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

[1]The 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 localization 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:

$$y(t) = \sum_{i=1}^{P} h_i(t) * x_i(t) = \sum_{i=1}^{P} \widetilde{x}_i(t), \tag{2.1}$$

where $*$ indicates convolution and $\widetilde{x}_i(t) = h_i(t) * x_i(t)$ is the i-th source in isolation following filtering; i.e., $x_i(t)$ is the i-th source measured in *source* space whereas $\widetilde{x}_i(t)$ 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] $\mathbf{Y} = \mathsf{TF}\{y(t)\}$, 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 TF such that $\mathbf{Y}$ is a real-valued matrix with column vectors, $\mathbf{y} \in \mathbb{R}^N$, denoting the short segments of the spectrogram in time—specifically, to constrain $\mathbf{Y}$ 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 trans-

[2] In this dissertation, **boldface** is used to indicate vectors and matrices (in lower- and upper-case letters, respectively).

formed mixture:

$$\mathbf{y} = \sum_{i=1}^{P} \mathbf{h}_i \bullet \mathbf{x}_i = \sum_{i=1}^{P} \widetilde{\mathbf{x}}_i, \tag{2.2}$$

where $\bullet$ indicates elementwise multiplication, $\mathbf{h}_i$ is the HRTF in the frequency domain, and $\mathbf{x}_i$ and $\widetilde{\mathbf{x}}_i$ 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 $\widetilde{\mathbf{x}}_i$ (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 $i = 1, \ldots, P$:

$$\widetilde{\mathbf{x}}_i = \sum_{j=1}^{R_i} \widetilde{\mathbf{d}}_{ij} c_{ij} = \widetilde{\mathbf{D}}_i \mathbf{c}_i, \tag{2.3}$$

where the j-th column of $\widetilde{\mathbf{D}}_i$ consists of neural feature $\widetilde{\mathbf{d}}_{ij}$ for the $\{ij\}$-th neuron, and the j-th element of $\mathbf{c}_i$ holds the corresponding neural activity c_{ij}. The "dictionary" $\widetilde{\mathbf{D}}_i$ 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 $\mathbf{y}$ is then given by:

$$\mathbf{y} = \sum_{i=1}^{P} \widetilde{\mathbf{x}}_i = \sum_{i=1}^{P} \sum_{j=1}^{R_i} \widetilde{\mathbf{d}}_{ij} c_{ij} = \left[\begin{array}{ccc} \widetilde{\mathbf{D}}_1 & \cdots & \widetilde{\mathbf{D}}_P \end{array} \right] \left[\begin{array}{c} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_P \end{array} \right] = \widetilde{\mathbf{D}} \mathbf{c}. \tag{2.4}$$

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 $\widetilde{\mathbf{D}}$ 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 $\widetilde{\mathbf{x}}_i$ in terms of dictionary sets $\widetilde{\mathbf{D}}_i$ and the corresponding coefficients $\mathbf{c}_i$, 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) $\widetilde{\mathbf{d}}_{ij}$ and the firing rate c_{ij} of a corresponding neuron.

The successful separation[3] of the sources $\widetilde{\mathbf{x}}_i$ from the mixture $\mathbf{y}$ using the dictionary methods (Eqs.(2.3) and (2.4)) requires:

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

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

[3]The underlying source $\mathbf{x}_i$ can then be obtained by the deconvolution, if necessary, for $\widetilde{\mathbf{x}}_i = \mathbf{h}_i \bullet \mathbf{x}_i$.

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 $\mathbf{x}_i$ can be expressed as a linear combination of (not necessarily orthogonal) basis elements $\mathbf{d}_j$:

$$\mathbf{x}_i = \sum_{j=1}^{\bar{R}} \mathbf{d}_j c_{ij} = \mathbf{D}\mathbf{c}_i, \tag{2.5}$$

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

$$\widetilde{\mathbf{d}}_{ij} = \mathbf{h}_i \bullet \mathbf{d}_j. \tag{2.6}$$

The basis elements $\mathbf{d}_j$ 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 $\widetilde{\mathbf{d}}_{ij}$ is then better thought

of as representing the hypothesis that an element $\mathbf{d}_j$ is present at position i. In the same way, neurons in the (primary) visual cortex can be thought of as representing the hypothesis ($\widetilde{\mathbf{d}}_{ij}$) that an oriented edge ($\mathbf{d}_j$) is present at a particular position (i) in the visual field. In other words, the elements $\mathbf{d}_j$ reflect the basic properties of neurons, which in turn reflects the statistical structure of stimulus, whereas the features $\widetilde{\mathbf{d}}_{ij}$ arise from the top-down modulation on the elements $\mathbf{d}_j$ according to the position information ($\mathbf{h}_i$) determined by other mechanisms (see also Section 2.4.4).

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

$$\begin{aligned}
\mathbf{y} &= \sum_{i=1}^{P} \mathbf{h}_i \bullet \mathbf{x}_i && \text{from (2.2)} \\
&= \sum_{i=1}^{P} \mathbf{h}_i \bullet \left(\sum_{j=1}^{\bar{R}} \mathbf{d}_j c_{ij} \right) && \text{from (2.5)} \\
&= \sum_{i=1}^{P} \sum_{j=1}^{\bar{R}} \widetilde{\mathbf{d}}_{ij} c_{ij} && \text{from (2.6)} \\
&= \widetilde{\mathbf{D}}\mathbf{c}. && (2.7)
\end{aligned}$$

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

(1) A set of dictionaries in source space $\mathbf{D}$ is learned from a training set of "unmixed" signals $\mathbf{x}_i$.

(2) Given a convolutional mixture $\mathbf{y}$ and position-dependent filters $\mathbf{h}_i$, 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 $\mathbf{d}_j$ associated with position i using Eq.(2.5).

Note that we no longer have to use "personalized" dictionaries $\widetilde{\mathbf{D}}_i$ for each source $\widetilde{\mathbf{x}}_i$ as in Eqs.(2.3) and (2.4) but could use any dictionary set $\mathbf{D}$ that captures the spectral correlations in the sources $\mathbf{x}_i$ 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 $\widetilde{\mathbf{D}}$ but reconstructing the signals using a pre-HRTF (source space) dictionary $\mathbf{D}$ (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 $\mathbf{c}$ could represent the stimulus $\mathbf{y}$ 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 $\mathbf{y}$, but to find a representation in which the underlying sources $\mathbf{x}_i$ 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,[4] 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 $\mathbf{y}$ using the *minimum number of spikes* (Figure 2.1C); formally, $\mathbf{c}$ should be optimized to minimize its L_1-norm: $\|\mathbf{c}\|_1 = \sum_{ij}|c_{ij}|$. Considering the balance between the sparseness prior and robustness to noise (or the accuracy of the fit), the neural representation $\mathbf{c}$ can be computed as:

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ subject to } \|\widetilde{\mathbf{D}}\mathbf{c} - \mathbf{y}\|_p \leq \beta \tag{2.8}$$

where β is proportional to the noise level and with $p = 1$, 2, or ∞. 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 linprog routine in the MATLAB Optimization Toolbox). The solutions presented in Section 2.3 all used $p = 1$. For this case, noise vectors $\mathbf{e}^+$ and $\mathbf{e}^-$ are introduced and Eq.(2.8) can then be rewritten in the standard form:

$$\begin{aligned} \mathbf{c}^+, \mathbf{c}^-, \mathbf{e}^+, \mathbf{e}^- &\geq 0 \\ \widetilde{\mathbf{D}}\mathbf{c}^+ - \widetilde{\mathbf{D}}\mathbf{c}^- + \mathbf{e}^+ - \mathbf{e}^- &= \mathbf{y} \\ \mathbf{1}^\top\mathbf{e}^+ + \mathbf{1}^\top\mathbf{e}^- &\leq \beta \end{aligned} \tag{2.8'}$$

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

[4] *Pluralitas non est ponenda sine necessitate* (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: $\|\mathbf{c}\|_0 = \sum_{ij} c_{ij}^0$ if we define $0^0 \stackrel{\text{def}}{=} 0$, 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:

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_2^2 \text{ subject to } \|\widetilde{\mathbf{D}}\mathbf{c} - \mathbf{y}\|_p \leq \beta. \tag{2.9}$$

The pseudoinverse $\widetilde{\mathbf{D}}^\dagger$ finds the solution $\mathbf{c}$ (for $\beta = 0$ and $p = 2$) that minimizes the L_2-norm, i.e., the squared neural activity: $\|\mathbf{c}\|_2^2 = \sum_{ij} c_{ij}^2$ (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.

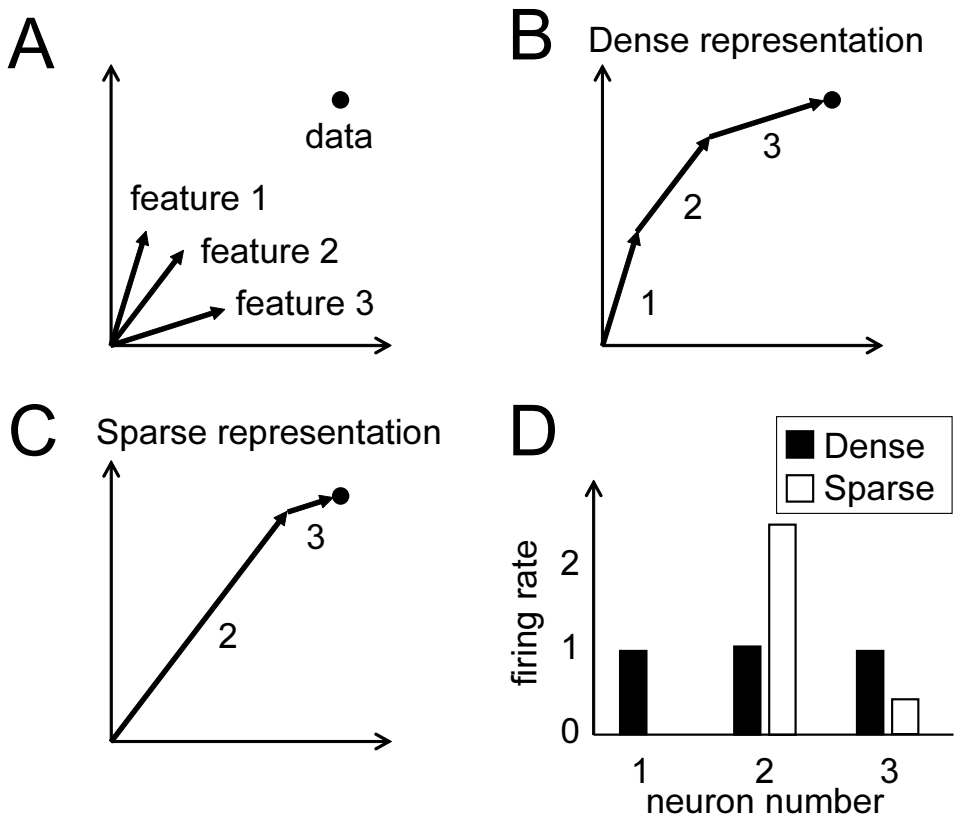

Figure 2.1: **Overcomplete representation in two dimensions**. **(A)** Three non-orthogonal basis vectors (neural features) $\widetilde{\mathbf{d}}_{ij} \in \mathbb{R}^2$ constitute an overcomplete representation, offering many possible ways to represent a data point $\mathbf{y}$ with no error. **(B)** 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): $\|\mathbf{c}\|_2^2 = \sum_{ij} c_{ij}^2$, 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: $\|\mathbf{c}\|_1 = \sum_{ij} |c_{ij}|$. **(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.

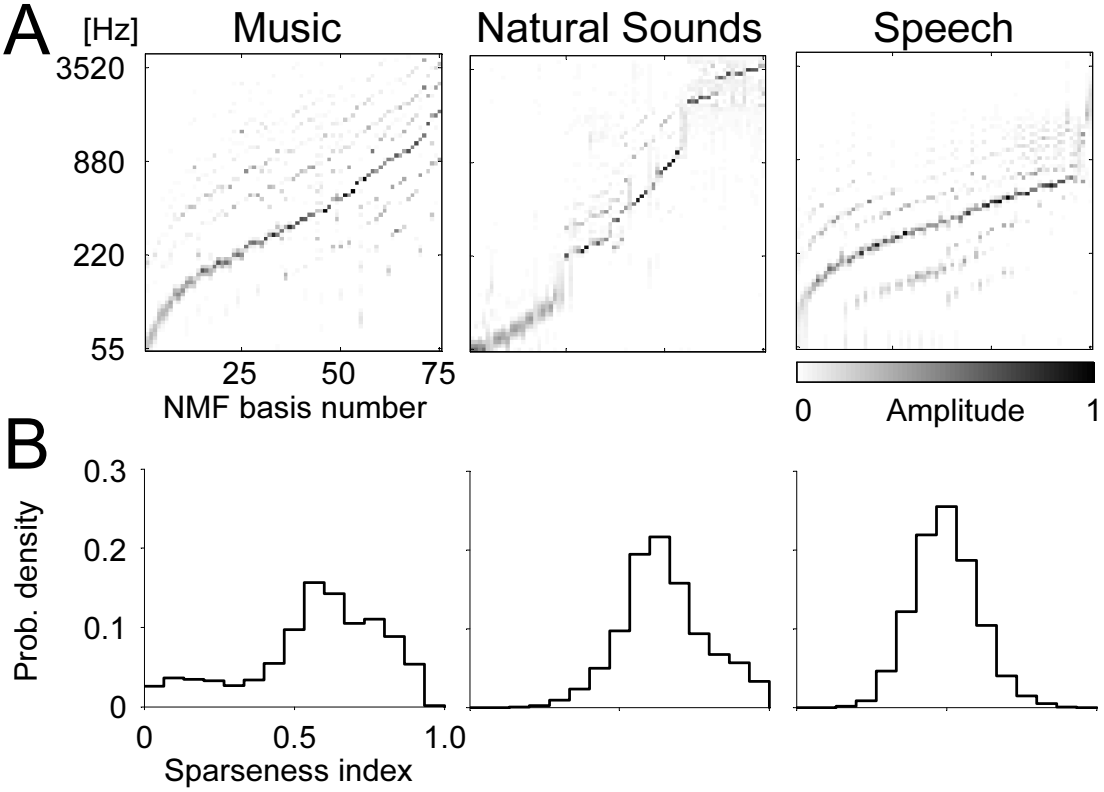

Figure 2.2: **Non-negative matrix factorization (NMF) can be used to find the parts of sound ensembles**. **(A)** NMF basis elements for three sound classes (music, natural sounds, and speech) were aligned in columns by the peak frequency. Power is concentrated in the fundamental frequency, but higher harmonics are clearly visible. Note that each column, which reflects statistical correlations present in the sources, is an example of $\mathbf{d}_j$ in Eq.(2.5); it is the filtered versions $\widetilde{\mathbf{d}}_{ij}$ 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: $\|\mathbf{c}_i\|_0 / \dim \mathbf{x}_i$, in the presence of a single (unmixed) source $\mathbf{x}_i$. Here this index is unity for a dense representation ($\dim \mathbf{c}_i / \dim \mathbf{x}_i = \bar{R}/N = 1$; see also Eq.(2.5)), and approaches zero as the representation becomes sparser. The distribution of the index was 0.61 ± 0.27, 0.64 ± 0.17, and 0.49 ± 0.13 (median $\pm$ 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 $\mathbf{X}$ (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 $\mathbf{X}$ 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 $\mathbf{A}$ obtained by NMF for each individual source were concatenated to form a (complete) source-space basis matrix of $\bar{R} = 5R = 75$ basis elements: $\mathbf{D} = \left[\mathbf{A}_1 \mid \mathbf{A}_2 \mid \cdots \right]$. Each column of $\mathbf{D}$ was then filtered through each of three different HRTFs ($P = 3$), resulting in a feature matrix with $P\bar{R} = 225$ columns: $\widetilde{\mathbf{D}} = \left[\mathbf{h}_1\mathbf{1}^\top \bullet \mathbf{D} \mid \cdots \mid \mathbf{h}_P\mathbf{1}^\top \bullet \mathbf{D} \right]$. 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 $-90°$, $0°$, and $90°$, 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 $\mathbf{X}$.

The ability of the NMF dictionaries to represent sounds can be quantified in terms of the "sparseness index" (Figure 2.2), defined as: $\|\mathbf{c}_i\|_0 / \dim \mathbf{x}_i \in \left(0, \bar{R}/N = 1\right]$, in the presence of a single (unmixed) source $\mathbf{x}_i$ (see also Eq.(2.5); in practice, $\|\mathbf{c}_i\|_0$ was computed with some tolerance, so the number of elements larger than 1×10^{-5}). The noise level was $\log_{10}\|\mathbf{y}\|_1/\beta = 1$ in Figure 2.2B, resulting in the reconstruction signal-to-noise ratio (SNR) of 18.3 ± 3.8, 16.0 ± 3.0, and 18.0 ± 3.6 (median $\pm$ 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 ± 0.27, 0.64 ± 0.17, and 0.49 ± 0.13 (mean $\pm$ 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: $\left\langle \|\mathbf{x}_i\|^2 / \|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|^2 \right\rangle_i$, where $\widehat{\mathbf{x}}_i$ is the estimate of $\mathbf{x}_i$, 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: $\widetilde{\mathbf{d}}_{ij} = \mathbf{h}_i \bullet \mathbf{d}_j$) 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 $\mathbf{d}_j$ in each panel (corresponding to the three filters $\mathbf{h}_i$). 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.

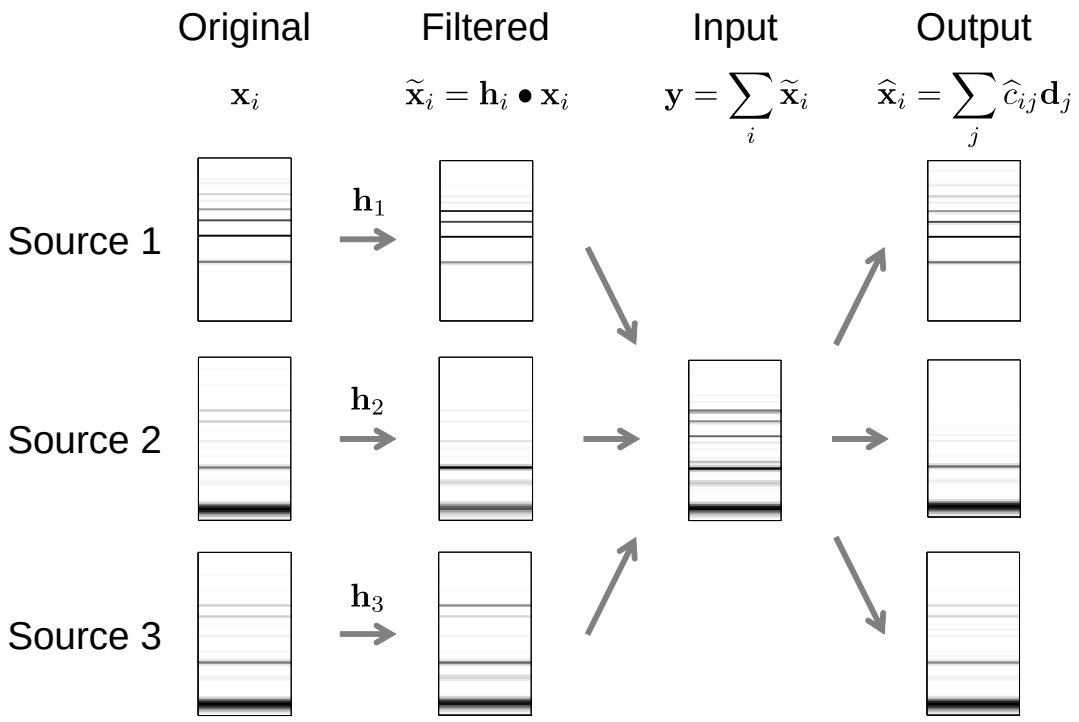

Figure 2.3: **Separation of three musical sources**. Three musical instruments $\mathbf{x}_i$ at three distinct spatial locations were filtered (by $\mathbf{h}_1, \ldots, \mathbf{h}_3$, corresponding to the HRTFs for azimuth $-90°$, $0°$, and $90°$ with zero elevation, respectively) and summed to produce the *input*: $\mathbf{y} = \sum_i \mathbf{h}_i \bullet \mathbf{x}_i$, and then separated using a sparse overcomplete representation to produce the *output*: $\widehat{\mathbf{x}}_i = \sum_j \widehat{c}_{ij}\mathbf{d}_j$. 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.

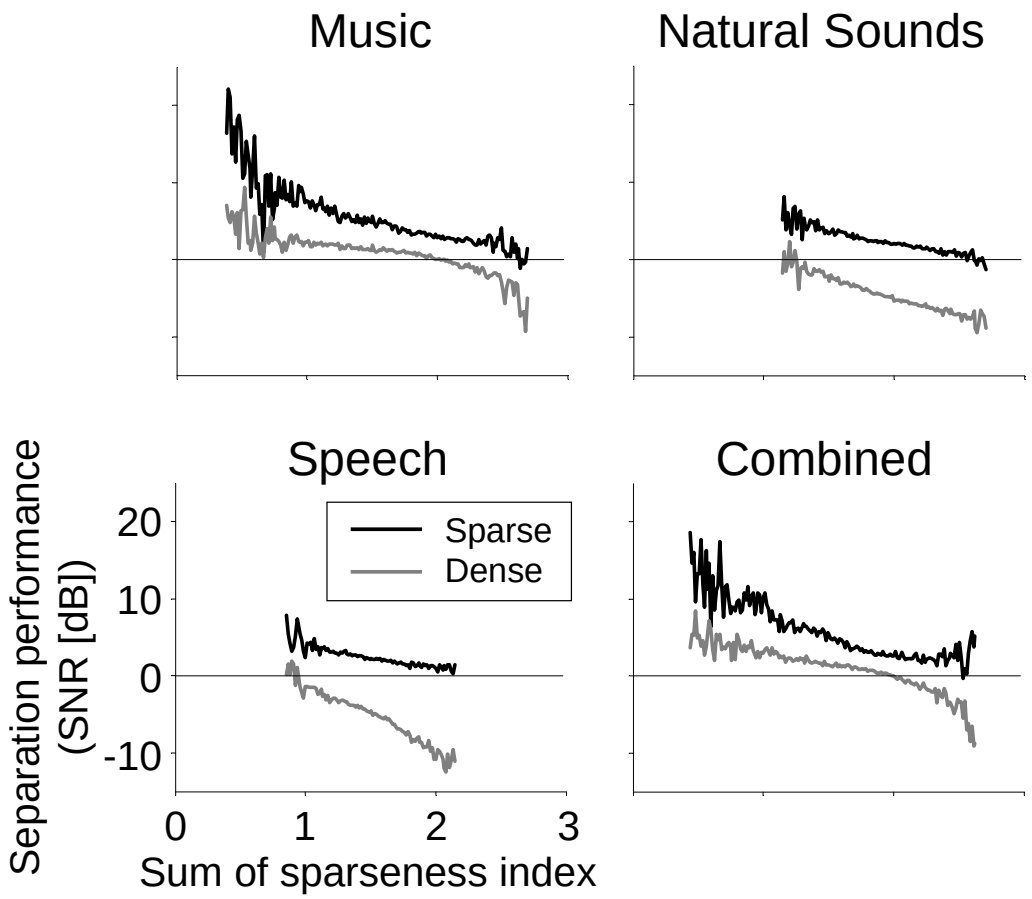

Figure 2.4: **Performance of different separation approaches with three sources**. The separation performance (SNR across sources) is shown as a function of the sum of the "sparseness index" of the three sources ($\sum_{i=1}^{P} \|\mathbf{c}_i\|_0 / \dim \mathbf{x}_i$ for $P = 3$, averaged over $20{,}000$ 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 $\sum_i \|\mathbf{c}_i\|_0 / \dim \mathbf{x}_i \leq 1$, 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.

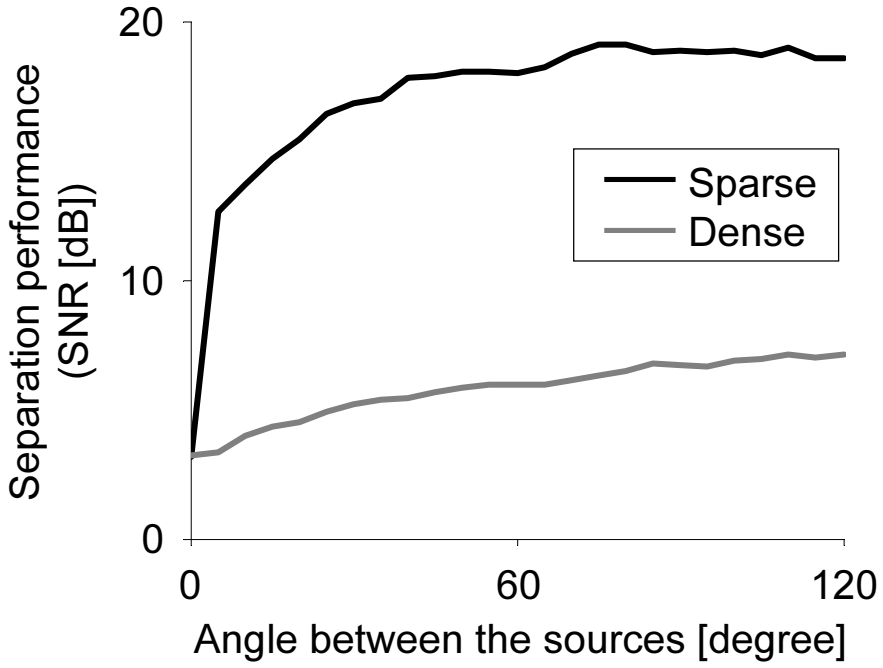

Figure 2.5: **Separation performance for different source locations**. Using a typical example of three novel stimuli (trumpet and two same harp; Figure 2.3), separation performance (vertical axis) was examined with all the possible combinations of the three sources (from 0 to 120 degrees apart in steps 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.

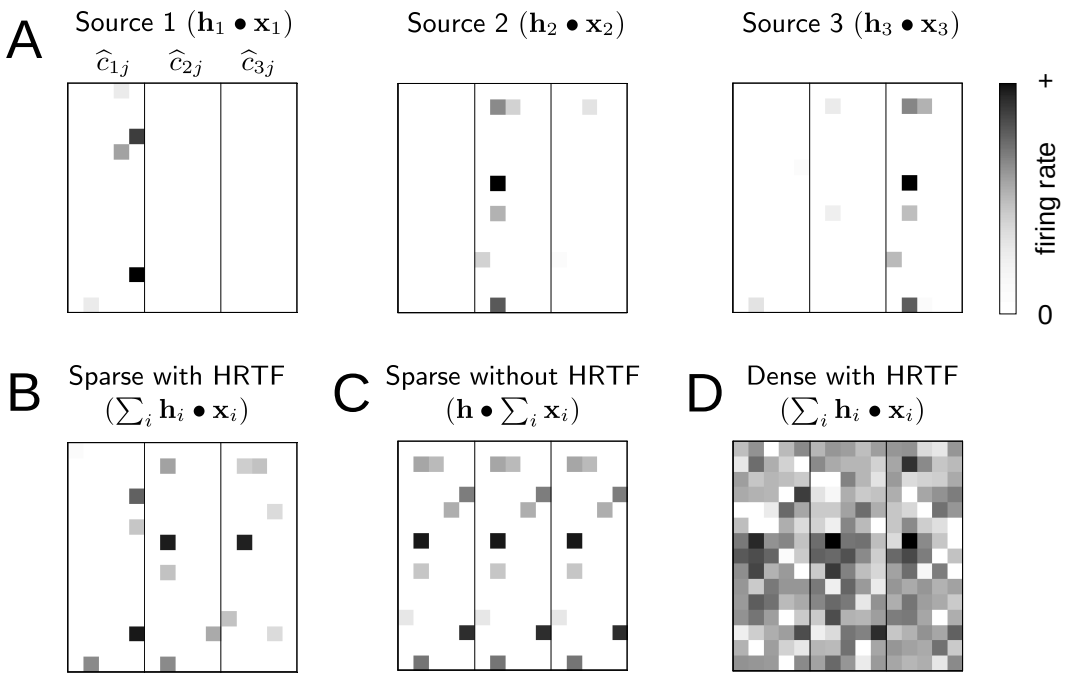

Figure 2.6: **Neural representations underlying source separation**. Each panel shows the activity of a population of 225 neurons, corresponding to the 225 features: $\widetilde{\mathbf{d}}_{ij} = \mathbf{h}_i \bullet \mathbf{d}_j$. The intensity of each dot in the 15×15 grid is proportional to the log of the firing rate of each neuron. Since the sources occupy three positions i, there are three copies of the basis $\mathbf{d}_j$ in each panel (corresponding to the three filters $\mathbf{h}_i$). The copies are arranged from left to right for convenience ($\widehat{c}_{1j}$, $\widehat{c}_{2j}$, and $\widehat{c}_{3j}$ 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 $\mathbf{y} = \widetilde{\mathbf{d}}_{ij}$. Since the feature $\widetilde{\mathbf{d}}_{ij}$ 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 $1{,}168\times3{,}600$ feature matrix $\widetilde{\mathbf{D}}$ 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.

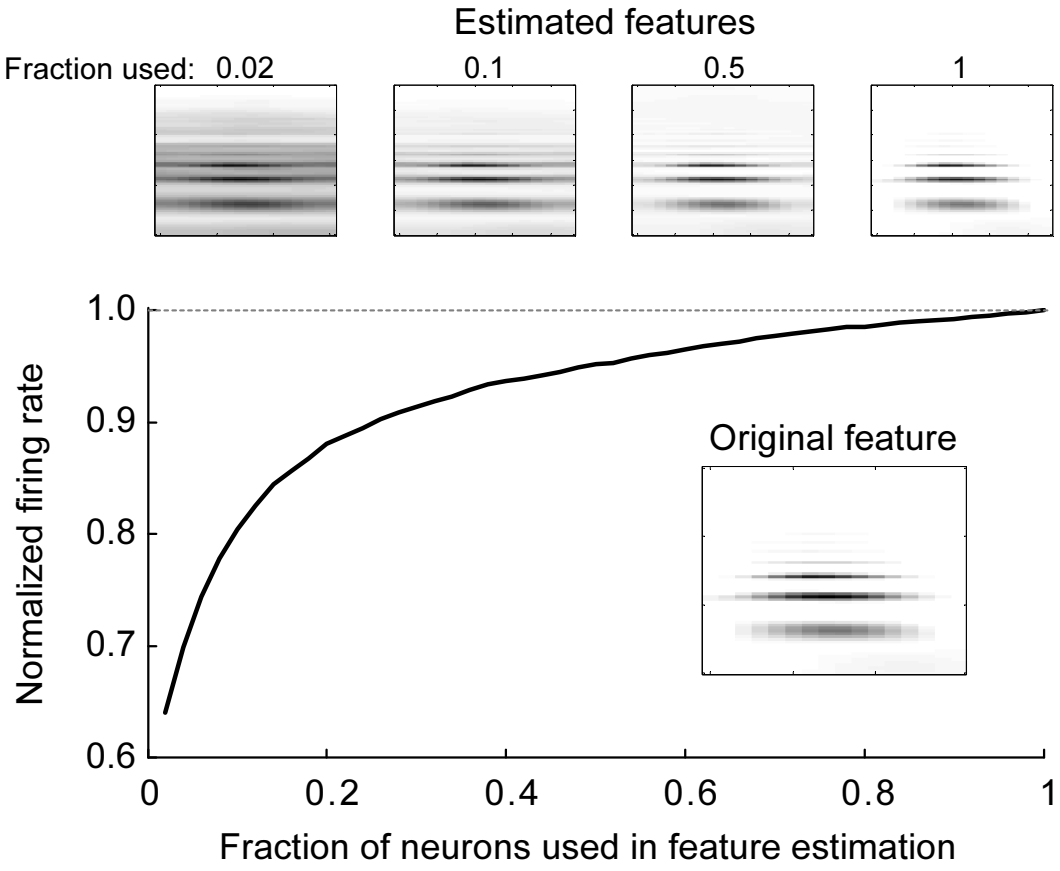

Figure 2.7: **Prediction 1: Stimulus optimization requires multi-neuron recording**. The vertical axis shows the simulated firing rate of a target neuron (normalized to its maximum firing rate) in response to the presentation of a stimulus corresponding to the optimal linear decoder constructed by recording 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 $\bar{\mathbf{D}}^{\dagger}$, where $\bar{\mathbf{D}}$ is a "packed matrix" whose columns are the subset of features involved in the sparse representation of a given stimulus $\mathbf{y} \in \mathbb{R}^N$, i.e., only the columns corresponding to the nonzero elements of $\mathbf{c}$. This matrix satisfies the decoding relation:

$$\mathbf{y} = \bar{\mathbf{D}}\bar{\mathbf{c}}, \tag{2.10}$$

where $\bar{\mathbf{c}}$ consists of $\mathbf{c}$ without zero elements. Because of the sparseness prior (L_1-norm minimization), however, the matrix $\bar{\mathbf{D}}$ 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:

$$\bar{\mathbf{c}} = \bar{\mathbf{D}}^{\dagger}\mathbf{y}. \tag{2.11}$$

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

The point in the prediction here is *not* in the piecewise linearity itself, but in the asymmetry between the encoding and decoding processes. In fact, any saturating nonlinearities must be introduced as a preprocessing in the model to make it more plausible; unlike physiology, doubling the stimulus $\mathbf{y}$ necessarily doubles the neural representation $\mathbf{c}$ in the current model; i.e., $\mathbf{y} = \widetilde{\mathbf{D}}\mathbf{c}$ in Eq.(2.7) implies $k\mathbf{y} = \widetilde{\mathbf{D}}(k\mathbf{c})$ 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 25×75 feature matrix $\widetilde{\mathbf{D}}$) were first generated randomly on the unit hypersphere. Neural activities $\mathbf{c}$ 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 $\mathbf{c}$ 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 $\mathbf{y}$ using the equation:

$$I(c, \mathbf{y}) = H(c) - H(c \,|\, \mathbf{y}) = H(c), \tag{2.12}$$

where $H(c)$ is the response entropy, and the conditional of the response given the stimulus satisfies: $H(c \,|\, \mathbf{y}) = 0$ 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 $\widehat{c}$ given the stimulus and the actual stimulus (*encoding*; $I(\widehat{c}, \mathbf{y})$); or between the optimal linear estimate of the stimulus $\widehat{\mathbf{y}}$ given the response and the actual response (*decoding*; $I(c, \widehat{\mathbf{y}})$). 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:

$$1 - \left\langle \frac{\|\text{reconstruction error}\|_2}{\|\text{response or signal}\|_2} \right\rangle = 1 - \left\langle \frac{1}{\text{SNR}} \right\rangle, \tag{2.13}$$

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.

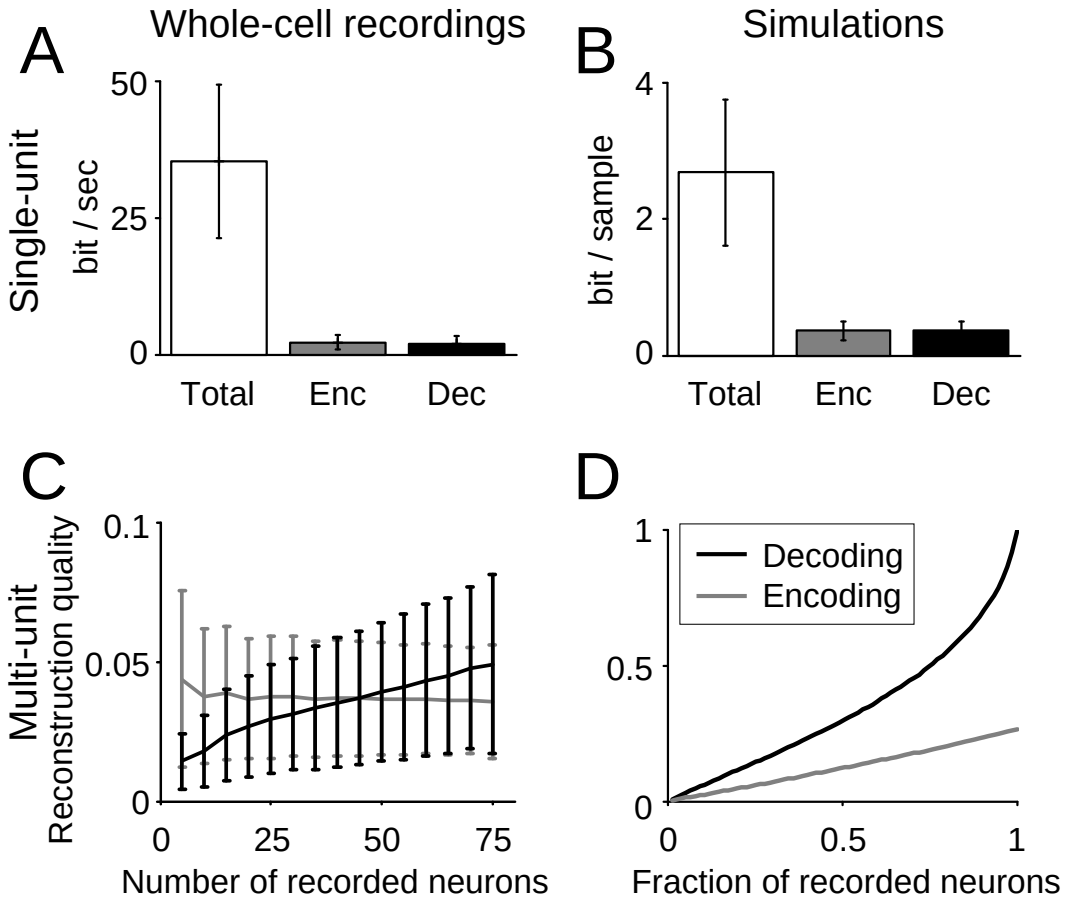

Figure 2.8: **Prediction 2: Asymmetry between linear decoding and linear encoding**. **(A, B)** The mutual information between the stimulus and the responses of single neurons (*Total*; Eq.(2.12); mean $\pm$ standard deviation) was compared with the mutual information captured by linear encoding (*Enc*) 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 $\pm$ 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.

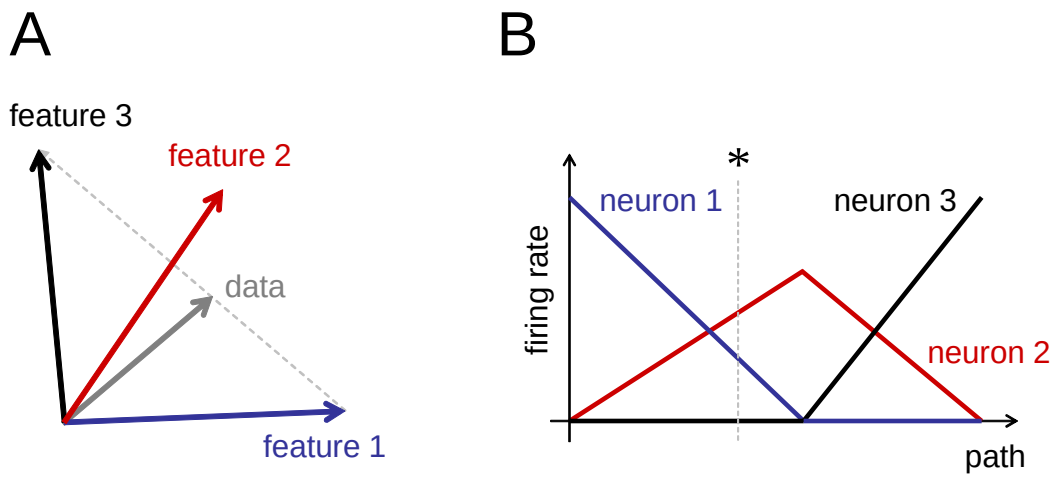

Figure 2.9: **Piecewise linear encoding**. **(A)** Three features (*blue*, *red*, and *black arrows*) in two dimensions constitute an overcomplete basis. A sample signal is indicated with a *gray arrow*. **(B)** Tuning curves for the three features are piecewise linear. The firing rate of each of the 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 respones 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

[5]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 $\widetilde{\mathbf{d}}_{ij}$ with respect to a stimulus $\mathbf{y}$ 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 $1{,}168\times3{,}600$ feature matrix $\mathbf{D}$ (the same one as in Figure 2.7), and used two different sets of 300 active features (i.e., $1{,}168\times300$ packed matrices $\bar{\mathbf{D}}$) 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.

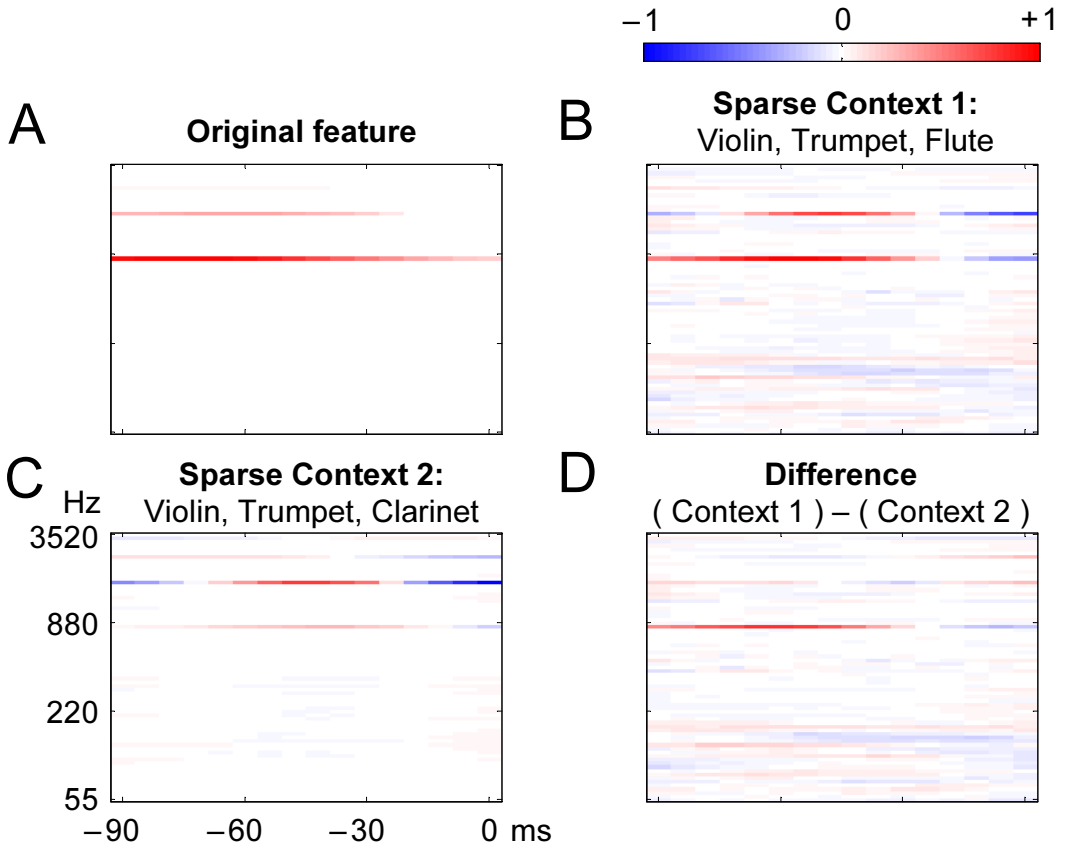

Figure 2.10: **Prediction 3: Dependence of STRF on stimulus context**. **(A)** Spectrogram of a trumpet feature, showing a strong fundamental around 880 Hz and some higher harmonics. **(B, C)** The STRFs corresponding to the feature in (A) when that feature is activated in two different contexts (clarinet 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 $\bar{D}^{\dagger}$ (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 $\widetilde{\mathbf{d}}_{ij}$ can be viewed as hypotheses about what a basis element $\mathbf{d}_j$ 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 $360^\circ/5^\circ = 72$-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.

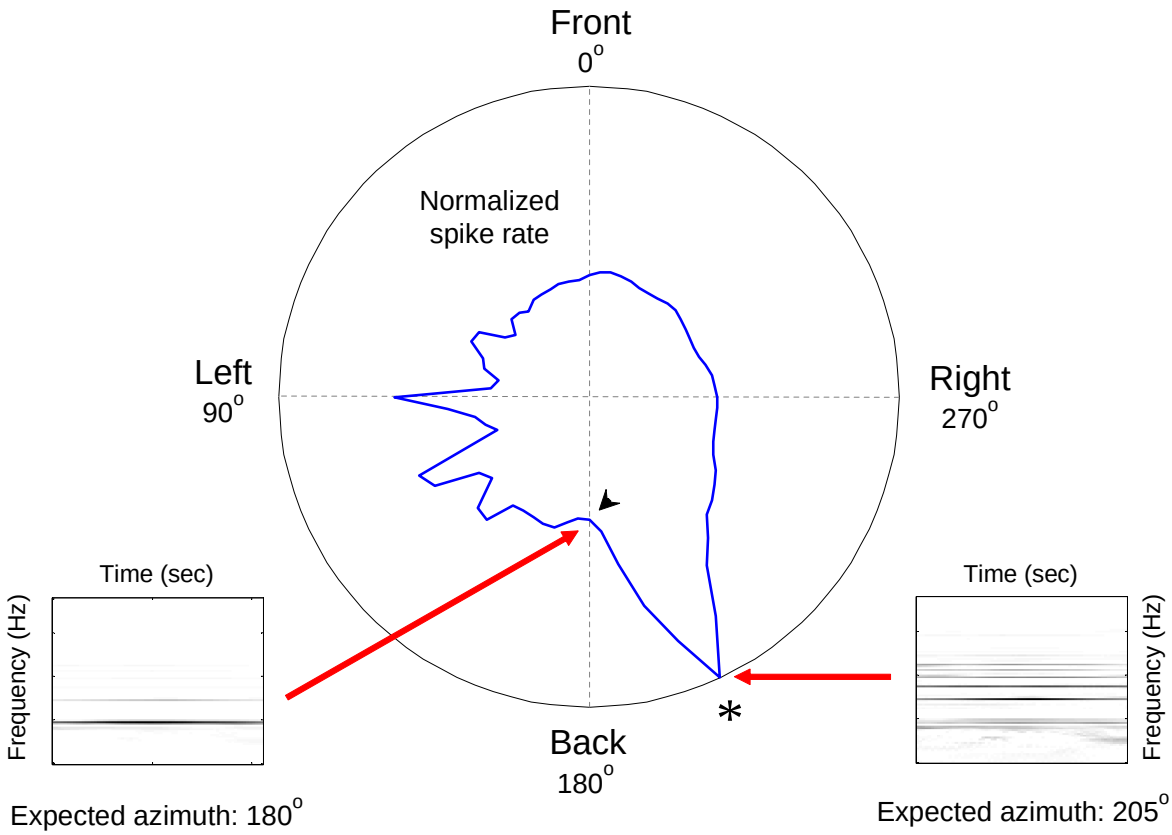

Figure 2.11: **Prediction 4: Top-down spatial expectation modulates neuronal tuning**. The *blue curve* shows the normalized activity of a model neuron tuned to a cello basis element $\mathbf{d}$ in response to the cello sound coming from $205°$ azimuth ("$*$"). When the expectation ($\mathbf{h}_{205°} \bullet \mathbf{d}$; spectrogram on the right) matches the actual sound, the response is maximal. When there is a mismatch between the actual and the expected sound ($\mathbf{h}_{180°} \bullet \mathbf{d}$; spectrogram on the left), however, the response is diminished (*arrowhead* at $180°$) 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 $\mathbf{c}$ for a given stimulus $\mathbf{y}$ is optimized to minimize its L_1-norm: $\|\mathbf{c}(\mathbf{y})\|_1$, or the total spike count. The triangle inequality gives: $\|\mathbf{c}(\sum_i \mathbf{y}_i)\|_1 \leq \sum_i \|\mathbf{c}(\mathbf{y}_i)\|_1$, 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: $\mathbf{c}(k\mathbf{y}) = k\mathbf{c}(\mathbf{y})$ 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).

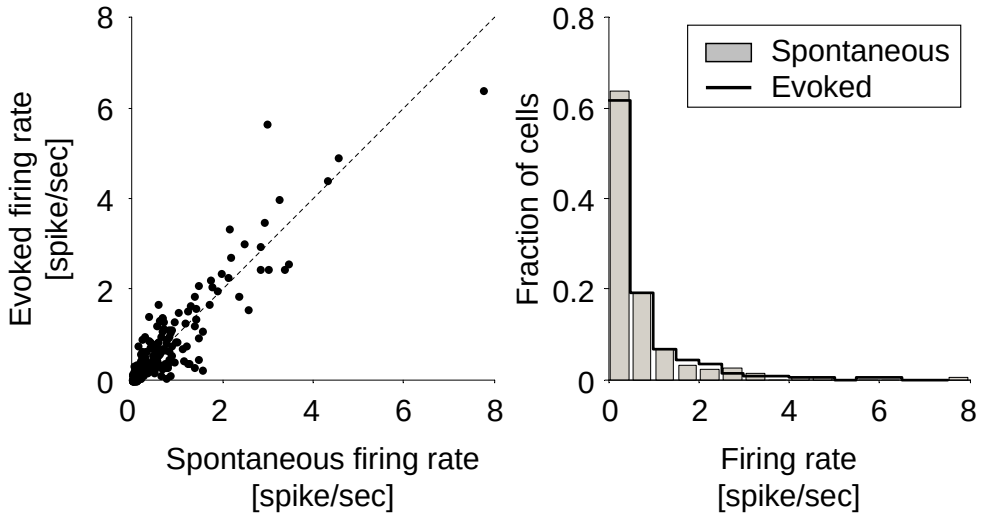

Figure 2.12: **Sparseness of responses in the primary auditory cortex**. The spontaneous firing rate was 0.25 ± 0.65 Hz (median $\pm$ interquartile range; 194 cells recorded by *in vivo* whole-cell patch-clamp techniques; see Section 3.2 in Chapter 3 for experimental details), whereas the firing rate evoked by natural sounds was 0.33 ± 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: $\|\mathbf{c}\|_1$. For illustrative purposes, let us consider a situation in $\mathbb{R}^2$ where basis functions $\widetilde{\mathbf{d}}_m$ for $m = 1, \ldots, M$ are on the unit circle at angles α_m, and a data point $\mathbf{y}$ is located at angle $\theta \in [\alpha_m, \alpha_{m+1}]$, i.e.,

$$\alpha_1 < \cdots < \alpha_m \leq \theta \leq \alpha_{m+1} < \cdots < \alpha_M \ (< \alpha_1 + 2\pi), \tag{2.14}$$

$$\|\mathbf{y}\|_2 = \|\widetilde{\mathbf{d}}_m\|_2 = 1, \quad (\text{for all } m) \tag{2.15}$$

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

$$\begin{aligned} C_m(\theta) &= \sum_{m=1}^{M} |c_m| \;=\; c_m + c_{m+1} \\ &= \frac{\sin(\alpha_{m+1} - \theta)}{\sin(\alpha_{m+1} - \alpha_m)} + \frac{\sin(\theta - \alpha_m)}{\sin(\alpha_{m+1} - \alpha_m)} \\ &= \cos\left(\theta - \frac{\alpha_m + \alpha_{m+1}}{2}\right) \sec\left(\frac{\alpha_{m+1} - \alpha_m}{2}\right). \end{aligned} \tag{2.16}$$

Let $P_m(\theta)$ be the probability density of $\mathbf{y}$ 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:

$$\begin{aligned} \left\langle \sum_{m=1}^{M} \int_{\alpha_m}^{\alpha_{m+1}} P_m(\theta)\, C_m(\theta)\, d\theta \right\rangle &= \left\langle \sum_{m=1}^{M} \frac{1}{\pi} \tan\left(\frac{\alpha_{m+1} - \alpha_m}{2}\right) \right\rangle \\ &\approx \frac{M}{\pi} \tan\left(\frac{\pi}{M}\right) \\ &\to 1 \quad \left(\text{as } M \to \infty\right) \end{aligned} \tag{2.17}$$

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: $\|\mathbf{c}\|_1 = \|\mathbf{y}\|_2 = 1$, 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.[6] 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 $\|d\mathbf{y}\|$ is given on average as (Figure 2.13A):

$$\left\langle \frac{|dc_m|}{\|d\mathbf{y}\|} \right\rangle = \left\langle \csc\left(\alpha_{m+1} - \alpha_m\right) \right\rangle \approx \frac{M}{2\pi}, \tag{2.18}$$

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 $\|d\mathbf{y}\|$) 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 evolutional viewpoint, the anatomical organization of the brain also implies that the computational power

[6] From a practical viewpoint, this noise susceptibility probelm can be alleviated to some extent by choosing an appropriate noise level (β) 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): $y(t) = \sum_{ij} c_{ij}(t) * \widetilde{d}_{ij}(t)$, or even into a continuous form: $y(t) = \int c(u,t) \circ \widetilde{d}(u,t)\, du$, 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).

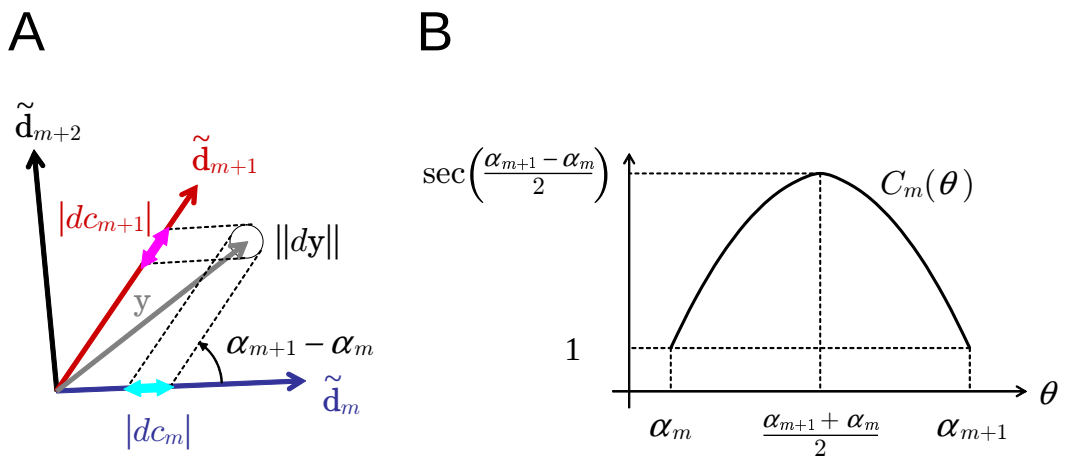

Figure 2.13: **Geometry of overcomplete representations in** $\mathbb{R}^2$. **(A)** Three features ($\tilde{\mathrm{d}}_i$ for $i = m$, $m+1$, and $m+2$) in $\mathbb{R}^2$ constitute an overcomplete basis. A sample signal $\mathbf{y}$ is located between the m-th and $(m+1)$-th features; i.e., at angle $\theta \in [\alpha_m, \alpha_{m+1}]$ in polar coordinate. The sensitivity of coefficients (*double-arrows*; $|dc_m|$) to perturbations on the signal (*black circle*; $\|d\mathbf{y}\|$) depends on the angle between the active features: $\alpha_{m+1} - \alpha_m$. **(B)** Coding cost: $C_m(\theta) = \|\mathbf{c}\|_1$ for data point $\mathbf{y} \in \mathbb{R}^2$ as a function of the angle θ; Eq.(2.16). Note the normalization: $\|\mathbf{y}\|_2 = 1$, in contrast to Figure 2.9.

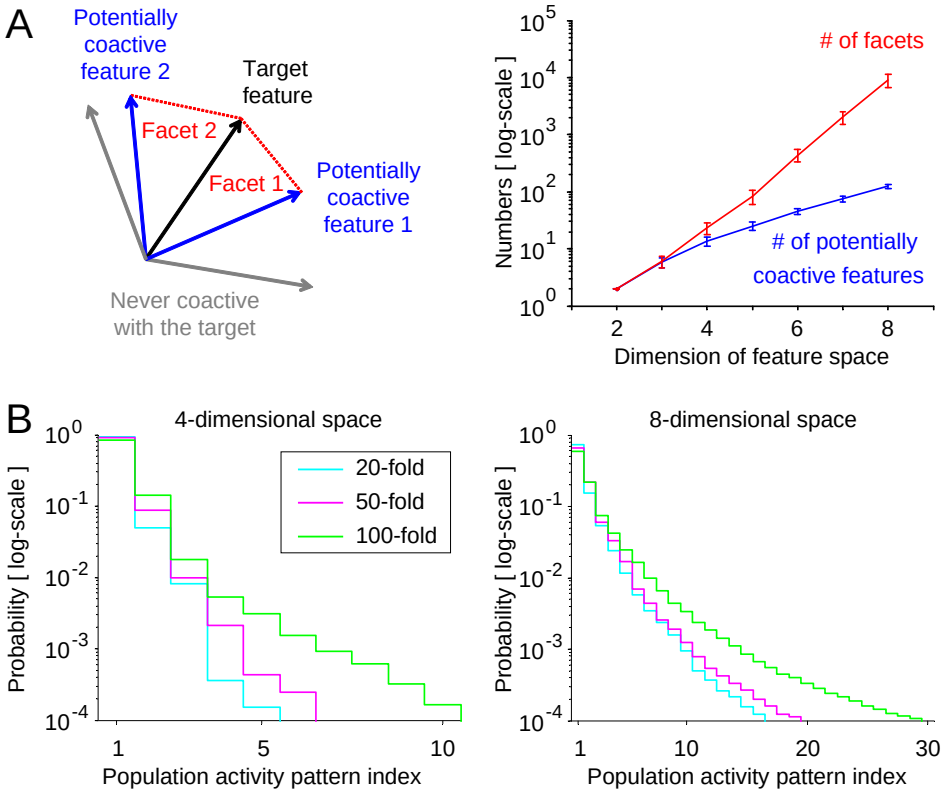

Figure 2.14: **Population variability in sparse overcomplete representations**. **(A)** The number of potentially coactive features and that of facets around a feature. *Left*: In $\mathbb{R}^2$ for example, the number of potentially coactive features (or "neurons;" *blue arrows*) with a target feature (*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 1×10^3 to 8×10^5) 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 $\mathbf{y}$, each examined over $10{,}000$ times with additive Gaussian noise: $\mathcal{N}[0, \|\mathbf{y}\|/100]$, 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)): $k\mathbf{y} = \widetilde{\mathbf{D}}(k\mathbf{c})$ for any scalar $k \in \mathbb{R}$. The objective in Eq.(2.8) then satisfies: $\|k\mathbf{c}\|_1 = |k| \, \|\mathbf{c}\|_1$, 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 $\mathbf{y}$ 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 $\mathbf{c}$ 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 ($\widetilde{\mathbf{d}}_{ij}$) 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 ($\mathbf{d}_j$), and they are differentiated by top-down modulations ($\mathbf{h}_i$) 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 overcomplete 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 estimated 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 Section 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 $\left(\mathbf{y}^{(t)}, c_1^{(t)}\right)$, 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 $\widehat{\mathbf{d}}_{m=1}$ can then be given by:

$$\text{minimize } E_{m=1} = \sum_t \mathcal{D}\left(\widehat{c}_{m=1}^{(t)}, c_1^{(t)}\right), \tag{2.19}$$

where

$$\widehat{\mathbf{c}}^{(t)} = \arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ subject to } \mathbf{y}^{(t)} = \widehat{\mathbf{D}}\mathbf{c}. \tag{2.20}$$

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

Least Squares

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

$$E_{\text{ls}} = \frac{1}{2}\sum_t \left(\widehat{c}_1^{(t)} - c_1^{(t)}\right)^2, \quad \text{and} \quad \nabla_{\widehat{\mathbf{d}}_m} E_{\text{ls}} = \sum_t \left(\widehat{c}_1^{(t)} - c_1^{(t)}\right) \nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}. \tag{2.21}$$

Using the "packed matrix" $\bar{\mathbf{D}}^{(t)}$ for the t-th input, we have: $\mathbf{y}^{(t)} = \bar{\mathbf{D}}^{(t)}\bar{\mathbf{c}}^{(t)}$ as in Eqs.(2.10) and (2.11) and thus the derivative $\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}$ in Eq.(2.21) can be approximated as:

$$\Delta\mathbf{y}^{(t)} = \bar{\mathbf{D}}^{(t)}\left(\Delta\bar{\mathbf{c}}^{(t)}\right) + \left(\Delta\bar{\mathbf{D}}^{(t)}\right)\bar{\mathbf{c}}^{(t)}, \quad \text{and} \quad \nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)} = \frac{\partial \widehat{c}_1^{(t)}}{\partial \widehat{\mathbf{d}}_m} \simeq -\widehat{c}_m^{(t)} \boldsymbol{d}_1^{(t)}, \tag{2.22}$$

where we assume: $\Delta\mathbf{y}^{(t)} = \mathbf{0}$, and $\left(\boldsymbol{d}_m^{(t)}\right)^\top$ is the m-th row of the pseudoinverse: $\left(\bar{\mathbf{D}}^{(t)}\right)^\dagger$.

Symmetrized Kullback-Leibler divergence

In least squares (Eqs.(2.21) and (2.22)), such signals have little effect on the gradient $\nabla_{\widehat{\mathbf{d}}_m} E_{\text{ls}}$ when $\widehat{c}_{m=1}^{(t)} = 0$, 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:

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

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

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

Optimization Algorithm

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

$$E_1 = E_{\text{ls}} + \gamma\, E_{\mathrm{KL}}, \tag{2.25}$$

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 $\widehat{\mathbf{d}}_m$ for $m = 1, \ldots, \widehat{M}$, we can write the iterative learning algorithm based on the gradient descent method as follows.

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

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 $\nabla_{\widehat{\mathbf{d}}_m} E_1$ by using Eqs.(2.21), (2.22) and (2.24), and update the estimates: $\widehat{\mathbf{d}}_m \leftarrow \mathsf{normal}\left(\widehat{\mathbf{d}}_m - \lambda \nabla_{\widehat{\mathbf{d}}_m} E_1\right)$.

Note that $\mathsf{normal}(\cdot)$ 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:

$$\mathbf{H}_m = \nabla^2_{\widehat{\mathbf{d}}_m}\big(E_{\text{ls}} + \gamma\, E_{\text{KL}}\big), \tag{2.26}$$

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

$$\nabla^2_{\widehat{\mathbf{d}}_m} E_{\text{ls}} = \sum_t \Bigg[\qquad \left(\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}\right)\left(\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}\right)^\top + \qquad \left(\widehat{c}_1^{(t)} - c_1^{(t)}\right) \qquad \nabla^2_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)} \Bigg], \tag{2.27}$$

$$\nabla^2_{\widehat{\mathbf{d}}_m} E_{\text{KL}} = \sum_t \left[\frac{c_1^{(t)} + \widehat{c}_1^{(t)}}{\big(\widehat{c}_1^{(t)}\big)^2} \left(\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}\right)\left(\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)}\right)^\top + \left(1 - \frac{c_1^{(t)}}{\widehat{c}_1^{(t)}} + \log \frac{\widehat{c}_1^{(t)}}{c_1^{(t)}}\right) \nabla^2_{\widehat{\mathbf{d}}_m} \widehat{c}_1^{(t)} \right]. \tag{2.28}$$

From Eq.(2.22), we also have:

$$\nabla^2_{\widehat{\mathbf{d}}_m} \widehat{c}_1 \;\simeq\; -\left(\nabla_{\widehat{\mathbf{d}}_m} \widehat{c}_1\right) \boldsymbol{d}_1^\top - \widehat{c}_m\, \nabla_{\widehat{\mathbf{d}}_m} \boldsymbol{d}_1^\top \;\simeq\; \widehat{c}_m \left(\boldsymbol{d}_m \boldsymbol{d}_1^\top + \boldsymbol{d}_1 \boldsymbol{d}_m^\top\right), \tag{2.29}$$

where the superscript (t) is omitted for brevity. Note that the variance of the estimates is given as the inverse of the Hessian matrix $\mathbf{H}_m$ (Eq.(2.26)).

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

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 $\mathbf{d}_m$ 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 $\mathbf{c}^{(t)}$ 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 $\widehat{\mathbf{d}}_1$ in the area where the target feature was active ($c_1^{(t)} > 0$), and the rest features $\widehat{\mathbf{d}}_{m\neq 1}$ surrounding the area in stimulus space. For simplicity, however, a set of features was randomly generated as an initial guess with $\widehat{M} = M$ 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.[7] Taking $N = 3$, $\widehat{M} = M = 20$, 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.

[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; $\mathbf{d}_1^\top \widehat{\mathbf{d}}_1 = 0.98 \pm 0.03$ 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, $\mathbf{d}_1^\top \widehat{\mathbf{d}}_1 = 0.99 \pm 0.01$, 0.91 ± 0.06, and 0.84 ± 0.08 over 10, 6, and 12 simulations, respectively; mean $\pm$ 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).

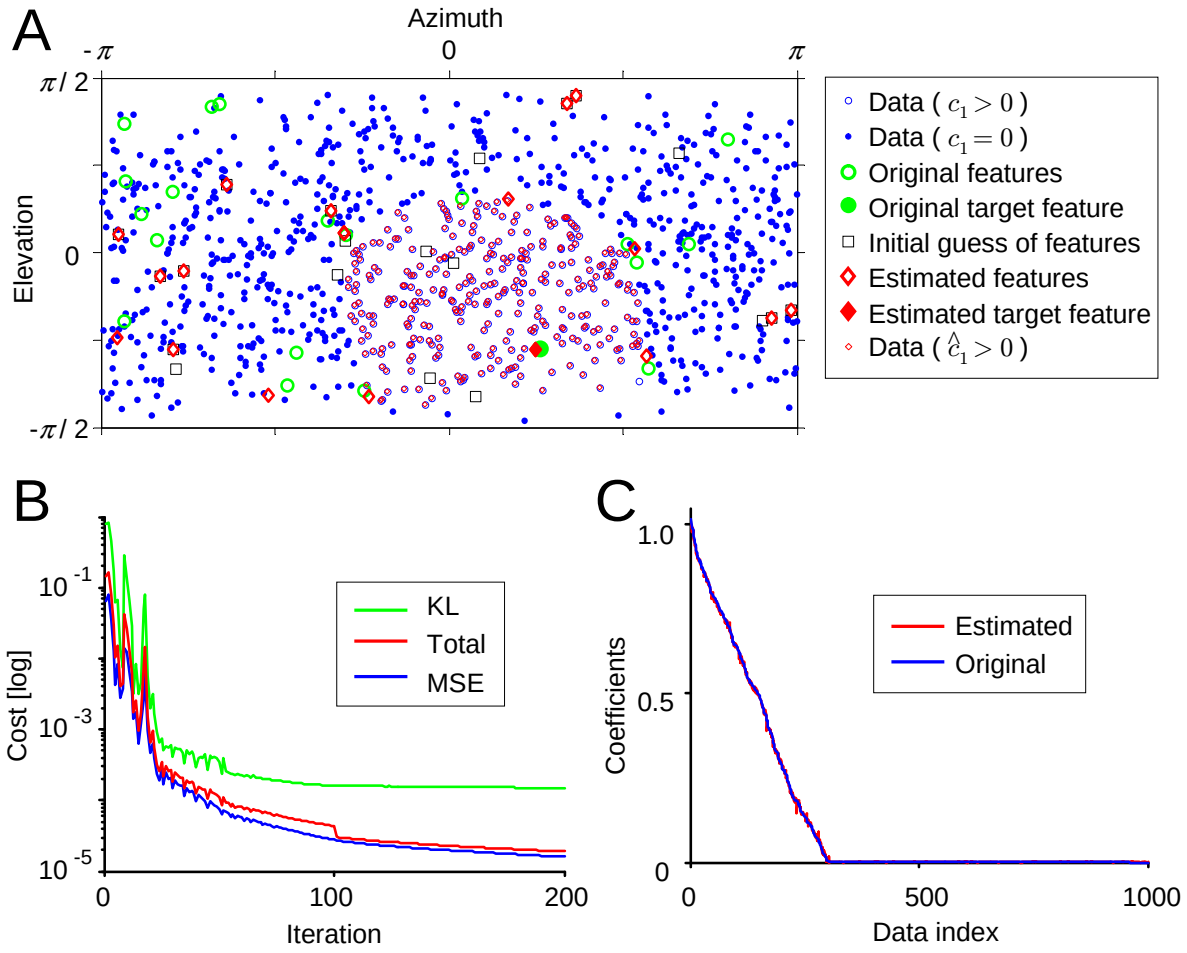

Figure 2.15: **Feature estimation in $\mathbb{R}^3$ from a simulated single-unit data set**. **(A)** Original features and estimated features in spherical coordinates ($N = 3$, $\widehat{M} = M = 20$, $T = 1{\times}10^3$). *Blue dots* and *small circles* show data ($c_1 = 0$ and $c_1 > 0$, respectively), while *small red open diamonds* are the data showing $\widehat{c}_1 > 0$. 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*; $E_{\rm ls}$ in Eq.(2.21)) and the symmetrized KL divergence (*KL*; $E_{\rm KL}$ in Eq.(2.23)) are shown in *blue* and *green*, respectively, and the total cost (*Total*; $E_{\rm ls} + \gamma\, E_{\rm KL}$ 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.

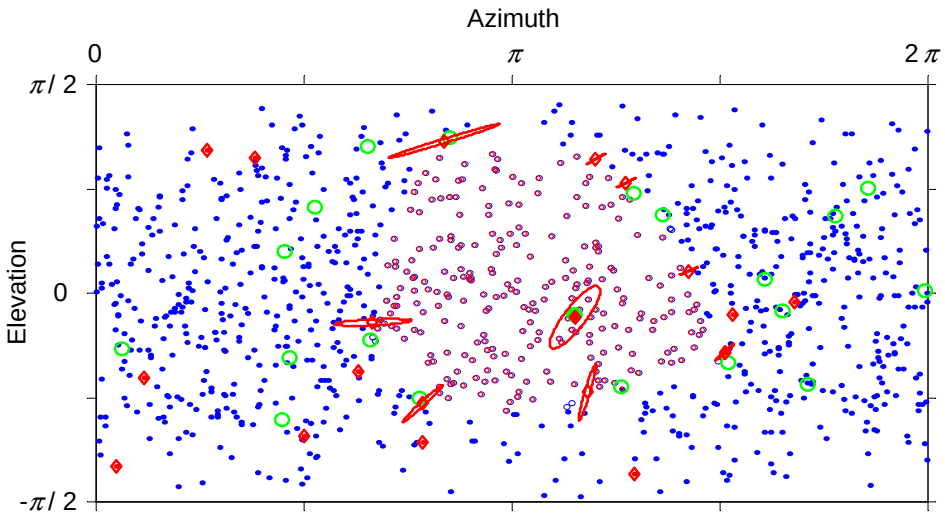

Figure 2.16: **Feature estimation and Fisher information matrices**. Another simulation result in $\mathbb{R}^3$ ($N = 3$, $\widehat{M} = M = 20$, $T = 1{\times}10^3$) was displayed in the same format as in Figure 2.15A. The *red ellipses* around estimated features (*red diamonds*) represent the Fisher 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).

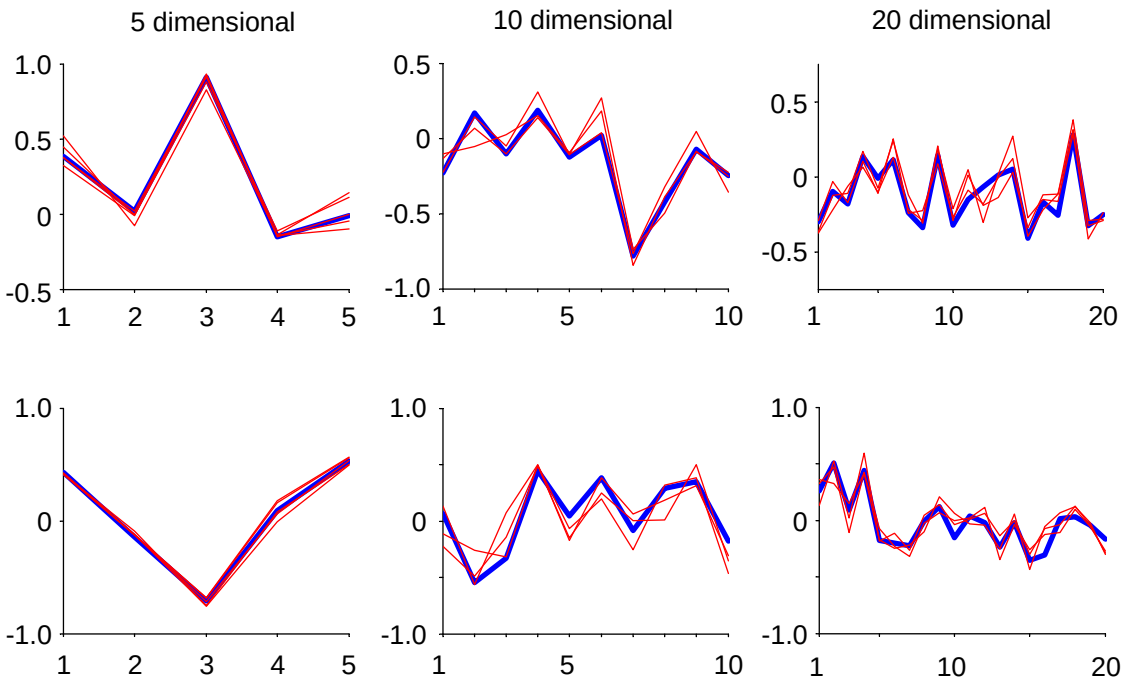

Figure 2.17: **Feature estimation for up to 20 dimension**. The *left*, *middle*, and *right* columns show the results for 5, 10, and 20 dimensional cases (two examples each), respectively, starting from several different initial conditions (5, 3, and 3 restarts, respectively). The *blue thick lines* and *red thin lines* show the coordinate values of the original and the estimated target feature, respectively. The parameters are $(N, \widehat{M}, M, T) = (5,\ 50,\ 50,\ 5{\times}10^3)$, $(10,\ 100,\ 100,\ 1{\times}10^4)$, $(20,\ 200,\ 200,\ 1{\times}10^4)$ 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 $\widehat{\mathbf{d}}_1$ is located in the area where the target is active, and the rest $\widehat{\mathbf{d}}_{m\neq 1}$ surrounding the area. Note that the estimated features $\widehat{\mathbf{d}}_{m\neq 1}$ far outside the area—i.e., those that are not coactive with $\widehat{\mathbf{d}}_1$—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:[8] $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$, 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.

[8]The Gamma function satifies $\Gamma(z) = z\Gamma(z-1)$ in general and is an extension of the factorial function: $\Gamma(N+1) = N!$ for all natural numbers N. Also note that $\Gamma(1) = 1$ 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.