sound properties. However, it remains to be addressed how neurons exploit the sound components on longer time-scales, and it is a
challenge for future works how we could build a plausible “forward” encoding model for A1 neurons.

3.6.3 Possible Mechanisms

Even though our main goal here is not to discover the underlying mechanisms of context-dependence but to characterize the time-course of auditory sensory memory in single-neuron response dynamics, it is always worth speculating the underlying mechanisms.

In contrast to much previous work (e.g., Abeles and Goldstein, 1972; Hocherman and Gilat, 1981; Phillips, 1985; Brosch and Schreiner, 1997, 2000; Bar-Yosef et al., 2002; Ulanovsky et al., 2003, 2004; Bartlett and Wang, 2005), here we examined context-dependence at the subthreshold level and observed significant effects on the time-scale of seconds (Figure 3.5). Although we cannot exclude the possibility that spike generation mechanisms were involved (Carandini and Ferster, 1997; Sanchez-Vives et al., 2000a,b; Gollisch and Herz, 2004), we would then expect that the observed dependence was caused by some cortical network effects (e.g., sensory memory) and by the mechanisms operating at the inputs to the neuron, such as synaptic depression or facilitation (Abbott et al., 1997; Tsodyks and Markram, 1997; Wehr and Zador, 2003; Zhang et al., 2003; Tan et al., 2004; Wehr and Zador, 2005). Such mechanisms could affect different parts of the dendrites, resulting in stimulus-specific effects presumably responsible for the observed nonmonotonicity in some cells (e.g., Figure 3.4; note also stimulus-specific adaptations reported by Ulanovsky et al., 2003, 2004).

We could also speculate that the observed effects would be mainly governed by stimulus-dependent “bottom-up” modulations—because we recorded from anesthetized rats, the semantics of presented sounds for rat behaviors or the state of the animal such as stress levels or attentions would have no effect in this study—and mainly cortical in origin because subcortical auditory neurons have typically much shorter time-scales than cortical neurons (Creutzfeldt et al., 1980; Miller et al., 2002). Then one interesting property of A1 potentially in close relation to the context-dependence is that neurons in A1 typically have short latencies
and precise timing of spiking activities for shorter time-scales (~10 msec; Phillips and Hall, 1990; DeWeese et al., 2003; Elhilali et al., 2004), while for rather longer time-scales (~1 sec) auditory cortical neurons are often considered to be “labile” because they tend to adapt quickly to the statistics of stimulus ensembles, or to consecutive presentations of the same stimulus (Condon and Weinberger, 1991; Malone et al., 2002; Ulanovsky et al., 2003, 2004). Small differences in stimulus history would then be accumulated together with response history, and later result in a large difference in the neural behavior where long-lasting effects would emerge in a chain reaction manner. However, such avalanche-like mechanisms could not explain well the nonmonotonic context-dependence effects we sometimes observed, and it remains to be addressed how and where the memory of stimulus history and its context is stored and retrieved.

### 3.6.4 Plausible Encoding Model

Successful curve fitting generally requires appropriate choices of the following three frameworks that allow us to formulate appropriate optimization problems (in the framework of maximum \textit{a posteriori} inference; see Appendix Section A.2.2): the model class, the loss/cost functions (noise distribution), and the regularizer (the prior). Here we discuss possible ways to develop better—and more plausible—encoding models with respect to these three points.

Two distinct classes of encoding models were examined in this chapter—i.e., classical STRF-based models and generic nonlinear models—but neither of them worked well as shown in Figure 3.11 and Table 3.2. One way to improve the models would be to incorporate observed properties of response dynamics, such as the slow adaptations (τ~1 sec) identified here. For example, we could employ some adaptive models (with a “forgetting factor” in Eq.(A.61) in Appendix Section A.2.3 (page 157) as: \( \xi \sim \exp[-\Delta t/\tau] \); see also Stanley, 2002). Another possible way would be to use recurrent models—such as Kalman filters (Kalman, 1960a,b; Roweis and Ghahramani, 1999), hidden Markov models (Baum and Petrie, 1966; Baum et al., 1970; Rabiner, 1989), and integrate-and-fire-like models (e.g., Paninski et al., 2004, 2005)—to exploit all the stimulus (and response) history of a target cell. Considering that even generic
nonlinear models failed to reach the upper bound estimate of the model performance, however, we would rather attribute the failure to inappropriate choices of the other two frameworks (see below): the loss/cost functions and the regularizer.

In this chapter, all models were trained under the assumption of additive i.i.d. Gaussian noise \((\varepsilon_{ij}(t) \sim \mathcal{N}[0, \sigma_{\text{noise}}^2])\), from Eq.(3.12) on page 97). Our data by and large support this assumption (Figure 3.2 on page 95), but we could use a different noise model—e.g., Laplacian or Cauchy distributions—because the noise distribution of our data was slightly more kurtotic than the Gaussian distribution (black vs. gray lines in Figure 3.2C, respectively). It might also be interesting to have a different perspective to fit a model to data, e.g., using the maximum informative dimension technique (Sharpee et al., 2004, 2006).

Regularization is critical in solving many problems in science because they are often ill-posed and/or data-limited. Here we imposed a power constraint (or equivalently a Gaussian prior; see Appendix Section A.2.1) on the STRF parameters, but it is not clear if this assumption would be the best for solving the neural encoding problem. One interesting observation is that the distribution of membrane potential is not symmetric but skewed in the positive (depolarization) direction (Figure 3.2A). It would then be interesting to devise or search for methods to exploit this “prior knowledge” for fitting a given class of models.

An alternative approach would be to transform the output responses into appropriate formats, in such a way that the transformed outputs have somewhat simpler—or “more linear”—relations to the input spectrograms. For example, because subthreshold dynamics consist of linear combinations of postsynaptic potentials that have stereotypical shapes in A1 (or “bumps;” Deweese and Zador, 2006)—be they stimulus-evoked or spontaneous—it would be reasonable to identify a template of the bumps and predict the position and scales of the template instead of predicting the membrane potential dynamics themselves. Such templates might be identified by feature extraction methods, and appropriate decompositions—or “bumpgrams”—might be obtained by, e.g., matching pursuit methods (Mallat and Zhang, 1993).
Chapter 4

Coda

The previous two chapters described the main results of this thesis on the auditory system characterization by taking top-down and bottom-up approaches, respectively. Here I will present a general discussion on how we could approach the sensory coding problems. Following a brief summary of the approaches and the results of this thesis, Section 4.1 will discuss further questions and future challenges in close relation to the topics of this thesis. As a concluding remark, Section 4.2 will brainstorm and argue what needs to be kept in mind for conducting theoretical studies on the brain.

4.1 Conclusions and Future Challenges

Sensory signal processing in the brain can be conceptually divided into several stages (see also Section 1.2); sensors at the very periphery, preprocessing at the subcortical levels, signal representations and perceptions at the cortical level, and output motor actions. This dissertation mainly targeted the representation—i.e., the cortical—level, and took two complementary approaches to understand the underlying logics of the auditory signal processing. On the one hand, in Chapter 2 we employed a top-down theoretical approach and considered principles that govern the activities of populations of neurons. Here we focused on the anatomical characteris-
tics that the auditory cortex has by far more neurons than the periphery does, and demonstrated that sparse overcomplete representations could contribute to difficult computations such as to solve the monaural cocktail party problem (Sections 2.2–2.3). We also derived several predictions on neural behaviors to provide ways to experimentally test the theory (Section 2.4). On the other hand, in Chapter 3 we took a bottom-up experimental approach and aimed at single-neuron dynamics. A major goal here was to study response properties of auditory cortex so as to tune up an encoding model, and we specifically assessed the time-course of neurons’ “memory” to limit the window length—and consequently the number of free parameters—of a model. We found that a slow stimulus adaptation over seconds would be the most relevant for response predictions (Section 3.5). However, the model performance did not improve by simply extending a window length, and none of the models we examined could even reach the estimated upper bound of the response prediction performance based on the experimental data.

Two comments are in order about the fundamental caveats we should be aware of. First, I generally assumed serial or sequential stages for the sensory signal processing, but the brain would most likely be a parallel processor. We could overlook this point as long as we focus only on the underlying principles—or the top two levels of the understandings in the Marr’s framework (Marr, 1982; see also Section 1.2). For example, the idea of sparse overcomplete representations was exploited in Chapter 2 as a possible principle of computation in the brain, but one would not be surprised even if the discussed implementations—i.e., feature extractions by non-negative matrix factorization and sparse representations by $L_1$-norm minimization—do not correspond to the actual algorithms employed in the nervous system. To fully understand the brain, however, we cannot avoid considering the physical instantiations—or the bottom level in the Marr’s classification (Marr, 1982)—and this would be one of the most important problems from biological viewpoints (see also Section 4.2 below). In the computer science community, parallel or distributed computing is now an area of extensive research because of its various advantages in cost and time savings and because of its ability to solve larger problems. Although it is less studied in the context of neuroscience so far, I believe that insights gained
from the one can be applied to the other (and vice versa), and that interdisciplinary research would open ways to the “parallelism.”

The second caveat is the processing with respect to “time.” From modeling viewpoints, we in general think of auditory signal processing as an analogue of visual signal processing (Massaro, 1972; Shamma, 2001)—i.e., frame-by-frame “auditory image” processing—and the aspects of continuous and/or sequential processing over time have been less studied so far despite a growing awareness of adaptive and/or dynamic models. It is not yet clear if we could treat “time” simply as an extra dimension in much the same way as “space” processing, or if we need a distinct theory for understanding temporal processing (see also Schwartz et al., 2007; Section 1.2). We should also be aware that changes in response properties could range over various time-scales depending on the properties of input sensory signals (or, individual sensory experiences; see e.g., Fritz et al., 2003; Ulanovsky et al., 2004; Weinberger, 2004)—from a rapid change over tens of milliseconds as in forward masking (Bregman, 1990) to a slow adaptation over seconds as shown in Chapter 3, and to the developmental and plastic changes (including cell death) in much longer time-scales—and that such changes can occur in both reversible and irreversible manners. While the former could be explained by changing parameters in a given model framework (e.g., by a gain control), the latter changes might then require different model frameworks—or “concepts”—to make sense of them. Because the system identification approach\(^1\) typically expects that a target system should be “unchanged,” it would thus be critical to look at appropriate time-scales for building plausible models that work well (at least within the given time-scales). It remains to be addressed, however, how many different classes of functions or models are required to fully account for neural behaviors, and if they can in fact be unified by introducing some meta-parameters.

\(^1\)Note however that multi-resolution multi-model analysis is now attracting much attention in various fields of science.
4.2 General Discussion

How useful would theory be for understanding biology? A theory (or model) can be made in data- and/or hypothesis-driven manners, but the main contributions of theory would be to provide comprehensive understandings of phenomena and predict outcomes for novel situations. It is then especially important to examine the correspondence between theory and biology by experiments, because biology is the study to identify what exactly nature selected among many possibilities that could all work well for achieving organisms’ goals. Therefore, biologically reasonable interpretations are indispensable; we could then design and perform (new) experiments to verify—or falsify—the models accordingly, and also seek theoretically interesting phenomena to improve the models and faithfully mimic mechanisms underlying functions or computations in the brain.

To see how important interpretations would be for a theory in biology, below I discuss some more conceptual issues on two prevailing ideas in neuroscience; feature extraction and optimization. First, a goal of sensory signal processing is often considered to be extracting features from received signals (Victor, 2005). Here I will not deny this idea but argue that it can be achieved by many different approaches. A straightforward method would be to search for structures of interest from backgrounds as “active” processes. Alternatively, we could also extract features by ignoring irrelevancies or removing uninteresting objects as a “passive” approach. From theoretical viewpoints, it would not matter whether we take active or passive strategies as long as the mathematical formulations are the same—e.g., a band-pass filter is nothing but a band-pass filter, and it does not matter whether it passes a particular bandwidth or attenuates the surroundings. However, this interpretation does matter from biological viewpoints because nature has presumably followed some particular strategies but not the others under certain selection pressures it has faced in evolution, leading to the nervous system organizations as we now have in this world.
The second example is the idea of optimization, e.g., with respect to “efficient” coding or redundancy reduction² (Attneave, 1954; Barlow, 1961, 2001). Again, I am not criticizing the optimization approach itself, but bringing out its usage in the context of biology. The point is that we do not know if the brain has been optimized, or is still in the process of optimization (from both phylogenetic and ontogenetic perspectives). Thus it does not mean a lot to simply measure some “neural correlates,” e.g., how much redundancy in “bits” there is in neural spiking patterns—it is even unclear if “bits” are what neurons care about—but we should compare it across species in different phylogenetic trees (ideally across those in different stages in evolution) and also across different developmental stages (preferably at different processing levels in the brain as well) so as to discuss the meaning of the optimization principle from biological viewpoints and justify or falsify it. It would also be interesting to think about activities/functions in suboptimal regimes, especially because human behaviors—even those of the “highest” animals—do not seem optimal or logical in many situations. One biologically interesting question would then be to examine when the optimality breaks down—e.g., by exploiting speed-accuracy trade-off or by lesion study approaches—and ask if higher animals are somehow “more optimized” than lower animals.

Finally, I would dare like to pose questions, some of which might even be philosophical and unanswerable by contemporary science (partly because of the lack of a solid—or mathematical—definition, and partly because they are “pseudoscientific” questions by themselves; Popper, 1934).

- To what extent are experimental animals (e.g., rats) similar to humans? Do we work in the same way as they do? How much we could know about us humans by studying experimental animals?

²It should be mentioned that “redundancy” in Barlow’s sense is defined as $1 - I/I_{\text{max}}$, where $I$ and $I_{\text{max}}$ are the observed Shannon information and the channel capacity, respectively (or, defined as: $1 - H/H_{\text{max}}$, where $H$ and $H_{\text{max}}$ are the observed and maximum entropies, respectively; Shannon, 1948) but not the difference between the sum of the information from individual cells and the total information (as is often defined in many literatures; see e.g., Schneidman et al., 2003; Nirenberg and Latham, 2003; Latham and Nirenberg, 2005; Averbeck et al., 2006), even though the point of the discussion here is not what “redundancy” means but how we could interpret a theoretically-interesting quantity from biological viewpoints.
• How big are individual differences between experimental animals, and between us? Could all contributions of nurture be explained as the results of “learning processes” once we understand everything about nature?

• Where does “conscious mind” arise from? Is there any neural correlate of consciousness in humans, and in animals (e.g., apes) as well? How many neurons (computational units) are required to have a conscious mind? Can we implement it in electronic devices?

• To what extent is the brain deterministic, and to what extent is it probabilistic/stochastic? How well can we describe the “the theory of the brain” at the level of classical mechanics, and how about at the level of quantum mechanics? Is everything—even the fact that I wrote this thesis and you are reading it—predetermined, or do we have “free will?”

• Can we build—in theory and even in practice—an “artificial brain?”

It might be nonsense as a biological science to ask any question unless there is a way to test it by (Gedanken) experiments. However, I would like to ask myself such questions as well because “to think is to live” (vivere est cogitare; Marcus Tullius Cicero) and because I am a “thinking reed” (roseau pensant; Blaise Pascal).
Appendix A

Technical Notes

This appendix overviews technical details and algorithms in close relation to the data analyses and simulations in this thesis. All these methods are widely used in various scientific fields such as statistical learning and psychophysics as well as neuroscience (for reviews, see Cover and Thomas, 1991; Hertz et al., 1991; Rieke et al., 1997; Dayan and Abbott, 2001; Duda et al., 2001; Hastie et al., 2001; Nelles, 2001; MacKay, 2003). Section A.1 describes matrix decomposition methods and their characteristics for extracting features in given data sets and for simplifying them in the sense of data compression or dimension reduction under arbitrary criteria of interest. In Section A.2, I explain linear regression and regularization techniques, which are closely related to matrix decomposition methods. Section A.3 then reviews the basics of artificial neural networks (multilayer perceptrons) and support vector regression, in order to provide a unifying view of (nonlinear) data fitting techniques often used in neuroscience. Finally, I give some backgrounds on information theory in Section A.4, and show a little survey on entropy estimation by exploiting file compression methods.
A.1 Matrix Decomposition

Let an $N \times M$ matrix $X$ be an original data matrix, each column of which contains the $N$-dimensional data or channels for one of $M$ observations or samples (typically, $M \geq N$). The matrix $X$ is then decomposed as:

$$X = AS + \text{noise}, \quad (A.1)$$

where the dimensions of the factors $A$ and $S$ are $N \times R$ and $R \times M$, respectively. The rank of factorization, $R$, is chosen as: $NR + RM < NM$, to compress the original $NM$ elements in $X$ into a smaller number of elements, $NR$ in $A$ plus $RM$ in $S$, with some “noise” level. In blind source separation (BSS), $A$ and $S$ are called mixing and source matrices, respectively, and we can express the solution to Eq.(A.1) using a projection or unmixing matrix $W$:

$$Y = WX, \quad (A.2)$$

where the matrix $Y$ is the estimate of the source $S$. For simplicity, we assume without loss of generality that each row of $X$ has zero mean.

A.1.1 Principal Component Analysis

Principal component analysis (PCA) projects $N$-dimensional data onto a lower $R$-dimensional subspace as in Eq.(A.2) in a way that is optimal in a sum-squared error sense. That is, PCA represents an original data set or variables $x \in \mathbb{R}^N$ by using a new set of variables $y \in \mathbb{R}^R$, called principal components, that captures most of the summed squared vector lengths of the data. All the principal components $y_r$ (for $r = 1, \ldots, R \leq N$) are orthogonal to each other, and the projection coefficients $w_{rn}$ are chosen so as to satisfy the following condition.

The variance of the first principal component $y_1$ is the maximum among all possible choices of $w_{1n}$, and the variance of the $r$-th principal component $y_r$ (for
\( r = 2, \ldots, R \) is the maximum among all possible linear combinations \( \sum_n w_{rn} x_n \)
that are uncorrelated to \( y_{r'} \) (for \( r' = 1, \ldots, r - 1 \)).

By “projection coefficients” we mean:

\[
\|w_r\|^2 = \sum_n w_{nr}^2 = 1. \tag{A.3}
\]

From Eq.(A.2), the first principal component \( y_1^\top = (y_{11}, \ldots, y_{1M}) \) can be written as:

\[
y_1^\top = w_1^\top X, \tag{A.4}
\]

where \( w_1^\top = (w_{11}, \ldots, w_{1N}) \). The mean and the variance of \( y_1 \) are given as:

\[
\mathbb{E}[y_1] \overset{\text{def}}{=} \frac{1}{M} \sum_{m=1}^M y_{1m} = \frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N w_{1n} x_{nm} = \frac{1}{M} \sum_{n=1}^N w_{1n} \left( \sum_{m=1}^M x_{nm} \right) = 0, \tag{A.5}
\]

\[
\mathbb{V}[y_1] \overset{\text{def}}{=} \frac{1}{M - 1} y_1^\top y_1 = \frac{1}{M - 1} (X^\top w_1)^\top X^\top w_1 = w_1^\top \Lambda w_1 \geq 0, \tag{A.6}
\]

respectively, where \( \Lambda = XX^\top / (M - 1) \) is the covariance matrix of \( X \). We can then use the Lagrange multiplier method to find a projection \( w_1 \) that maximizes the variance \( \mathbb{V}[y_1] \) subject to Eq.(A.3):

\[
w_1 = \arg \max_{\|w\|=1} \mathbb{V}[w^\top X] = \arg \max_w w^\top \Lambda w - \lambda_1 \left( w^\top w - 1 \right), \tag{A.7}
\]

where \( \lambda_1 \) is a Lagrange multiplier. Let \( J_1 \) be the objective in Eq.(A.7), and we have:

\[
\frac{\partial J_1}{\partial w_1} = 2\Lambda w_1 - 2\lambda_1 w_1 = 0, \quad \text{that is,} \quad (\Lambda - \lambda_1 I) w_1 = 0, \tag{A.8}
\]

where \( I \) is the identity matrix. Substituting Eq.(A.8) into Eq.(A.6), we have:

\[
\mathbb{V}[y_1] = w_1^\top \Lambda w_1 = w_1^\top \lambda_1 w_1 = \lambda_1, \tag{A.9}
\]
where the last equality holds from Eq.(A.3). Therefore, from Eqs.(A.8) and (A.9), \( \lambda_1 \) and \( w_1 \) can be given as the maximum eigenvalue of \( \Lambda \) and the corresponding eigenvector, respectively.

Similarly, we can find the projections \( w_r \) for the \( r \)-th principal components \( y_r \) by mathematical induction. Let us assume that the projections \( w_i \) for \( i = 1, \ldots, r - 1 \) satisfy the following conditions:

\[
\begin{align*}
(A.10) & \quad (\Lambda - \lambda_i I) w_i = 0, \\
(A.11) & \quad w_i^\top w_j = \delta_{ij},
\end{align*}
\]

where \( \delta_{ij} \) is the Kronecker delta:

\[
\delta_{ij} \overset{\text{def}}{=} \begin{cases} 
1 & (i = j) \\
0 & (i \neq j)
\end{cases}
\]

For finding \( w_r \), the objective function \( J_r \) to be maximized is then given as:

\[
J_r = w_r^\top \Lambda w_r - \lambda_r (w_r^\top w_r - 1) - \sum_{i=1}^{r-1} \mu_i w_r^\top w_i
\]

where \( \lambda_r \) and \( \mu_i \) are Lagrange multipliers. Note that the last term on the right side of Eq.(A.13) indicates that \( y_r \) is uncorrelated to \( y_i \) for all \( i = 1, \ldots, r - 1 \). Setting the derivative of \( J_r \) equal to 0, we have:

\[
\frac{\partial J_r}{\partial w_r} = 2\Lambda w_r - 2\lambda_r w_r - \sum_{i=1}^{r-1} \mu_i w_i = 0.
\]

From Eqs.(A.10), (A.11), and (A.14), we have \( \mu_j = 0 \) for \( j = 1, \ldots, r - 1 \), and:

\[
(A.15) \quad (\Lambda - \lambda_r I) w_r = 0.
\]

Because the largest \( r - 1 \) eigenvalues of \( \Lambda \) and the corresponding eigenvectors are already used for representing the first to \( (r - 1) \)-th principal components, the variance of \( r \)-th principal
components is then given as the \( r \)-th largest eigenvalue, and the projection \( w_r \) as the corresponding eigenvector. Note that the matrix \( W = (w_1, \ldots, w_r)^\top \) is orthonormal, and the data matrix \( X = W^\top Y \) can be factorized by the matrices \( A = W^\top \) and \( S = Y = (y_1, \ldots, y_r)^\top \) as in Eq.(A.1).

A.1.2 Singular Value Decomposition

The previous Section A.1.1 described that the principal components are highly related to the eigenvalue problem on the covariance matrix \( \Lambda \) of an appropriately standardized data matrix \( X \). Here we show that PCA is also highly related to the singular value decomposition (SVD), a powerful tool for data mining and linear regression (see below Section A.2).

As before, let us assume we have an \( N \times M \) data matrix \( X \) containing \( M \) samples on the \( N \)-dimensional variables, and consider the following eigenvalue problem:

\[
\frac{1}{M - 1} XX^\top w_i = \lambda_i w_i \quad (i = 1, \ldots, N)
\]  

(A.16)

where \( \lambda_i \) and \( w_i \) are the \( i \)-th largest eigenvalue of the covariance matrix of \( X \) and its corresponding eigenvector normalized to have unit length (\( L_2 \)-norm), respectively. We then have:

\[
U \Sigma = X^\top V,
\]

(A.17)
where

\[
V = \begin{pmatrix} w_1 & \cdots & w_N \end{pmatrix} = W^T \quad (A.18)
\]

\[
\sigma_i = \sqrt{(M-1)\lambda_i} \quad (i = 1, \ldots, N) \quad (A.19)
\]

\[
\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ \vdots & \ddots \\ 0 & \sigma_N \end{pmatrix} \quad (A.20)
\]

\[
u_i = \frac{1}{\sigma_i} X^T w_i \quad (i = 1, \ldots, N) \quad (A.21)
\]

\[
U = \begin{pmatrix} u_1 & \cdots & u_N \end{pmatrix}. \quad (A.22)
\]

Because the matrix \(V\) is orthonormal, we have the SVD of \(X\) from Eq.(A.17):

\[
X = V \Sigma U^T. \quad (A.23)
\]

Note that the matrix \(U\) is also orthonormal:

\[
u_i^T u_j = \frac{1}{\sigma_i} (X^T w_i)^T \cdot \frac{1}{\sigma_j} X^T w_j = \frac{1}{\sigma_i \sigma_j} w_i^T X X^T w_j = \delta_{ij}, \quad (A.24)
\]

where the last equality holds from Eqs.(A.16) and (A.19). Suppose the rank of \(X\) is \(R\), Eq.(A.23) can be written as:

\[
X = \sum_{r=1}^{R} \sigma_r w_r u_r^T. \quad (A.23')
\]

Hence, the projection \(w_r\) for the \(r\)-th principal components can also be given as the left singular vector corresponding to the \(r\)-th largest singular value \(\sigma_r\) of \(X\).
A.1.3 Independent Component Analysis

While PCA seeks directions in “feature” space that best represent the data in a sum-of-squared-error sense, leading to *uncorrelated* directions, independent component analysis (ICA) seeks linear projections—not necessarily orthogonal to each other—that are most statistically *independent* from each other (Comon, 1994; Hyvärinen and Oja, 2000). Note that the latter is a much stronger condition than the former: uncorrelatedness involves only second-order statistics and implies independence only in the Gaussian case, whereas independence depends on all higher-order statistics and always implies uncorrelatedness.

One natural measure of the dependence between $R$ random variables $y_r$ (for $r = 1, \ldots, R$) is the mutual information (see also Section A.4):

$$I(y_1, \ldots, y_R) \overset{\text{def}}{=} \sum_{r = 1}^{R} H(y_r) - H(y), \quad \text{(A.25)}$$

where $H$ is the differential entropy as defined in Eq.(A.80) on page 167. Note that the mutual information $I(y_1, \ldots, y_R)$ is equivalent to the Kullback-Leibler divergence between the joint density $p(y)$ and the product of its marginal densities $\prod_r p(y_r)$, and that Eq.(A.25) is always non-negative and equals zero if and only if the variables $y_r$ are statistically independent.

Given an invertible linear transformation: $y = Wx$, Eq.(A.25) becomes:

$$I(y_1, \ldots, y_R) = \sum_{r = 1}^{R} H(y_r) - H(x) - \log|\det W|, \quad \text{(A.26)}$$

where $|\det W|$ is the absolute value of the determinant of $W$. If we then constrain $y_r$ to be *uncorrelated* and to have unit variance, we have:

$$I(y_1, \ldots, y_R) = \sum_{r = 1}^{R} H(y_r) + C_0 = -\sum_{r = 1}^{R} Q(y_r) + C_0', \quad \text{(A.27)}$$

---

1In fact we will perform “whitening” on $x$ and thus can constrain $W$ to be orthogonal, so $|\det W| = 1$. 

141
where $C_0$ and $C'_0$ are constants not dependent on $W$, and $Q(y)$ is the “negentropy” of $y$:

$$Q(y) \overset{\text{def}}{=} H(y_{\text{gauss}}) - H(y), \quad (A.28)$$

where $y_{\text{gauss}}$ is a set of Gaussian random variables that has the same covariance matrix as $y$. Because a Gaussian variable has the largest entropy among all random variables of equal variance (see Eq.(A.83) on page 168), the negentropy is non-negative and can be considered as a natural measure of non-gaussianity. Hence, maximizing independence between $y_r$ is equivalent to minimizing the mutual information $I(y_1, \ldots, y_R)$, which in turn is equivalent to maximizing the sum of the negentropies of $y_r$, i.e., the non-gaussianity.

One problem of using mutual information or negentropy as objective functions is that they are hard to estimate. Several approximation methods based on polynomials or higher-order statistics have then been proposed, and below I will describe one such algorithm for illustration—called “FastICA”—that exploits the fourth-order cumulant or kurtosis to maximize the non-gaussianity of $y_r$ (Hyvärinen and Oja, 1997, 2000).

**Whitening**

Whitening is a preprocessing step where the observed data $x$ is linearly transformed into a new vector $\tilde{x}$ of unit variance and uncorrelated elements:

$$\mathbb{E}[\tilde{x}\tilde{x}^T] = I. \quad (A.29)$$

This is useful for some ICA algorithms as it makes the target $W$ a “rotation” matrix (see below). Using SVD as in Eq.(A.23), we can decompose the data matrix into: $X = V\Sigma U^T$. Then such transformation that satisfies Eq.(A.29) is given by $M = \Sigma^{-1}V^T$, i.e.,

$$\tilde{x} = Mx \approx MAs. \quad (A.30)$$
The last approximation is from Eq.(A.1) with \( s \) being a corresponding column of \( S \), and we can see that \( B = MA \) is orthogonal:

\[
I = E[\tilde{X}\tilde{X}^\top] \approx B E[ss^\top] B^\top \approx BB^\top. \tag{A.31}
\]

The last approximation holds because we assume independent sources, \( s \), and because the sign and magnitude of the sources are arbitrary. Therefore, a goal in ICA can now be considered as to find appropriate orthogonal projection \( W = B^\top \) on the whitened data \( \tilde{X} \).

**FastICA algorithm**

The fourth-order cumulant—or kurtosis—of \( y \) is defined as:

\[
\mathbb{K}[y] \overset{\text{def}}{=} E[y^4] - 3(E[y^2])^2. \tag{A.32}
\]

The kurtosis is zero for a Gaussian variable, positive for heavy-tailed super-Gaussian distributions, and negative for light-tailed sub-Gaussian distributions.

Using the Lagrange multiplier \( \lambda_1 \), the objective function for the first projection \( y_1 = w_1^\top \tilde{X} \) becomes:

\[
J_1 = \mathbb{K}[y_1] + \lambda_1 (\|w_1\|^2 - 1) = E\left( (w_1^\top \tilde{X})^4 \right) - 3\|w_1\|^4 + \lambda_1 (\|w_1\|^2 - 1). \tag{A.33}
\]

Setting the derivative of \( J_1 \) equal to zero:

\[
\frac{\partial J_1}{\partial w_1} = E\left[ 4 (w_1^\top \tilde{X})^3 \tilde{X} \right] - 12\|w_1\|^2 w_1 + 2\lambda_1 w_1 = 0, \tag{A.34}
\]

the update rule based on the fixed-point algorithm is then given as:

\[
w \leftarrow \frac{-2}{\lambda_1} \left( E\left[ (w_1^\top \tilde{X})^3 \tilde{X} \right] - 3\|w_1\|^2 w_1 \right). \tag{A.35}
\]
Considering the normalization $\|w_1\| = 1$ from Eq.(A.3), we then have:

\[ w_1 \leftarrow \text{normal}\left(\mathbb{E}\left[\left(\overline{w_1^\top \tilde{x}}\right)^3\tilde{x}\right] - 3w_1\right), \tag{A.36} \]

where normal(·) is the operator for column-wise normalization.

For finding more than one independent projections $w_r$ (for $r = 2, \ldots, R$), we can use the same update rule as Eq.(A.36) but must decorrelate $w_r$ with all the other projections $w_{r'}$ (for $r' = 1, \ldots, r - 1$). Using $w_\top r w_{r'} = \delta_{r,r'}$, the Gram-Schmidt-like decorrelation gives the following additional step after every iteration by Eq.(A.36):

\[ w_r \leftarrow w_r - B_{r-1}B_{r-1}^\top w_r, \tag{A.37} \]

where $B_{r-1} = (w_1, \ldots, w_{r-1})$. Finally, let $B = B_R$, and we have independent components $Y$ from Eqs.(A.2) and (A.30):

\[ Y = B^\top \tilde{X} = B^\top M X. \tag{A.38} \]

Note that $Y$ can be given by rotating the principal components of $X$, i.e., $MX = U^\top$, by the orthogonal matrix $B^\top$.

It should also be mentioned that this FastICA algorithm is equivalent to replacing $-\log p(y_r)$ with $y_r^4$ in the definition of differential entropy $H(y_r) \overset{\text{def}}{=} \mathbb{E}[-\log p(y_r)];$ Eq.(A.80) on page 167) used in the information theoretic objective in Eq.(A.27) on page 141, and seeking an orthogonal transformation $W$ that gives an extremum on the fourth-order moment: $\mathbb{E}[y_r^4]$. Taking the first projection $w_1$, for example, the objective function using the Lagrange multiplier $\lambda_1$ is:

\[ J = \mathbb{E}[y_1^4] - \lambda_1 2\left(\overline{w_1^\top w_1} - 1\right). \tag{A.39} \]

Setting the derivative of $J$ equal to zero, we have:

\[ \frac{\partial J}{\partial w_1} = \mathbb{E}[4\overline{\tilde{x}} \tilde{y}_1^3] - \lambda_1 w_1 = 0. \tag{A.40} \]
Using the Newton-Raphson method, the update rule to solve Eq. (A.40) is:

\[ w_1 \leftarrow w_1 - f'(\tilde{x})^{-1} f(\tilde{x}) \], \quad \text{where} \]
\[ f(\tilde{x}) = E[4\tilde{x}y_1^3] - \lambda_1 w_1, \]
\[ f'(\tilde{x}) = E[12\tilde{x}\tilde{x}^T y_1^2] - \lambda_1 I \approx E[\tilde{x}\tilde{x}^T] E[12y_1^2] - \lambda_1 I \approx (12 - \lambda_1)I. \]

Note the whitening as in Eq. (A.29), and the constraint on \( y_1 = w_1^T\tilde{x} \) to be uncorrelated and to have unit variance for obtaining the objective as in Eq. (A.27). Hence, substituting Eqs. (A.42) and (A.43) into Eq. (A.41) followed by (the multiplication with \( \lambda_1/4 - 3 \) and) the normalization on \( w_1 \), we have the same update rule as Eq. (A.36).

### A.1.4 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) achieves the approximation of a data matrix \( X \) as in Eq. (A.1) on page 136 under non-negativity constrains on each element of the factors \( A \) and \( S \) (Lee and Seung, 1999, 2001). With the Lagrange multiplier \( \lambda \), the objective function to be minimized is given as:

\[ J_{\text{NMF}} = D(X, AS) + \lambda \cdot \text{regularizer} \quad \text{for} \quad A \geq 0, \quad S \geq 0, \tag{A.44} \]

where \( D(\cdot, \cdot) \) is a distance measure such as (matrix) norms or Kullback-Leibler divergence, and the regularizer can be, e.g., a sparseness constraint on \( S \): \( \| S \|_1 = \sum_{r,m} |s_{rm}| \), or a smoothness constraint on \( A \): \( \sum_{n,r} |a_{n,r} - a_{n-1,r}| \). For simplicity, however, here we ignore the regularizer (or \( \lambda = 0 \); see also Section A.2.1), and show below an algorithm that alternatively updates \( A \) and \( S \) where \( D(\cdot, \cdot) \) is the squared Frobenius norm (Lee and Seung, 2001). Taking the gradient of \( J_{\text{NMF}} \) with respect to \( S \):

\[ \nabla_S J_{\text{NMF}} = 2A^T (AS - X), \tag{A.45} \]
and setting a step-size matrix as:

$$\Delta S = \frac{S}{2A^TAS},$$ (A.46)

a steepest descent gives the following update rule for $S$:

$$S \leftarrow S - \Delta S \cdot \nabla_S J_{NMF} = S \cdot \frac{A^TX}{A^TAS},$$ (A.47)

where the operators $\cdot$ and $\div$ indicate elementwise multiplication and division, respectively. Similarly, the update rule for $A$ can be written as:

$$A \leftarrow \text{normal} \left( A \cdot \frac{XS^T}{A S^T} \right).$$ (A.48)

Note that the normalization of the columns of $A$ to unit length is required for finding a unique solution, and that Eqs.(A.47) and (A.48) converge to a local minimum of Eq.(A.44).

### A.1.5 Characteristics of Each Factorization Method

The three factorization methods described above (PCA, ICA, and NMF) all decompose a data matrix $X$ into a new basis matrix $A$ and a corresponding coefficient matrix $S$, but have different characteristics because of the different constraints imposed on $A$ and $S$.

PCA constrains the columns of $A$ to be orthonormal and the rows of $S$ to be orthogonal. The resulting basis set $A$ then has a good statistical interpretation as the subspace of largest variance, but it is often difficult to qualitatively interpret $A$ because PCA allows negative values on $A$ and $S$.

ICA chooses the basis not as orthogonal but as independent as possible. This technique is thus more powerful than PCA to capture the statistical structures of the sources. Note however that, rather than directly optimize the independence, ICA algorithms typically maximize higher-order moments (e.g., kurtosis for FastICA in Section A.1.3), and thus the resulting components are not necessarily independent. Nevertheless, ICA is quite successful in many
cases—e.g., in the case of natural images and sounds (Simoncelli and Olshausen, 2001)—and has been applied to solve difficult problems such as the BSS problem (Choi et al., 2005).

In contrast to PCA and ICA, NMF does not allow negative entries in $A$ and $S$. Therefore, only additive combinations are allowed for data reconstruction, leading to a “parts”-based representation. (To my knowledge, however, no mathematical explanation is given for why this would be the case.) That is, NMF can even extract correlated features, and the basis often has intuitive interpretations.

Below I will show simulation results (done in MATLAB) where PCA, ICA, and NMF were applied to sample data sets such as Kanji Chinese characters to see the characteristics of each matrix decomposition method.

**Example 1: Kanji Chinese characters**

A data set of kanji Chinese characters was generated by rasterizing the “MS gothic” fonts (30 points), where some characters consist of combinations of “left” and “right” parts whereas others “top” and “bottom” parts. The data set contains images ($30 \times 30$ pixels) of 1006 characters that students in Japan are supposed to learn by the age of 12 (National Curriculum Standards for Elementary Schools in Japan issued by the Ministry of Education, Culture, Sports, Science, and Technology; as of July, 2007). The list\(^2\) was downloaded from http://www.mext.go.jp/b_menu/shuppan/sonota/990301b/990301d.htm (in Japanese).

When applied to the data set, NMF extracted those “meaningful” parts that are used frequently in the original data set, whereas PCA and ICA failed to find any “meaningful” basis elements but learned holistic representations (Figure A.1; data not shown for PCA).

\(^2\)Also available at the website of AOZORA-BUNKO: http://www.aozora.gr.jp/kanji_table/kyouiku_list.zip.
Figure A.1: NMF found “meaningful” parts in Chinese characters while ICA failed. NMF and ICA were applied to a data set of 1006 kanji Chinese characters (30×30 pixels). Some of the basis images obtained by NMF showed “meaningful” parts as shown in the bottom panel. In contrast, ICA learned a holistic representation.
Example 2: Images comprised of non-overlapping arbitrary basis

To confirm if NMF can always extract the basis images for the original data set, a set of 1000 images (25×25 pixel) were generated by an additive combination of 25 arbitrary non-overlapping basis images $A_{\text{ini}}$ using the coefficients drawn from $\mathcal{N}[0, 1]$, where $\mathcal{N}[\mu, \sigma^2]$ is the Gaussian—or Normal—distribution with mean $\mu$ and variance $\sigma^2$. The basis images $A_{\text{ini}}$ were generated by the two-dimensional Normal distribution $\mathcal{N}[(\mu_x, \mu_y), \sigma^2]$ with mean $(\mu_x, \mu_y) = (5i - 2, 5j - 2)$ for $i, j = 1, 2, 3, 4, 5$ and standard deviation $\sigma = 2.5$ pixels. Each basis image was then normalized to sum to unity, resulting in a not completely but essentially non-overlapping basis image set.

When the correct rank of factorization—i.e., the exact number of the underlying basis images ($R = 25$ in this example)—was chosen, the basis images obtained by NMF had one-to-one correspondence to the original basis $A_{\text{ini}}$ (Figure A.2). When the rank was chosen by far smaller or larger than the correct value, however, the obtained basis images showed quite different patterns from $A_{\text{ini}}$. In contrast, when the rank was chosen a little smaller or larger than the correct rank size, some images showed redundancy but the others remained to have good one-to-one correspondence to $A_{\text{ini}}$. These results show that it is important to choose an appropriate rank size $R$ for the factorization, although it remains to be addressed what a good criterion would be. For data compression, $R$ should satisfy: $NR + RM < NM$, but this inequality would not be very helpful for determining an appropriate rank size.
Figure A.2: The original non-overlapping basis images and those obtained by NMF. A set of 25 non-overlapping arbitrary basis images ($A_{ini}$; top-left) was randomly combined to generate 1000 image data set (top-right), which in turn was decomposed back into basis images by NMF (bottom-right). Note that there is one-to-one correspondence between the two basis sets. The numbering was done by hand.
Example 3: Images comprised of overlapping arbitrary basis

The factorization methods was next tested on data sets $X$ (1000 samples) generated by additive *dense* combinations of arbitrary overlapping basis images $A_0$ ($25\times25$ pixels) weighted with $|\mathcal{N}[0,1]|$, and on data sets $X_S$ (1000 samples) generated by additive *sparse* combinations of randomly-chosen 10 out 25 basis images with the weights of $|\mathcal{N}[0,1]|$. To make the overlapping basis images $A_0$, two random points were first chosen for each basis image within a two-dimensional plane $(x,y)$ for both $x$ and $y \in [0,25]$ pixels. A line segment was generated by connecting these two points, and the value for each pixel on the plane was determined by: $\exp[-d^2/2]$, where $d$ is the Euclidean distance between the line segment and the pixel of interest, followed by the normalization to sum to unity. In total, we generated 25 overlapping basis images $A_0$ for simulations (Figure A.3, top-left).

Figure A.3 shows that NMF with the correct rank size ($R = 25$) worked well to extract underlying features $A_0$ from the sparse data set $X_S$ but not from the dense data set $X$. In contrast, PCA and ICA failed to break even $X_S$ back into the original basis $A_0$ and corresponding coefficients, although some of the ICA basis images had one-to-one correspondence to $A_0$. These simulation results then suggest that NMF would be preferable to ICA or PCA for extracting features of sound signals because natural sound ensembles are supposedly made up of a sparse combination of sound elements (see Section 2.2.4).
Figure A.3: **Original overlapping basis images** $A_0$ and those obtained by NMF, PCA, and ICA. The basis images obtained by NMF from a sparse data set $X_S$ showed good one-to-one correspondence to the original basis $A_0$ (except for #18), while the other methods failed to show good fidelity.
A.2 Linear Regression

To understand how the brain processes sensory information, it is important to study the relationship between input stimuli and the output neural responses (for review, see e.g., Simoncelli et al., 2004; Wu et al., 2006). Here I briefly overview linear regression, one of the basic and the simplest methods for modeling input-output functions. Section A.2.1 reviews regularization methods to find the best (static) linear models (for details, see e.g., Hastie et al., 2001; Duda et al., 2001; Chen and Haykin, 2002), and Section A.2.2 describes the regression techniques from the probabilistic viewpoints. Section A.2.3 then shows some ways for incorporating temporal dynamics to better characterize the system (for further extensions, see e.g., Nelles, 2001).

A general goal in a regression model is to predict an output \( y \) from a vector of inputs \( x \in \mathbb{R}^N \). The linear regression model assumes that the regression function \( f_L \) is linear\(^3\) and has the form:

\[
\hat{y} = f_L(x) = \alpha_0 + \sum_{n=1}^{N} \alpha_n x_n = \alpha_0 + x^T \alpha
\]  
(A.49)

where \( \hat{y} \) is the estimated output, and \( \alpha \) (and \( \alpha_0 \)) are unknown parameters or coefficients. Typically we have a set of training data: \( (y_m, x_m) \) for \( m = 1, \ldots, M \geq N \) to estimate the coefficients \( \alpha \) (and \( \alpha_0 \)). The most common estimation method is to minimize the residual sum of squared errors between the estimated output \( \hat{y} \) and the original output \( y \):

\[
E(\alpha) = \|y - \hat{y}\|^2 = (y - X^T \alpha)^T (y - X^T \alpha),
\]  
(A.50)

where the \( m \)-th column of the matrix \( X \) consists of an \( m \)-th input vector \( x_m \). For simplicity, here we assume that the outputs have zero mean: \( \sum_m y_m = 0 \), that is, \( \alpha_0 = 0 \) in Eq.(A.49).

\(^3\)For a member of any vector space \( x \) (and \( y \)), a function \( f(x) \) is linear if it satisfies \( f(x + y) = f(x) + f(y) \) and \( f(\alpha x) = \alpha f(x) \) for all scalar \( \alpha \).
The least square solution is then given by:

\[ \hat{\alpha}_{ls} = (XX^\top)^{-1}Xy. \] (A.51)

Note that \( X^\dagger = (XX^\top)^{-1}X \) is called the pseudoinverse of \( X^\top \), and that \( XX^\top \) and \( Xy \) are sometimes referred as _auto-correlation_ and _cross-correlation_, respectively, and Eq.(A.51) as _reverse correlation_ in the neurophysiological jargon (de Boer and Kuyper, 1968; Eggermont et al., 1983; Ringach and Shapley, 2004).

### A.2.1 Regularization Methods

In practice, the auto-correlation \( XX^\top \) in Eq.(A.51) could have some eigenvalues close to zero, leading to an overfitting and a very noisy estimate of the coefficients \( \alpha \). To address this issue,\(^4\) a regularizer is often introduced to place constraints on the coefficients so that we do not suffer much from high variability in the estimation. The cost function to be minimized is then, from Eq.(A.50),

\[ E_{\text{reg}}(\alpha, \lambda) = E(\alpha) + \lambda \cdot \text{regularizer}, \] (A.52)

where the parameter \( \lambda \geq 0 \) determines the strength of the constraint.

Ridge regression is one of the shrinkage methods to penalize strong deviations of the parameters from zero; i.e., the “regularizer” in Eq.(A.52) is given as: \( \|\alpha\|_2^2 = \alpha^\top\alpha \). The solution for ridge regression can then be expressed as:

\[ \hat{\alpha}_{\text{ridge}} = (XX^\top + \lambda I)^{-1}Xy. \] (A.53)

\(^4\)Another common technique is to weight the contribution of the square residuals on the cost function for each data; i.e., to minimize \( E_{\text{wls}}(\alpha, \Omega) = (y - X^\top \alpha)^\top \Omega (y - X^\top \alpha) \), where \( \Omega \) is a diagonal matrix whose \((i, i)\)-element consists of the weight \( \omega_i \) on the \( i \)-th data. The solution of weighted least squares is given as:

\[ \hat{\alpha}_{\text{wls}} = (X\Omega X^\top)^{-1}X\Omega y, \] and \( \hat{\alpha}_{\text{wls}} = \hat{\alpha}_{\text{ls}} \) when \( \Omega = I \). Note that \( \Omega \) has many free parameters to be estimated, and thus the regularizer approach would be preferable that often has less free parameters; e.g., ridge regression in Eq.(A.53) has only a single parameter \( \lambda \).
Note that the solution adds a positive constant $\lambda$ to the diagonal of $XX^\top$ before the inverse, which makes the matrix well conditioned (nonsingular) even if $XX^\top$ is poorly conditioned (or is not practically a full-rank matrix). Also note that the ridge parameter $\lambda$ can be determined by common model selection techniques (Hastie et al., 2001; MacKay, 2003), such as Akaike or Bayesian information criteria (Akaike, 1974; Schwarz, 1978), minimal description length (Rissanen, 1983, 1986), or (generalized) cross-validation (Stone, 1974; Efron, 1983).

The least square solution $\hat{\alpha}_{ls}$ in Eq.(A.51) and the ridge regression solution $\hat{\alpha}_{ridge}$ in Eq.(A.53) are highly related to the SVD described in Section A.1.2. Following the notation in Eq.(A.23) on page 140, the pseudoinverse $X^\dagger$ can be written as:

$$X^\dagger = (XX^\top)^{-1} X = (V \Sigma^2 V^\top)^{-1} V \Sigma U^\top = V \Sigma^{-1} U^\top.$$  (A.54)

Therefore, from Eqs.(A.51) and (A.53),

$$\hat{\alpha}_{ls} = V \Sigma^{-1} U^\top y;$$  (A.51’)
$$\hat{\alpha}_{ridge} = V (\Sigma^2 + \lambda I)^{-1} \Sigma U^\top y.$$  (A.53’)

Note that the $(r, r)$-element of the diagonal matrix $(\Sigma^2 + \lambda I)^{-1} \Sigma$ is $\sigma_r/(\sigma_r^2 + \lambda)$, where $\sigma_r$ is the $r$-th largest singular value. The estimated outputs $\hat{y} = X^\top \hat{\alpha}$ can then be written as:

$$\hat{y}_{ls} = X^\top (XX^\top)^{-1} X y = U U^\top y,$$  (A.55)
$$\hat{y}_{ridge} = X^\top (XX^\top + \lambda I)^{-1} X y = U \Sigma (\Sigma^2 + \lambda I)^{-1} \Sigma U^\top y = \sum_r u_r \frac{\sigma_r^2}{\sigma_r^2 + \lambda} u_r^\top y.$$  (A.56)

Note that $U^\top y$ in least squares (Eq.(A.55)) are the coordinates of $y$ with respect to the orthogonal basis $U$, and the coordinates are shrunk by the factor of $\sigma_r^2/(\sigma_r^2 + \lambda)$ in the ridge regression (Eq.(A.56)). In least squares for example, the estimation noise is then given as:

$$\eta_{ls} = y - \hat{y}_{ls} = (I - U U^\top) y.$$  (A.57)
A.2.2 Probabilistic Interpretation

The regression framework described above is based on a frequentist viewpoint, but it can be readily interpreted from a probabilistic or Bayesian viewpoint as well. Suppose that a sample \( y \) is derived from a joint density \( p(y, \theta) \) —or likelihood—with some parameters \( \theta \), the principle of maximum likelihood (ML) assumes that the most reasonable values for \( \theta \) are the ones for which the probability of obtaining the observed samples is the largest.

Least squares regression is then the same as ML under the assumption of additive i.i.d. Gaussian noise. Using the conditional likelihood: \( p(y|x, \theta) = \mathcal{N}[x^\top \alpha, \sigma^2] \), where \( \theta \equiv \{\alpha, \sigma\} \) for the linear regression as in Eq.(A.49), the log-likelihood of the observed data set \((y_m, x_m)\) for \( m = 1, \ldots, M \) can be written as, from Eq.(A.50),

\[
\sum_{m=1}^{M} \log p(y_m|x_m, \theta) = -\frac{E(\alpha)}{2\sigma^2} - M \log \sqrt{2\pi\sigma^2}. \tag{A.58}
\]

Maximizing Eq.(A.58) is thus equivalent to minimizing \( E(\alpha) \); i.e., \( \hat{\alpha}_{ML} = \hat{\alpha}_{ls} \) as in Eq.(A.51), and the unbiased estimate of \( \sigma^2 \) is given as, from Eq.(A.57): \( (M - N - 1)\hat{\sigma}^2 = \|\eta_{ls}\|_2^2 \sim \sigma^2 \chi_{M-N-1}^2 \), where \( \chi_{M-N-1}^2 \) is a chi-square distribution with \( M - N - 1 \) degrees of freedom. Different cost functions are obtained by the ML approach for other noise distributions.

Regularization methods in Section A.2.1 can be considered as maximum a posteriori (MAP) approach where the parameters \( \theta \) in ML are also treated as random variables. Let \( p(\theta) \) be a prior distribution of \( \theta \), then a posterior distribution \( p(\theta|y) \) can be given by the Bayes theorem:

\[
p(y, \theta) = p(\theta|y)p(y) = p(y|\theta)p(\theta), \text{ hence, Posterior } \propto \text{ Likelihood } \times \text{ Prior.} \tag{A.59}
\]

The prior can be written in the exponential form: \( p(\theta) \propto \exp[-\lambda \cdot \text{regularizer}] \), without loss of generality. Taking the negative logarithm of Eq.(A.59), we then have the equivalent objective
as Eq. (A.52):

\[- \log p(\theta|y) \propto - \log p(y|\theta) - \log p(\theta) = E_{\text{reg}}(\alpha, \lambda). \tag{A.60}\]

Therefore, MAP with a Gaussian prior is equivalent to ridge regression with an $L_2$ penalty on $\alpha$, and in the case of a Laplacian prior, MAP is equivalent to lasso (least absolute shrinkage and selection operator) regression where an $L_1$ penalty is imposed on $\alpha$, encouraging sparsity (see also Section 2.2.3). Note that ML (without regularizer) is a special case of MAP with a “flat” prior.

### A.2.3 Dynamic Models

Because the encoding properties of neurons typically vary over time, static models such as Eq. (A.49) are often too simple to account for the neural behaviors in response to sensory signals, even though we have a reasonable interpretation; e.g., $\alpha$ corresponds to the neuron’s “receptive field” (Barlow, 1953; Kuffler, 1953; Ringach, 2004). Here I explain two ways to extend the static models by incorporating temporal dynamics.

#### Adaptive Filters

One common technique to incorporate temporal dynamics for extending the static models is to make the filter $\alpha$ adaptive (Stanley, 2002):

\[\hat{\alpha}_m = \arg \min_{\alpha} \sum_{t=0}^{T} \xi^t (y_{m-t} - \hat{y}_{m-t})^2, \tag{A.61}\]

where $\hat{y}_m$ follows Eq. (A.49), and $\xi \in [0, 1]$ is a “forgetting factor” that down-weights past information in an exponential manner. Note that the estimation is based only on the past information up to time $T$, instead of using the whole data as in Eq. (A.51) or Eq. (A.53).
If $T$ is large enough, the auto-correlation $\Phi_m$ and the cross-correlation $\psi_m$ at time $m$ can be approximated as:

\[
\Phi_m = \sum_{t=0}^{T} \xi^t x_{m-t}^\top x_{m-t} \approx \xi \Phi_{m-1} + x_m x_m^\top, \tag{A.62}
\]

\[
\psi_i = \sum_{t=0}^{T} \xi^t x_{m-t} y_{m-t} \approx \xi \psi_{m-1} + x_m y_m. \tag{A.63}
\]

The adaptive filter $\hat{\alpha}_m$ in Eq.(A.61) can then be estimated as:

\[
\hat{\alpha}_m = \left( \Phi_m + \lambda I \right)^{-1} \psi_m \approx \hat{\alpha}_{m-1} + \left( \Phi_m + \lambda I \right)^{-1} x_m (y_m - x_m^\top \hat{\alpha}_{m-1}), \tag{A.64}
\]

where the first equality is by the ridge regression technique (see above Section A.2.1). Note that the development of the adaptive filter consists of the estimate at time $m - 1$ plus the update (error correction) based on the new information at time $m$.

**Integrate-and-fire-like Model**

Another way to extend the static model is to consider both input- and output-history dependence (up to time $\tau_x$ and $\tau_y$, respectively), i.e., to incorporate a recursive term into Eq.(A.49):

\[
\hat{y}_m = \alpha_0 + \sum_{t=1}^{\tau_x} x_{m-t}^\top \alpha_t + \sum_{t=1}^{\tau_y} y_{m-t} \beta_t. \tag{A.65}
\]

Eq.(A.65) implements an infinite impulse response (IIR) filter whereas the earlier formulations are all finite impulse response (FIR) filters, and that linear regression technique can be exploited to estimate the parameters $\alpha_t$ and $\beta_t$ after some rearrangement (Nelles, 2001). This recursive least squares method is essentially equivalent to the Kalman filter, although the latter is usually applied for the estimation of states rather than just parameters (Kalman, 1960a,b; Roweis and Ghahramani, 1999).
We could also rewrite Eq.(A.65) as follows:

\[
\frac{\Delta y_m}{\Delta m} = \alpha_0 + \sum_{t=1}^{\tau_x} x^\top_{m-t}\alpha_t - \sum_{t=1}^{\tau_y} y_{m-t}\bar{\beta}_t ,
\]  

(A.66)

which is a variant of an integrate-and-fire (IF) model (Lapicque, 1907; Stevens and Zador, 1998; Gerstner and Kistler, 2002; Paninski et al., 2005). The parameters could then be interpreted more readily from the viewpoint of neurons—as before, \(\alpha_t\) corresponds to the neuron’s “receptive field,” and \(\bar{\beta}_t\) response-dependent factors such as refractoriness or adaptation effects—and more elaborate models have been proposed and applied in this context (see e.g., Fishbach et al., 2001; Keat et al., 2001; Gerstner and Kistler, 2002; Fishbach et al., 2003; Paninski et al., 2004; Pillow et al., 2005).

### A.3 Nonlinear Regression

Linear models have been widely used in sensory systems neuroscience due to their simplicity and interpretability (Paninski, 2003a; Simoncelli et al., 2004; Wu et al., 2006). However, neural responses to sensory stimuli cannot be fully explained by such “simple” encoding models in general, even though neurons sometimes show high trial-to-trial reliability (for the primary auditory cortex, see e.g., Machens et al., 2004; Deweese and Zador, 2004; see also Chapter 3). We thus need more “complex” models, and below I will briefly overview some nonlinear regression techniques, especially from a viewpoint of artificial neural networks. Section A.3.1 shows that incorporating (static) nonlinearities into the linear models (see above Section A.2) leads to an equivalent model structure as a “perceptron.” Section A.3.2 then describes a general framework—multi-layer neural networks—that has more “units/layers” and higher expressive power. Note that the central idea of the model framework is: (1) to extract features as linear combinations of inputs, and (2) to model outputs as a nonlinear function of these features. Finally, Section A.3.3 shows one technique—support vector regression—developed recently for obtaining better generalization performance.
A.3.1 Perceptron

(Static) nonlinearities can be given as a nonlinear transformation \( f \) that acts on the output of the linear model (\( f_L \) as in Eq.(A.49) on page 153) to form a new better estimate (see e.g., Paninski, 2003a; Simoncelli et al., 2004; Machens et al., 2004):

\[
\hat{y} = f(f_L(x)) = f(\alpha_0 + x^T \alpha).
\]  \hspace{1cm} (A.67)

For finding a nonlinear function \( f \), we typically plot the actual output \( y \) against the linear estimates \( f_L(x) \) and generate a calibration curve, e.g., by robust locally weighted scatter-plot smoothing (LOWESS; Cleveland, 1979; Cleveland and Devlin, 1988). Note that this approach does not formally yield the optimal estimate of \( f \) when \( \hat{\alpha} \) is a biased estimator due to data limitations and/or sampling biases, and simultaneous updates of \( f \) and \( f_L \)—e.g., to maximize mutual information between \( y \) and \( f_L(x) \) by gradient descent—are required in general for optimal estimation (Paninski, 2003a; Sharpee et al., 2004, 2006).

The model structure in Eq.(A.67) is known as a “neuron”—or a “perceptron”—in artificial neural network studies (McCulloch and Pitts, 1943; Rosenblatt, 1958), and has been widely applied in pattern classifications (Hertz et al., 1991; Hastie et al., 2001; Duda et al., 2001) as well as in neuroscience; e.g., cerebellar functions can be well explained as perceptrons (Marr, 1969; Albus, 1971; Ito et al., 1982). However, (single-layer) perceptrons work only for “linearly separable” problems (Minsky and Papert, 1969)—e.g., the logical function of exclusive-OR (XOR) cannot be learned—and “hidden layers” are required to overcome the limitations (see below Section A.3.2).

A.3.2 Artificial Neural Network

Multi-layer networks can provide optimal solutions to an arbitrary fitting problem (Hecht-Nielsen, 1989; Hertz et al., 1991), where each “unit” in each “layer” extracts features as linear combinations of inputs (from units in the previous layer) and passes outputs by a nonlinear
function of these features to units in the following layer. In the case of three-layer feedforward networks that consist of input, hidden, and output layers interconnected with modifiable weights, the regression function can be written as a cascade of mappings:

$$\hat{y} = f(\alpha_0 + z^T\alpha), \quad \text{where} \quad z_r = f_r(\alpha_{r0} + x^T\alpha_r) \quad \text{for} \quad r = 1, \ldots, R.$$  \hspace{1cm} (A.68)

Note that this formulation is in essence no different from the basic framework of Eq.(A.67); i.e., Eq.(A.68) performs linear regression with $\alpha$—followed by nonlinear transformation $f$—in a “feature” space of $z \in \mathbb{R}^R$ where the inputs $x \in \mathbb{R}^N$ have been mapped nonlinearly by $\alpha_r$ and $f_r$.

One of the most popular methods for training parameters in such artificial neural networks is known as “backpropagation” or the generalized delta rule (Widrow and Hoff, 1960; Rumelhart et al., 1986), based on gradient descent in the sum squared errors (see also least squares; Eq.(A.49)). Let us suppose we have a training data set: $(y_m, x_m)$ for $m = 1, \ldots, M$, then the learning algorithm can be written as:

$$E(\alpha) = \sum_{m=1}^{M} (y_m - \hat{y}_m)^2 \quad \text{and} \quad \alpha \leftarrow \alpha - \lambda \frac{\partial E}{\partial \alpha}, \hspace{1cm} (A.69)$$

where $\lambda(>0)$ is the learning rate—indicating the relative size of the change in parameters or “weights”—and the gradient $\partial E/\partial \alpha$ can be computed by using the chain rule for differentiation:

$$\frac{\partial E}{\partial \alpha_r} = - \sum_{m=1}^{M} (y_m - \hat{y}_m) f'(\alpha_0 + z_m^T\alpha) z_{rm},$$ \hspace{1cm} (A.70)

$$\frac{\partial E}{\partial \alpha_{rn}} = - \sum_{m=1}^{M} (y_m - \hat{y}_m) f'(\alpha_0 + z_m^T\alpha) \alpha_r f'_r(\alpha_{r0} + x_m^T\alpha_r) x_{nm}. \hspace{1cm} (A.71)$$
Let us write Eqs. (A.70) and (A.71) as:

\[
\frac{\partial E}{\partial \alpha_r} = \sum_{m=1}^{M} \Delta_{\text{output}}^m z_{rm},
\]

(A.72)

\[
\frac{\partial E}{\partial \alpha_{rn}} = \sum_{m=1}^{M} \Delta_{\text{hidden}}^m x_{rn},
\]

(A.73)

where \(\Delta_{\text{output}}^m\) and \(\Delta_{\text{hidden}}^m\) are the “errors”—or the “sensitivity”—from the current model at the output and hidden layer units, respectively. Then we have:

\[
\Delta_{\text{hidden}}^m = f'_r (\alpha_{r0} + x_m^T \alpha_r) \Delta_{\text{output}}^m,
\]

(A.74)

showing that the sensitivity is propagated “back” from the output unit to the hidden units.

Several comments for the application of neural networks are in order (for details, see e.g., Hertz et al., 1991; Hastie et al., 2001; Duda et al., 2001). First, the learning algorithm described above was the most straightforward one, and more sophisticated algorithms have been developed, e.g., by using conjugate gradient descent based on a second-order analysis of the error function. Second, common choices of (differentiable) functions \(f\) and \(f_r\) are Gaussian (radial basis), logistic/sigmoidal, polynomial, and Heaviside/rectification functions. Third, neural networks often have too many parameters and will easily overfit the data. Several regularization methods have been developed to overcome this drawback—e.g., early stoppings, cross-validation, complexity adjustment/pruning, and weight decay (which is essentially equivalent to ridge constraints as described in Section A.2.1), and so on—but it is also important to choose appropriate number of hidden units/layers and initial conditions to find optimal solutions. Note also that the error function often possesses many local minima. Finally, it is often useful to preprocess data—e.g., standardization or dimension reduction by PCA—before training a network.

In sensory systems neuroscience, artificial neural networks were applied to characterize neural response patterns (Lehky et al., 1992; Lau et al., 2002; Prenger et al., 2004), but
criticized for the lack of biological interpretability and plausibility (Grossberg, 1987; Stork, 1989) and thus less popular despite their high expressive power. However, it should be mentioned that most neural encoding/decoding models have the same structure as Eq.(A.68); i.e., cascades of linear and nonlinear transformations (Hunter and Korenberg, 1986; Korenberg and Hunter, 1986, 1996). Because the dimension of input stimuli is typically huge, we often search for the subspace—or feature space—relevant for neural behavior on the basis of, e.g., orientation, contrast, spatial frequency or phase for visual stimuli (e.g., Enroth-Cugel and Robson, 1966; Movshon et al., 1978; Jones and Palmer, 1987; Tolhurst and Dean, 1990; Reid and Shapley, 1992; DeAngelis et al., 1993a,b; Ringach et al., 1997; Brenner et al., 2000; Touryan et al., 2002; David et al., 2004; Rust and Movshon, 2005), and frequency, spectrogram, or ripple domains for auditory stimuli (e.g., Klein et al., 2000; Theunissen et al., 2000, 2001; Depireux et al., 2001; Escabí and Schreiner, 2002; Machens et al., 2004). Such subspaces can then be interpreted as “tuning curves” and/or “receptive fields” of neurons of interest.

Models in neuroscience typically have a relatively small number of “hidden units,” and many feature extraction—or dimension reduction—techniques have been proposed and applied in sensory systems research, such as spike triggered average/covariance (or white-noise analysis; Bussgang, 1952; de Boer and Kuyper, 1968; Marmarelis and Naka, 1972; Marmarelis and Marmerelis, 1978; Aertsen and Johannesma, 1981; Eggermont et al., 1983; Eggermont, 1993; Ringach and Shapley, 2004; Schwartz et al., 2002; Simoncelli et al., 2004), and maximally informative dimensions (Paninski, 2003a; Sharpee et al., 2004, 2006). The fundamental idea here is to find a subspace that can best differentiate the stimulus distribution of interest—such as those evoked spikes $p(x|\text{spike})$—from the whole stimulus distribution $p(x)$, and the nonlinearities can then be recovered by a direct comparison between the two distributions using Bayes rule $p(\text{spike}|x) \propto p(x|\text{spike})/p(x)$; de Ruyter van Steveninck and Bialek, 1988; Pillow and Simoncelli, 2006), or by assuming some specific/judicious form of gain controls (e.g., divisive normalization; Schwartz and Simoncelli, 2001). However, it is generally very difficult to find out $a\ priori$ what form of nonlinear transformations neurons would perform (but
see e.g., Adelson and Bergen, 1985; Victor, 1992, 2005; Chichilnisky, 2001; Fishbach et al., 2001, 2003; Keat et al., 2001; Nykamp and Ringach, 2002; Herz et al., 2006)—even if target phenomena could be explained faithfully, we should not overinterpret any model unless it is verified to be (most likely) the case by enough experimental evidences. We should remember that many classes of models could exist that all work well but have nothing to do with biology, and that we could be deceived by such “wrong”—or nonbiological—models.

**A.3.3 Support Vector Regression**

Support vector machine (SVM)—a technique based on the statistical learning theory and the Vapnik-Chervonenkis (VC) dimension (Vapnik and Chervonenkis, 1971)—is a powerful tool for data classification (Boser et al., 1992; Vapnik, 1995), and can be applied for regression, maintaining the main advantages of the support vector methods (for tutorial, see Smola and Schölkopf, 2004). The basic idea is to map the inputs \( x \) into a high-dimensional space by nonlinear transformations \( f_r \), and perform linear regression in this feature space. Formally, we have:

\[
\hat{y} = \alpha_0 + \sum_{r=1}^{R} \alpha_r f_r(x). \tag{A.75}
\]

The goal in “epsilon-support vector regression (\( \varepsilon \)-SVR)” is then to find such \( \hat{y} \) that has at most \( \varepsilon \) deviation from the actual output \( y \) in feature space, but at the same time is as flat as possible.

To estimate parameters \( \alpha_r \) for \( r = 0, \ldots, R \), we thus minimize:

\[
E(\alpha) = \sum_{m=1}^{M} |y_m - \hat{y}_m|_\varepsilon + \lambda \cdot \frac{\|\alpha\|^2}{2}, \tag{A.76}
\]

where \( |\cdot|_\varepsilon \) is the so-called \( \varepsilon \)-insensitive loss function:

\[
|\xi|_\varepsilon \overset{\text{def}}{=} \max\{0, |\xi| - \varepsilon\} = \begin{cases} 
0 & \text{if } |\xi| \leq \varepsilon \\
|\xi| - \varepsilon & \text{otherwise.}
\end{cases} \tag{A.77}
\]
Note the similarity of the formulations between SVR (Eqs. (A.75) and (A.76)) and artificial neural networks (Eqs. (A.68) and (A.69)), and that flatness is ensured by minimizing \( \| \alpha \|^2 \), equivalent to maximizing the “margin” \( \sim 1/\| \alpha \| \). Also note that Eq. (A.76) has the form: “loss + regularizer” as in Eq. (A.52) on page 154 and can be understood from the regularization viewpoint (Smola et al., 1998), and that the loss function is not necessarily \( \varepsilon \)-insensitive but can be chosen arbitrarily, with the corresponding density models from a probabilistic viewpoint (Section A.2.2).

The beauty of the support vector methods is to solve the “dual” problem—instead of directly searching the solution of the “primal” problem as in Eq. (A.76)—by using the Lagrange multiplier; i.e.,

\[
\begin{align*}
\text{arg min}_{\beta^+, \beta^-} & - \frac{1}{2} (\beta^+ - \beta^-)^\top K (\beta^+ - \beta^-) - \varepsilon \sum_{m=1}^{M} (\beta^+_m + \beta^-_m) + \sum_{m=1}^{M} (\beta^+_m - \beta^-_m) y_m, \\
\text{subject to} & \sum_{m=1}^{M} (\beta^+_m - \beta^-_m) = 0, \quad \text{and} \quad 0 \leq \beta^+_m, \beta^-_m \leq 1/\lambda,
\end{align*}
\]

(A.78)

where the \((i, j)\)-element of the kernel matrix \( K \) is the inner product (between \( x_i \) and \( x_j \)) in feature space: \( K_{ij}(x_i, x_j) = \sum_r f_r(x_i)^r f_r(x_j) \). The dual problem in Eq. (A.78) can be solved by quadratic programming methods, and the regression estimate then takes the form:

\[
\hat{y} = \alpha_0 + \sum_{m=1}^{M} (\beta^+_m - \beta^-_m) K(x, x_m).
\]

\[(A.75')\]

Therefore, we do not have to specify and work on the large set of (nonlinear) functions \( f_r \), but only the inner product kernel \( K_{ij} \) needs to be evaluated (“kernel trick;” Mercer, 1909).

Common choices of the kernels include Gaussian: \( K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \) with \( \gamma > 0 \)—resulting in equivalent model structure as radial basis function (RBF) network—and sigmoids: \( K(x_i, x_j) = \tanh(-\gamma x_i^\top x_j + \gamma_0) \), with positive \( \gamma \), resulting in multilayer perceptrons. From the neural network viewpoint, only those weights from hidden to output layers are modifiable in SVR, and the connection pattern from input to hidden layers is auto-
matically determined, depending on the choice of the kernels. Nevertheless, SVR generally has high expressive power because of the nonlinear projection of inputs—specified by the choice of the kernel—to a high-dimensional feature space (i.e., with a large number of hidden units or support vectors), while at the same time it avoids the “curse of dimensionality” (i.e., the loss of generality due to having a high-dimensional input or feature space; Bellman, 1961) and maintains a high generalization performance by selecting as small number of support vectors as possible.

An appropriate choice of loss functions and regularizers is in most cases critical for the application of SVR in practice. A new parameter $\nu$ can be introduced to choose $\varepsilon$ and effectively control the fraction of support vectors and errors ($\nu$-SVR; Schölkopf et al., 2000)—by imposing an additional cost: $\nu\varepsilon$ on the primal problem in Eq.(A.76), or constraint: $\sum_{m=1}^{M} (\beta_m^+ + \beta_m^-) \leq \nu/\lambda$ on the dual problem in Eq.(A.78)—but there is no “free lunch” for these parameter selection procedures, so serendipity and trial-and-error tuning are often required.

### A.4 Information Theory

This section overviews the basics and logics of information theory, a powerful tool often used for examining the quality of stimulus-response models in neuroscience (for review, see e.g., Cover and Thomas, 1991; Rieke et al., 1997; Borst and Theunissen, 1999; Paninski, 2003b). I will discuss the connection between correlation functions and entropy/information in Section A.4.1, and a way to compute information by exploiting the SVD and the Fourier transform in Section A.4.2. Finally, Section A.4.3 shows a little survey on entropy estimation by exploiting file compression methods.
A.4.1 Entropy

Entropy is a measure of the randomness—or unpredictability—of observed variables, or the degree of information that the observation gives on the variables (Shannon, 1948). The entropy $H$ of a discrete set of symbols $z = \{z_1, \ldots, z_M\}$ with associated probabilities $p_m$ (for $m = 1, \ldots, M$) is defined as:

$$H(z) \overset{\text{def}}{=} -\sum_m p_m \log_2 p_m = \mathbb{E}[-\log_2 p_m] \quad \text{bits.}$$  \hspace{1cm} (A.79)

Similarly, the differential entropy for $M$ continuous variables or vector $x \in \mathbb{R}^M$ with (Riemann integrable) density $p(x)$ is defined as:

$$H(x) \overset{\text{def}}{=} -\int p(x) \log_2 p(x) \, dx = \mathbb{E}[-\log_2 p(x)] \quad \text{bits.}$$  \hspace{1cm} (A.80)

Note the relation between Eqs.(A.79) and (A.80). Let us consider the case with $M = 1$ in Eq.(A.80) for simplicity, where $x \in \mathbb{R}$ is a random variable with density $p(x)$. With the bin size of $\Delta$, the middle value of each bin $x_i$ approximately has the probability: $p_\Delta(x_i) = p(x_i) \Delta$. Thus the entropy of the quantized random variable $x_\Delta = \{x_i\}$ is given as, from Eq.(A.79),

$$H(x_\Delta) = -\sum_{i=-\infty}^{\infty} p_\Delta(x_i) \log_2 p_\Delta(x_i) \simeq -\log_2 \Delta - \sum_{i=-\infty}^{\infty} p(x_i) \Delta \log_2 p(z_i) \quad \text{bits.}$$  \hspace{1cm} (A.81)

From Eqs.(A.80) and (A.81), we then have:

$$H(x_\Delta) + \log_2 \Delta \rightarrow H(x), \quad \text{as} \quad \Delta \rightarrow 0.$$  \hspace{1cm} (A.82)

Also note that, given the number of symbols $M$, the maximum entropy discrete distribution is the uniform distribution: $H = \log_2 M$ bits from Eq.(A.79). For continuous distribution, it is instead the Gaussian distribution $x_{\text{gauss}} \in \mathbb{R}^M$ that has the maximum entropy given covariance $\Lambda$. 

167
When we discuss discrete functions of time, we can think of the correlation function as the analog of the covariance matrix. Hence, in the case of a single Gaussian signal $x(t)$, we have:

$$x = \begin{pmatrix} x(1) \\ x(2) \\ \vdots \\ x(T) \end{pmatrix}, \quad \mathbf{\Lambda} = \mathbb{E}[\mathbf{x} \mathbf{x}^T] = \begin{pmatrix} C(0) & C(1) & \cdots & C(T-1) \\ C(-1) & C(0) & \cdots & C(T-2) \\ \vdots & \vdots & \ddots & \vdots \\ C(1-T) & C(2-T) & \cdots & C(0) \end{pmatrix} \quad (A.84)$$

where $C(\tau)$ is the auto-correlation of $x(t)$:

$$C(\tau) \overset{\text{def}}{=} \lim_{T \to \infty} \frac{1}{T - |\tau|} \sum_{t=1}^{T} x(t) x(t - \tau). \quad (A.85)$$

Note that $C(\tau) = C(-\tau)$ and thus the Toeplitz matrix $\mathbf{\Lambda}$ in Eq.(A.84) is in fact symmetric.

In the case of multiple Gaussian signals $x_m(t)$ for $m = 1, \ldots, M$, we can replace $\mathbf{\Lambda}$ in Eq.(A.83) with the following $M \times M$ block matrix:

$$\mathbf{\Lambda} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1M} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \mathbf{C}_{M2} & \cdots & \mathbf{C}_{MM} \end{pmatrix} \quad (A.86)$$

where $\mathbf{C}_{ij}$ is the $T \times T$ cross-correlation matrix—an analog of $\mathbf{\Lambda}$ in Eq.(A.84)—between $i$-th signal $x_i(t)$ and $j$-th signal $x_j(t)$. Note that $\mathbf{C}_{ij}^T = \mathbf{C}_{ji}$ and thus the block matrix $\mathbf{\Lambda}$ in Eq.(A.86)
is symmetric. Alternatively, we can first look at between-set covariances at time $\tau$:

$$
C(\tau) = \begin{pmatrix}
C_{11}(\tau) & C_{12}(\tau) & \cdots & C_{1M}(\tau) \\
C_{21}(\tau) & C_{22}(\tau) & \cdots & C_{2M}(\tau) \\
\vdots & \vdots & \ddots & \vdots \\
C_{M1}(\tau) & C_{M2}(\tau) & \cdots & C_{MM}(\tau)
\end{pmatrix},
$$  \hspace{1cm} (A.87)

where $C_{ij}(\tau)$ is the cross-correlation—an analog of $C(\tau)$ in Eq.(A.85)—between $x_i(t)$ and $x_j(t)$:

$$
C_{ij}(\tau) \overset{\text{def}}{=} \lim_{T \to \infty} \frac{1}{T - |\tau|} \sum_{t=1}^{T} x_i(t) x_j(t - \tau).
$$  \hspace{1cm} (A.88)

We have the covariance matrix $\Lambda$ as the following $T \times T$ block matrix:

$$
\Lambda = \begin{pmatrix}
C(0) & C(1) & \cdots & C(T-1) \\
C(-1) & C(0) & \cdots & C(T-2) \\
\vdots & \vdots & \ddots & \vdots \\
C(1-T) & C(2-T) & \cdots & C(0)
\end{pmatrix}.
$$  \hspace{1cm} (A.89)

Note the similarity to Eq.(A.84), and that $\Lambda$ in Eq.(A.89) is symmetric because $C(\tau) = C(-\tau)$.

### A.4.2 Mutual Information

While entropy measures uncertainty, “information” is defined as the difference of entropies, i.e., a reduction of uncertainty (Shannon, 1948; Cover and Thomas, 1991). In this way, information theory determines how much information about inputs $X$ is contained in the outputs $Y$, and can be used to calculate the rates of information transfer. Mutual information between $X$ and $Y$ is defined as:

$$
I(X, Y) \overset{\text{def}}{=} H(X) - H(X|Y),
$$  \hspace{1cm} (A.90)
where the entropy $H(X)$ represents the maximum information that could be encoded in the inputs, and $H(X|Y)$ is the conditional entropy of inputs $X$ given the outputs $Y$. Because mutual information is symmetric between $X$ and $Y$, we can also define $I(X,Y) = I(Y,X)$ as:

$$I(X,Y) \overset{\text{def}}{=} H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X), \quad (A.90')$$

where $H(X,Y)$ is the joint entropy of $X$ and $Y$. In the latter expression in Eq.(A.90’), the output entropy $H(Y)$ represents the maximal information that could be carried by the system, and $H(Y|X)$ is the entropy in the outputs given the inputs, or the system noise.

Estimation of mutual information is often very difficult because it is hard to measure the entropy of experimental data in practice. As a compromise, we often estimate its lower- and/or upper-bound by introducing some assumptions on the distribution (e.g., Gaussianity), and below I will explain some estimation methods commonly used in neuroscience (for details, see e.g., Rieke et al., 1997; Borst and Theunissen, 1999; Paninski, 2003b).

**Direct method and upper-bound estimate of mutual information**

The direct method calculates information by estimating $H(Y)$ and $H(Y|X)$ from sample data. This is done by separating outputs $Y$ into a deterministic part $Y_{\text{det}}$ and a random component by repeating the (same) inputs $X$ many times. Under the additive Gaussian noise assumption for example, $Y_{\text{det}}$ can be estimated as the average of $Y$. We can then calculate $I(Y, Y_{\text{det}})$, which gives an estimated upper-bound of $I(Y, X)$ if we further assume that $Y$ is also Gaussian.

**Lower-bound estimate of mutual information**

From the data processing inequality theorem, we have: $I(Y, X) \geq I(Y, \hat{Y})$, where $\hat{Y}$ is the estimated output of $Y$ from inputs $X$. If we define: $I_{\text{gauss}} = H(Y) - H(N_{\text{gauss}})$, where $N_{\text{gauss}}$ is the Gaussian process with the same dimension and covariance as the estimated noise $N =$
\( Y - \hat{Y}, \) then \( I(Y, \hat{Y}) \) is bounded below by:

\[
I(Y, \hat{Y}) = H(Y) - H(Y|\hat{Y}) = H(Y) - H(N) \geq H(Y) - H(N_{gauss}) = I_{gauss}.
\] (A.91)

The inequality holds because the Gaussian distribution has the maximum entropy given the covariance. From Eq.(A.83), under the Gaussian assumption on \( Y, \) an estimate of mutual information is given as:

\[
I_{gauss} = \frac{1}{2} \log_2 \frac{|\det \Lambda_Y|}{|\det \Lambda_N|}
\] (A.92)

where \( \Lambda_Y \) and \( \Lambda_N \) are the covariance matrices of the output \( Y \) and the noise \( N, \) respectively. Note that \( I_{gauss} \) gives the lower-bound of \( I(Y, X) \) if and only if \( Y \) is in fact Gaussian.

**Computation of information using SVD**

The most straightforward way to evaluate Eq.(A.92) is to measure the covariance matrices (as in Eq.(A.84) for a single channel, or in Eq.(A.86) or Eq.(A.89) for multiple channels) and to compute their eigenvalues, because \( |\det \Lambda| = \prod_i \lambda_i, \) where \( \lambda_i \in \mathbb{R} \) are the eigenvalues of \( \Lambda. \)

However, there are two major difficulties in directly computing \( |\det \Lambda| \) for large data sets: (1) it can be computationally expensive, and (2) it can require huge memory resources. One way to avoid these issues is to introduce a window to approximate the covariance matrices and exploit SVD (see Section Section A.1.2) to compute their eigenvalues.

Consider a single Gaussian signal \( x(t) \) for \( t = 1, \ldots, T \) whose covariance matrix \( \Lambda \) is given as in Eq.(A.84). Then the symmetric matrix \( \Lambda \) can be approximated as:

\[
\Lambda \approx XX^\top, \text{ where } X = \frac{1}{\sqrt{T}} \begin{pmatrix} x(1) & x(2) & \cdots & x(T) \\ x(1) & \cdots & x(T-1) & x(T) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & x(1) & x(2) & \cdots & x(T) \end{pmatrix}.
\] (A.93)
By applying SVD to $X$, we have the spectral decomposition of $XX^\top$:

$$X = V\Sigma U^\top, \quad \text{and thus} \quad XX^\top = V\Sigma^2 V^\top,$$  

(A.94)

where we follow the notations in Eq.(A.23) on page 140. Although $XX^\top$ in Eq.(A.93) is a biased estimate of $\Lambda$ for the off-diagonals, we can then easily compute the determinant of the covariance matrix $\Lambda$ as:

$$|\det \Lambda| = |\det V\Sigma^2 V^\top| = \prod_i \sigma_i^2,$$  

(A.95)

where $\sigma_i$ are the singular values of $X$ and $\sigma_i^2$ correspond to the eigenvalues of $\Lambda$. In the case of multiple Gaussian signals $x_m(t)$ for $m = 1, \ldots, M$, we can evaluate Eq.(A.92) in a similar manner. That is, an $X$ that satisfies $\Lambda \approx XX^\top$ is given as the following block matrix:

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_M \end{pmatrix}$$  

(A.96)

where $X_i$ is the analog of $X$ in Eq.(A.93) for the $i$-th signal $x_i(t)$.

Although an efficient algorithm for the SVD has been provided elsewhere (e.g., svd function—with the “econ” option—in MATLAB), it would not necessarily be computationally preferable to apply SVD directly to $X$ because the $TM \times (2T - 1)$ matrix $X$ can be bigger than the $TM \times TM$ covariance matrix $\Lambda$ in the single channel case ($M = 1$). Instead, by assuming that there is no correlation between the signals far apart, we can introduce a window of length $K (\leq T)$ to approximate the covariance matrix $\Lambda \approx \bar{X}X^\top$, where $\bar{X}$ is the $K \times (T + K - 1)$ matrix corresponding to the upper-left corner—the first $K$ rows of $X$ in Eq.(A.93) in essence—of $X$ in Eq.(A.93) in the single channel case; in the multiple channel case, we have the $KM \times (T + K - 1)$ matrix $\bar{X}$ as an analog of Eq.(A.96). Note that having the window length $K$ results in the same approximation level as having the bin size $2\pi K/T$ for the analysis in the Fourier domain (see below).
Furthermore, we can (randomly) pick up \( L (\leq T - K + 1) \) samples (columns) to obtain an \( \bar{X} \) analog, resulting in the \( K \times L \) and \( KM \times L \) matrices in the single and multiple channel case, respectively. In the former case, we have:

\[
\frac{1}{\sqrt{L}} \left( \begin{array}{cccc}
  x(i_1 + K - 1) & x(i_2 + K - 1) & \cdots & x(i_L + K - 1) \\
  x(i_1 + K - 2) & x(i_2 + K - 2) & \cdots & x(i_L + K - 2) \\
  \vdots & \vdots & & \vdots \\
  x(i_1) & x(i_2) & \cdots & x(i_L)
\end{array} \right)
\]  
(A.97)

In this way, we can reasonably approximate the covariance matrices and readily evaluate Eq.(A.92) in the time domain.

**Computation of mutual information in the Fourier domain**

Although Eq.(A.92) holds in any orthonormal basis, it is in many cases evaluated in the Fourier domain under the assumption of stationary (time translation-invariant) ensembles. The main reason is that the covariance matrices in the Fourier domain are diagonal because the Fourier transform is an expansion using a set of orthogonal basis functions. Therefore, different frequency components can be thought of as independent variables, and the power spectrum measures the variances of these independent variables:

\[
\log_2 |\det \Lambda_Y| = \sum_{\omega} \log_2 P_Y(\omega), \quad \log_2 |\det \Lambda_N| = \sum_{\omega} \log_2 P_N(\omega),
\]  
(A.98)

where \( P_Y(\omega) \) and \( P_N(\omega) \) are the power spectral densities of the outputs \( Y \) and the noise \( N \), respectively. The power spectral density can be obtained by the squared Fourier coefficients of the signals, or the Fourier transform of the auto-correlation function (that is, for large \( T \) in Eq.(A.84), the eigenvalues of \( \Lambda \) correspond to the power spectral density of \( x \); the Wiener-
Khintchine theorem). Then we have, from Eqs.(A.92) and (A.98),

$$I_\omega = \frac{1}{2} \log_2 \frac{P_Y(\omega)}{P_N(\omega)}, \quad I_{\text{gauss}} = \sum_\omega I_\omega = \sum_{\omega=0}^{f_c} \log_2 \frac{P_Y(\omega)}{P_N(\omega)} \text{ bits} \quad (A.99)$$

where $f_c$ is the Nyquist frequency and $I_\omega$ is the information at frequency $\omega$. Note the relation:

$$I_\omega = I_{2f_c-\omega}$$

In the case of multiple dynamic channels, it requires huge computational power and resources to evaluate Eq.(A.92) in the time domain using Eq.(A.86), because we need to consider the correlation between the channels as well. For example, in a linear decoding (reconstruction) model where neural response $r(t)$ is used to estimate input spectrogram $S(t,f)$, the covariance matrix of $S$ has $O(T^2 M^2)$ elements where $T$ is the sample size in time and $M$ is the number of frequency bands in $S$ or the number of channels—we introduced the SVD method to avoid this issue. In contrast, two-dimensional Fourier transform leads to a diagonal covariance matrix of the Fourier coefficients. Therefore, it is much easier to evaluate Eq.(A.92) in the Fourier domains, and the estimate of mutual information $I(S,r)$ under the Gaussian assumption is:

$$I_{\text{gauss}} = \sum_{t,m} I(\omega_t, \omega_m) = \sum_{t,m} \log_2 \frac{P_S(\omega_t, \omega_m)}{P_N(\omega_t, \omega_m)} \text{ bits}, \quad (A.100)$$

where $\omega_t$ and $\omega_m$ are the Fourier domains corresponding to time and frequency in the spectrogram, respectively. $I(\omega_t, \omega_m)$ is the information at $(\omega_t, \omega_m)$, and $P_S$ and $P_N$ are the squared Fourier coefficients (power) of the input spectrogram and the reconstruction noise, respectively.

**Signal-to-noise ratio and coherence function**

Several equivalent formulae for Eq.(A.99) can be derived in the linear system, using the signal-to-noise ratio (SNR) and coherence function (Gabbiani, 1996; Rieke et al., 1997). Let $Y(t)$ and $\hat{Y}(t)$ be the output and its estimate from inputs $X(t)$, respectively. Then the estimated noise is
given as: \( N(t) = Y(t) - \hat{Y}(t) \), and the SNR is defined as:

\[
\text{SNR}(\omega) = \frac{P_Y(\omega)}{P_N(\omega)} = \frac{P_Y(\omega)}{P_N(\omega)} - 1, \tag{A.101}
\]

where \( P_Y, P_{\hat{Y}}, \) and \( P_N \) are the power spectral densities of \( Y, \hat{Y}(t), \) and \( N(t) \), respectively. The Gaussian estimate of mutual information between \( X \) and \( Y \) can then be written as, from Eqs. (A.99) and (A.101),

\[
I_{\text{gauss}} = \sum_{\omega} \log_2 [1 + \text{SNR}(\omega)] \text{ bits.} \tag{A.99’}
\]

The SNR can also be defined as:

\[
\text{SNR}(\omega) = \frac{P_Y(\omega)}{P_{N_{\text{eff}}}(\omega)}, \tag{A.101’}
\]

where \( P_{N_{\text{eff}}} \) is the power spectral density of the effective noise \( N_{\text{eff}}(t) \) uncorrelated to the original output \( Y(t) \):

\[
\hat{Y}(t) = g(t) * (Y(t) + N_{\text{eff}}(t)). \tag{A.102}
\]

Here * denotes convolution, and the function \( g(t) \) is chosen so that the cross-correlation between \( Y(t) \) and \( N_{\text{eff}}(t) \) is equal to zero for any time \( t \). The Fourier transform of \( g(t) \) is then called the coherence function \( \tilde{g}(\omega) \), and the Gaussian estimate can be given as:

\[
I_{\text{gauss}} = -\sum_{\omega} \log_2 [1 - \tilde{g}(\omega)] \text{ bits.} \tag{A.99''}
\]

In a linear system, the coherence function between the inputs \( X(t) \) and the outputs \( Y(t) \) can be rewritten as:

\[
\tilde{g}(\omega) = \frac{|P_{XY}(\omega)|^2}{P_X(\omega)P_Y(\omega)} = \frac{\text{SNR}(\omega)}{1 + \text{SNR}(\omega)}, \tag{A.103}
\]

where \( P_{XY} \) is the Fourier transform of the cross-correlation between \( X(t) \) and \( Y(t) \), and \( P_X \) and \( P_Y \) are the power spectral densities of \( X(t) \) and \( Y(t) \), respectively.
### A.4.3 File Compression Methods

Entropy (mutual information) estimation has been widely investigated, and various methods have been proposed in the context of neuroscience: e.g., for estimating the entropy carried by a spike train (e.g., histogram method, Strong et al., 1998; vector space method, Victor, 2002; Lempel-Ziv complexity method, Amigó et al., 2004). The methods described above have also been widely used, but the Gaussian assumption does not hold in most cases and thus the estimates could be far different from the real values. Considering that the entropy is in some sense the coding length of random variables, i.e., the number of bits required to encode the data (the “source coding” theorem; Cover and Thomas, 1991), we could instead estimate the entropy of data by examining how well the data can be compressed.

Say that each sample point in raw data is encoded by \( K \) bits/sample. Then the original file size for a given data of length \( M \) is \( X_{\text{orig}} = MK \) bits. Suppose the actual entropy of the data is \( H \) bits/sample. Based on the source coding theorem, the maximally compressed file size for the data (\( X_{\text{comp}} \) bits) would then be asymptotically equivalent to \( HM \) bits. Thus the estimated entropy can be given as:

\[
\hat{H} = K \frac{X_{\text{comp}}}{X_{\text{orig}}} = \frac{X_{\text{comp}}}{M} \quad (\geq H) \quad \text{bits/sample.} \tag{A.104}
\]

For better entropy estimation, \( X_{\text{comp}} \) can be normalized by subtracting the compressed file size for zero entropy data of length \( M \).

In practice, there are a couple of points we should be aware of. First, there is no best compression algorithm in general and thus the entropy based on Eq.(A.104) would give overestimates. Nevertheless, the compression approach surely has an advantage because it takes temporal structures of data into account more directly than discretization approaches such as the direct method.

The second point is that higher data precision requires larger data size. To avoid this issue, floating-point data would be converted into integers by multiplying data with a constant.
\(\alpha/\beta\), which gives data precision of \(\beta/\alpha\) with the data range of \([-\beta, \beta]\). Note however that this quantization process is *lossy*.

**File compression algorithms**

To encode a given data with as small file size as possible, various lossless file compression methods have been developed for the past decades, such as the Run-Length method and the Huffman coding method (Huffman, 1952; for review, see e.g., Salomon, 2006). These methods are typically based on Shannon’s information entropy (Shannon, 1948), and here we examined the performance of several commercially available compressors. The **gzip** algorithm,\(^5\) which is widely used in the UNIX and Windows platforms, is based on the Deflate/Inflate compression algorithm—Lempel-Ziv algorithm (Ziv and Lempel, 1977, 1978) with a sliding dictionary method and the Huffman coding—and often shows a good performance. The **bzip2** algorithm\(^6\) uses the Burrows-Wheeler transform (BWT) algorithm (Burrows and Wheeler, 1994) and Huffman coding. The **bzip2** algorithm often outperforms the more conventional Lempel-Ziv (LZ77/LZ88 or LZW; Welch, 1984) algorithms, and approaches the performance of the prediction-by-partial-matching (PPM) family of statistical methods (Cleary and Witten, 1984). We also used the **rk archiver**\(^7\) which often shows a very good compression performance.\(^8\)

**Entropy estimation for Gaussian distribution**

To examine the performance of entropy estimation by the file compression method, we first applied it to an uncorrelated Gaussian distribution (white noise; Figure A.4A–B) with various variances as well as correlated Gaussian distributions (pink noise; Figure A.4C–E). Note that the entropy of a Gaussian distribution is given by Eq.(A.83) on page 168, as the sum of entropies for each Fourier component (see Eq.(A.98) on page 173). As is expected, the obtained

---

\(^{5}\)The latest distribution gzip 1.2.4 (as of July, 2007): [http://www.gzip.org/](http://www.gzip.org/)

\(^{6}\)The latest distribution bzip2 1.0.4 (as of July, 2007): [http://sources.redhat.com/bzip2/](http://sources.redhat.com/bzip2/)


\(^{8}\)For benchmark, see e.g., [http://www.maximumcompression.com/](http://www.maximumcompression.com/)
values overestimated the true entropy in all the tested examples, but bzip2 and rk performed better than gzip. The estimated entropy as in Eq.(A.104) by the three algorithms depended little on the data length (tested data length ranged from $1 \times 10^4$ up to $1 \times 10^7$; data not shown).

**Entropy estimation for natural sounds**

We then examined the performance of entropy estimation for natural sound fragments. All fragments were obtained from commercially available audio CDs, originally sampled at 44.1 kHz and resampled at 97.656 kHz; *The Diversity of Animal Sounds* and *Sounds of Neotropical Rainforest Mammals* (Cornell Laboratory of Ornithology, Ithaca, NY, USA). For comparison, the upper-bound entropy was also estimated as the sum of entropies for each Fourier component under the Gaussian assumption. Here we multiplied the data in 16-bit encodings by $\alpha/\beta = 1, 10, 100, 2^{15}/100, \text{and } 2^{16}/100$, and rounded to the nearest integers for quantization.

As shown in Figure A.4F, the estimated entropies by bzip2 and rk were consistently smaller than the upper-bound estimates calculated under the Gaussian assumption, whereas those by gzip were sometimes not. Considering that the file compression methods return somewhat overestimates, the estimated entropies should be closer to the real values than those obtained under the Gaussian assumption. This makes sense because natural sounds presumably have rich temporal and spectral structures and differ much from Gaussian distributions.

These results suggest some advantages of the file compression method where we have no assumption on the data structures. Because the compression method considers the temporal structure of data more directly than discretization approaches, it would potentially give a good estimate for natural sounds or natural images. Another advantage would be the ease of estimation. Since the performance depends on the compression algorithm, however, the entropy estimation would be slightly biased and often overestimated. It should also be noted that the file compression methods typically require a large data set to estimate the entropy, and that our survey here illustrates well the difficulties of estimating entropy in general.
Figure A.4: **Entropy estimation by file compression.** The compression level was chosen to be the best (switches: -9 for gzip and bzip2, and -mx for rk). (A–B) White noise of the data length $1 \times 10^7$ in 16 and 32 bit encodings. Relatively lower performances around every 8 bits (i.e., 1 byte) would be due to the specification of the file system. (C–E) Pink noise (data length; $1 \times 10^7$) in 16 bit encodings. The rk (green) and bzip2 (magenta) performed better than the gzip (cyan) for estimating entropy in all examples we tested; low-pass, high-pass (data not shown, but the results were very similar to those of low-pass filtered case), band-pass, and band-stop filtered noise. (F) Natural sounds (16 bit encodings; 16 seconds; sampling frequency, 97.656 kHz) with quantization factor $\alpha/\beta = 1, 10, 100, 2^{15}/100,$ and $2^{16}/100.$ In most cases rk gave the best (i.e., the smallest) estimates.
Bibliography


Descartes, R. (1637). Discours de la méthode.


Pascal, B. (1670). *Pensées*.


