# Hierarchical statistical models of computation in the visual cortex

Yan Karklin

CMU-CS-07-159

November 2007

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Thesis Committee:**

Michael S. Lewicki, *Chair*
Zoubin Ghahramani
Tai Sing Lee
Bruno Olshausen, U. C. Berkeley

*Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.*



This research was sponsored Department of Energy Computational Science Graduate Fellowship, the National Science Foundation under grant no. IIS-0413152, the Department of Health and Human Services under grant no. T32 NS07433-05, the Office of Naval Research under contract no. N00014-07-1-0747 and subcontract no. 1968767038469A. It was also supported by a generous fellowship from the Litton Corporation. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

**Keywords:** natural scene statistics, hierarchical statistical models, visual neuroscience

**Abstract**

How does the visual system form stable, coherent representations of image structure (edges, textures, shapes) from the coarse and noisy patterns of light collected at the retina? A common view is that neurons in the visual pathway act as feature detectors, with a hierarchy of increasingly complex features represented in V1, V2, and higher cortical areas. This approach has defined most experimental and modeling work to date (and inspired much computer vision research). However, it fails when applied to natural scenes, where object boundaries do not always produce clear edges, and surface properties like texture are defined by their intrinsic variability rather than fixed configurations of shapes. Models formulated in terms of feature processing also fail to account for a large number of subtle behaviors exhibited by neurons in the visual cortex.

In this dissertation we develop an alternative theory: rather than encoding preferred features, neurons describe entire distributions over their inputs, and thus capture the patterns of variability that underlie textures, contours, and other image elements. This allows the neural code to represent more abstract aspects of the image and remain invariant across fixations within local regions. We develop hierarchical models that implement this idea, and show that they yield better statistical descriptions of natural images than standard unsupervised learning techniques. The proposed models use distributed representations of image structure, a strategy likely employed in the brain. Although we do not fit the models to neural data, they exhibit a number of classical properties of "complex cells" in V1, as well as more subtle effects observed in V2 and V4. These results thus provide the first functional account for several previously unexplained neural behaviors. Finally, we demonstrate how model encoding of natural images can be used to analyze data from physiological experiments and predict neural responses to novel stimuli.

**Acknowledgments**

I owe the deepest gratitude to my advisor Mike Lewicki, whose insight, dedication, patience, and unfazed optimism will serve as a model for any scientific (or other) enterprise I undertake. His ability to see through the details to the important looming questions has been truly inspiring. I would also like to thank my parents for their unwavering support throughout my long academic journey.

My time in Pittsburgh was made pleasurable by many extraordinary people. I would like to acknowledge the company of my office compatriot Dan Bohus, whose intelligence and humor brightened the darkest office days. Finally, I have been very fortunate to know Arjun Rajeswaran, Anna Goldenberg, and Valérie and Kahlua Ventura. They have provided invaluable friendship and support during these years.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

The visual system is a remarkable mechanism. We perceive, navigate, and interact with a complex, noisy, and rapidly changing environment, our vision guided only by the coarse input of light registered on the retina. Crucial information that guides our decisions — the location of fruit on a tree, the speed of an approaching predator — is only implicitly represented in this signal. How are stable, coherent percepts of objects, surfaces, textures formed in the cortex? More specifically, what pieces of information are conveyed by the activity of individual neurons? How do neural populations jointly represent image structure? What computational principles are implemented by neural circuits?

A hundred years of experimental research have identified brain areas responsible for processing visual information, but the exact computations underlying this process remain largely unknown. The most direct approach is to probe the system experimentally, by measuring electro-physiological responses of visual neurons to the presented images. This method has served the primary role in mapping out processing in the early visual system. In the retina and in the lateral geniculate nucleus (the first stages of the visual pathway), the basic computational steps have largely been identified (though many phenomena are still unexplained). Neurons in these areas can be characterized by their responses to simple stimuli, and models can account for many of the response properties (see for example Carandini et al., 2005). However, further down the visual pathway, in cortical visual areas, neural activity signals increasingly more complex, abstract aspects of the visual input. Responses become more selective — cells might respond poorly to the majority of images presented in an experiment, and the response to simple stimuli is not necessarily related to processing of structurally rich, ecologically relevant, natural images. A proper characterization requires recording responses to all possible images; this is not feasible for single cells, let alone for entire populations of neurons. A common way to deal with this is to probe with a restricted set of images (e.g. gratings, bars, dots) in the hope that dimensions relevant to neural response are well sampled. This, of course, requires a priori knowledge about the type of image structure encoded by the neurons, and this approach can break down when our intuitions are misleading.

A more principled approach is to employ theoretical models, developed around fundamental computational goals, to direct experimental research and aid the interpretation of results. The aim is to identify the organizing principles behind the observed neural properties while making explicit the underlying assumptions. Theoretical models can make predictions regarding which dimensions in the stimulus space are important for neural response and should be explored; they can explain how an individual cell's activity fits within a population representation; and most importantly, they can propose functional roles that the observed response properties serve.

One theoretical approach that makes few assumptions is based on the *efficient coding* hypothesis. It holds that neural systems are adapted to the statistical structure of their sensory inputs, and have evolved to encode such inputs optimally. Therefore, by characterizing the statistics of the sensory environment, we can reveal organizing principles behind computation in the visual cortex, as well as constraints imposed by the specific nature of the input. Recent applications of this approach have been successful in accounting for properties of cells in the early cortex, but many experimental findings remain unexplained. At the same time, there is much room for further development of statistical models employed in this approach, and in the characterization of natural scene statistics, and it is the aim of this dissertation to address some of these limitations.

## 1.2 Organization of this dissertation

In the next chapter I give an overview of the basic physiology of the visual system, with emphasis on the cortical areas pertinent to the models developed in later chapters. Neural processing in the visual cortex has been studied extensively in the last several decades and a variety of complex response properties identified, but a coherent picture of neural function has yet to emerge. In the brief review below, I identify a number of phenomena for which no functional account exists, and which this dissertation attempts to address. Next I describe in more detail the hypothesis that neural systems are organized to efficiently process sensory information and show how early work has employed this principle to develop models of early visual processing. This work illustrates how the characterization of natural scene statistics can lead to specific predictions of neural function; this approach also forms the foundation for models developed in this dissertation. In the last section, I point out the limitations of existing models, both in terms of capturing the statistics of natural scenes and behavior of cortical visual neurons.

Chapter 3 describes a hierarchical statistical model that addresses some of these limitations. Trained on natural images, the model accounts for dependencies observed in linear models and automatically learns higher-order statistical regularities. Model representations capture more abstract properties of the scene and are more invariant over image regions containing similar structure. The model is general and can be applied to any high-dimensional, structured data, and we demonstrate this using analysis of speech waveforms.

Next I describe a model that extends this work to capture a wider range of statistical regularities, including arbitrary correlational patterns in the data. I relate the model to previous (mechanistic) models of cortical neurons and suggest how inference in the model might be approximated by neural circuits. I analyze the parameters of a model trained on image patches sampled from natural scenes and examine the way model representations generalize over similar types of images. This chapter also compares the proposed hierarchical models to standard generative models of natural images and shows that the hierarchical models yield better density estimates of the data, and thus can also be used to improve performance on statistical image restoration tasks.

Chapter 5 directly relates the proposed models to computation in the visual cortex. Units in the model are shown to exhibit classical response properties of complex cells, as well as a number of more subtle effects observed in V1 and V2. Preliminary results suggest parallels between a select set of model units and neurons in V4 that are broadly tuned for orientation or spatial frequency. In the last section I show how model encoding of images, together with neural data collected in physiological experiments, can be used to derive new descriptions of non-linear neurons and predict their responses to novel stimuli. Chapter 6 summarizes the contributions of this dissertation and offers concluding remarks regarding unresolved issues and directions for future research.

# Chapter 2

# Background

## 2.1 The visual pathway

Visual information is processed by a hierarchy of computational stages, starting in the retina and continuing through the lateral geniculate nucleus (LGN), to the primary visual cortex (also called the striate cortex), and then to higher level "extrastriate" visual areas (Fig. 2.1). In the cortex, a number of anatomically distinct areas contain neurons that respond selectively to visual stimulation. Anatomically defined areas V1 and V2 (which share many characteristics in humans and non-human primates) comprise the early stages of cortical processing. From here the pathway splits into what are typically called the ventral and the dorsal streams. While the ventral stream, proceeding to V4 and IT (inferior temporal cortex) is hypothesized to encode form and color, the dorsal stream passes through area MT (V5) and codes motion and spatial relationships. This description is somewhat cartoonish, since a complete picture includes feedback and mutual connections among practically all cortical areas (Felleman and Van Essen, 1991).

Although our understanding of early visual areas (the retina and the LGN) is far from complete, plausible computational functions have been identified (e.g. image decorrelation in the LGN, Dong and Atick, 1995; Dan et al., 1996) and neural responses characterized in a variety of synthetic and natural settings (Dan et al., 1996; Reinagel and Reid, 2000; Lesica and Stanley, 2004; Bonin et al., 2005). Cells in these areas encode single visual features in one location of the visual field (their *receptive field*), and their response is (approximately) a linear function of the strength of this feature in the stimulus. For example, the response of an LGN neuron can be modeled by convolving its preferred stimulus, a center-surround pattern of light and dark, with the image, and then passing the output through a non-linear rectifying and saturating function (Fig. 2.2a).

The first cortical area to receive visual input, the primary visual cortex (V1), contains a variety of neuron types, including cells originally described as "simple" and "complex" (Hubel and Wiesel, 1962, 1968). Simple cells typically prefer oriented, localized patterns and are also well modeled with a linear stage followed by a non-linearity (Fig. 2.2b). Stimuli that optimally drive V1 simple cells are well-fit by 2D Gabor functions (a 2D sinusoid multiplied by a Gaussian envelope; Jones and Palmer, 1987). Because of their strong response to bars and gratings, simple cells have been hypothesized to encode edges.

Complex cells also respond strongly to bars and edges, but are much more insensitive to the precise position of an edge in their receptive field, typically respond equally well to edges of opposite polarity, and when probed with sinusoidal gratings, are insensitive to the underlying phase. In a standard model of complex cells, the "energy" model, two localized and oriented features (typically 90° out of phase Gabor functions) are convolved with the image, and their outputs are squared and summed to give the neuron's response (see Fig. 2.2b; Movshon et al., 1978; Adelson and Bergen, 1985; Heeger, 1992; Heeger et al., 1996).

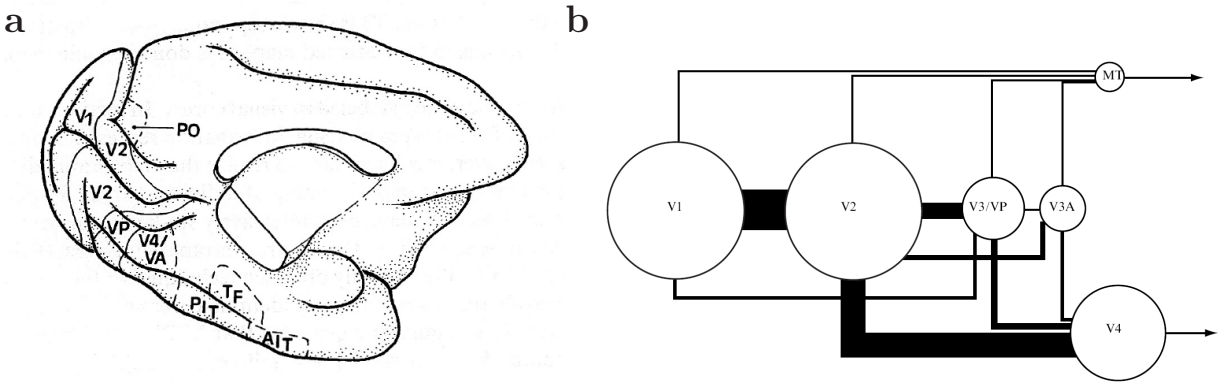


Figure 2.1: ***a***. Cortical visual areas of the macaque monkey (right hemisphere, anterior is to the right, reproduced from Maunsell and Newsome, 1987). ***b***. Relative sizes of and connections among visual areas in monkey occipital cortex (from Lennie, 1998).

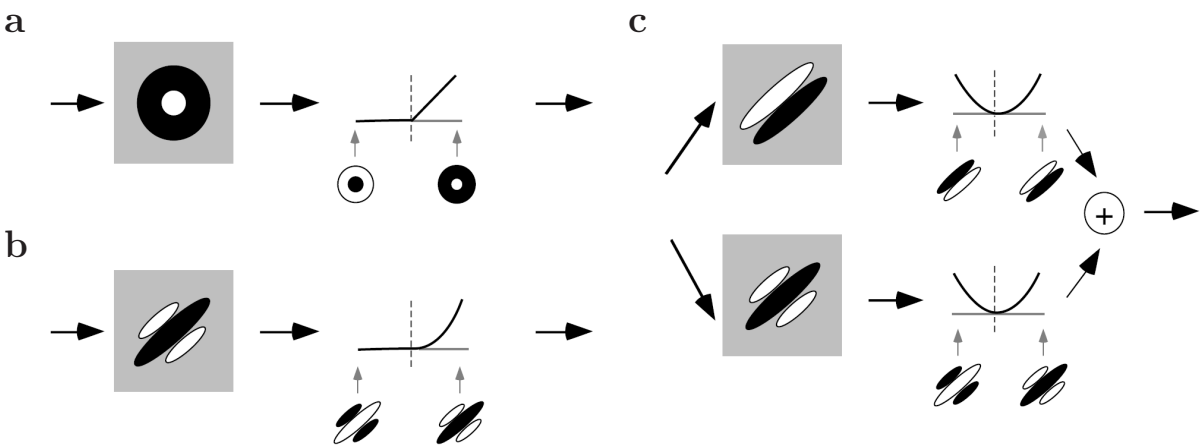


Figure 2.2: Standard models of early visual neurons (reproduced from Carandini et al., 2005). ***a***. Lateral geniculate nucleus neuron. ***b***. V1 simple cell. ***c***. V1 complex cell.

However, most complex cells are not perfectly phase invariant, and many simple cells have significant non-linear components, and it is more likely that the classical dichotomy of "simple" and "complex" simply covers the extrema of a spectrum of neurons of varied properties in V1 (Mechler and Ringach, 2002). Most V1 neurons also exhibit a variety of non-linear effects not captured by the standard models. These include suppression in response to image structure at orientations orthogonal to the "preferred" orientation (Sillito, 1975; Morrone et al., 1982; Bonds, 1989), as well as a number of effects dependent on image structure outside the classical receptive field (Heeger, 1992; Knierim and van Essen, 1992; Cavanaugh et al., 2002; Jones et al., 2002). To account for these effects, models have incorporated input from additional linear subfields and non-linear output stages. These models can accurately reproduce neural response to a range of stimuli (Carandini, 2004), though it is unclear how much of processing in V1 has been fully explained (Olshausen and Field, 2005), or how well such methods capture neural responses to complex dynamic stimuli (Smyth et al., 2003; Yen et al., 2007).

The classical account of visual processing holds that higher visual areas, such as V2 and V4, encode increasingly more complex shapes (Felleman and Van Essen, 1991; Lennie, 1998). Neurons in these areas are increasingly selective in their responses, and simple stimuli effective for characterizing simple or complex cells do not adequately drive many higher level neurons. Similarly, models designed around these features do a

poor job predicting responses to new categories of images or movies (David and Gallant, 2005). In order to investigate coding properties, experimentalists have had to rely on parameterized (and often idiosyncratic) synthetic stimuli designed to test specific hypotheses of neural coding. Cells in V2 largely mirror properties of V1, though their receptive fields are larger and they may be encoding more complex shapes than V1 neurons (Levitt et al., 1994; Hegdé and Van Essen, 2007). Because these areas are part of the ventral pathway that includes the high-level form sensitive areas such as the infra-parietal area, their possible coding of shape, contour, and texture has been explored. Many neurons in V2 and V4 seem to respond well to non-grating stimuli, such as polar or hyperbolic patterns (Gallant et al., 1993), angles (Ito and Komatsu, 2004), shapes (Hegdé and Van Essen, 2000), and curved contours (Pasupathy and Connor, 2001). However, these individual studies are difficult to reconcile, as they investigate different aspects of the neurons' representation of an image and their conclusions are not easy to interpret in the context of coding complex *natural* scenes. To further complicate this picture, recent work suggests that differences among these cortical areas are in fact quite subtle; for example, a large number of V1 neurons also have complicated response properties, and response properties across V1, V2, and V4 defy clear categorical segregation (Hegdé and Van Essen, 2007).

Early models of cortical neurons were developed around such experiments, attempting to account for the observed response properties to a limited set of stimuli — bars, gratings, and simple shapes — that best activated neurons and were consistent with experimenters' intuitions about the functional roles of these cells. More recently, methods have been developed for automatic discovery of linear subfields and identification of appropriate non-linearities by random sampling of the stimulus space to characterize neural response functions (de Boer and Kuyper, 1968; Jones and Palmer, 1987; de Ruyter van Steveninck and Bialek, 1988; Schwartz et al., 2006). They make few assumptions about the shape of the encoded image features and the non-linearities that define the neural response.

These methods have recovered receptive fields for simple cells consistent with earlier descriptions (Ringach, 2002), and have begun to uncover the detailed structure of non-linear components of higher-order neurons (Rust et al., 2005; Touryan et al., 2005). Because of computational constraints, this approach is only feasible when neurons have relatively simple response properties. Furthermore, the stimulus distribution must be random and obey certain properties (e.g. it must be spherically symmetric, Paninski, 2003). This is significant because it precludes (or at least introduces significant biases into) the use of these methods when studying responses to natural images, which are highly non-Gaussian and whose statistics are poorly understood (Rust and Movshon, 2005). On the other hand, random stimuli that are designed to evenly sample the input space and have regular statistics (e.g. white noise images or flashing checkerboard patterns) fail to elicit sufficient response in neurons in later stages of the visual hierarchy, which are increasingly selective in the images that drive them.

Clearly, progress is hampered by several unresolved issues. Fully random sampling of the input space is not a feasible approach; manual selection of stimuli is difficult when the computational goal of the area under investigation is unknown; in addition, the selection of a particular set brings with it an interpretation of neural coding (e.g. Fourier analysis, edge detection, shape coding) that is not necessarily appropriate. If the visual areas are encoding more abstract properties of the image, the neural activity will be invariant across a range of stimuli that satisfy some underlying property. Testing with individual stimuli might not reveal this invariance or uncover the encoded abstract property. At the same time, interactions between neurons can produce complex response patterns not easily explained with limited recording of individual neurons. All this suggests that computational theories, designed around specific hypotheses of neural function, are necessary to guide the exploration and interpretation of neural encoding of complex visual input.

Recent work has begun to address this problem, using theoretical models to make predictions about the features of individual cells, as well as those of neural populations. In the following section I describe one approach, based on the efficient coding hypothesis, which forms the foundation for models developed in this dissertation.

## 2.2 Efficient coding and the statistics of natural scenes

What computational principles underlie processing in the visual cortex? One idea that makes few assumptions about a sensory system's goals and mechanisms is that such a system should preserve information about its input while reducing the redundancy of the employed code (Attneave, 1954; Barlow, 1961; Simoncelli and Olshausen, 2001). This hypothesis, known as the *efficient coding theory*, is a general principle that can be applied to any sensory modality and input signals. Although this hypothesis has its limitations (e.g. it assigns equal value to all incoming information, some of which might be more or less important for an organism functioning in its environment), it serves as a sensible principle for the analysis of early stages that deal with raw input from sensory organs.

The efficiency of a sensory code depends on how well it is matched to the statistics of the messages it is used to transmit (Shannon and Weaver, 1949); in the context of neural coding, this implies that in order to be efficient, neural representations should be specifically adapted to the statistical structure of their input. For example, sensory input containing a specific pattern of correlations, if decorrelated by early stages of processing, can be transmitted more efficiently. This also means that representations that are optimal for one set of inputs, or a certain type of sensory environment, are not necessarily best for conveying other types of information. Therefore, by characterizing the statistics of the input to the visual system — natural images — we can characterize an optimal system, and use this description to gain insight into processing in biological vision.

Our sensory environment is rich with statistical structure. Images of outdoor scenes contain edges and shapes, textures and boundaries. The appearance and configuration of these elements produces statistical regularities in the sensory input that are very different from the statistics of random patterns. These patterns are distributed across visual space and interact in complex non-linear ways, so finding a compact description is not trivial. Nevertheless, efficient coding provides a statistical framework which naturally deals with many aspects of sensation in an uncertain world, such as the presence of noise, and inference of quantities implicitly represented in the signal.

### 2.2.1 Linear generative models

How do we describe the statistical structure of the visual world? To begin, we will disregard color and motion (surely important cues, but beyond the scope of this work) and focus on the analysis of static image structure – form, shape, texture, contours, borders. A grayscale image is an array of pixel intensities; this we can unroll into a vector of scalar values. Each image is then represented as a point in a vector space, and a collection of images from an ensemble (e.g.a "natural" set of photographs of outdoor scenes) comprises the distribution to model.

Images are rich with structure, but this structure is not obvious in the raw distribution of pixel intensities (Fig. 2.3a). Early work noted that the statistics of natural images are consistently different from those of random images. Adjacent pixels are correlated (Fig. 2.3b); on the whole, the second-order, correlational, structure follows a "$1/f$" distribution: the correlations among pixels are described by the power spectrum of the image, and the spectrum is characteristic in that amplitudes of frequencies ($f$) fall off roughly as $1/f$ (Tolhurst et al., 1992; Ruderman and Bialek, 1994).

However, a Gaussian model defined specifically by these second-order statistics does a poor job modeling natural image distributions. While data drawn from a multi-variate Gaussian, when projected onto a random vector will have a Gaussian histogram, such a projection of image data results in a more sparse distribution – more peaked at zero and heavier at the tails (Field, 1987; Daugman, 1989). A measure of peakiness is kurtosis, related to the fourth moment of the data and sometimes defined as $K(x) = E[(x-\mu)^4]/\sigma^4 - 3$ (where $\mu$ is the data mean and $\sigma$ the standard deviation). According to this definition, the kurtosis of Gaussian-distributed data is 0, with positive kurtosis indicating peakier distributions. Natural images tend to be significantly more

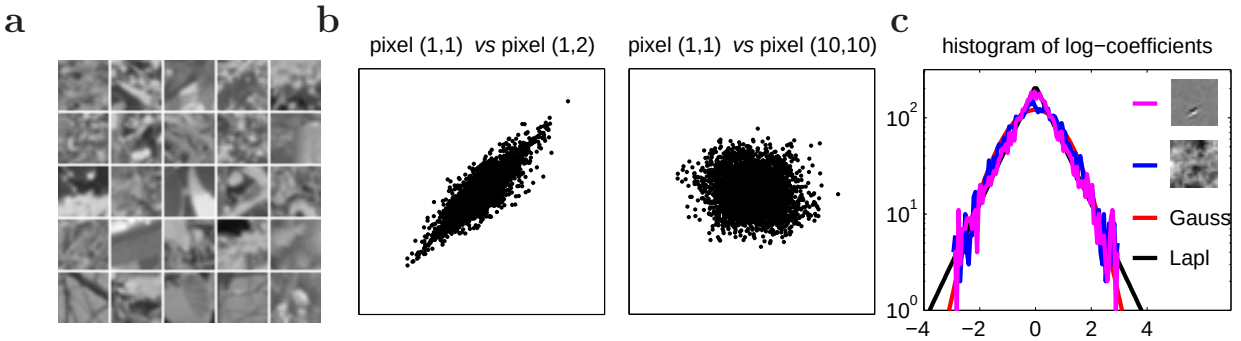


Figure 2.3: ***a***. Example image patches. Our goal is to model these data. ***b***. Scatter plots of image data. Neighboring pixels are clearly correlated, but long-range structure, evident by eye, is not easily discerned. ***c***. Histograms of coefficient distributions for natural images. Projected onto a random vector (shown as an image filter in the legend, blue line), the distributions are slightly more peaky and heavy-tailed (kurtosis = 0.7) than a Gaussian (red line, kurtosis = 0). Projected onto a Gabor wavelet (legend, magenta), the distributions are quite sparse (kurtosis = 5.6) and better fit by a Laplacian distribution (black).

peaky than Gaussian data (Fig. 2.3c). It has been argued that such sparse distributions lead to codes that are more efficient for transmitting information (Field, 1987) and forming associative memories (Zetzsche, 1990; Field, 1994), and early work related this to cortical representations by noting that the outputs of filters resembling simple cell receptive fields are especially sparse (Field, 1987; Zetzsche, 1990; Daugman, 1989; Field, 1994).

A closer link between the theory of efficient coding and visual representations in the cortex was established by two related modeling approaches, independent component analysis (ICA, Comon, 1994; Bell and Sejnowski, 1995, 1997) and sparse coding (Olshausen and Field, 1996, 1997), that showed that among all possible linear codes, ones that employed filters resembling simple cell receptive fields are in fact optimal for natural images. In these models, each image $\mathbf{x}$ (here a vector of pixel intensities) is represented as a linear combination of basis functions $\mathbf{A}_j$ weighted by coefficients $s_j$,

$$\mathbf{x} = \sum_j^J \mathbf{A}_j s_j \,. \tag{2.1}$$

ICA attempts to find the basis $\mathbf{A}$ that results in the most independent (and thus least redundant) coefficients. Typically, the number of basis functions is equal to the dimensionality of the inputs ($\mathbf{A}$ is square and invertible), and the coefficients are computed as $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$. Because empirical measures of independence are difficult to compute, in practice approximations are used, for example by optimizing the basis under the assumption of a factorial probability density for the coefficients $p(\mathbf{s}) = \prod_j p(s_j)$.

The sparse coding model incorporates additive Gaussian noise, $\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}$ and does not restrict the number of encoding coefficients ($J$) to the dimensionality of the input. If the noise is non-zero, or $J$ is larger than the dimensionality of the data (the over-complete case), the units' activities are no longer a deterministic function of the data, because many states can give rise to the data, and the best one must be inferred. This requires approximation techniques to learn the optimal basis and makes more difficult the computation of the posterior over the state of the coefficients $p(\mathbf{s}|\mathbf{A}, \mathbf{x})$, but it also allows the model to deal with noisy sensory input and a variable number of neurons for representing an image.

Both models, when adapted to small patches sampled from natural images produce oriented features that resemble simple cell receptive fields (Fig. 2.4). These results are derived directly from the postulated computational principle (efficient coding), and thus lead to a novel hypothesis for the *function* of neurons in V1: rather than acting as edge detectors, or analyzing images in terms of frequency spectra, neurons recode the retinal input to reduce redundancy, and the resulting code is tuned specifically to the statistics of natural scenes. Importantly, this model makes theoretical predictions not only about the properties of individual neurons, but also about the way populations of neurons collectively represent the stimulus (see for example

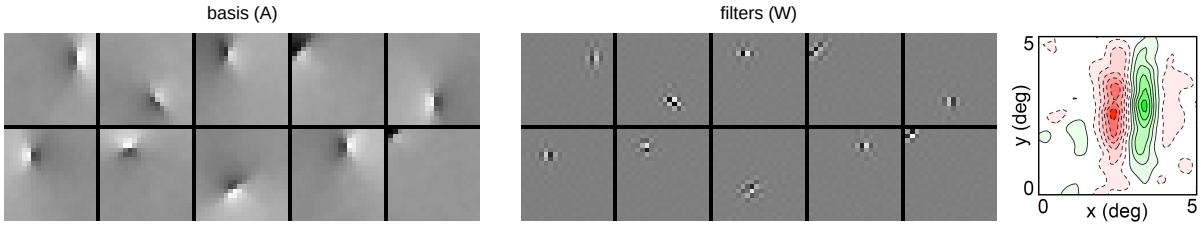


Figure 2.4: ICA applied to natural image patches yields simple cell-like "receptive fields". Left: A subset of 10 ICA basis functions. Center: the associated ICA filters ($\mathbf{W} = \mathbf{A}^{-1}$), which correspond to the stimulus pattern used for computing coefficients $\mathbf{s}$. They are the model's analogue of neural receptive fields. Right: reconstructed receptive field (optimal stimulus) of a V1 simple cell derived from physiological measurements (green is light, red is darkness), reproduced from (DeAngelis et al., 1995).

van Hateren and van der Schaaf, 1998; Olshausen and Field, 1997; Lewicki, 2002).

Because these methods are firmly grounded in a general statistical framework and make few assumptions about the data, they have also proved useful in a wide variety of image processing applications. Good models of the data are useful for compression, dimensionality reduction, and statistical image restoration, and these problems have been tackled using these and related models (Lewicki and Olshausen, 1999; Hyvärinen, 1999; Hyvärinen and Oja, 2000).

It should also be noted that reducing the redundancy of the neural code is only one theoretical principle that can be applied to neural processing. Other coding objectives and neural constraints, some related to redundancy reduction, have also been applied to natural images. Linear models that minimize energy expenditure (Baddeley, 1996), force their activity to vary smoothly in time (Hurri and Hyvärinen, 2003), or maximize quasi-orthogonality of an over-complete "basis" (Inki and Hyvärinen, 2001) also result in qualitatively similar, oriented and localized components. As a result, we have multiple possible functional explanations for the shape of neural receptive fields, and a variety of theoretical predictions testable by physiological experiments.

### 2.2.2 Limitations of linear models

Although these models have successfully provided a functional account of early processing in the visual system, they do suffer some limitations. First, because the underlying generative models are linear, they can only account for statistical dependencies that result from linear combinations of independent quantities. If this were sufficient to account for all structure, we would expect 1) images sampled from these models to be indistinguishable from real natural images and 2) the coefficients used to encode images (and assumed to be independent) to be, in fact, independent. It is easy to show that (1) does not hold; while images sampled from these models are somewhat closer in appearance to real world images than Gaussian-generated images (Fig. 2.5b), they lack much of the visible structure of real images. Specifically, randomly generated images lack image structure such as elongated contours, regular textures, and heterogeneous regions.

It has also been shown that the coefficients in linear models are not independent. One observation is that the magnitudes of coefficients are correlated (Fig. 2.6a) — a non-linear relationship that these models cannot capture — a pattern first observed among wavelet coefficients (Simoncelli, 1997). The strength of this effect depends on the relationship between the two filters; pairs at adjacent locations in the image or at nearby orientations exhibit the strongest positive correlations in magnitude. Another piece of evidence for higher-order dependencies is that the joint distributions in the coefficients are not well described by factorized distributions, i.e. their joint distributions cannot be obtained from products of the marginals (Fig. 2.6b, Zetzsche and Röhrbein, 2001). These provide clear indications that even from a pure statistical modeling perspective, the linear coding models do not adequately capture the complexity of natural image distributions.

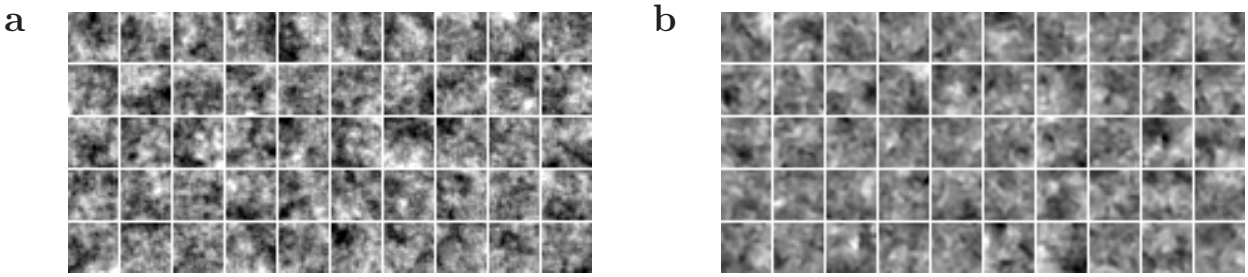


Figure 2.5: Images sampled from a Gaussian model (***a***) and an ICA model with sparse coefficients (***b***). Neither set resembles natural images (Fig. 2.3)

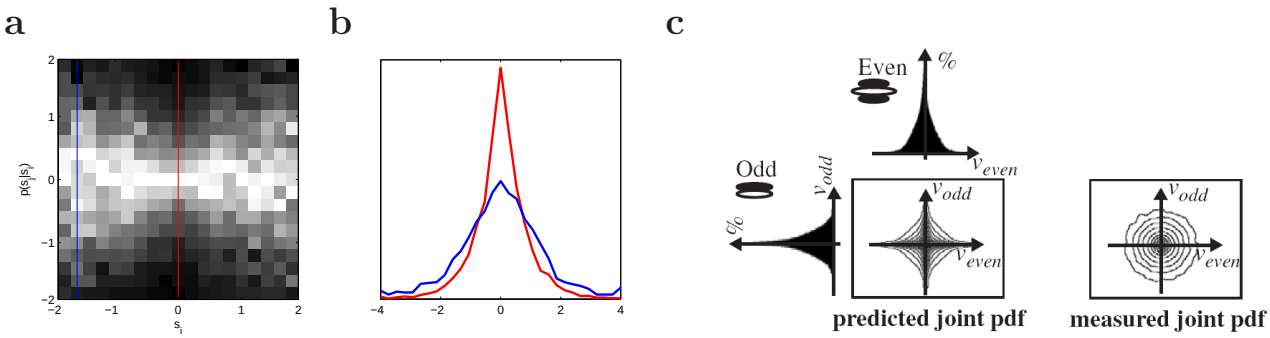


Figure 2.6: Magnitude dependence among pairs of coefficients. ***a***. Conditional histogram; each vertical slice shows the distribution of $s_j$ conditional on values of $s_i$ (shown on the ordinate, intensity rescaled independently for each slice). The "bow-tie" shape (originally described for wavelets in Simoncelli, 1997) indicates that variance of $s_j$ increases with larger values of $|s_i|$. ***b***. Probability density of $s_j$ when $|s_i| < .2$ (red line) and $|s_i| > 1.6$ (blue line). ***c***. Joint distributions of coefficients are not consistent with factorial density functions (from Zetzsche and Röhrbein, 2001).

The second shortcoming of these models is that they assume that the statistics of the data do not change, i.e. they describe stationary probability distributions. For example, once model parameters are adapted in ICA, both the prior and the basis functions are fixed, leading to a stationary distribution over the image ensemble. This does not depend on the form of the prior, and also applies to models with adaptive or entirely non-parametric priors. However, the statistics of the images change depending on the context (Fig. 2.7) or as the physical properties of the environment or conditions for data acquisition vary. While the stationary prior assumption gives a valid approximation of the true density over a large corpus of training images, it does not reflect the variation that is observed across different subsets of the dataset.

Also, linear transformations of image patches do not offer tractable representations of the visual structure we observe in natural images. Textures, edges, corners, and other seemingly salient features are not clearly isolated in the representations. Individual fixations across images of a particular type, e.g. a homogeneous texture, result in wild fluctuations in the output of linear filters. In order to form a representation that is stable (invariant) across individual fixations, a model must extract more abstract properties of the image. Latent variables in the model should represent unobserved but significant characteristics that can be used for grouping data points or finding statistically similar data. Current models do not provide such compact descriptions of higher-order structure. There is a clear need for more powerful models that can account for the observed dependencies, describe higher-order statistical regularities, and form useful representations of complex image structure.

As computational accounts of neural processing, the models are most useful in the predictions they make about neural codes and the interpretations they lend to observed properties of cells. Thus, application of these methods to poorly understood brain areas is of greatest importance, but to date few models make predictions

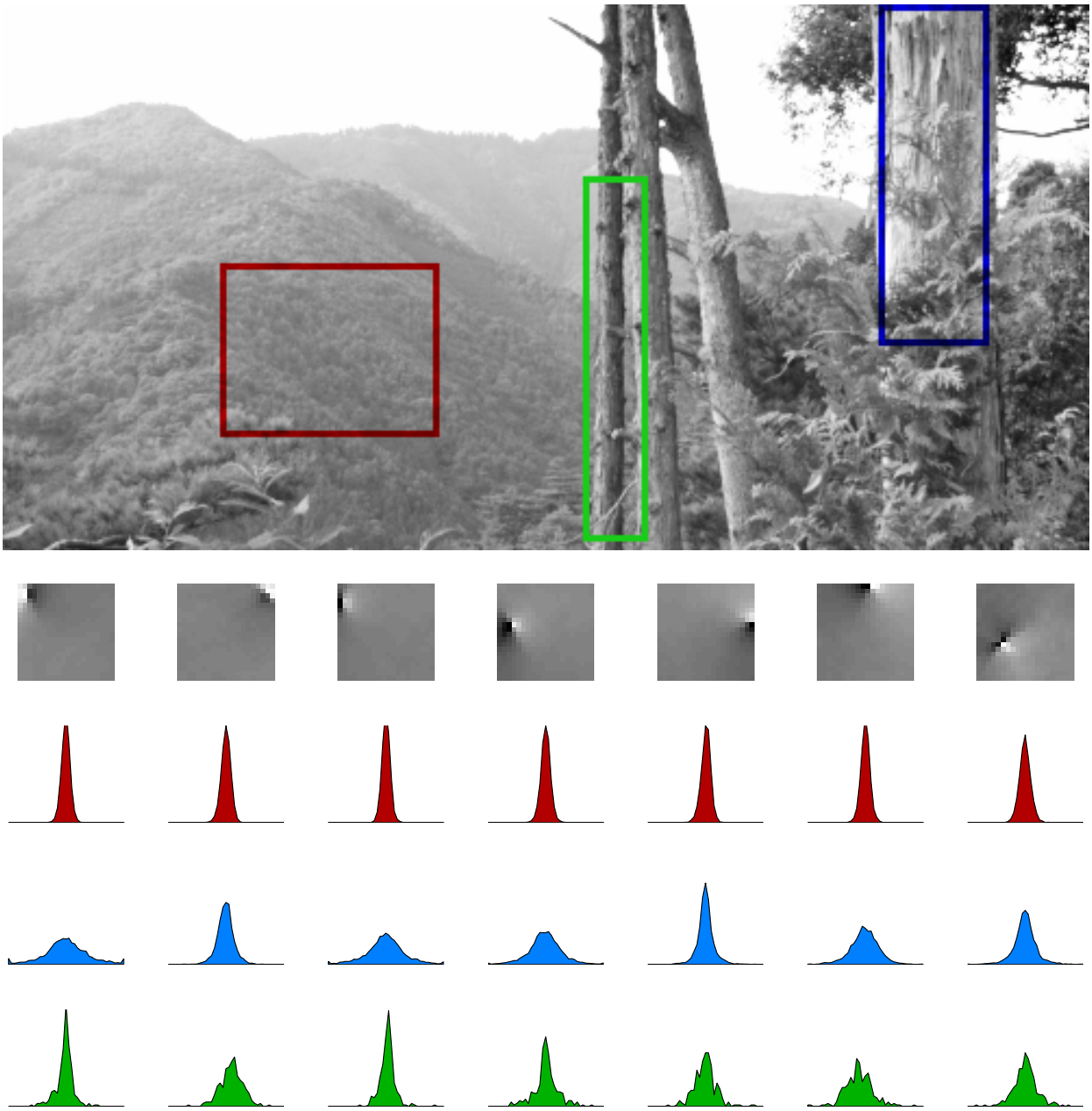

Figure 2.7: The variances of ICA coefficients change from context to context. Image patches were sampled from three regions in a natural image (top), and histograms for 7 ICA basis function coefficient computed (bottom).

about processing in higher visual areas like V2 and V4, or provide functional accounts of non-linear effects in V1. It is true that in the case of sparse coding, when the representation is over-complete or the noise level is significant, the encoding step (deriving the optimal representation of a given image) is non-linear, with basis function coefficients competing to best represent the image and maximize the sparseness of the code. This competitive behavior leads to non-linearities in the model response that matches some properties of simple cells in V1, such as end-stopping and surround suppression (Raina et al., 2007). This suggests that global competition and very sparse representation can account for some of non-linear properties; nevertheless this model does not explain the full range of non-linear behaviors, or responses in higher visual areas.

The key contribution of this dissertation is to address the limitations of these models by developing novel,

hierarchical, statistical models that automatically learn regularities in the input beyond the simple linear relationships. An equally important component is the detailed analysis of model parameters and encoding of natural images, and the comparison between these and the experimentally observed properties of neurons in the visual cortex.

# Chapter 3

# A hierarchical model for variance dependencies

## 3.1 Related work

Linear factor models, as described in the previous chapter, provide a good foundation for more powerful statistical models of natural images, and several extensions have been proposed to capture the observed dependence among the linear coefficients. For example, in the subspace ICA model (Hyvärinen and Hoyer, 2000), linear coefficients are no longer assumed to be independent; instead, groups of basis functions form neighborhoods (or subspaces), within which energies of the coefficients tend to be correlated, so that pooled activity is either very high or close to zero. The linear basis functions are then adapted to maximize the independence of the vector norms of the neighborhoods, rather than the independence of individual coefficients. This structure captures the dependence among coefficient magnitudes, since variables within a subspace are assumed to have correlated power, and it allows the model to find the optimal subspaces to describe this dependence.

In the more general form of the model, called topographic ICA, the disjoint sets of dependent basis functions are replaced by a topographic arrangement that defines magnitude dependencies locally (Hyvärinen et al., 2001). These models yield several interesting findings: the linear basis functions are again oriented and localized features, but are now organized into related sets or topological maps; the pooled neighborhood units represent more complex features of image patches, and replicate some invariant properties of complex cells (Hyvärinen and Hoyer, 2000, 2001; Hyvärinen et al., 2001). The main limitation of these models is that the pattern of dependencies is specified in advance, rather than learned from the data, and is limited to the discrete (and positive-only) relationships supported by subset groupings or topographic relationships. If activation within a subspace is considered a model of complex cells, then none of the complex *suppressive* effects are captured by this model. Also, representation of higher-order image structure is restricted to pooling local energy, so these models lack a representation of global statistical regularities underlying complex images such as textures,

Another set of models employs more flexible architectures but relies on a *fixed* linear transform, such as a multi-scale wavelet decomposition, and models dependencies among its coefficients. The Gaussian Scale Mixture model (Andrews and Mallows, 1974; Wainwright et al., 2001) and some models derived from it (Buccigrossi and Simoncelli, 1999; Romberg et al., 2001) have been used to describe coefficients of linear transforms (e.g. the multi-scale wavelet pyramid) as products of *independent* Gaussian variables and multiplier variables that are *mutually dependent*. The dependence among the multiplier variables is propagated along pre-specified structures, typically a tree of wavelet coefficients, but the strength of this de-

pendence can be learned from the data. These models have been shown to improve image compression (Buccigrossi and Simoncelli, 1999) and denoising (Wainwright et al., 2001; Portilla et al., 2003), suggesting that they provide better descriptions of image statistics.

While these models were designed to account for statistical dependencies observed in natural images, they have also been used to explain non-linear properties of sensory neurons, providing more evidence that early neural representations are optimized to the statistics of natural scenes. For example, these models compute local estimates of variance for outputs of linear filters. Normalizing the filter outputs yields a more independent code that also matches neural responses better than simple linear models, replicating complex behaviors of simple cells, such as contrast response saturation, cross-orientation inhibition, and mask suppression (Schwartz and Simoncelli, 2001). However, this approach relies on a fixed linear representation (not optimized for the structure of natural images), and is also limited to describing local pair-wise dependencies among linear filter outputs.

Most importantly, none of these methods provide a very rich description of higher-order dependencies — dependence among magnitudes are represented locally in topographic maps or coefficient trees. But higher-order image structure, as well as lower level, requires a distributed code capable of capturing a wide variety of complex visual features. We have developed a hierarchical model that addresses some of these problems (Karklin and Lewicki, 2003, 2005). It uses a parametric density model to learn statistical regularities from the data and makes no assumptions about the type of structure it expects to find. The model is general, it is not specific to any domain and can be applied to any dataset with rich statistical structure. Because the model forms distributed representations at all levels of its hierarchy, it scales well to large dimensional data. Adapted to patches from natural images or samples from speech data, the model is able to learn non-linear statistical regularities, a distributed representation of context, which included higher-order spatial relationships for image data, and frequency and harmonic structure for audio data. Sampling from the model produced data with the same statistical regularities observed in the training datasets.

## 3.2 Model

### 3.2.1 Definitions

The proposed model accounts for the observed residual dependencies and non-stationarity by automatically learning dependencies among the *variances* of coefficients of the linear model. Instead of fixing the scope of dependencies, as in previous models, it employs a flexible and distributed representation of coefficient variances. The data are again assumed to have been generated by a linear transformation from latent variables (possible including a noise term), $\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}$, but the latent coefficients $\mathbf{s}$ are now modeled with a hierarchical prior that reflects non-stationary variance. Specifically, we take the element-wise logarithm transform of the vector of variances $\boldsymbol{\sigma}^2$, and model the result as a linear function of higher-order random variables $\mathbf{v}$,

$$\log \boldsymbol{\sigma}^2 = \mathbf{B}\mathbf{v}\,. \tag{3.1}$$

When $\mathbf{v} = \mathbf{0}$ the variances of all the linear coefficients are 1, and the joint density reduces to the standard i.i.d. form of the standard linear models. Each column $\mathbf{B}_j$ describes a common *pattern of variances* of linear coefficients. As its coefficient $v_j$ moves away from zero, the pattern in the joint distribution $P(\mathbf{s})$ (and the deviation from the i.i.d. distribution) becomes more pronounced (Fig. 3.1b). Because the generic case when $\mathbf{v} = 0$ can describe an "average" distribution of data points, activation of the higher-order variables is only required to represent data in particular statistically salient contexts.

The schematic in Fig. 3.1a illustrates the full generative model. The only observed quantities in the model are the data $\mathbf{x}$. The next layer contains a set of coefficients $\mathbf{s}$, here referred to as *linear coefficients* because they generate the data through the linear transform, although in the recognition direction — from the data

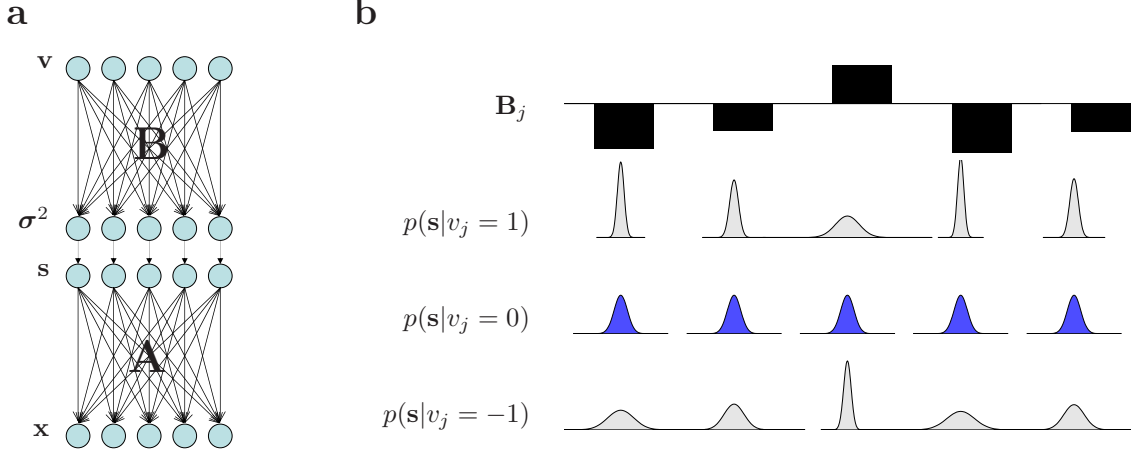


Figure 3.1: ***a***. A graphical representation of the hierarchical model for variance dependence. The variances of coefficients $\mathbf{s}$ are functions of higher-order latent variables $\mathbf{v}$. Model parameters $\mathbf{B}$ specify the linear transformation from $\mathbf{v}$ to $\log \boldsymbol{\sigma}^2$, and the parameters $\mathbf{A}$ from $\mathbf{s}$ to $\mathbf{x}$ (Gaussian additive noise can be incorporated at this stage, $\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}$, but is not shown in the graph). ***b***. An illustration of the effect of the $j$th column in $\mathbf{B}$ on the joint distribution $p(\mathbf{s})$, assuming all other higher-order variables $v_k$ $(k \neq j)$ are zero (for illustration, the conditional densities $p(\mathbf{s}|\mathbf{B}, \mathbf{v})$ were chosen to be Gaussian, but other parameterizations are possible).

to the coefficients — the function might not be linear (for example, if there are more linear coefficients than data dimensions and we must choose which subsets should represent the data). The parameters in the matrix $\mathbf{A}$ make up the *linear basis*, which may include more basis functions than input dimensions. The upper layer in the model consists of the higher-order *variance coefficients* $\mathbf{v}$, that, through the *variance components* in $\mathbf{B}$ specify the joint *linear coefficient likelihood* $p(\mathbf{s}|\mathbf{B}, \mathbf{v})$. For the moment we leave the number of variance coefficients unspecified; in theory it can be greater or smaller than the number of linear coefficients.

Variance components (columns in $\mathbf{B}$) combine in a continuous fashion to describe a variety of possible joint probability distributions. They provide a distributed representation of statistical regularities, instead of relying on exclusive classes to describe the data. For each data sample, the variance coefficients specify an estimate of the probability density from which the sample was generated. The vector of all the variance coefficients scales and combines the variance components, generating the vector of variances for the joint distribution. This generating density is a statistical description of the context of the data sample, and can be used to formulate similarity metrics or draw other inferences about each image patch.

This model bears some resemblance to Mixture-of-Gaussians models for ICA (Girolami, 2001; Davies and Mitiar 2004) which describe sparse joint distributions by combining different multivariate Gaussians. However, these models are not hierarchical and utilize a set of Gaussian densities only to approximate fixed sparse multivariate distributions. They do not learn any higher-order dependencies in the data.

The matrix of variance components $\mathbf{B}$ is unconstrained; adapting it to the data allows us to automatically learn higher-order statistical regularities without making assumptions about the type of structure we expect to find. This entails maximizing the expected value of the log-posterior of model parameters over the entire data ensemble, $\langle \log p(\mathbf{A}, \mathbf{B}|\mathbf{x}_n) \rangle_n$. For a single data point $\mathbf{x}_n$, the posterior distribution is expressed as

$$p(\mathbf{A}, \mathbf{B}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{A}, \mathbf{B}) p(\mathbf{A}) p(\mathbf{B}) \tag{3.2}$$

$$\propto \int_{\mathbf{s}} p(\mathbf{x}, \mathbf{s}|\mathbf{A}, \mathbf{B}) p(\mathbf{A}) p(\mathbf{B}) d\mathbf{s} \tag{3.3}$$

$$\propto \int_{\mathbf{s}} p(\mathbf{x}|\mathbf{A}, \mathbf{s}) p(\mathbf{s}|\mathbf{B}) p(\mathbf{A}) p(\mathbf{B}) d\mathbf{s} \tag{3.4}$$

$$\propto \int_{\mathbf{s}} p(\mathbf{x}|\mathbf{A}, \mathbf{s}) \left( \int_{\mathbf{v}} p(\mathbf{s}|\mathbf{B}, \mathbf{v}) p(\mathbf{v}) d\mathbf{v} \right) p(\mathbf{A}) p(\mathbf{B}) d\mathbf{s} \tag{3.5}$$

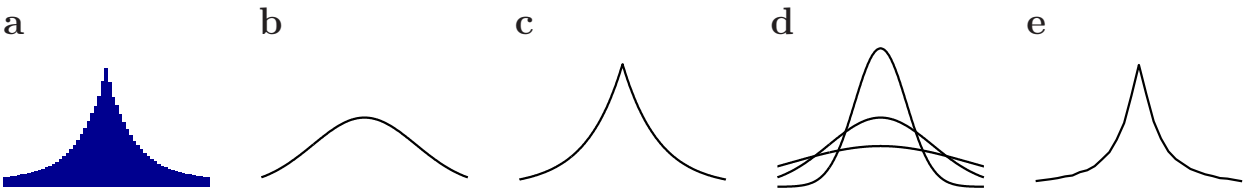


Figure 3.2: The observed distribution of coefficients in ICA (***a***) is not well described by a Gaussian density (***b***), but is very similar to the Laplacian density (***c***). If we take Gaussians of different variances (***d***) and compute the distribution marginalized over the probability of each value of the variance (e.g. by assuming a log-normal prior for the variance), we get a similar peaky, heavy-tailed function (***e***).

(dropping the $n$ data index for simplicity). This is the objective function we want to maximize. Below we break down the posterior and focus on its component probability functions.

*Data likelihood* $p(\mathbf{x}|\mathbf{A},\mathbf{s})$. If the linear stage of the model is complete and noise-free, as in the original ICA, the data likelihood given the coefficients collapses to a delta function,

$$p(\mathbf{x}|\mathbf{A},\mathbf{s}) = \delta(\mathbf{x} - \mathbf{A}\mathbf{s}) \tag{3.6}$$

and the posterior is computed by marginalizing over the latent variables $\mathbf{v}$ only,

$$p(\mathbf{A},\mathbf{B}|\mathbf{x}) \propto \frac{1}{|\det \mathbf{A}|}\left(\int_{\mathbf{v}} p(\mathbf{s}|\mathbf{B},\mathbf{v})p(\mathbf{v})d\mathbf{v}\right)p(\mathbf{A})p(\mathbf{B})\,. \tag{3.7}$$

Alternatively, we can include noise or an over-complete set of coefficients $\mathbf{s}$, as in sparse coding. If the noise is assumed to be i.i.d. Gaussian,

$$p(\mathbf{x}|\mathbf{A},\mathbf{s}) = \mathcal{N}(\mathbf{A}\mathbf{s},\sigma_\epsilon^2\mathbf{I})\,, \tag{3.8}$$

where $\sigma_\epsilon^2$ is the noise variance. This allows us to do probabilistic denoising, or evaluate the effect of over-complete representations on the recovered parameters (Lewicki and Olshausen, 1999). In this case we must infer the state of both $\mathbf{s}$ and $\mathbf{v}$ given an input image, and marginalize over both sets of unknowns when doing parameter estimation (Eqn. 3.5). In practice the marginalizations are not feasible to compute, and we must rely on approximations (see below for details).

*Linear coefficient likelihood* $p(\mathbf{s}|\mathbf{B},\mathbf{v})$. In the linear factor models, $p(\mathbf{s})$ was a fixed (and typically sparse) factorisable prior density, and the linear coefficients were assumed to be independent. Now, however, the variance of each coefficient is allowed to vary from sample to sample, and the linear coefficients are independent only *given the variance coefficients* $\mathbf{v}$,

$$p(\mathbf{s}|\mathbf{B},\mathbf{v}) = p(\mathbf{s}|\boldsymbol{\sigma}^2) = \prod_i p(s_i|\sigma_i^2)\,. \tag{3.9}$$

We have several choices for the exact form of this likelihood: we can, as in ICA, choose a sparse density, such as the Laplacian (figure 3.2c), and include $\sigma_i^2 = \exp([\mathbf{B}\mathbf{v}]_i)$ as the scale parameter,

$$p(s_i|\sigma_i^2) = \frac{1}{\sqrt{2\sigma_i^2}}\exp\left(-\frac{\sqrt{2}|s_i|}{\sqrt{\sigma_i^2}}\right)\,, \tag{3.10}$$

or we can choose to model $s_i$ with a Gaussian with variance $\sigma_i^2$,

$$p(s_i|\sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{s_i^2}{2\sigma_i^2}\right)\,. \tag{3.11}$$

This might seem like an invalid assumption — in the previous section we noted that the Gaussian distribution is not an appropriate prior for modeling our data, which tends to be sparse (Fig. 3.2a). In this hierarchical

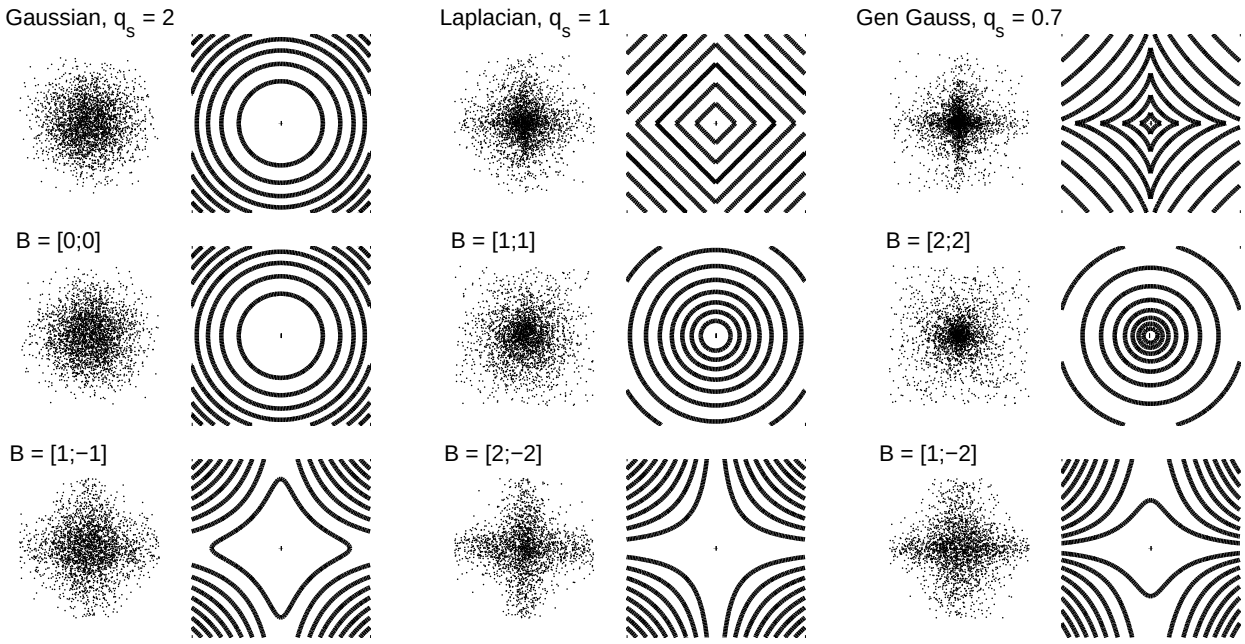


Figure 3.3: The joint distribution $p(\mathbf{s})$, marginalized over higher-order variables $\mathbf{v}$, can have a variety of shapes. Top row: some examples of 2D joint distributions of fixed linear models with an increasingly sparse prior (these distributions all belong to the generalized Gaussian family, $p(s_i) \propto \exp(-|s_i|^q)$). Middle and bottom row: hierarchical models with different sets of parameters $\mathbf{B}$. For illustration, the dimensionality of $\mathbf{v}$ is 1, and the matrix $\mathbf{B}$ is simply a column vector. In these examples, $v$ is assumed to follow a Gaussian distribution $u$ is conditionally Gaussian, (i.e. $p(\mathbf{s}|\mathbf{B}, v) \sim \mathcal{N}(0, \mathrm{diag}(\exp(\mathbf{B}v)))$. When $\mathbf{B} = [0; 0]$, the model density is a multivariate Gaussian. Even with this simple hierarchy, the model can generate sparse star-shaped (bottom row) or radially symmetric (middle row) densities, as well as more complex non-symmetric densities (bottom right). In higher dimensions, it is possible to describe more complex joint distributions, with different marginals along different projections.

model, however, each data point is generated from a multivariate Gaussian distribution with a different variance (and not necessarily isotropic). Marginalizing over all possible values of the variance (by integrating over the hyper-parameters $\mathbf{v}$) can in fact yield a very sparse distribution (Fig. 3.2e). Choosing the Gaussian likelihood function (Eqn. 3.11) implies that the marginal density $p(\mathbf{s})$ is in the class of Gaussian Scale Mixture models, which includes the generalized Gaussian ($p(x) \propto e^{-|x|^q}$), the Cauchy, and other distributions (Wainwright et al., 2001). Whether the likelihood in (3.10) or (3.11) is more appropriate in our setting is an empirical question. If we allow some level of noise in the data, the true distribution of the coefficients is unknown, and the best model depends how sparse we believe this true distribution to be. Empirically, the outputs of linear filters are well fit by a Laplacian (Fig. 2.3c) or the slightly more sparse generalized Gaussian with $q = 0.7$, but we might be interested in recovering a more sparse set of coefficients. It is difficult to obtain answers to these model selection questions when no specific task is in mind. In our simulations, the learned higher-order dependencies among linear coefficients were not significantly affected by the choice of this likelihood function.

This hierarchical model, even when the prior and the coefficient likelihood functions are Gaussian, can produce a variety of shapes for the joint distribution in data space (Fig. 3.3). It is interesting to note that the hierarchical model can generate peaky but spherically symmetric probability functions with sparse distributions along all projections. The plotted projections only hint at the greater complexity of the high-dimensional joint distribution, which is adapted to best describe the data ensemble.

*Prior on variance coefficients* $p(\mathbf{v})$. We would like the parameter matrix $\mathbf{B}$ to capture the dependencies among linear coefficient variances, so we assume that the variance coefficients themselves are mutually

independent,

$$p(\mathbf{v}) = \prod_j p(v_j)\,, \tag{3.12}$$

and we can place a Gaussian or a sparse prior on $v_j$, depending on how densely we think the variance components will contribute to the conditional density $p(\mathbf{s}|\boldsymbol{\sigma}^2)$. A Gaussian prior on $v_j$ yields a multivariate log-normal distribution for the vector of variances,

$$\log \boldsymbol{\sigma}^2 \sim \mathcal{N}(0, \mathbf{B}\mathbf{B}^T)\,, \tag{3.13}$$

meaning that the estimated variance components ($\mathbf{B}$) capture correlational structure in the log-space of the variances. Portilla et al. (2001) noted that in the case of 1D data, it is possible to estimate empirically the distribution of the latent variance parameter, and that this distribution is approximately log-normal for wavelet coefficients that encode natural images. However, if distributions of log-variances are not spherically symmetric and have directions of higher or lower density, a Gaussian prior will not allow us to recover these directions (a multivariate Gaussian model does not yield identifiable axes, as combinations of Gaussian variables are also Gaussian). Another potentially important issue arises when we consider that symmetric priors on $\mathbf{v}$ imply that the learned variance patterns are symmetric — a pattern of high and low variances is as likely as its converse, low and high (see figure 3.1). Alternatively, we can restrict $\mathbf{v}$ to be all-positive, dropping this assumption, but this can be computationally tricky when gradient methods for MAP estimation are employed.

*Priors on parameters $p(\mathbf{A})$ and $p(\mathbf{B})$.* We do not have strong prior beliefs about the structure of parameter matrices $\mathbf{A}$ and $\mathbf{B}$, and would like to place fairly agnostic priors. For example, we can choose i.i.d. Gaussian priors on the elements of these matrices. In practice, this has little effect, because the number of training samples in the experiments is very large. On the other hand, adding weak weight decay or fixing the norm of the vectors in $\mathbf{A}$ and in $\mathbf{B}$ does help alleviate degenerate conditions introduced by the approximations of marginalization over the latent variables (see below).

*Marginalization over the variance coefficients $\int_v d\mathbf{v}$.* In order to estimate the optimal parameters we need to marginalize over the latent variables $\mathbf{v}$. In practice, this is quite difficult to do. Solving the integral analytically is intractable, but it is possible to approximate this optimization by replacing the integral with its value at the mode, i.e. at the maximum *a posteriori* value of $\mathbf{v}$,

$$\begin{aligned} p(\mathbf{s}|\mathbf{B}) &= \int p(\mathbf{s}|\mathbf{B},\mathbf{v})p(\mathbf{v})d\mathbf{v} && (3.14)\\ &\approx p(\mathbf{s}|\mathbf{B},\hat{\mathbf{v}})p(\hat{\mathbf{v}}) && (3.15)\\ \text{where } \hat{\mathbf{v}} &= \arg\max_{\mathbf{v}} p(\mathbf{v}|\mathbf{B},\mathbf{s})\,. && (3.16) \end{aligned}$$

This means that instead of finding model parameters best for the distribution of latent variables, we will recover optimal parameters for the best values of latent variables. The approximation — replacing the volume under the surface of the integral with the height of its peak — is fairly gross. It also introduces a degeneracy into the learning process, in which the weights in the matrix $\mathbf{B}$ grow without limit while optimal values of $\mathbf{v}$ rescale to be smaller (and more likely as unique MAP solutions). Nevertheless, this method works in practice (see below for empirical verification); a comparison to sampling methods, e.g. Markov Chain Monte Carlo, is left for future work.

In the case of a noiseless linear stage, a Gaussian linear coefficient likelihood, and a Laplacian prior for the

variance coefficients, the final approximation to the log-likelihood function is given by

$$L = \log p(\mathbf{x}|\mathbf{A}, \mathbf{B}) \tag{3.17}$$

$$= -\log|\det \mathbf{A}| + \sum_i \log \int p(s_i|\mathbf{B}, \mathbf{v}) p(\mathbf{v}) d\mathbf{v} \tag{3.18}$$

$$\hat{L} = -\log|\det \mathbf{A}| + \sum_i \log p(s_i|\mathbf{B}, \hat{\mathbf{v}}) + \sum_j \log p(\hat{v}_j) \tag{3.19}$$

$$\propto -\log|\det \mathbf{A}| + \sum_i \left( -\frac{[\mathbf{B}\hat{\mathbf{v}}]_i}{2} - \frac{\sqrt{2}|s_i|}{e^{[\mathbf{B}\hat{\mathbf{v}}]_i/2}} \right) - \sum_j |v_j|\,, \tag{3.20}$$

where $[\mathbf{Bv}]_i$ refers to the $i^{th}$ element of the product vector $\mathbf{Bv}$.

When the linear stage is over-complete or noisy, the coefficients are not deterministically obtained from the data, and we approximate the log-likelihood using the MAP estimate $\hat{\mathbf{s}} = \arg\max p(\mathbf{s}|\mathbf{x}, \mathbf{A}, \mathbf{B})$.

### 3.2.2 Latent variable inference

**Variance coefficients**

For each data sample, we would like to compute the higher-order representation $\mathbf{v}$ that best describes the pattern in the scale of coefficients $\mathbf{s}$. This transformation is non-linear, and cannot be expressed in closed form. In the case where the linear coefficients are unknown, they too must be inferred.

We compute the best value of $\mathbf{v}$ by maximizing the posterior distribution (Eqn. 3.16). For the simulations below, $\hat{\mathbf{v}}$ was derived by gradient ascent. The inference gradient for the variance coefficients is

$$\frac{\partial \hat{L}}{\partial v_j} = \frac{1}{2} \sum_i \left( -B_{ij} + B_{ij} \frac{\sqrt{2}|s_i|}{e^{[\mathbf{Bv}]_i/2}} \right) - \phi_v'(v_j)\,, \tag{3.21}$$

where $\phi_v'(v_j) = -\partial \log p(v_j)/\partial v_j$ (see appendix for derivation). In the case of independent Laplacian prior, $\phi'(v_j) = \sqrt{2}\mathtt{sign}(v_j)$. What is this gradient computing? The first part of the equation measures the deviation in the magnitude of each coefficient (normalized by the current scale estimate), and adjusts the variance coefficient according to its weights $B_{ij}$ to the linear coefficients, so that

$$\hat{v}_j^{new} \longleftarrow \hat{v}_j^{old} + \epsilon_v \left( \sum_i B_{ij}(|\bar{s}_i| - 1) - \phi_v'(\hat{v}_j^{old}) \right) \tag{3.22}$$

where $\epsilon_v$ is the step size and $\bar{s}_i = \sqrt{2}s_i/e^{]\mathbf{Bv}]_i/2}$. We can think of the inference process as attempting to match the scale of the linear coefficients by adjusting the representation $\mathbf{v}$, subject to the sparsification term $\phi_v'(\cdot)$.

How well-behaved is this likelihood function? Are we guaranteed to obtain the global maximum using gradient ascent? The Hessian of the negative log-likelihood is

$$\frac{\partial^2 \hat{L}}{\partial v_j \partial v_k} = -\frac{\partial}{\partial v_k} \frac{1}{2} \sum_i B_{ij} \frac{\sqrt{2}|s_i|}{e^{[\mathbf{Bv}]_i/2}} \tag{3.23}$$

$$= \frac{1}{4} \sum_i B_{ij} B_{ik} \frac{\sqrt{2}|s_i|}{e^{[\mathbf{Bv}]_i/2}} \tag{3.24}$$

$$\frac{\partial^2 \hat{L}}{\partial \mathbf{v} \partial \mathbf{v}^T} = \frac{1}{4} \mathbf{B}^T \bar{\mathbf{S}} \mathbf{B} \tag{3.25}$$

where $\bar{\mathbf{S}}$ is a diagonal matrix containing the variance-normalized coefficient magnitudes, $\bar{\mathbf{S}} = \text{diag}\left(|\bar{s}_1|, |\bar{s}_2|, \ldots, |\bar{s}_I|\right)$. For a Laplacian prior $p(\mathbf{v})$, the last term of Eqn. 3.21 is constant and the second derivative is 0, but it is defined only when $v_j \neq 0, \forall v_j$. The Hessian of Eqn. 3.25 is positive definite as long as it holds that

$$\bar{\mathbf{S}}^{1/2}\mathbf{B}\mathbf{x} = 0 \quad \text{iff} \quad \mathbf{x} = 0 \quad \text{for any} \quad \mathbf{x}\,. \tag{3.26}$$

This is satisfied when the columns of $\bar{\mathbf{S}}^{1/2}\mathbf{B}$ are linearly independent. We can safely assume the columns in $\mathbf{B}$ are linearly independent, and the diagonal matrix $\bar{\mathbf{S}}$ only rescales these vectors. However, if any linear coefficient $s_j$ is exactly 0, the determinant of the Hessian can be zero. (We can see that this could drive some variance coefficients to extreme values.) Thus it is possible for the log-likelihood to be non-convex, but only when linear coefficients are exactly zero. In practice, the number of variance components was always significantly smaller than the dimensionality of the input, which alleviated this problem.

We also used the diagonal terms of the Hessian to stabilize and speed up the inference procedure by adjusting the step size along each dimension of $\mathbf{v}$ (see the appendix for details).

Because the prior is zero-centered and sparse, only a few non-zero values will contribute to the representation of each data sample. The inference of optimal variance component coefficients is analogous to estimating sample variance based on single observations, but the problem is further constrained by the structure of the learned variance components. Because the model is constrained to describe the pattern of variance with a sparse combination of variance components, the value $\mathbf{v}$ for a typical pattern is usually well-determined, though it is not the case that there is always a single global maximum. (This can be verified by starting inference at different random initial values, which leads to convergence to the same estimates $\hat{\mathbf{v}}$). In addition, the high dimensionality of the input facilitates the inference process, as it provides more directions of variation that make up the variance pattern.

**Linear coefficients**

When the linear coefficients $\mathbf{s}$ are unknown (e.g. when we include noise in the model), the MAP estimates $\hat{\mathbf{s}}$ can be obtained as in sparse coding (Olshausen and Field, 1997),

$$\frac{\partial \hat{L}}{\partial \mathbf{s}} = \frac{1}{\sigma_\epsilon^2}\mathbf{A}^T\left(\mathbf{x} - \mathbf{A}\mathbf{s}\right) - \phi_s'(\mathbf{s}|\mathbf{B}, \hat{\mathbf{v}})\,, \tag{3.27}$$

except that the prior $p(\mathbf{s})$ is now a non-stationary conditional function (Eqns. 3.10 and 3.11) instead of a fixed sparse distribution, and the term $\phi_s'(\cdot)$ must reflect this.

### 3.2.3 Parameter estimation

The linear basis functions and the variance components are adapted to the data ensemble by maximizing the expected likelihood over the data ensemble $\langle p(\mathbf{x}_n|\mathbf{A}, \mathbf{B})\rangle_n$. We assume that samples in the data ensemble are independent, so that

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N|\mathbf{A}, \mathbf{B}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{A}, \mathbf{B})\,. \tag{3.28}$$

The posterior is broken down in Eqn. 3.5. Marginalization over latent variables is avoided by using the MAP approximation. The matrices $\mathbf{A}$ and $\mathbf{B}$ can be optimized concurrently by interleaving gradient ascent steps on $\mathbf{A}$ and $\mathbf{B}$. When the matrix $\mathbf{B}$ is fixed, adapting $\mathbf{A}$ is equivalent to optimization in linear models, such as ICA and sparse coding, except the priors on linear coefficients are scaled by the variance given by $\mathbf{B}\mathbf{v}$. In the noiseless complete case, $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$, we optimize the inverse of the linear basis functions, the filter $\mathbf{W}$,

using the natural gradient

$$\frac{\partial \hat{L}}{\partial \mathbf{W}} = (\mathbf{I} + \phi'(\mathbf{s})\mathbf{s}^T)\mathbf{W} \tag{3.29}$$

(Amari, 1999); otherwise we use the sparse coding update rule,

$$\frac{\partial \hat{L}}{\partial \mathbf{A}} = (\mathbf{x} - \mathbf{A}\hat{\mathbf{s}})\mathbf{s}^T \,. \tag{3.30}$$

The higher-order parameters in $\mathbf{B}$ are obtained by following the gradient

$$\frac{\partial \hat{L}}{\partial B_{ij}} = -\, v_j + v_j \frac{\sqrt{2}\,|s_i|}{e^{[\mathbf{Bv}]_i/2}} \,. \tag{3.31}$$

(see appendix for derivation). Again, this gradient strives to match the scale of the linear coefficients of the training data set by adjusting the parameters, just like latent variable inference was attempting to match the scale of individual encodings.

As in other models that rely on a MAP approximation scheme, there is a degeneracy in the model. Because the volume of the integral over the latent variables is replaced by its value at the peak, it is possible to improve the likelihood without bound by scaling the parameters up; this would produce smaller magnitude latent variables that are more likely under the sparse priors. One way to address this is to manually adjust the norm of the parameters to maintain the desired level of variance for the latent variables (Olshausen and Field, 1997). This method was was used in most of the simulations below. Other work has explored alternatives in the context of statistical models for sparse coding, e.g. using approximations to the integral based on the likelihood function's curvature information (Lewicki and Olshausen, 1999). However, these methods are slightly more complex and were not implemented for the hierarchical models.

### 3.2.4 Testing parameter recovery

In order to verify that the learning algorithm, and the approximations employed, reliably produce a valid solution, we adapted model parameters to a synthetic dataset for which the true parameter settings were known. The data were generated by constructing a set of variance components, generating random $\mathbf{v}$'s, and then sampling linear coefficients according to $p(\mathbf{s}|\mathbf{B}, \mathbf{v})$. An illustration of the process and the obtained results are shown in Fig. 3.4. Beginning with small random initial values and computing the maximum likelihood parameter estimates produced a matrix that was identical (up to a permutation of its columns) to the true values (Fig. 3.4a,b). The synthetic data also illustrate how variance components specify non-linear dependencies among coefficient magnitudes; there are no linear correlations among coefficients sampled from the model (even when the same $\mathbf{v}$ is used to generate the coefficients). Linear models like ICA are unable to recover these statistical regularities.

As a control, we adapted the model to a pure noise dataset in which coefficients $\mathbf{s}$ were random samples from independent sparse distributions. In this case, no regularities in the magnitudes of coefficients existed, and the resulting variance components consisted of small, random values.

## 3.3 Results on natural images

How does this model represent image structure? In what way is it an improvement over previous work? How does it contribute to our understanding of visual processing? And what are its limitations? Below I summarize the main findings resulting from applying the model to natural images, highlight their implications for neural coding, and discuss the limitations of this model.

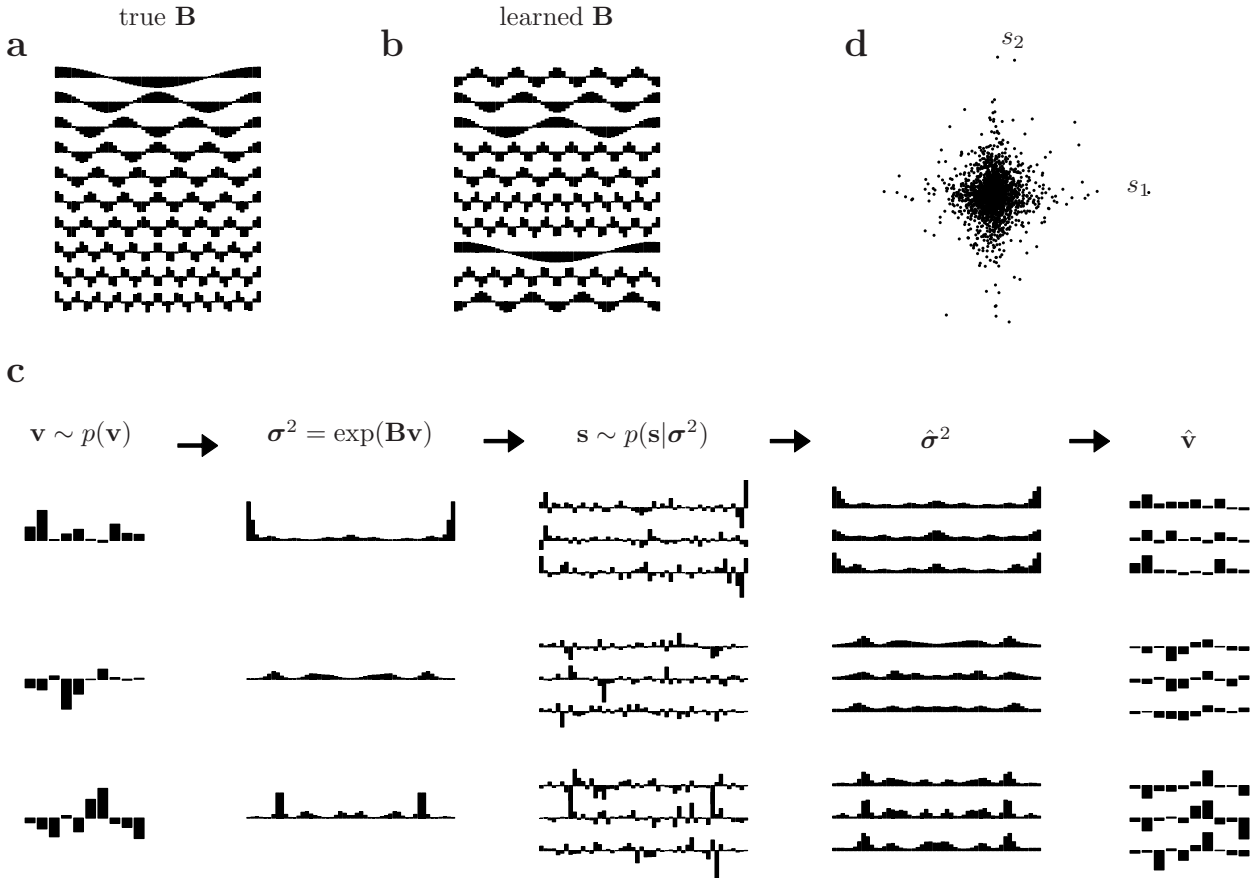


Figure 3.4: The model correctly recovers the variance components used to generate synthetic data. ***a***. We constructed a 50×10 matrix composed of 10 cosine-shaped variance components. ***b***. After 3000 iterations, the model recovers (up to a permutation) the correct parameters. ***c***. The generative and inference steps of the algorithm. Three 10-dimensional variance coefficients are drawn from a sparse distribution; each state of $\mathbf{v}$ specifies a different vector of scaling variables $\boldsymbol{\sigma}^2$ through the nonlinear transformation $\boldsymbol{\sigma}^2 = \exp(\mathbf{B}\mathbf{v})$. The scaling variables are hyper-parameters for non-stationary distributions $p(\mathbf{s})$. In order to emphasize that each vector of scaling variables $\boldsymbol{\sigma}^2$ specifies a distribution, not fixed values of $\mathbf{s}$, we plotted several $\mathbf{s}$'s drawn from the distribution $p(\mathbf{s}|\boldsymbol{\sigma}^2)$; in actual simulation each data point was generated independently. Using the learned variance components estimates of $\hat{\mathbf{v}}$ and $\hat{\boldsymbol{\sigma}}^2$ were obtained for each data sample. Because the inference problem involves the estimation of density parameters from single data points, $\hat{\mathbf{v}}$ and $\hat{\boldsymbol{\sigma}}^2$ only approximately match true parameters. Note that we omitted another linear transformation from the coefficients to the data. ***d***. The scatter plot of 1000 samples of $s_1$ and $s_2$ drawn from the model shows that there is no linear dependence among basis function coefficients.

### 3.3.1 Simulation details

We applied the learning algorithm to small (20×20) image patches sampled from from 40 images of outdoor scenes (Doi et al., 2003). The images were filtered with a low-pass radially symmetric filters to eliminate high frequencies that are artifacts of the square sampling lattice, and the DC component was removed from each image patch. We used a complete set of linear basis functions (399), and the number of variance components was set to 100. The image ensemble was whitened by premultiplying with $\mathbf{C}^{-1/2}$, where $\mathbf{C}$ is the data covariance matrix, but all the results were analyzed by projecting back into the original image space. For some of the simulations the linear stage was noiseless, i.e. $\mathbf{x} = \mathbf{A}\mathbf{s}$, which allowed us to compare coefficients $\mathbf{s}$ to their variance-normalized values ($\bar{\mathbf{s}}$, see below) and avoid one of the marginalizations, while other simulations included small independent Gaussian noise at the pixel level. We used the Laplacian (Eqn. 3.10) as the coefficient likelihood function.

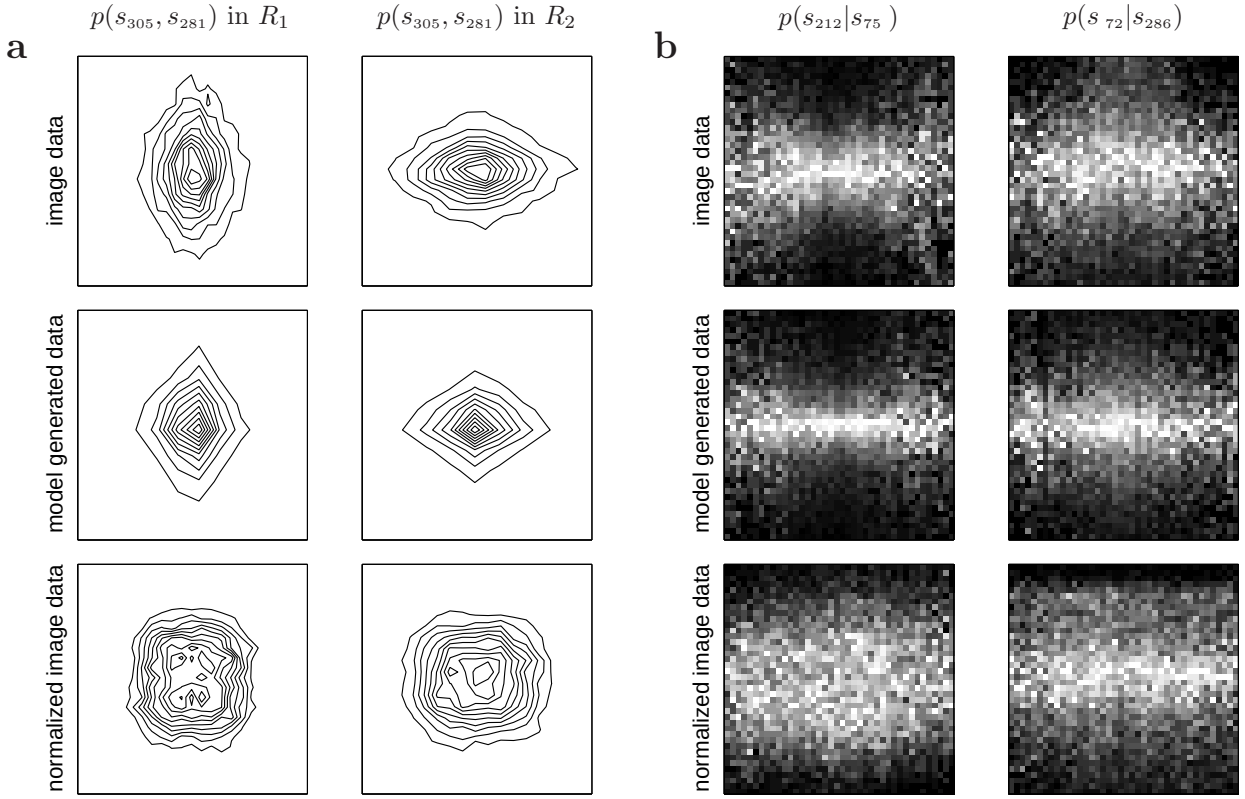


Figure 3.5: Dependence in the magnitudes of linear basis function coefficients are captured by the variance components. ***a***. The joint distributions of linear coefficients are different in two regions of a natural scene, i.e. the data distribution is not stationary. Sampling from the model under the estimated higher-order representation of each context results in similar distributions. Normalizing the image data by the estimated scale parameters, $\bar{s}_i = (1/\sqrt{\hat{\sigma}_i^2})s_i$, eliminates the non-stationarity. ***b***. Over the full data ensemble, empirical conditional histograms for pairs of coefficients show statistical dependencies in the magnitude. Sampling from the model adapted to this data ensemble produces similar dependencies, and normalizing by the estimated scale parameters removes the magnitude correlations. See text for more details.

### 3.3.2 Modeling residual dependencies

First, does the model account for the observed dependencies and non-stationary statistics? One of the motivations for the proposed model was that coefficients of linear models with fixed priors exhibited magnitude dependence and non-stationary variance. In the hierarchical model, the variance coefficients describe a different joint distribution for each image patch. Bivariate joint distributions sampled from the model are consistent with those observed in the data. For example, joint distributions of coefficients sampled using the MAP estimates $\hat{\mathbf{v}}$ in two different regions of a natural image have the same shape as the empirical joint distributions (Fig. 3.5a). If we normalize the linear coefficients of each image patch by the variances inferred for that sample, their distributions become much more uniform across contexts. Normalized coefficients all have variance of approximately 1, and they no longer exhibit pair-wise magnitude dependence (Fig. 3.5), suggesting that the hierarchical model is able to capture the observed dependencies.

### 3.3.3 Learned linear filters

Next we examined whether the optimal lower-level representation (the linear stage of the model) for natural images is different when trained in the context of such non-linear hierarchical models (Karklin and Lewicki,

2006). We compared the basis functions in $\mathbf{A}$ learned by the sparse coding algorithm Olshausen and Field (1996) and the hierarchical model described above (which, like sparse coding, included a data noise term).

All the results and analyses are reported in the original data space. Noise variance $\sigma_\epsilon^2$ was set to 0.1, and the basis functions were initialized to small random values and adapted on randomly sampled batches of 300 patches. In this simulation, the prior on $\mathbf{v}$ was Gaussian (and thus the patterns in $\mathbf{B}$ were not localized); the conditional distribution $p(\mathbf{s}|\mathbf{v})$ was also Gaussian (Eqn. 3.11). We used 40 images from the Kyoto dataset (Doi et al., 2003). We ran the algorithm for 10,000 iterations with a step size of 0.1 (tapered for the last 1,000 iterations once model parameters were relatively unchanging). The parameters of the hierarchical model were estimated in a similar fashion. Gradient descent on $\mathbf{A}$ and $\mathbf{B}$ was performed in parallel using MAP estimates $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$. The step size for adapting $\mathbf{B}$ was gradually increased at the beginning of the simulation because emergence of the variance patterns requires some stabilization in the basis functions in $\mathbf{A}$.

Because encoding in the noisy linear stage is a non-linear process, it is not possible to compare the inverse of the learned $\mathbf{A}$ directly to physiological data. Instead, we estimated the corresponding *filters* using reverse correlation to derive a linear approximation to a non-linear system, which is also a common method for characterizing V1 simple cells. We analyzed the resulting filters by fitting them with 2D Gabor functions, then examining the distribution of their frequencies, phase, and orientation parameters.

The shapes of basis functions and filters obtained with sparse coding have been previously analyzed and compared to neural receptive fields van Hateren and van der Schaaf (1998); Ringach (2002). However, some of the reported results were in the whitened space or obtained by training on filtered images. In the original space, sparse coding basis functions have very particular shapes: except for a few large, low frequency functions, they are localized, predominantly odd-symmetric, and span a single period of the sinusoid (Fig. 3.6, top left). The estimated filters are similar but smaller (Fig. 3.6, bottom left), with peak spatial frequencies clustered at higher frequencies (Fig. 3.7c).

In the hierarchical model, both sets of functions are significantly different (Fig. 3.6, right panels). Both the basis and the filters span a wider range of spatial scales (Fig. 3.7a), a result previously unobserved for models trained on non-preprocessed images, and one that is more consistent with physiological data De Valois et al. (1982); Ringach (2002). The learned filters still exhibit clustering along the vertical, horizontal, and diagonal orientations (Fig. 3.7a); this might be an effect of the dataset used, although it is more likely that preprocessing of the images did not fully correct for corner frequency artifacts.

The shapes of the basis functions and the filters are different — the envelopes often include more oscillations of the sinusoid, and the basis functions more closely resemble Gabor functions, although they tend to be less smooth than the sparse coding basis functions. These results more closely resemble simple cell receptive fields, which exhibit a variable number of oscillating subfields Ringach (2002). We also compared the distributions of spatial phases for filters obtained with sparse coding and the hierarchical model (Fig. 3.7b). While sparse coding filters exhibit a strong tendency for odd-symmetric phase profiles, the hierarchical model results in a much more uniform distribution of spatial phases. Although some phase asymmetry has been observed in simple cell receptive fields, their phase properties tend to be much more uniform than sparse coding filters Ringach (2002). Finally, frequency distributions of filters in the hierarchical model are slightly less clustered at the high frequencies than in the sparse coding model (Fig. 3.7c). It has previously been observed that the peak spatial frequency distributions of learned filters do not match physiological measurements (V1 simple cell receptive fields with low spatial frequencies are quite numerous) (van Hateren and van der Schaaf, 1998). The distribution of frequencies learned with the hierarchical model is closer to neural properties, but is still very different from that observed in V1. Recent work has shown that very sparse (in the $L_0$ norm sense) priors on linear basis functions yields much more physiologically realistic filter characteristics, suggesting that such codes might be more appropriate for modeling cortical neurons.

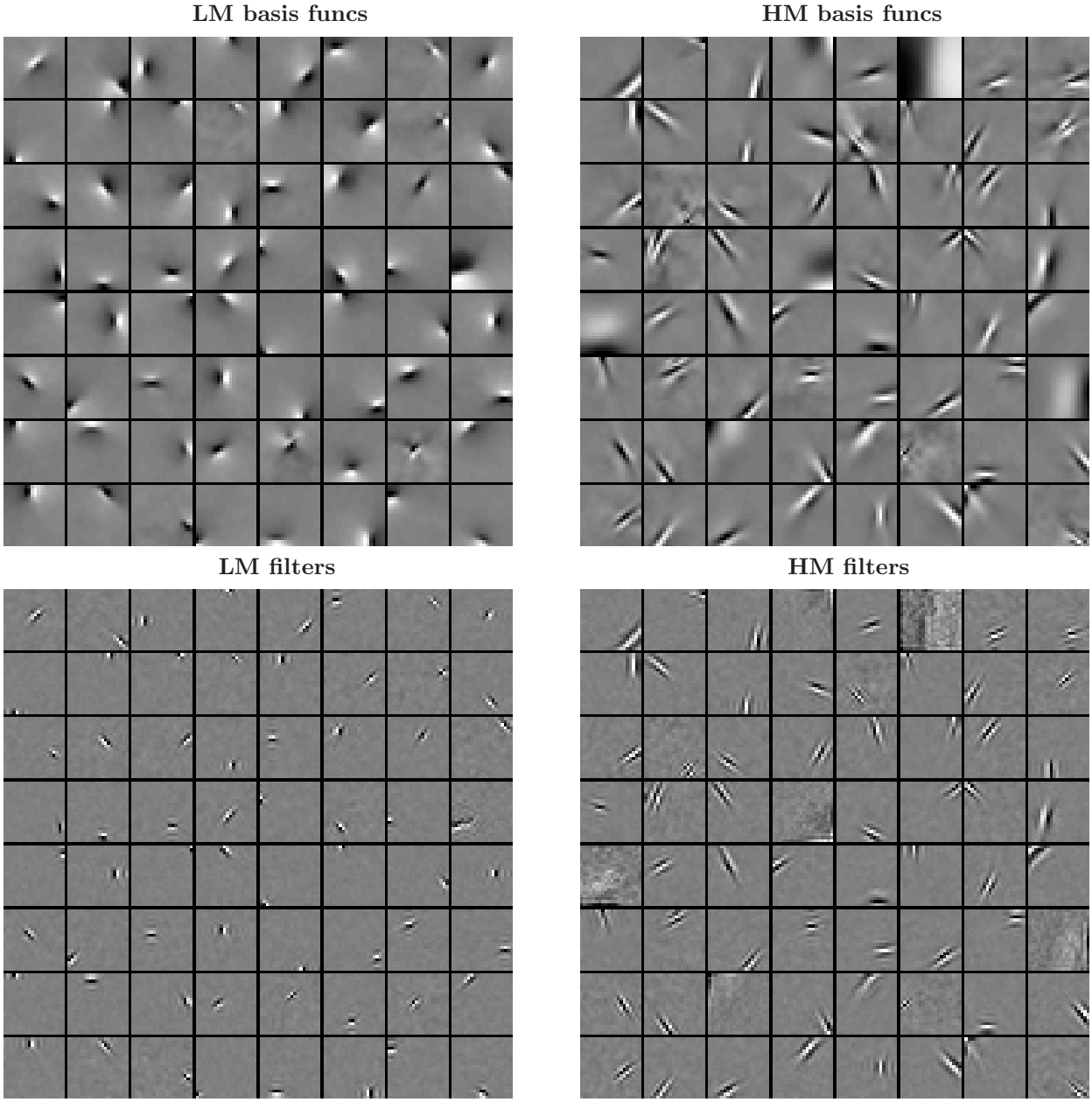


Figure 3.6: The lower-level representations learned by a linear model (LM) and the hierarchical variance model (HM). Shown are subsets of the learned basis functions and the estimates for the filters obtained with reverse correlation.

### 3.3.4 Variance components

What kind of higher-order statistical regularities does this model discover in the data? The learned parameters in the matrix **B** are meaningless on their own because they do not directly correspond to pixel intensity values. However, we can examine how the pattern of learned weights in **B** relates to the linear basis functions whose coefficients they affect. We first analyze the fits of the 2D Gabor function to the linear basis functions; using these parameters we represent the higher-order weights as a function of spatial location and frequency or orientation. A subset of the variance components is shown in Fig. 3.8c. Many reflect a change of variation (enhancement or suppression of activity, relative to the default distribution) localized to a part of the image patch. The component shown in the first panel, for example, represents a pattern of smaller variances for

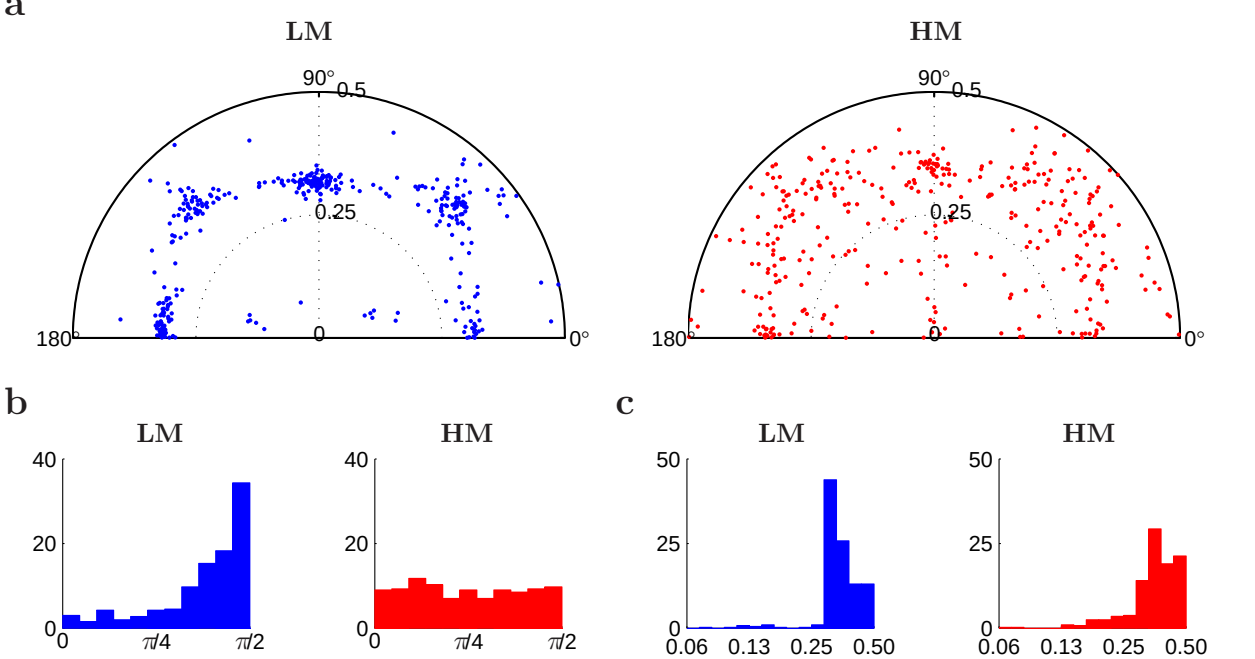


Figure 3.7: ***a***. Scatter plots of peak frequencies and orientations of the Gabor functions fitted to the estimated filters. The units on the radial scale are cycles/pixel and the solid line is the Nyquist limit. Although both sparse coding (LM) and hierarchical model (HM) filters exhibit predominantly high spatial frequencies, the hierarchical model yields a representation that tiles the spatial frequency space much more evenly. The distributions of phases (***b***) and frequencies (***c***) for the fitted Gabor functions, also differ markedly for sparse coding (LM) and hierarchical model (HM) filters. The phase units specify the phase of the sinusoid in relation to the peak of the Gaussian envelope: 0 is even-symmetric, $\pi/2$ is odd-symmetric. The frequency axes are in cycles/pixel.

linear basis functions centered at the top-left of the image patch, and larger variances for basis functions at the top-right, thus coding for spatially localized image contrast. This variance component and the associated coefficient $v_j$ is insensitive for any image structure outside the colored regions. Several other components describe similar localized structure, often shaped by Gabor-like envelopes (c.f. second and fourth panels), though this pattern is not one of alternating luminance values, but rather high and low contrast subfields.

The third variance component in the figure describes a pattern of high variances for one diagonal orientation, and low variance for the orthogonal orientation, but is not localized to any part of the image patch. The fifth panel shows a highly localized and oriented component, with high variance for the vertical orientation along a single contour and lower variance for orthogonal image structure at that location. In the last panel, a variance component codes for a boundary between oriented textures — when its coefficient is large and positive, the input image contains prominent horizontal structure near the top of the patch and vertical structure near the bottom. Note that because the variance coefficients can be positive or negative, the components encode both the indicated pattern and its converse, and the pattern in image structure is reversed when the coefficient is negative. These plots also help explain how a combination of components can describe the structure in each image — the activation of the first component (location) and the third (orientation) describes oriented image structure localized to the top corner of the image patch.

The full set of the variance components, shown in Fig. 3.9, reveals how the higher-order code can represent a wide range of statistical regularities in the image patch. Most components code for areas of localized contrast, which happen to have Gabor-like envelopes, but span a much larger range of spatial scales than the lower-level representation. More localized components are also specific for orientation (although the dot diagrams do not show this). Others represent global contrast levels or orientation, while some are more subtle (but spatially regular, perhaps encoding texture elements) and cannot be clearly categorized. Because the model is non-linear, it is difficult to precisely characterize the behavior of the higher-order units. Arranging the

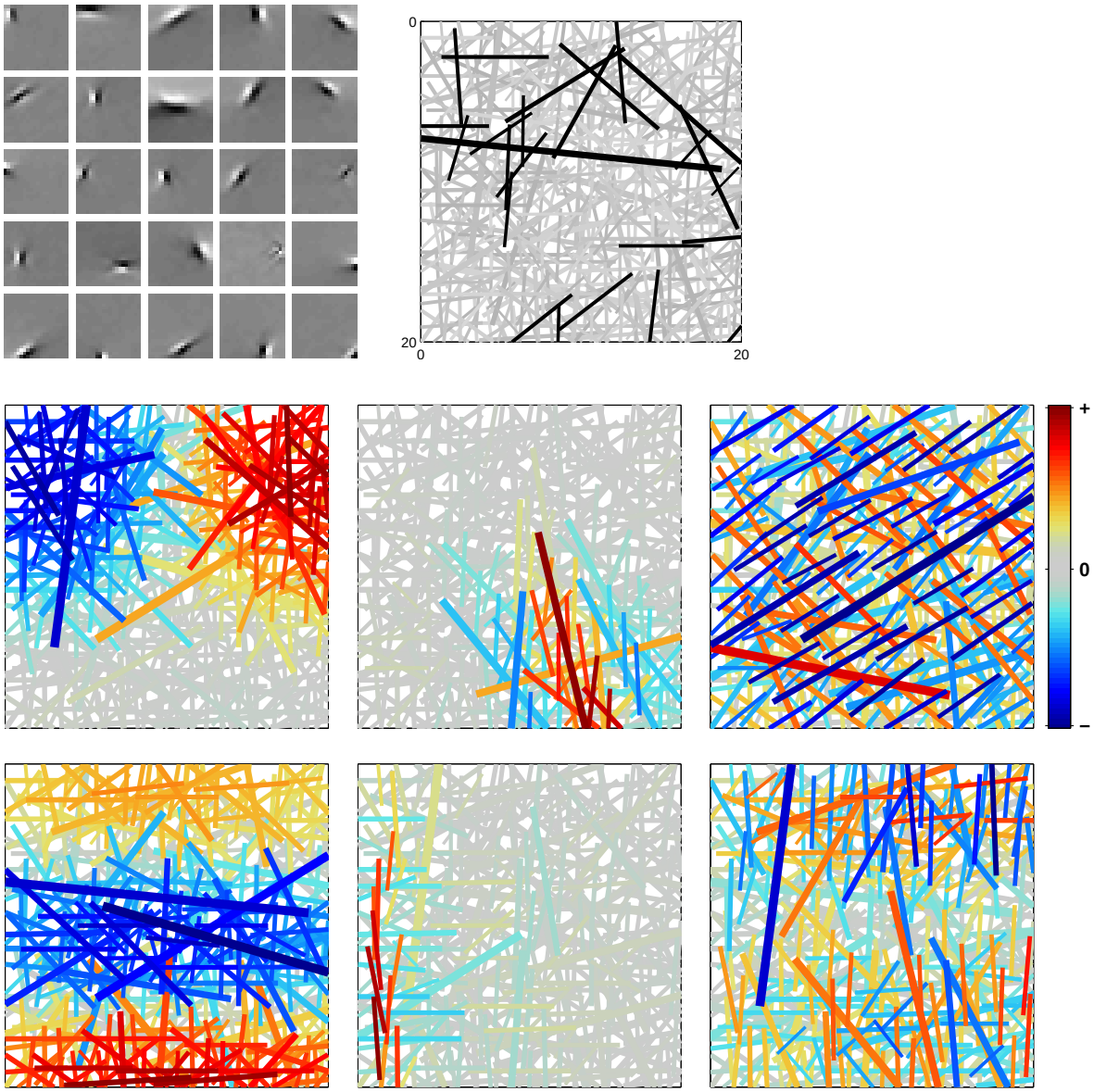


Figure 3.8: A convenient way to analyze the higher-order code learned by the model is to represent each linear basis function as a line in the space of the image patch (20×20 pixels) that reflects the position, orientation, and frequency (length and thickness) of the Gabor-like feature. A subset of 25 linear basis functions (***a***) is shown in black among the full set of 399 linear features (***b***). ***c***. Six representative higher-order basis functions. Each square contains lines corresponding to all the linear basis functions (as in ***b***), colored according to the weights in the column of **B**. Each panel is rescaled independently, colorbar shown on the right.

weights by the spatial location (as in Fig. 3.9) ignores other properties of the linear representation to which these components might be sensitive, such as phase or fine spatial relationships at the scale of overlapping linear basis functions. There are other ways to examine these networks, like looking at image patches that maximally activate the variance coefficients (Karklin and Lewicki, 2003). The difficulty highlights the similar challenge of assigning concise and intuitive functional roles to visual neurons that are sensitive to unknown intrinsic dimensions of visual stimuli.

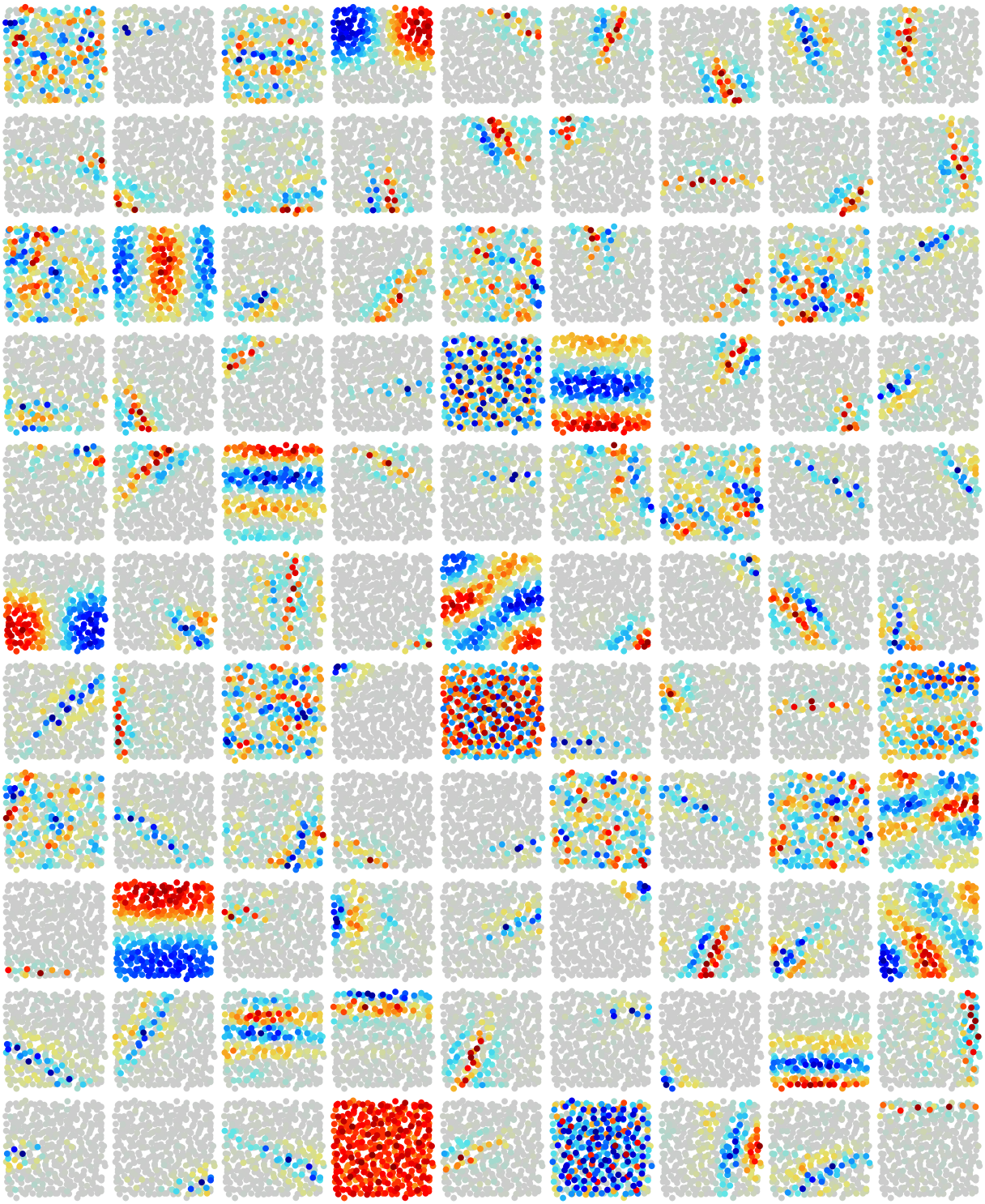

Figure 3.9: Spatial structure captured in the higher-order components in $\mathbf{B}$. The representation is the same as in Fig. 3.8c, but instead of lines, only a single dot is drawn at the center (in image patch space) of the basis function associated with each weight. The colormap is the same as in Fig. 3.8c. (For space considerations, only 99 out of 100 components are shown.) Most components describe spatial relationships and capture co-activation of the linear basis functions localized to a particular area of the image patch; some describe global orientation patterns or contrast levels; the properties of others are not so clear.

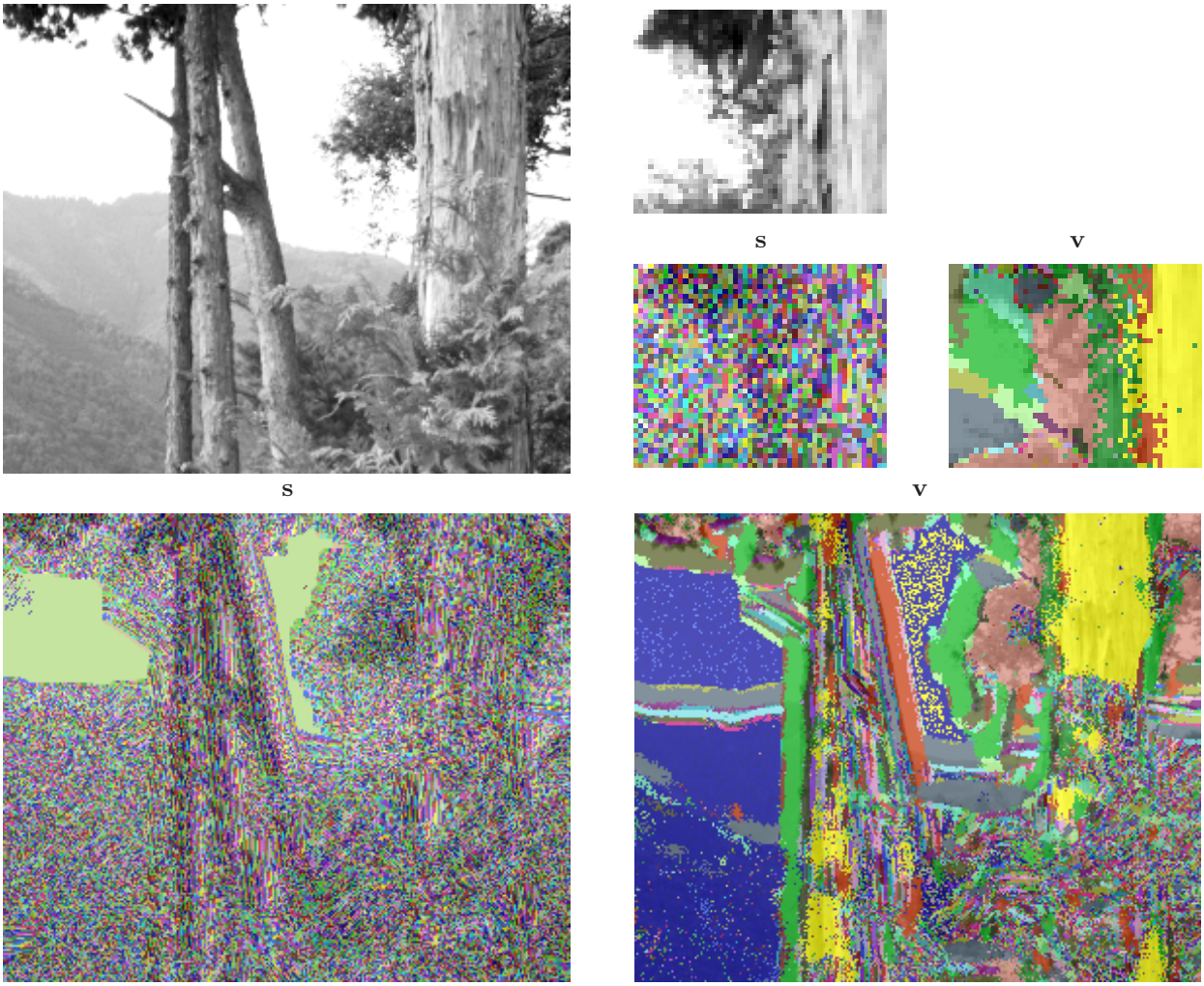


Figure 3.10: Variation of the model's representation over a large natural scene. Image patches were sampled by sliding a window across a natural image (top-left). In the "winner maps" (**s** - bottom left, **v** - bottom right), each color uniquely identifies the maximally active coefficient for the image patch centered at that location. The three smaller panels show a magnified area of the image. The linear representation **s** changes radically over the image, while **v** varies more slowly, resulting in a more homogeneous "winner map". For example, over most of the tree bark on the right of the image, a single unit in **v** is most active, in spite of large variations in the raw visual pattern, suggesting that it captures more invariant properties of the scene.

### 3.3.5 A more invariant code

Another interesting observation is that, although the model is trained on unordered patches randomly sampled from many images, the activity of the variance coefficients is much more invariant across a large natural scene. Fig. 3.10 illustrates the variation of the linear representation and the higher-order code by plotting the identity of the maximally active **s** and maximally active **v** as a sliding sampling window is applied to a large natural image. The linear representation is simply the output of linear filters, and changes rapidly from patch to patch — the maximally active coefficient is consistently different from pixel to pixel. On the other hand, the higher-order representation changes more slowly; several coefficients are active across large regions of homogeneous texture, such as the tree bark, the leaves, or the hill-side, while others capture a salient edge (and ignore small variations in its appearance). No assumption of spatial smoothness is built into the model, but the model captures higher-order statistical regularities that are naturally more stable in the visual world, resulting in a more invariant representation.

Although it is not the focus of this thesis, these observations suggest that this model can be very useful for computer vision tasks such as image segmentation or classification. In fact, related but more restricted models have already been successfully applied to segmentation (Lee and Lewicki, 2000) and image denoising (Portilla et al., 2003; Park and Lee, 2004), and have been shown to result in more efficient codes (Buccigrossi and Simoncelli, 1999).

## 3.4 Results on speech sounds

### 3.4.1 Simulation details

We also applied the model to speech data from the TIMIT database[1]. Linear basis functions were adapted to band-pass filtered speech segments of 256 samples (16 msec of 16kHz sound). The number of variance components was set to 100, and the parameters were optimized using stochastic learning on data batches of 1000 for 10000 iterations. The optimal linear basis functions for speech are mostly localized band-pass functions (Lewicki and Sejnowski, 2000); these are shown in Fig. 3.11a. In order to display the weights in the variance components as they relate to the linear code, we first computed the Wigner distributions (WD) of the linear basis functions (Abdallah and Plumbley, 2001; Cohen, 1989) using the DiscreteTFDs MATLAB package (O'Neill, 1999). The Wigner distribution of a basis function is a surface in the time-frequency space that localizes the power of the time-varying function. We took a contour at 95% peak value for each basis function and drew all these contours on a single time-frequency plot (time on the abscissa, 0 to 16 msec, and frequency on the ordinate, 0 to 8kHz, see Fig. 3.11c for two isolated examples). Because the linear basis functions adapted to speech tile most of the the time-frequency space, the contours also exhibit relatively even tiling of the axes.

### 3.4.2 Variance components

A representative set of the learned variance components is shown in Fig. 3.12. Here, as for image data, the shading of each patch corresponds to the value of the weight (colormap as in Fig.3.8c). Some components describe co-activation of linear basis functions of adjacent frequency bands, while others are localized in time within the sample window. Most components capture periodic higher-order structure and regularities across multiple frequencies or time intervals, and a few are tuned specifically to shifts in dominant frequency over the sample window.

### 3.4.3 Invariance on speech

Applied to audio data, the model also captures more abstract properties of the stimulus. In Fig. 3.13, we plot an example speech signal (the sentence "She has a bad flu"), along with the activities of three linear coefficients and three variance coefficients. We emphasize that, as for the images, the model is trained on segments drawn randomly from the dataset, and the values of the coefficients for each sample position in the signal shown in the figure are determined independently. The higher-order representation varies more slowly than responses of the linear filters and captures structural elements that extend well beyond the small sampling window. This may reflect a general property of natural signals – fast fluctuations in their exact values are caused by interactions of underlying physical properties, which themselves change more slowly.

[1] available from the LDC Corpus catalog at `http://www.ldc.upenn.edu/`

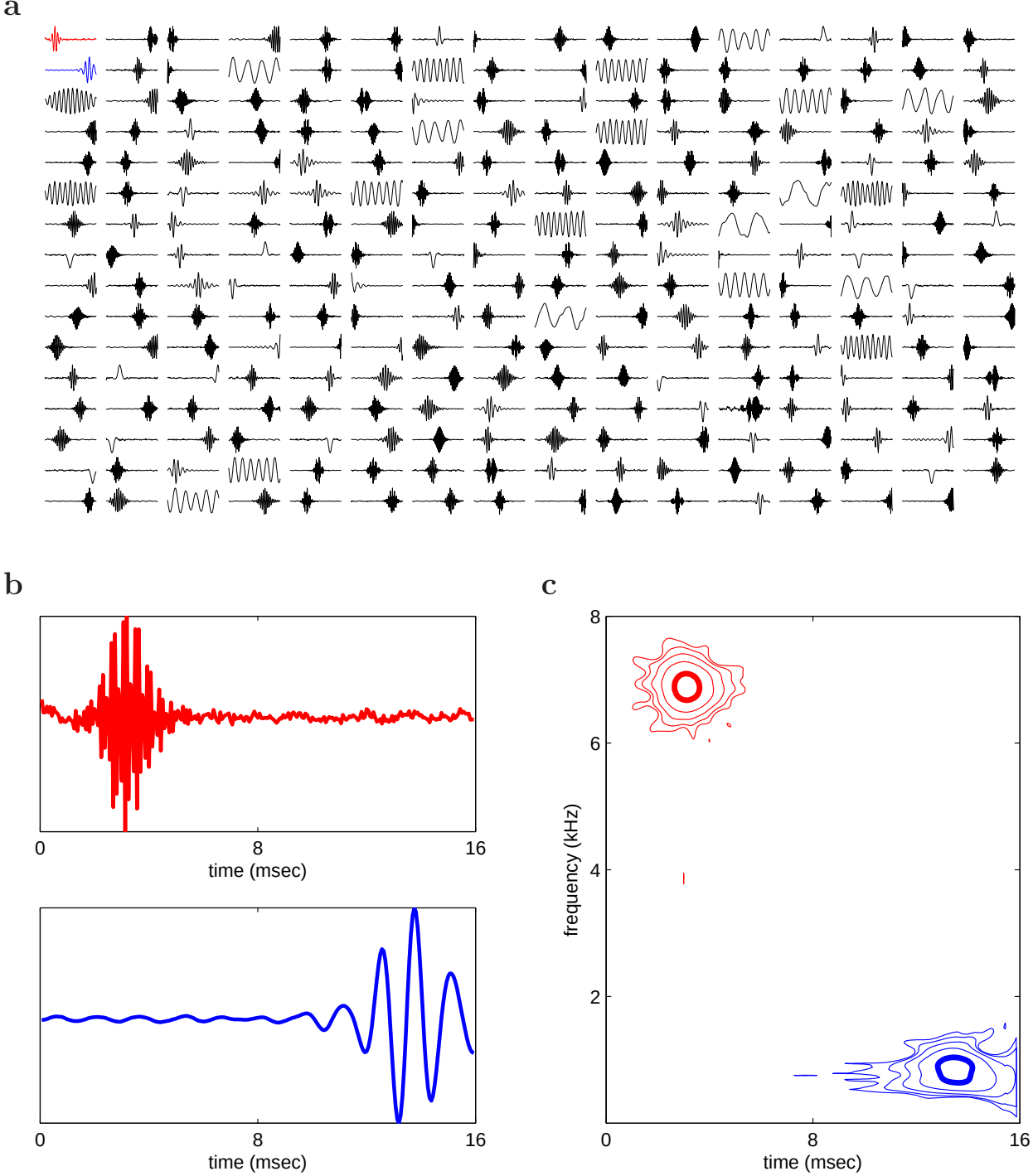


Figure 3.11: ***a***. Basis functions derived from ICA applied to speech data. Each function is rescaled to span the height of the panel. ***b***. Two basis functions, highlighted in (***a***). Each basis function is localized in time and frequency, and relative power is plotted as contours of a Wigner distribution (***c***, see text for details). Inner contour, in bold, is used to represent the basis function in Fig. 3.12.

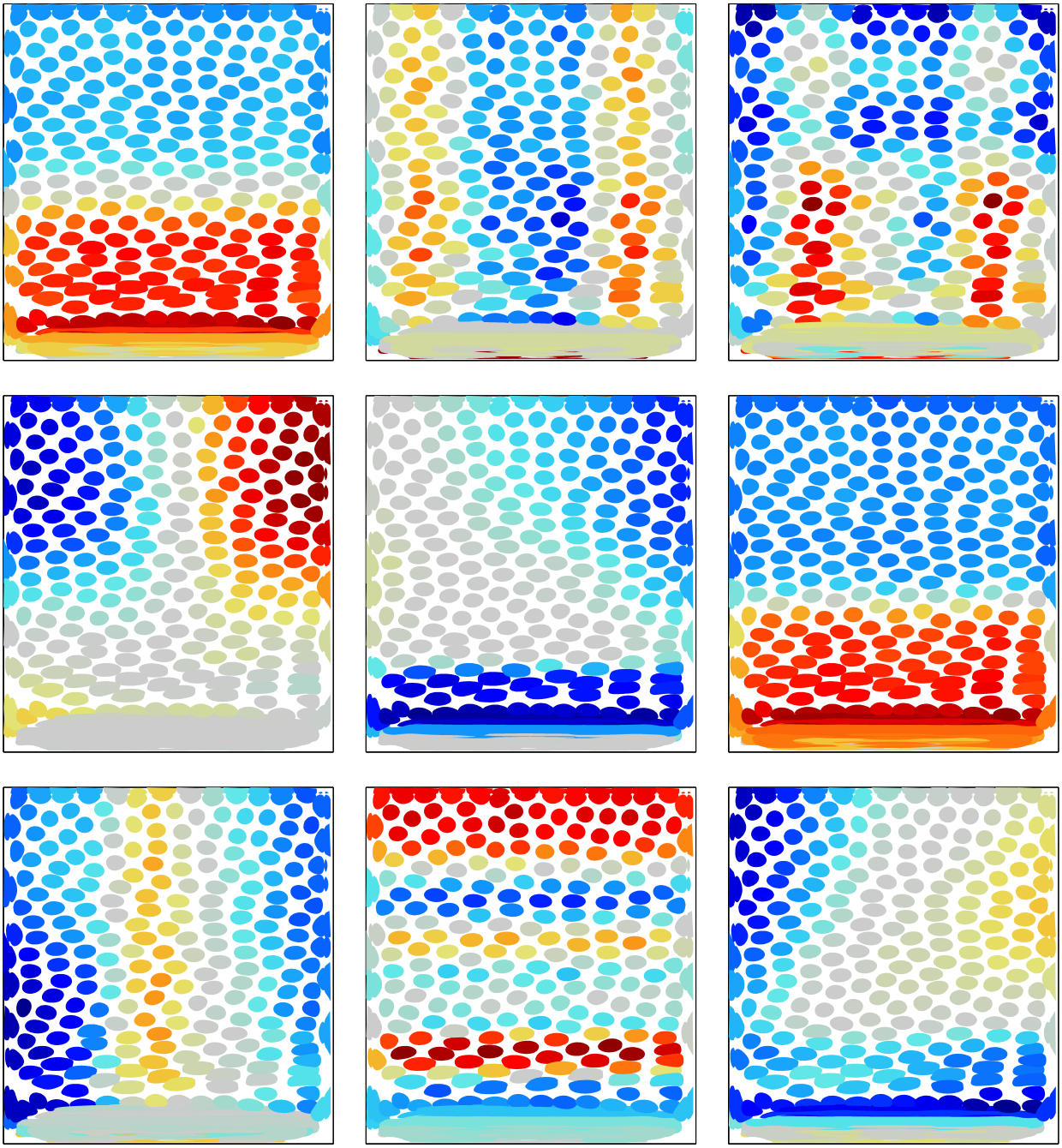

Figure 3.12: A subset of nine variance components of speech. The weights in a column of **B** are plotted as shaded patches in one of the nine panels. Each patch is placed according to the temporal and frequency distribution of the associated linear basis function (see Fig. 3.11c for key) and shaded according to the value of the weight (colormap as in Fig. 3.8c). The axes, as in Fig. 3.11, represent time, 0 to 16 msec, , and frequency, 0 to 8kHz. The variance components form a distributed representation of the frequency of the signal and the location of energy within the sample window. Variance components coding for multiple frequencies might capture harmonic regularities in the speech signal (see text for details).

## 3.5 Discussion

This chapter introduced a novel hierarchical model that can capture complex statistical patterns while making few assumptions about the structure of the data. It accounts for non-linear dependencies observed

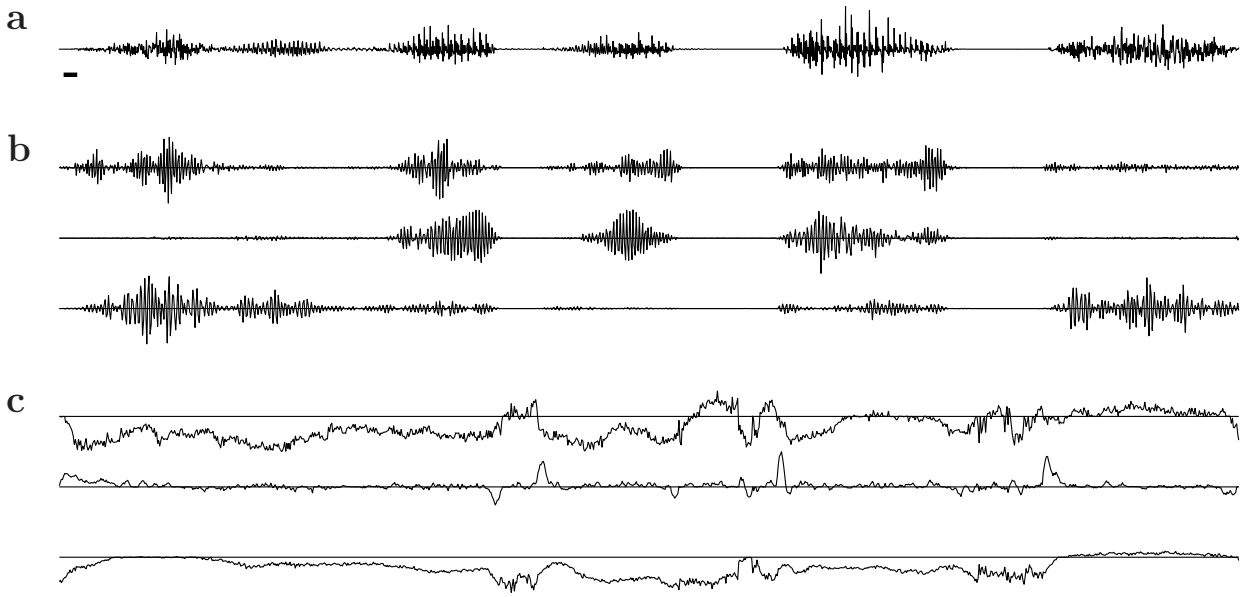


Figure 3.13: The higher-order representation formed by the hierarchical model trained on speech data is more invariant than outputs of linear filters. A sliding window was applied to a speech signal ($\boldsymbol{a}$, size of window indicated by the short bar). At each point, the linear basis function coefficients $\mathbf{s}$ were computed ($\boldsymbol{b}$) and MAP estimates of higher-order coefficients $\hat{\mathbf{v}}$ were inferred ($\boldsymbol{c}$). Values of $\hat{\mathbf{v}}$ change slowly and represent more abstract properties, such as the presence of silence or the onset of vocalization.

in the data, and automatically learns compact representations of higher-order structure that scale well to large dimensional data. Although we analyzed the model primarily in the context of modeling natural images (and showed some results for speech sounds), the model is general and can be applied to any multi-dimensional data that contain rich statistical structure.

Recently, a related set of work has argued that higher-order properties of natural signals change slowly across time or space, and that this spatial and temporal *coherence* can be utilized to extract higher-order structure from the data (Foldiak, 1991; Kayser et al., 2001; Wiskott and Sejnowski, 2002; Hurri and Hyvärinen, 2003). Our results indicate that, at least in some cases, simply learning higher-order statistical regularities in the data leads the model to recover more abstract properties that tend to vary slowly with time or space. This raises the possibility that the explicit computational goal of extracting coherent (slowly changing) parameters is helpful, but not necessary to learning intrinsic structures that underlie the variation in the data.

One result of learning global statistical regularities is that the learned structure is not necessarily obvious; for example, variance components adapted to natural images describe a variety of statistical regularities, some of which are not easily interpreted. This is true for other unsupervised learning models that do not specify in advance the structure to be learned. For example, ICA applied to natural images yields a matrix of basis functions whose functional interpretation has ranged from edge detectors (Bell and Sejnowski, 1997) to models of biological sensory systems (van Hateren and van der Schaaf, 1998). The work presented here suggests that as more powerful unsupervised learning models are developed, the analysis of learned parameters and data representations will gain in importance.

# Chapter 4

# Modeling covariance dependencies

In the previous chapter, I described a hierarchical model that accounted for the observed dependencies among coefficient magnitudes. This was accomplished by allowing the variance of coefficient distributions to change; in data space, this amounts to stretching the joint likelihood along the axes of the linear basis functions. However, the *correlational structure* of the linear coefficients also changes from context to context (Fig. 4.1), and a more rich model should be able to capture these context-specific correlations, as well as the changing variances. In this section, I introduce a generalization of the hierarchical model in which the full covariance matrix of the linear coefficient likelihood is shaped by the latent random variables.

This approach is fundamentally different from the variance-dependence model described above, as well as the related models. Whereas dependencies among *magnitudes* or *variances* of linear coefficients have been incorporated into generative models, energy-based models, Gaussian Scale Mixtures, and temporal coherence models, none of the previous approaches have explicitly modeled non-stationary correlational structure in the data. It can be argued that a correlated distribution in one set of axes is simply a non-isotropic but an uncorrelated distribution in a properly rotated space. For example, random Gaussian variables with a diagonal (stationary) covariance matrix $\mathbf{D}$, when transformed into a new space through a linear transformation $\mathbf{A}$ yield a covariance of $\mathbf{ADA}^T$, which is not necessarily diagonal. Thus, models that change the variance of the coefficients already introduce changing correlations among pixel intensity values of the image. However, we would like to model the observed correlations without relying on a fixed linear transformation, and would like to express correlations in directions not aligned with a fixed set of basis functions.

First, we assume a conditionally Gaussian form for the linear coefficient likelihood. For each data sample, the model should describe the covariance of the coefficient likelihood $\mathbf{C}$ as a function of a set of latent variables $\mathbf{y}$,

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \tag{4.1}$$

$$\mathbf{C} = f(\mathbf{y})\,. \tag{4.2}$$

Because the covariance matrix is not constrained, it can absorb any linear transformation $\mathbf{C} = \mathbf{AC}_s\mathbf{A}^T$. Therefore we will attempt to model directly the covariance matrix of the data $\mathbf{C}$ (i.e. image pixels), though the resulting model can be applied to the distribution of linear coefficients as well.

The key problem is formulating a hierarchical prior, that is choosing a parameterization of the covariance matrix in terms of latent variables, $\mathbf{y}$. As in the model for variances, we would like a distributed representation that captures the intrinsic dimensions of the variation in $\mathbf{C}$ rather than learning a fixed set of different covariance matrices, and these dimensions should be captured in parameters extracted automatically from the data. Because the model now generates the full covariance matrix, the number of free parameters will be much larger; for a covariance matrix of size $D \times D$, the size of the space of possible matrices is $D(D+1)/2$. However, we expect most of this space to be irrelevant for describing natural image structure, with only

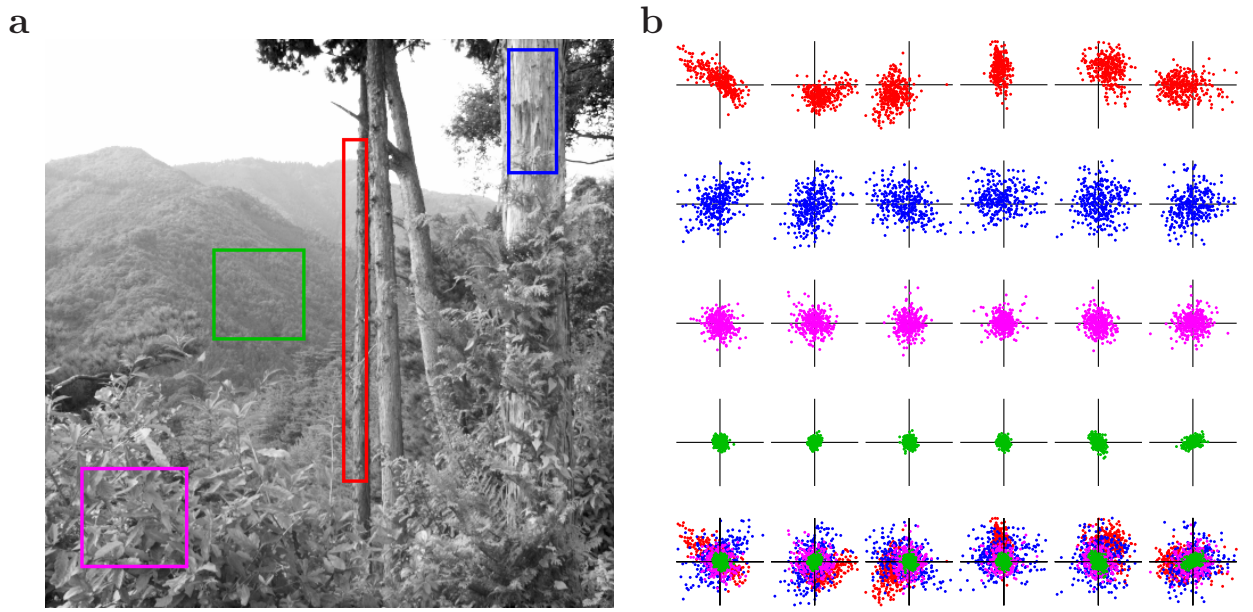


Figure 4.1: Local image distributions in natural scenes show different correlational patterns. ***a***. A natural scene with four distinct regions outlined (Image courtesy of E. Doi). ***b***. Each column shows the joint output of a pair of linear feature detectors or filters from 20×20 image patches sampled from the different scene regions (rows 1-4). Both edges (row 1) and textures (rows 2-4) have high variability. Different visual features yield different distributions, but all of them overlap (row 5) and cannot be used to distinguish between the regions.

a small region containing correlational structures encountered in natural scenes. One of the challenges is formulating a sufficiently powerful model that is not restricted to diagonal matrices, yet is scalable enough to be applied to high-dimensional data like natural images.

## 4.1 Related Work

In addition to the variance-component model described above, other hierarchical models for natural images have incorporated non-stationary variances or scale parameters (Wainwright et al., 2001; Valpola et al., 2004), but non-stationary correlations in image data (to the best of our knowledge) have not been modeled.

However, covariance modeling and estimation is relevant to a number of fields and different parameterizations of the covariance matrix have been explored in signal processing, financial models, kernel machine learning methods, as well as a variety of Bayesian estimation and shrinkage tasks that require sensible priors on data covariance (Dempster, 1972; Wax et al., 1984; Leonard and Hsu, 1992; Daniels and Kass, 1999; Pourahmadi, 2004; Asai et al., 2006). Some of the approaches to modeling covariance include spectral decomposition of the covariance matrix (see Boik, 2002, and citations therein) or its inverse (Dempster, 1972); factorization of the inverse into inverse partial variances and partial correlations (Wong et al., 2003) or Cholesky decomposition factors (Smith and Kohn, 2002); Givens rotation matrices (Yang and Berger, 1994; Daniels and Kass, 1999), matrix-logarithmic transforms (Leonard and Hsu, 1992; Chiu et al., 1996), Cholesky decomposition (Pourahmadi, 1999, 2000), and the standard error-correlations decompositions (Barnard et al., 2000). Many of the relevant techniques are reviewed in (Pourahmadi, 2004; Daniels and Kass, 1999).

One of the main difficulties of parameterizing a covariance matrix is the positive definite constraint that the matrix must satisfy. This often results in unwieldy conditions on the elements of component matrices. Statistical interpretability is another important issue. This is especially critical for this work, where ultimately we would like to use the latent variables and the model's representation to analyze possible strategies of

coding by visual neurons.

The most relevant prior work to our problem is Bayesian estimation of the parameters of a covariance matrix (Leonard and Hsu, 1992; Chiu et al., 1996). These methods aim to provide a sensible prior when estimating a covariance matrix; a parsimonious way of incorporating prior knowledge is essential to efficient estimation of these parameters. Putting a prior directly on the elements of the covariance matrix is problematic because the resulting matrix must be positive definite. One possible approach is to use the Wishart distribution, often employed as a prior in Bayesian covariance modeling (Evans, 1965). This distribution has two sets of parameters, a degree of freedom parameter, and a scale matrix. However, this description does not lend itself well to the structure of our problem – while the scale matrix represents the true covariance of the sample covariance matrices described by the distribution, it does not capture the structure in the variation around this matrix, something we would like to discover and represent. Therefore, below we explore parameterizations that are based on various factorizations of the covariance matrix that permit the learning of structured knowledge.

Leonard and Hsu (1992) proposed the matrix logarithm transformation, $\mathbf{A} = \log(\mathbf{C})$, convenient because any symmetric matrix $\mathbf{A}$, when exponentiated, yields a positive definite matrix (i.e. a valid covariance matrix). The matrix exponential is defined by the series

$$\exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \,, \tag{4.3}$$

and has several useful properties. If we express a symmetric matrix in terms of a spectral decomposition

$$\mathbf{A} = \mathbf{V}\left[\mathtt{diag}(d_1, d_2, \ldots)\right]\mathbf{V}^T \tag{4.4}$$

(where $\mathbf{V}$ collects the eigenvectors and $d_i$ are the eigenvalues of $\mathbf{A}$), then the effect of the matrix exponential is restricted to an element-wise exponentiation of the eigenvalues,

$$\exp(\mathbf{A}) = \mathbf{V}\left[\mathtt{diag}(e^{d_1}, e^{d_2}, \ldots)\right]\mathbf{V}^T \,. \tag{4.5}$$

Thus estimation of parameters in the log-covariance space retains the eigenvectors of the original space. The matrix exponential of the $\mathbf{0}$ matrix yields the identity matrix, which means that zero-centered priors in the log-covariance space favor an identity covariance matrix. This type of prior was proposed by Leonard and Hsu (1992); the upper-diagonal elements of the log-covariance matrix $\mathbf{A}$ were vectorized, and a multivariate Gaussian prior placed on the elements. Chiu et al. (1996) extended this approach by modeling the elements by a linear combination of design matrices,

$$\mathbf{A} = \sum_j y_j \mathbf{A}_j \,. \tag{4.6}$$

If the number of design matrices is small, this can limit the number of parameters (and latent variables if $y_j$ are different for different data points), but does not provide an interpretable description of data structure. Below I introduce a model that is based on this work, but extends it to capture non-stationary correlational patterns using parameters that are clearly interpretable.

## 4.2 Model

### 4.2.1 Definitions

**Data likelihood**. We begin by modeling the pixel data with a multivariate Gaussian probability density and a set of linear basis functions on the log-covariance matrix,

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \qquad\qquad \log \mathbf{C} = \sum y_j \mathbf{A}_j \,. \tag{4.7}$$

Each $\mathbf{A}_j$ is a symmetric matrix of size equal to the covariance matrix. As in previous models, we assume the mean in the likelihood distribution $p(\mathbf{x}|\mathbf{y})$ is zero. It is desirable to relax this assumption, and perhaps future work will tackle this issue, but the current computational approach (using gradient ascent on the latent variable posterior) would deal poorly with the local minima introduced by varying both the mean and the variance of a distribution. Another outstanding question is, in a model flexible enough to place the mean of a non-stationary distribution right at each training sample, what use is encoding the local covariance, rather than reducing the variance and tightening the density around the point?

Any symmetric matrix can be represented as a summation over rank 1 matrices, e.g. by doing a singular value decomposition and computing a weighted sum of outer products of its (unit norm) eigenvectors,

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_{jk} \mathbf{b}_{jk}^T \tag{4.8}$$

(in this case the weights are the eigenvalues). Choosing outer product matrices leads to clear interpretability for the parameters. Consider the case when the final log-covariance is composed of only one matrix $\log(\mathbf{C}) = \mathbf{A}_j$, and its component vectors $\mathbf{b}_{jk}$ are orthogonal. Because matrix exponentiation works only on the eigenvalues of a matrix, it will not affect the *directions* specified by these vectors, instead acting only on their eigenvalues. Thus these vectors represent directions of elongation or contraction both in the original and the exponentiated space, and the scale along each direction is given by $\{e^{w_{jk}}\}$. However, if the set of vectors is not orthogonal, or if multiple component matrices are added $\log(\mathbf{C}) = \sum_j \mathbf{A}_j = \sum_{jk} w_{jk} \mathbf{b}_{jk} \mathbf{b}_{jk}^T$ whose components are not mutually orthogonal, interactions between vectors $\mathbf{b}_k$ must be considered. These will interact by redefining the eigenvectors of the final sum; near-orthogonal vectors will not produce large cross-terms and will not affect the spectral structure of the resulting matrix, while less orthogonal vectors will modify the spectral structure to reflect the cross-terms.

If we use Eqn. 4.8 directly and employ a different set of vectors $\mathbf{b}_{jk}$ for each $\mathbf{A}_j$, we have not reduced the space of parameters (i.e. we would still have to estimate $O(D^2)$ parameters). We will make an assumption that a limited number of directions, if combined in different ways, can explain many common correlational patterns observed in natural scenes. In other words, image distributions can be described by altering the canonical distribution (the spherical Gaussian density) along a limited number of directions though this number might be much larger than the dimensionality of the data (i.e. the number of directions $K$ will be $D \ll K \ll D^2$. All the covariance components will share this pool of possible density-manipulating directions, but each will be free to use a different weighted combination of the vectors,

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T \,. \tag{4.9}$$

Because the coefficients combining $\mathbf{A}_j$ are assumed to be independent, the weights $w_{jk}$ effectively capture correlated changes in the shape of distributions along directions $\mathbf{b}_k$. For example, if elongation of the density along one direction typically co-occurs with a tightening of the density along another direction, both of the vectors $\mathbf{b}_k$ will be linked to a single $y_j$, with weights of opposite signs.

The log-covariance matrix is therefore defined as

$$\log(\mathbf{C}) = \sum_j y_j \mathbf{A}_j = \sum_{jk} y_j w_{jk} \mathbf{b}_k \mathbf{b}_k^T \,. \tag{4.10}$$

To clarify, the weights $w_{jk}$ are required because we would like the variables $y_j$ to encode independent correlational patterns; the vectors $\mathbf{b}_k$ manipulate the distribution along single directions in data space, and these manipulations might (on average) be correlated. The vectors $\mathbf{b}_k$ are fixed to be unit-length, $\|\mathbf{b}_k\| = 1, \forall k$, so the weights also account for the magnitude of modeled deviations in the covariance — large deviations from the canonical distribution are mediated by weights with large magnitudes. The sign of the weights reflect the direction of the effect, allowing a single $y_j$ to increase variation along one direction (positive $w_{jk}$) and decrease variation along a different direction (negative $w_{jk}$). I will refer to the variables

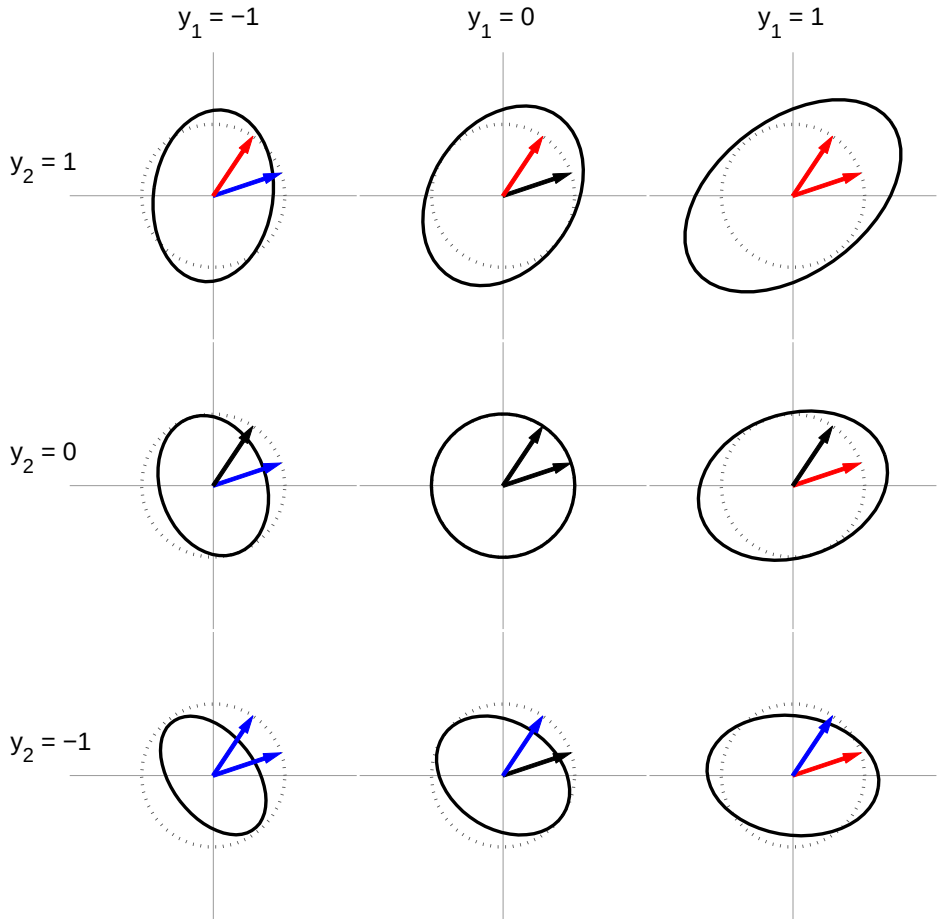


Figure 4.2: A schematic illustrating the effect of covariance coefficients on the covariance structure of the data. Each ellipse shows an iso-probability contour for the 2-dimensional Gaussian distribution $p(\mathbf{x}|\mathbf{y}, \theta)$, where $\theta$ collects model parameters. This toy model is comprised of two covariance components, each linked with weight $w_{jk} = 1$ to only one vector $\mathbf{b}_1$ or $\mathbf{b}_2$ (i.e. the weight matrix is a two-dimensional identity matrix $\mathbf{W} = \mathbf{I}_2$). Positive covariance coefficients act to expand (red arrows), and negative act to contract (blue arrows) the density along the associated directions in image space. When $\mathbf{y} = 0$, the distribution is the default spherical Gaussian density (center panel).

in the vector $\mathbf{y}$ as *covariance coefficients*. The effect of the covariance coefficients on the joint distribution is illustrated in Fig. 4.2.

When latent variables are set to zero, the covariance matrix is equal to the identity matrix $\mathbf{I}$, corresponding to the canonical distribution, the spherical Gaussian. Before training this model on image data, we pre-process the images by "whitening" them, eliminating global correlational structure. Model-defined covariance will therefore only reflect deviations from the global statistics of the data ensemble. This is slightly more natural than the "default" case in the *variance* model described above, which yields the covariance $\mathbf{A}\mathbf{A}^T$ when all the higher-order coefficients are zero. This is not necessarily equal to the identity matrix (unless the linear basis is orthonormal), and cannot be so when $\mathbf{A}$ is over-complete (though when the number of dimensions is large, even an over-complete set of vectors tends to close to orthogonal). The covariance model, on the other hand, allows us to use an over-complete set of linear features $\mathbf{b}_k$ and keep the default covariance as the identity and the canonical distribution a spherical multivariate Gaussian. Another important benefit of this model is that, even when the number of linear features is over-complete, there is no marginalization over the linear coefficients (they are not explicitly computed in the model). However, this does mean that any linear encoding and correspondence to fairly linear neurons like simple cells is implicit in model computations. This point is discussed in more detail in the next chapter.

The log-likelihood is defined as

$$L = -\frac{1}{2}\log\det\mathbf{C} - \frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} \tag{4.11}$$

Using the relation $\log(\det\mathbf{C}) = \mathrm{Tr}(\log\mathbf{C})$ (see the appendix for details),

$$L = -\frac{1}{2}\mathrm{Tr}\left(\sum y_j\mathbf{A}_j\right) - \frac{1}{2}\mathbf{x}^T\exp\left(-\sum_j y_j\mathbf{A}_j\right)\mathbf{x} \tag{4.12}$$

$$= -\frac{1}{2}\sum_{jk} y_j w_{jk} - \frac{1}{2}\mathbf{x}^T\exp\left(-\sum_{jk} y_{jk}w_{jk}\mathbf{b}_k\mathbf{b}_k^T\right)\mathbf{x}\,. \tag{4.13}$$

Because the vectors $\mathbf{b}_k$ are unit-norm, they drop out from the log-determinant term.

**Latent variable prior**. The coefficients $\mathbf{y}$ are the latent variables in the model, while $\mathbf{b}_k$ and the weights $w_{jk}$ are parameters to be estimated. As above, we can assume the independent Laplacian density as a prior on $\mathbf{y}$,

$$\log p(\mathbf{y}) = \sum_j \log p(y_j) \propto -\sum_j |y_j|\,. \tag{4.14}$$

Such a symmetric prior makes an implicit assumption that for a given correlational pattern, as defined by $\mathbf{A}_j$, the positive and negative coefficients are equally probable; and more significantly, it assumes that both sets of patterns (those associated with positive and negative coefficients) comprise an optimal set of covariance components. Such assumptions of symmetry were also made in linear models, but in this model image structures represented by the two polarities of the coefficients are quite different. Fig. 4.2 illustrates this problem: one covariance component represents distributions to the right and to the left of the middle panel, depending on the sign of its coefficient, and another morphs distributions into those above and below the middle panels. Thus the model assumes a symmetry that might not exist in the data — in fact the model has no way of capturing, for example, only the types of distributions in the top center panel — and the learned representations might not be optimal for the training data.

This issue is also relevant to the variance model introduced in the previous chapter. Below I explore this issue with simulations designed to test the goodness of the statistical models, but current methods for fitting the models make several approximations that preclude conclusive results. In the future, improved methods for estimating model parameters might resolve these issues; for now, except where specifically stated, the simulations were performed using the Laplacian (symmetric) prior on the covariance coefficients.

### 4.2.2 Latent variable inference

#### Exact gradient

The exact gradient is computed using the directional derivative of the matrix exponential. The directional derivative, w.r.t. an arbitrary matrix $\mathbf{Q}$ is defined as

$$\nabla_{\mathbf{Q}} e^{\mathbf{A}} = \mathbf{V}\left[(\mathbf{V}^{-1}\mathbf{Q}\mathbf{V}) \odot \Phi(\mathbf{A})\right]\mathbf{V}^{-1} \tag{4.15}$$

where $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ is the spectral decomposition, '$\odot$' is the element-wise (Hadamard) product of two matrices, and $\Phi(\mathbf{A})$ is composed of

$$[\Phi(\mathbf{A})]_{ij} = \begin{cases} \frac{e^{d_i} - e^{d_j}}{d_i - d_j} & i \neq j \\ e^{d_i} & i = j \end{cases}\,, \tag{4.16}$$

where $d_i$ is the $i^{th}$ element of the diagonal of $\mathbf{D}$ (see Eqn. 144 in Najfeld and Havel, 1995).

Each variable $y_j$ affects the log-covariance matrix linearly through $\mathbf{A}_j$, so we use the directional derivative $\nabla_{\mathbf{A}_j} \exp(\mathbf{A})$ to obtain the gradient of the log-likelihood function,

$$\frac{dL}{dy_j} = \frac{1}{2}\left(\mathbf{x}^T\left(\mathbf{V}\left[(\mathbf{V}^{-1}\mathbf{A}_j\mathbf{V}) \odot \Phi(\mathbf{A})\right]\mathbf{V}^{-1}\right)\mathbf{x} - \mathrm{Tr}(\mathbf{A}_j)\right) \tag{4.17}$$

Note that this requires the computation of the spectral decomposition of $\mathbf{A} = \sum y_j \mathbf{A}_j$ at every update of $\mathbf{y}$ and is quite slow for large matrices.

### Approximating using series expansion

We can use the series approximation to the matrix exponential to speed up computation (specifically, we would like to avoid having to recompute the spectral decomposition of $\mathbf{A}$ at each iteration). From Eqn. 4.3, the second term of the log-likelihood can be expressed as,

$$-\frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} = -\frac{1}{2}\mathbf{x}^T e^{-\mathbf{A}}\mathbf{x} = -\frac{1}{2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{A}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{A}\mathbf{x} - \frac{1}{6}\mathbf{x}^T\mathbf{A}\mathbf{A}\mathbf{A}\mathbf{x} + \dots\right). \tag{4.18}$$

Let $\mathbf{B}$ be the matrix collecting vectors $\mathbf{b}_k$ and $\mathbf{w}_j$ be the column vector of weights $\mathbf{w}^j = (w_{j1}, w_{j2}, \dots, w_{jK})^T$. Then the gradient w.r.t. the approximate (series-defined) log-likelihood is

$$\frac{d\hat{L}}{dy_j} = \frac{1}{2}\mathbf{w}_j^T\left(-\mathbf{1} + \left(\mathbf{B}^T\mathbf{x}\right)^2 - \left(\mathbf{B}^T\mathbf{x}\right) \odot \left(\mathbf{B}^T\mathbf{A}\mathbf{x}\right) + \frac{1}{3}\left(\mathbf{B}^T\mathbf{x}\right) \odot \left(\mathbf{B}^T\mathbf{A}\mathbf{A}\mathbf{x}\right) - \dots\right), \tag{4.19}$$

where $\mathbf{1}$ is a vector of length $K$ containing all 1s (see the appendix for derivation). This is easy to implement in MATLAB (and can even be done without any for-loops!). In practice, we truncated this series and others to the fifth order and smaller (computing high order terms does not add much computational burden on top of the early terms in the series). Note that other, faster or more stable approximations based on alternative series expansions exist (Najfeld and Havel, 1995); exploring these is a worthwhile direction for future research.

This form makes more clear the computation underlying inference in this model and its relationship to feed-forward computation and classical models for non-linear neurons. We know the model attempts to steer the covariance matrix to cover a given data point. Mechanistically, inference in the model approximately corresponds to projecting the stimulus onto the linear features $\mathbf{B}\mathbf{x}$, squaring the result, and comparing it to 1, $(\mathbf{B}\mathbf{x})^2 - \mathbf{1}$; the result then drives the value of $y_j$ up or down depending on the associated weights $\mathbf{W}^j$. Higher order terms that include the full sum inside $\mathbf{A}$ ($\mathbf{B}^T\mathbf{A}\mathbf{x}$, etc.) reflect the covariance that is already accounted for given the current state of $\mathbf{y}$, and act to cancel the effect of the first order term.

This computation is similar to classical "energy" models of complex cells, in which the stimulus is projected onto two features, and the resulting values are squared and summed to give the neuron's output. Here, however, the number and relative influence of the different projections are determined by the weights $w_{jk}$, learned automatically from the data.

### Closed-form approximate solution

We might want to approximate the solution obtained by iterative gradient ascent with a single step "feed-forward" solution. This can be useful when inference must be performed quickly (e.g. in online applications), and it might provide a more direct link between computation in the model and fast feed-forward neural processing of visual information. Although an analytical solution for the MAP estimate $\hat{\mathbf{y}}$ cannot be computed, there are several ways to obtain estimates.

*Single sample log-covariance.* One approach is to use the single-sample covariance estimate,

$$\hat{\mathbf{C}} = \mathbf{x}\mathbf{x}^T\,, \tag{4.20}$$

and compute a set of coefficients that best approximate this estimate, such that $\hat{\mathbf{y}} \approx \log(\hat{\mathbf{C}})$. However, the single-sample covariance is singular and has only a single eigenvector with a non-zero eigenvalue (lying along the vector $\mathbf{x}$); all the other dimensions have no volume and the ideal encoding should tighten all these dimensions while matching the scale of the input. During MAP inference, the zero-centered prior prevents this divergence by penalizing extreme values of $\mathbf{y}$.

If we augment the single-sample covariance matrix with a small variance ($\epsilon$) isotropic covariance model, then

$$\hat{\mathbf{C}} = \mathbf{x}\mathbf{x}^T + \epsilon\mathbf{I}\,. \tag{4.21}$$

The eigenvalues of this matrix are $(\|\mathbf{x}\|^2 + \epsilon, \epsilon, \ldots, \epsilon)$ and the eigenvectors are comprised of $\mathbf{x}/\|\mathbf{x}\|$ and an arbitrary set of complimentary orthogonal vectors. The matrix logarithm has the same eigenvectors and its eigenvalues are given by $(\log(\|\mathbf{x}\|^2 + \epsilon), \log\epsilon, \ldots, \log\epsilon)$. Because the model employs a set of linear basis function in the space of log-covariance matrices, we can compute the set of coefficients that approximate this non-singular log-covariance matrix,

$$\sum_j \hat{y}_j \mathbf{A}_j \approx \log\hat{\mathbf{C}}\,. \tag{4.22}$$

We vectorize the basis functions ($\mathbf{a}_j = \mathrm{vec}(\mathbf{A}_j)$), vectorize the log-covariance matrix, and estimate the vector $\mathbf{y}$ that best approximates it,

$$[\mathbf{a}_1; \mathbf{a}_2; \ldots; \mathbf{a}_J]\,\mathbf{y} = \mathrm{vec}(\log\hat{\mathbf{C}}) \tag{4.23}$$

$$\hat{\mathbf{y}} = [\mathbf{a}_1; \mathbf{a}_2; \ldots; \mathbf{a}_J]^\dagger\,\mathrm{vec}(\log\hat{\mathbf{C}})\,. \tag{4.24}$$

This relies on minimizing the squared error in the log-covariance space (weighting all the elements of log-covariance matrix equally), which might not be correct given the structure of the space — some deviations in the elements of the log-covariance affect the log-likelihood more severely that others.

*Matrix exponential approximation.* Another approach is to assume that the vectors $\mathbf{b}_k$ are almost orthogonal (in a high dimensional space, they will be approximately orthogonal), ignore the cross terms, and deal with log-likelihood terms where the exponential is evaluated element-wise instead of relying on the matrix exponential computation. Let $\mathbf{Z}$ be a diagonal matrix with elements $z_{kk} = \sum_j y_j w_{jk}$ and let the matrix $\mathbf{B}$ collect the column vectors $\mathbf{b}_k$. Then

$$L = \log p(\mathbf{x}|\mathbf{y}) = -\frac{1}{2}\mathrm{Tr}(\mathbf{Z}) - \frac{1}{2}\mathbf{x}^T e^{-\mathbf{B}\mathbf{Z}\mathbf{B}^T}\mathbf{x}\,. \tag{4.25}$$

If $\mathbf{B}$ is orthonormal (in fact its component vectors are normalized but not exactly orthogonal), then we can rewrite this as

$$L = -\frac{1}{2}\mathrm{Tr}(\mathbf{Z}) - \frac{1}{2}\mathbf{x}^T\mathbf{B}[\mathtt{diag}(e^{-z_{11}}, e^{-z_{22}}\ldots)]\mathbf{B}^T\mathbf{x} \tag{4.26}$$

$$= -\frac{1}{2}\sum_j y_j \sum_k w_{jk} - \frac{1}{2}\sum_k \left(e^{-z_{kk}}[\mathbf{B}^T\mathbf{x}]_k[\mathbf{B}^T\mathbf{x}]_k\right)\,. \tag{4.27}$$

The derivative with respect to $y_j$ is

$$\frac{\partial L}{\partial y_j} \propto -\sum_k w_{jk} + \sum_k \frac{\partial z_{kk}}{\partial y_j} e^{-z_{kk}}[\mathbf{B}^T\mathbf{x}]_k^2 \tag{4.28}$$

$$\propto -\sum_k w_{jk} + \sum_k w_{jk} e^{-z_{kk}}[\mathbf{B}^T\mathbf{x}]_k^2\,. \tag{4.29}$$

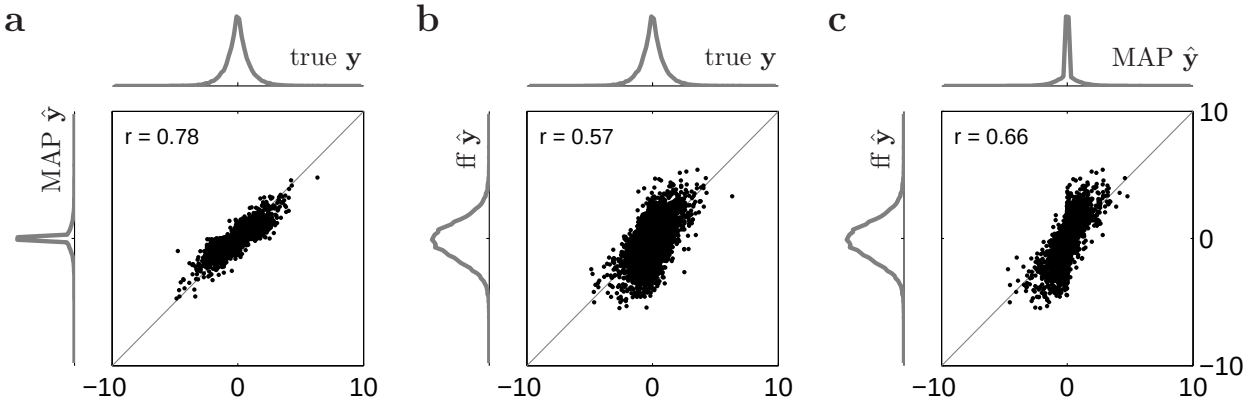


Figure 4.3: True generating values of latent variables (true $\mathbf{y}$) compared to MAP estimates (MAP $\hat{\mathbf{y}}$) and their feed-forward approximation (ff $\hat{\mathbf{y}}$) obtained using Eqn. 4.34 in a synthetic dataset (sampled directly from the model). The dimensionality of the data was 99, the dimensionality of $\mathbf{y}$ was 20, and model parameters were drawn from standard normal distributions. We compared MAP estimates to the true values (***a***), the feed-forward approximation to the true values (***b***), and the feed-forward approximation to the MAP estimates (***c***). Correlation coefficients are given in the insets. The MAP values were highly correlated with the true values, and were much more sparse, as expected (the MAP distribution should not necessarily match the prior, as it picks out the maximum of the posterior). The feed-forward approximation was correlated to the MAP values (***c***), though it was much less sparse than either the MAP estimates or the true values.

Setting this to 0, we get the following equality at the optimal value of $y_j$,

$$\sum_k w_{jk} = \sum_k w_{jk} e^{-z_{kk}} [\mathbf{B}^T \mathbf{x}]_k^2 \,. \tag{4.30}$$

One solution is given when $e^{-z_{kk}}[\mathbf{B}^T\mathbf{x}]_k^2 = 1$, $\forall k$. This leads to the estimate

$$e^{z_{kk}} = [\mathbf{B}^T\mathbf{x}]_k^2 \tag{4.31}$$

$$z_{kk} = \log([\mathbf{B}^T\mathbf{x}]_k^2) \tag{4.32}$$

$$[\mathbf{W}\mathbf{y}]_k = \log([\mathbf{B}^T\mathbf{x}]_k^2) \tag{4.33}$$

$$\hat{\mathbf{y}} = \mathbf{W}^\dagger \log([\mathbf{B}^T\mathbf{x}]^2) \tag{4.34}$$

where $\mathbf{W}$ is the matrix of all weights $w_{jk}$ and $\mathbf{W}^\dagger$ is its pseudo-inverse. This approximation essentially reformulates the covariance component model as the variance component model described in the previous chapter, with log-variances along projections $\mathbf{b}_k$ defined through a linear transform (here $\mathbf{W}$).

Neither of the two approximate feed-forward encodings described above incorporate the sparse prior on $\mathbf{y}$. In order to compute feed-forward estimates that are also sparse, we could employ methods developed for linear models with sparseness constraints, such as basis pursuit methods (Chen et al., 2001) or shrinkage estimators (Hyvärinen, 1999). These methods could be incorporated by revising the cost function to include a penalty for non-sparse solutions, e.g. $L = \| \log([\mathbf{B}^T\mathbf{x}]^2) - \mathbf{W}\mathbf{y}\|_2^2 + \phi(\mathbf{y})$.

We tested the approximations derived above (without the sparseness constraint) on synthetic data, in which the true generating values of coefficients were known (they were drawn from independent Laplacian distributions), and on natural image data, with model parameters adapted to a large set of images. The estimates based on independent projection and log-variance computation (Eqn. 4.34) much better approximated the MAP values than the single-sample covariance estimate, Eqn. 4.24. (Further work is necessary to provide a theoretical justification for these results. It is likely that small-error approximations in the space of log-covariance matrix *elements* do not reflect the structure of the problem.) The correlations between the estimates and the iteratively computed MAP values are shown in Figs. 4.3 and 4.4.

For synthetic data, we can also compare the MAP estimates to the generating values of $\mathbf{y}$ (Fig. 4.3a).

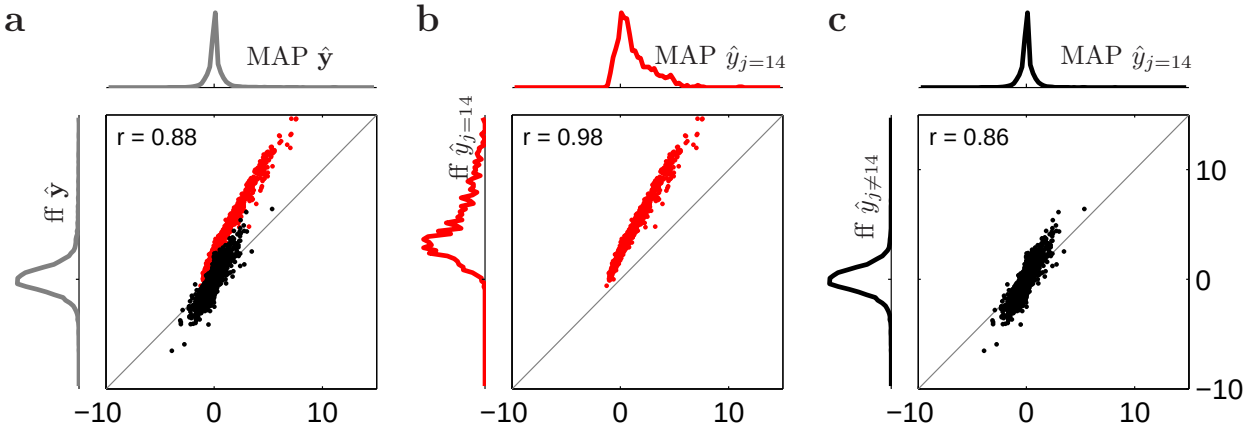


Figure 4.4: Feed-forward approximation to MAP estimates derived using Eqn. 4.34 compared to MAP estimates obtained using gradient ascent for a *model trained on 10×10 image patches* (model structure as in Fig. 4.3). Because the covariance component associated with global changes in contrast ($y_{14}$, the DC variance unit) exhibited a different distribution of MAP values, it is drawn in red (and separately in ***b***). When the model is adapted to image data, the feed-forward approximation is better (though more biased) than when model parameters are random (Fig. 4.3***c***) because the model's representation has spread out to represent independent covariance patterns.

The MAP estimate corresponds fairly well to the true values (correlation coefficient is 0.78), while the correspondence between the MAP estimate and feed-forward approximation is more modest (Fig. 4.3c, r=0.66). As expected, the distribution of MAP values is much more sparse than the Laplacian prior. The feed-forward estimate is approximately Gaussian and does not capture the large concentration of MAP values near zero.

A similar pattern is observed for the model trained on natural images, but the structure of the learned parameters helps constrain the feed-forward estimates and at the same time introduces bias. On average, the feed-forward estimates are highly correlated with the MAP values (Fig. 4.4a). However, the MAP values of the coefficient responsible for global contrast has a different distribution, and feed-forward estimates of this coefficient, though highly correlated with the MAP values, are not on the same scale (Fig. 4.4b). The feed-forward approximation gives good estimates for the rest of the coefficients ($r = 0.86$), though the lack of the prior means that they are not sparse (Fig. 4.4c).

These simulations indicate that the proposed feed-forward approximations perform fairly well, and suggest that they can be useful when fast inference is required. Their performance can be improved further by incorporating sparseness constraints and by taking into account dependence among non-orthogonal covariance components. The asymmetric distribution of the DC variance coefficient suggests that the Laplacian prior is not appropriate for the DC contrast unit. In linear models like ICA, a similar effect is observed with the DC luminance coefficients, which are also not sparse, and this component is typically excluded during training. Thus it is likely that we can improve the fit of the current model by using a flexible prior for this coefficient, or modifying the model to exclude this direction in the space of covariance matrices.

### Mapping approximate inference to neural computation

A feed-forward approximation to the inference is also useful for mapping computation in the model to processing in the visual system. The visual task we are modeling — a course, rapid processing of images that appear at the fovea — is performed very quickly, leaving little time for signals to propagate or bounce among cortical areas. The underlying neural mechanisms must integrate information across the visual field in a feed-forward process, which might also be gated by local inhibitory connections.

According to Eqn. 4.34, these computations consist of projecting onto a set of linear features $\mathbf{B}^T\mathbf{x}$, roughly

corresponding to activation of simple cells in V1 (and the features $\mathbf{b}_k$ do resemble simple cell receptive fields). The output of this first layer of neurons is then squared, and the value compared to 1: the larger the value, the more positive the term $\log([\mathbf{B}^T\mathbf{x}]^2)$ will be, and the smaller the value, the more negative it will be. This information is then integrated by the second later of neurons, $\mathbf{y}$, weighted by the connection strengths represented by the matrix $\mathbf{W}^\dagger$. What do these weights look like? In our implementation, the matrix $\mathbf{W}$ does not span the space of all linear features (the number of covariance dimensions is typically much smaller than the expanded space of $\mathbf{b}_k$'s), and the pseudo-inverse produces a matrix with weights qualitatively similar to the original matrix.

Any sparsification, which is part of the model but not included in the approximate inference schemes, results from lateral inhibition among the neurons at the second stage.

### 4.2.3 Parameter estimation

Model parameters are estimated by maximizing the data log-likelihood; collected in $\theta = \{w_{jk}, \mathbf{b}_k\}$, the ML estimates are

$$\hat{\theta} = \arg\max \langle \log P(\mathbf{x}_n|\theta) \rangle_n \tag{4.35}$$

(we drop the $n$ below). Here it is again possible to do this in a Bayesian framework and specify priors on these parameters, but we will only seek the maximum likelihood estimates. As above, we need to marginalize over the latent variables $\mathbf{y}$,

$$p(\mathbf{x}|\theta) = \int p(\mathbf{x}|\theta, \mathbf{y}) p(\mathbf{y}) d\mathbf{y} \,. \tag{4.36}$$

However, because of the highly non-linear form of the likelihood as well as the non-Gaussian prior $p(\mathbf{y})$, this integral is intractable and as in the previous chapter, we again resort to approximating it with the value at the maximum,

$$p(\mathbf{x}|\theta) \approx p(\mathbf{x}|\theta, \hat{\mathbf{y}}) p(\hat{\mathbf{y}}) \qquad \text{where} \qquad \hat{\mathbf{y}} = \arg\max p(\mathbf{y}|\mathbf{x}, \theta) \,. \tag{4.37}$$

As above, this approximation can be avoided by using sampling methods, such as Markov chain Monte Carlo, to approximate the posterior distribution (over both the latent variables and model parameters). In practice, the MAP estimation was effective at recovering parameters, so we leave the development of alternative methods for future work.

We used gradient ascent to obtain the maximum likelihood estimates for these parameters. As in the estimation of the latent variables, we can use the exact expression for the gradient, which requires the spectral decomposition of $\mathbf{A}$ at each iteration, or a faster approximation based on the series expansion. The approximate gradients employed in the simulations below (and based on the series expansion) are

$$\frac{\partial \hat{L}}{\partial \mathbf{b}_k} = \Big( \sum_j w_{jk} y_j \Big) \left( \mathbf{I} + \mathbf{x}\mathbf{x}^T - \frac{1}{2} \left( \mathbf{x}\mathbf{x}^T\mathbf{A} + \mathbf{A}\mathbf{x}\mathbf{x}^T \right) + \frac{1}{6} \left( \mathbf{x}\mathbf{x}^T\mathbf{A}\mathbf{A} + \mathbf{A}\mathbf{A}\mathbf{x}\mathbf{x}^T + \mathbf{A}\mathbf{x}\mathbf{x}^T\mathbf{A} \right) + \ldots \right) \mathbf{b}_k \tag{4.38}$$

$$\frac{\partial \hat{L}}{\partial w_{jk}} = \frac{1}{2} \left( -1 + \left( \mathbf{b}_k^T\mathbf{x} \right)^2 - \left( \mathbf{b}_k^T\mathbf{x} \right) \left( \mathbf{b}_k^T\mathbf{A}\mathbf{x} \right) + \frac{1}{3} \left( \mathbf{b}_k^T\mathbf{x} \right) \left( \mathbf{b}_k^T\mathbf{A}\mathbf{A}\mathbf{x} \right) + \frac{1}{6} \left( \mathbf{b}_k^T\mathbf{A}\mathbf{x} \right)^2 - \ldots \right) y_j \,. \tag{4.39}$$

(Derivations are included in the appendix.)

For an ensemble of images, gradients are evaluated for each image patch, and the average is computed. This maximizes the expected log-likelihood over the entire data ensemble. After each gradient step, we normalized the vectors $\mathbf{b}_k$ to make sure $\|\mathbf{b}_k\| = 1, \forall k$. As in the variance component model, the norm of the vector of weights to each latent variable, $\|\mathbf{w}_j\|$ was adjusted gradually to maintain the desired variance of $y_j$. If we allow the weights to change freely, the MAP approximation to latent variable marginalization leads to a degenerate condition where the parameters continue to grow while the variance of the latent variables decreases.

### 4.2.4 Relationship to the variance component model

Although this model has been described in terms of representing covariance matrices, it bears a close relationship to the variance model developed in the previous chapter. Changing variance along one dimension, if the space is rotated, means introducing correlation terms between pairs of dimensions. These models are applied in the whitened space (so the canonical distribution has the global $1/f$ statistics), so the changing variances in that model did not affect individual pixels, but worked along directions that were combinations of pixels.

If the variance model is conditionally Gaussian, i.e.

$$p(s_i|\mathbf{v}) = \mathcal{N}(0, e^{[\mathbf{B}\mathbf{v}]_i}) \tag{4.40}$$

then the distribution in data space after projecting through the basis functions $\mathbf{A}$ is

$$p(\mathbf{x}|\mathbf{v}) = \mathcal{N}(\mathbf{0}, \mathbf{A}\left[e^{\mathbf{B}\mathbf{v}}\right]\mathbf{A}^T), \tag{4.41}$$

where the matrix $\left[e^{\mathbf{B}\mathbf{v}}\right]$ is a diagonal matrix whose elements are $e^{[\mathbf{B}\mathbf{v}]_i}$.

Therefore the variance component model described covariances in data space of the form $\mathbf{C} = \mathbf{A}\left[e^{\mathbf{B}\mathbf{v}}\right]\mathbf{A}^T$. There is a close relationship between this form and the matrix exponential employed in the covariance model, $\mathbf{C} = \exp(\sum y_j w_{jk} \mathbf{b}_k \mathbf{b}_k^T)$. If the set of vectors $\mathbf{b}_k$ is complete and orthogonal, the eigenvalues of the log-covariance matrix $\mathbf{A} = \sum y_j w_{jk} \mathbf{b}_k \mathbf{b}_k^T$ are $\mathbf{b}_k$ themselves, and the matrix exponential reduces to the element-wise exponential of the eigenvalues, which are $\sum_j y_j w_{jk} = \mathbf{W}\mathbf{y}$. Thus the covariance matrix is $\mathbf{C} = \mathbf{B}\left[e^{\mathbf{W}\mathbf{y}}\right]\mathbf{B}^T$, a form which is exactly the same as in the variance model.

The equivalence no longer holds, however, when the vectors $\mathbf{b}_k$ are under-complete or over-complete and not orthogonal. In the variance model, the effect of individual basis functions sums in the data space, but in the covariance model, the contribution from each vector $\mathbf{b}_k$ alters the eigenvector structure of $\mathbf{A}$, and the contraction and elongation of the density is effected through the exponentiation of the eigenvalues of the entire matrix. The result is that for the variance model, an over-complete set of basis functions $\mathbf{A}$ gave a canonical covariance matrix ($\mathbf{v} = 0$) of $\mathbf{A}\mathbf{A}^T$, whereas in the covariance model, the distribution defaults to the isotropic Gaussian case $\mathbf{C} = \mathbf{I}$. This might not have a significant effect when the dimensionality of the data is large and the $\mathbf{b}_k$'s are almost orthogonal, but increasing the over-completeness of the linear features makes this effect more prominent. When there is an under-complete number of linear features, the variance model simply does not span the data space (there will be dimensions along which the variance is 0); the covariance model, on the other hand, will describe a default variance of 1 along all the "non-covered" directions in space, although it will be unable to describe any changes in variation along them.

Another benefit of the covariance model is that it implicitly marginalizes the linear coefficients $\mathbf{s}$. It is also possible to compute the marginalized posterior $p(\mathbf{v}|\mathbf{x})$ in the variance model, though we did not do this, instead computing the MAP estimates of the linear coefficients together with the variance component coefficients. A possible downside is that the covariance component model makes the assumption that the conditional data distribution is Gaussian, while the variance model is fairly general and can include a variety of distributions for the linear coefficients, as long as we can model the logarithm of the scale parameter and estimate the desired quantities.

## 4.3 Results on natural images

We trained the model on a large set of image patches (20×20 pixels) sampled randomly from 110 grayscale photographs of outdoor scenes (van Hateren and van der Schaaf, 1998). We set the number of covariance coefficients ($\mathbf{y}$) in the model to 150 and the number of image features used for describing distributions ($\mathbf{b}_k$) to 1000. We found that for 20×20 image patches, a larger number of covariance components did not yield

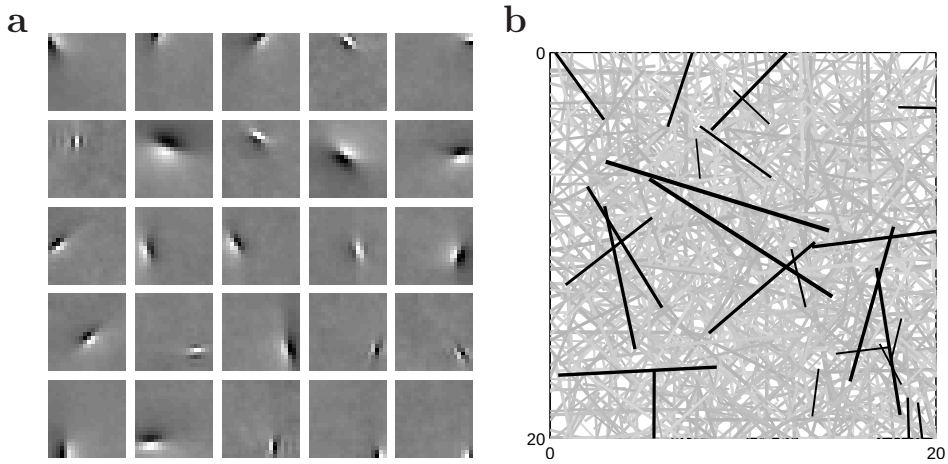


Figure 4.5: When adapted to natural images, the vectors $\mathbf{b}_k$, used by the model to describe image distributions, are oriented, localized image features. ***a***. 36 representative vectors (from a total of 1000), shown in image form. Each vector changes the correlational pattern in the image distribution: pixels that are the same color, e.g. two white pixels, become more correlated, while pixels of opposite colors become more anti-correlated. ***b***. The full set of image features spans the spatial extent of the 20×20 pixel image patch. Each line reflects the orientation, spatial position within the image patch, and scale (length of line) of one of the image features. Features in (***a***) are drawn in black.

as clean and interpretable set of parameters, with many covariance components shrinking in magnitude. We also experimented with different numbers of vectors $\mathbf{b}_k$ and found that a complete set (400) produced worse likelihood for the data and less clean and varied variance components (see below), while increasing their over-completeness above 2.5 times ($> 1000$) produced duplicate vectors. We simultaneously optimized the set of vectors $\mathbf{b}_k$, as well as the weights $w_{jk}$ linking neurons to them, to best represent image distributions.

As training data we used 110 grayscale images of outdoor scenes (van Hateren and van der Schaaf, 1998). Pixel intensities were log-transformed (corresponding roughly to the transformation at the retinal cone cells (van Hateren, 1997)), and the images were low-pass filtered and down-sampled to remove corner frequency sampling artifacts. We extracted random 20×20 image patches from the entire dataset. The mean luminance value was subtracted from each patch (this sped up model training but had no significant influence on the results). We "whitened" all image patches to remove global correlations and normalize the variance; this allowed the model to encode only the deviations of each image distribution from the global statistics (not consisting of white covariance structure). For visualization of image features, the results were projected back into the original image space.

The vectors $\mathbf{b}_k$ encode the directions of common expansion or contraction in the shape of the image distribution; Fig. 4.5a shows a representative subset after training. When drawn as image patches, each is an oriented and localized edge-like feature. The full set of 1000 tiles the spatial extent of the image patch (Fig. 4.5b) and spans the range of orientations and spatial frequencies of natural images (not shown). These oriented, band-pass image features are consistent with the optimal images for exciting simple cells in the primary visual cortex (Jones and Palmer, 1987; van Hateren and van der Schaaf, 1998; Ringach, 2002). Similar representations have been derived previously using linear statistical models adapted to natural images that maximize the efficiency of the resulting codes (Olshausen and Field, 1996; Bell and Sejnowski, 1997). In the model proposed here, however, these features are not used explicitly to reconstruct the original image, but instead function to modify the encoded distributions (arrows, Fig. 4.2). Thus, while the traditional interpretation of early sensory codes is that they are adapted for faithful reconstruction of the stimulus, these results suggest an additional interpretation: that these codes convey variations in image distributions and allow downstream visual areas to form more abstract representations.

Inspection of the set of linear features also revealed that, as in the hierarchical variance model, the population properties of these features differ from those of linear models like ICA. The full set of the vectors $\mathbf{b}_k$ contains

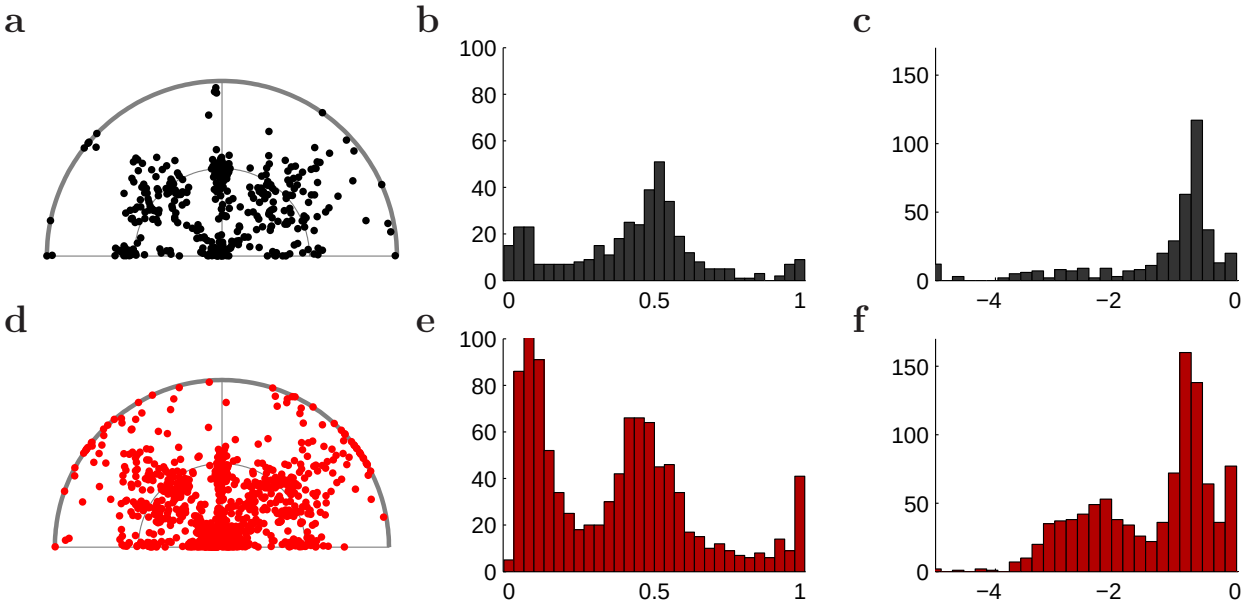


Figure 4.6: Comparison of linear features $\mathbf{b}_k$ in complete (400, black) and over-complete (1000, red) settings. ***a***, ***d***. Scatter plots of peak frequencies and orientations of the Gabor functions fitted to the linear features. The units on the radial scale are cycles/pixel and the thick line is the Nyquist limit. ***b***, ***e***. Histograms of spatial frequencies. ***c***, ***f***. Histograms of log-frequencies. Note that these plots reflect the properties of the features $\mathbf{b}_k$, which are more analogous to the basis functions in ICA. Because the model does not linearly re-encode the stimulus, it lacks features equivalent to filters in linear models.

more low frequency components than either ICA basis functions or the vectors obtained when their number is equal to the dimensionality of the data (Fig. 4.6).

This model provides another possible explanation for the benefit of over-complete representations. Units in the model encode image distributions and rely on the linear features to describe these distributions (Fig. 4.2). Different covariance regimes require different sets of underlying directions used for manipulating the distributions. For example, a covariance component that changes only the frequency content of the image (see below for an example) must differentially affect sets of frequency-organized linear features $\mathbf{b}_k$, and a sufficient number of low and high frequency features must be available for it to describe the necessary covariance structure. In order for another unit to independently encode changes in orientation content of the image, there must be a set of linear features that span a range of orientations. A complete basis, derived by ICA or a hierarchical model with a limited number of linear features, does not contain a sufficient set of features. Another insight is that oriented, localized features are not only optimal for efficient encoding of natural images, but they also provide the best, most compact descriptions of image distributions and associated higher-order image structure.

### 4.3.1 Analysis of individual covariance units

The second set of parameters, the weights linking model neurons to the set of vectors $\mathbf{b}_k$, describe the each neuron's role in shaping the encoded image distribution. A set of learned weights for a typical model neuron is shown in Fig. 4.7a. This neuron exerts the strongest effect on features localized in the top left of the image patch, increasing the variability (i.e. activation) of those oriented at a 45° angle (its "preferred" orientation), decreasing the variability of those at the orthogonal orientation, as well as those at the preferred orientation but at an offset location. Rather than responding to a few excitatory or suppressive image features, the neuron integrates a large number of them (a subset is shown in Fig. 4.7b) in order to describe a pattern of variability that underlies a particular image distribution. While the functional significance of these subunits

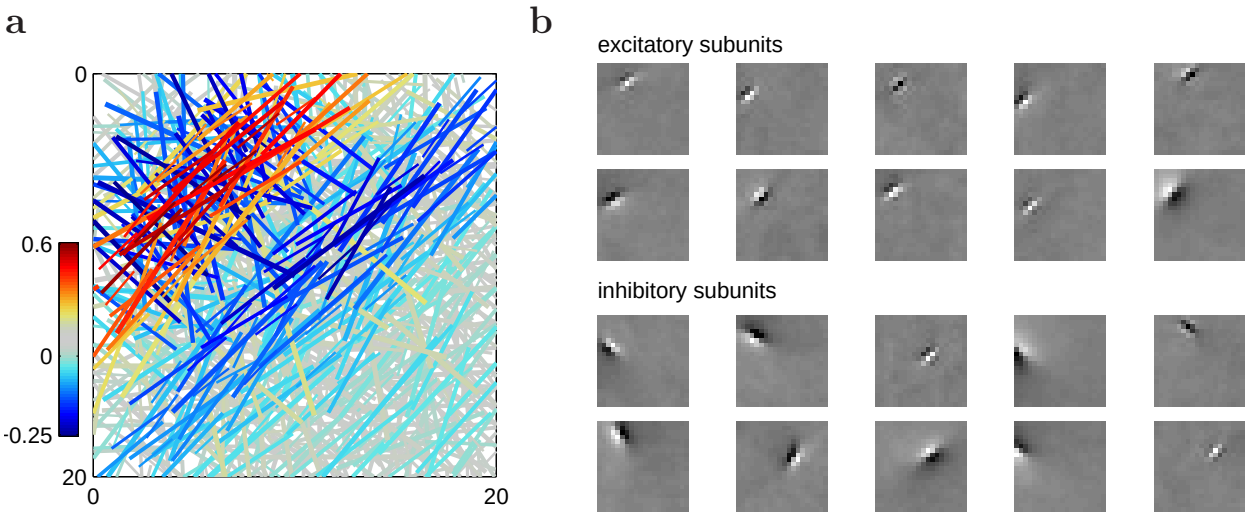


Figure 4.7: ***a***. Weights of one typical model neuron to the vectors $\mathbf{b}_k$ plotted as in Fig. 4.5b. The color indicates the sign and magnitude of the weight (see colorbar). Positive weights (red hues) indicate increased variability in the corresponding feature; negative weights (blue hues) indicate decreased variability; features not affected by this neuron are shaded gray. ***b***. The vectors $\mathbf{b}_k$ corresponding to the ten most positive (top rows) and the ten most negative (bottom rows) weights in (***a***). These act as excitatory and inhibitory subunits for this neuron. When active, this neuron signals image structure of specific orientation in the upper left part of the image patch, as well as the *absence* of image structure at the orthogonal orientation, or at the parallel orientation and offset location in the image patch.

is to modify the statistical structure of the encoded distribution, they also reflect features of the best and worst stimuli for exciting this model neuron. Note that a model neuron has no single best excitatory stimulus; instead it is activated by all images from the distribution it encodes. Conversely, stimuli that lie in parts of image space assigned low probability by this neuron will inhibit its activity.

The model allows us to compute analytically each neuron's most activating and most suppressive directions (where activating and suppressive are defined w.r.t the magnitude of positive $y_j$ values). By setting neuron $j$'s activity to 1 and all others' to 0, we obtain the covariance matrix encoded by this neuron, $\mathbf{C} = \exp(\mathbf{A}_j)$. Spectral analysis of this matrix reveals the dimensions of greatest expansion of the distribution and greatest contraction: eigenvectors associated with the largest eigenvectors correspond to directions in image space along which the variance is most increased, relative to the global distribution (when $\mathbf{y} = 0$). Small eigenvalues indicate that the distribution is contracted. Such an analysis is shown for one model unit in Fig. 4.8, and for a subset of the population in Fig. 4.9b.

### 4.3.2 Population coding

For each input image, it is the *joint* activity of the population of model neurons that describes the inferred image distribution. In order to understand this population code, we performed cluster analysis to identify groups of neurons with similar function (Fig. 4.9a) and then looked at selected neurons' representations (Fig. 4.9b).

In order to cluster the neural population, we examined which parameters of image features $\mathbf{b}_k$ could best account for the values of the weights for each neuron, $w_{jk}$. For example, the neuron in Fig. 4.7 is sensitive to oriented and localized structure, and accordingly its weights to the underlying image features (i.e. the colors in Fig. 4.7a) are explained best by the location and orientation of features $\mathbf{b}_k$ (93% of variance in $w_{jk}$'s explained by regressing on these two parameters). For each neuron, we computed a vector indicating how much the feature parameters (location, orientation, frequency, and all their combinations) contributed to explaining the neuron's weights. Elements in the 8-dimensional vector corresponded to the *null* feature (the

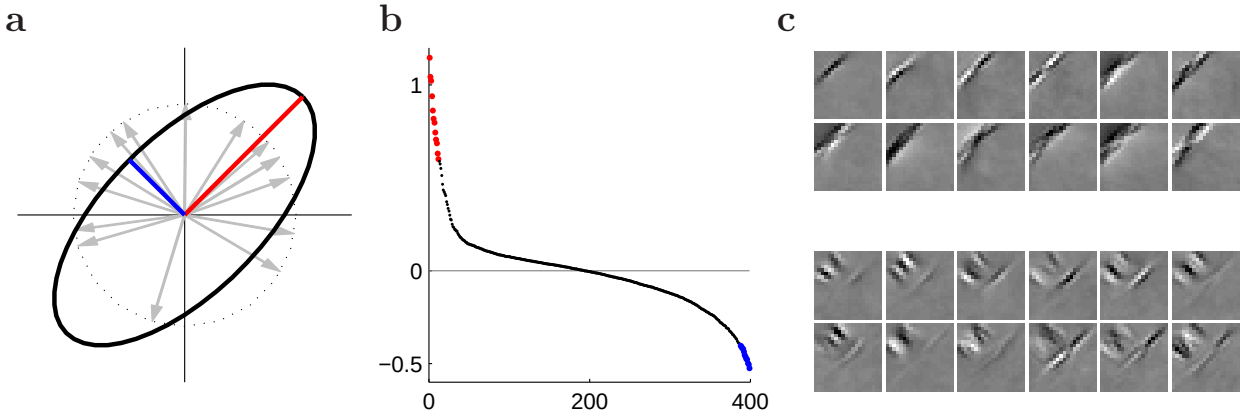


Figure 4.8: ***a***. A schematic of one model neuron's effect on the encoded distribution. The neuron uses the underlying image features (gray arrows) to transform the canonical distribution (dotted circle) into a different distribution (black ellipse). The final effect of the neuron on the distribution is given by the spectral decomposition of the resulting log-covariance matrix (see text). The most expanded and most contracted directions correspond to the largest and smallest (respectively) eigenvalues (red and blue lines). ***b***. The full set of 400 eigenvalues describes the scale of all directions in image space. ***c***. Eigenvectors associated with the most positive 12 (top) and the most negative 12 (bottom) eigenvalues, drawn in image form. The corresponding extreme eigenvalues are highlighted in color in ***b***.

bias term in the regression), *loc*, *fr*, *or*, *loc-fr*, *loc-or*, *fr-or*, and *loc-or-fr*, and ranged from 0 to 1, according to how much variance in $\{w_{jk}\}$ each term explained. Because the relationship between these parameters and the weights were often non-linear and possibly multi-modal, we used nonparametric multi-dimensional histogram regression. We divided each set of parameter values into several (typically 5) equally-spaced bins, and used the mean value in each bin (or hyper-bin when multiple dimensions were regressed on) as the estimated regression coefficient.

Once the regression coefficients were obtained, we subtracted from the values of the nested terms the maxima of their parents, so that the result reflected only the gain (in variance explained) attained by adding a new term. For example, if regressing on *loc* and *or* already explained 60% of the variance and adding *fr* only increased that to 63%, that element of the vector was set to 0.03. Finally, we used standard hierarchical clustering methods (the single linkage algorithm, applied to a matrix of *cityblock*-metric distances between input vectors, using MATLAB's `pdist` and `linkage` routines); this produced the dendrogram in Fig. 4.9a.

We also considered several alternative methods for clustering, such as ones used in identifying groups of functionally distinct neurons (Hegdé and Van Essen, 2003; Gallant et al., 1996). However, this is difficult when the function of neurons is not known in advance and the stimuli used to characterize neurons are as rich and complex as natural images. A typical method involves centering the receptive field structure of the neuron with respect to the dominant orientation and measuring the broadness of orientation tuning, the extent of excitatory and suppressive features, as well as their spatial localization. This would work well for some units in our model, but not for others. Visual inspection suggested that orientation tuning and extent of suppression were not sufficient to describe the heterogeneous population, and other analysis choices depended too much in prior expectation of unit function. Nevertheless, an analysis that closely corresponds to methods in physiological research might be appropriate in the future, when closer correspondence between cortical neurons and model representations is investigated.

The population clustering analysis revealed two large groups and a small number of specialized units; the population of neurons exhibits a range of properties observed in cortical visual cells. One large set contains orientation-sensitive, localized neurons, such as the one analyzed in Fig. 4.7 (also shown in Fig. 4.9b, neuron 8). Most exhibit the inhibitory cross-orientation and surround regions described in Fig. 4.7 associated with orientation-selective V1 and V2 neurons, while encoding a variety of image types, some with curvature or more complex patterns (5-9). Another large set of neurons is employed by the model to indicate localized contrast (energy) in the stimulus (10-12). Individually, each of these specifies only coarsely the location of

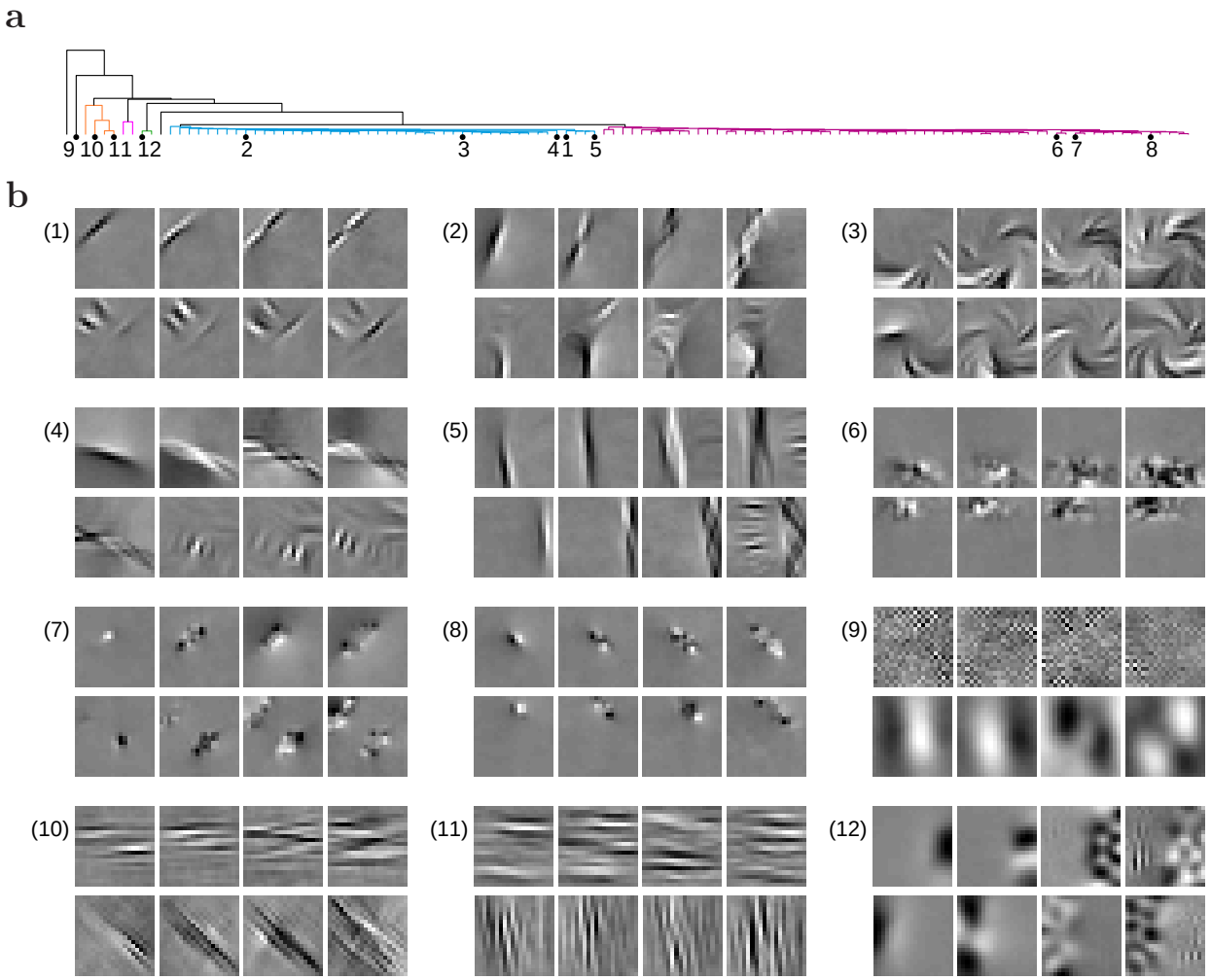


Figure 4.9: An analysis of the population of units learned by the model. ***a***. 120 most active (out of a total of 150) neurons were hierarchically clustered according to the different aspects of image structure they encode (see Methods and Supplementary Materials). The clustering reveals two large categories of neurons, as well as some specialized neurons. Subtrees are distinguished in color for ease of discrimination. ***b***. To obtain a concise description of each neuron, we identified its most activating and most suppressive image features (see supplementary materials for details). Here, for twelve model neurons that are representative of the learned population, we show four activating (top row of each panel) and four suppressive (bottom row) image features. Numbers indicate the neuron's position in the dendrogram in (***a***). Neuron (8) was analyzed in Figure 4.

contrast energy in the stimulus (and corresponds to a broad set of image distributions), but their joint activity acts a set of constraints that input images must satisfy to belong to the encoded distribution. Although cortical neurons have not been analyzed in a framework that could identify such a code, localized contrast subfields are consistent with observations that some cortical neurons are sensitive to second-order (energy) patterns in the image (Zhou and Baker, 1994; Mareschal and Baker, 1998b).

Among the neurons in the model, some analyze the spatial frequency content of the image (e.g. neuron 1). When neuron (1) is active, the input image is inferred to come from a set of images with given frequency statistics. Note that each neural activity in the model can be both positive and negative; positive activity here signals high frequency (fine) image structure, while negative activity signals low spatial frequency (coarse) structure. This neuron does not signal anything about the spatial localization of structure in the image or its dominant orientation, and images that activate it can be quite different, as long as they satisfy the spatial frequency constraints. Other neurons in the population convey global orientation structure (2,

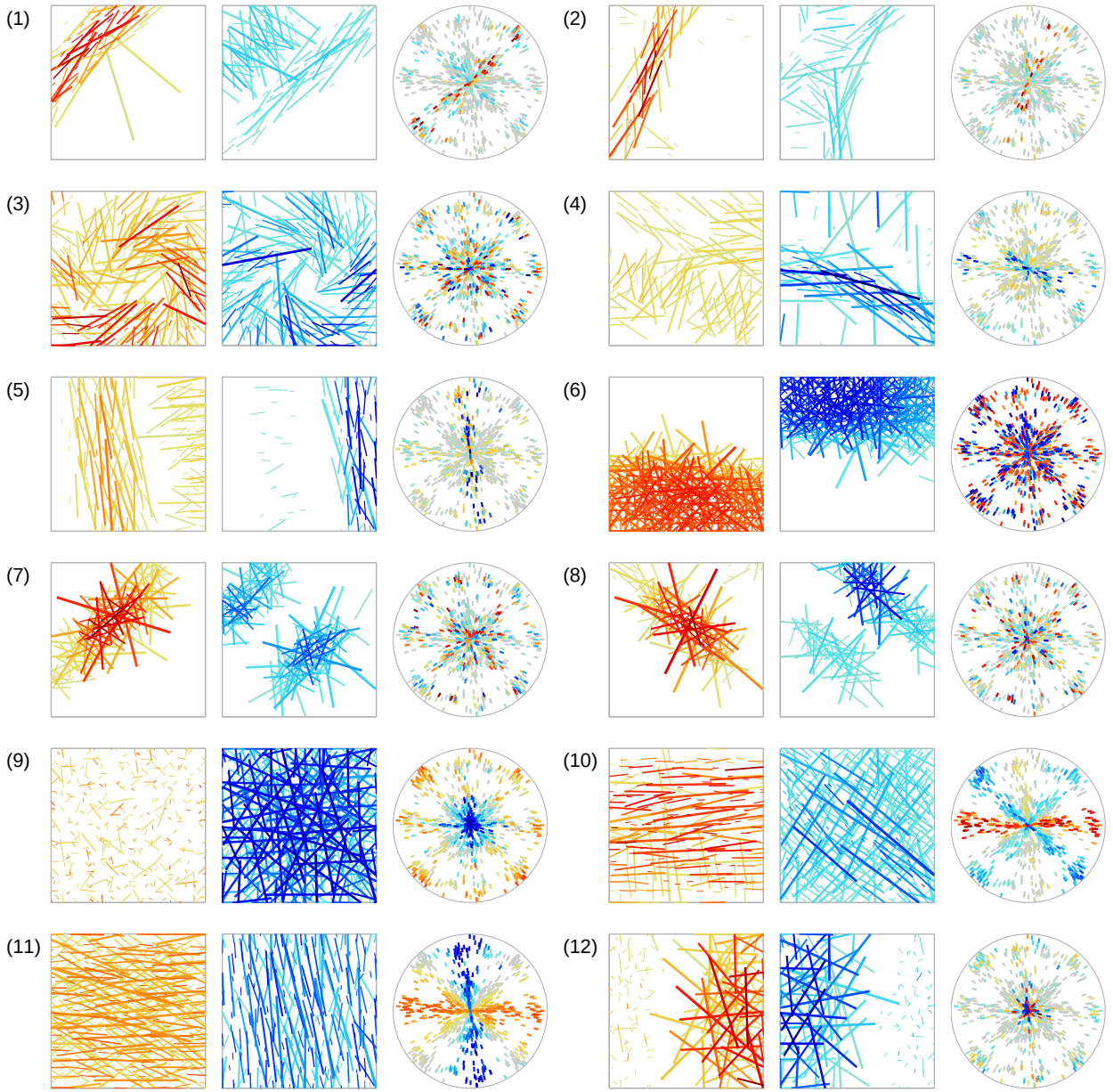


Figure 4.10: The same twelve units as in Fig. 4.9, but plotted as in Fig. 4.7 (unit numbers shown in parentheses). The first two panels of each three-panel block show weights by location in the image patch. Small weights are omitted; only $w_{jk} > \max_j |w_{jk}|$ (left panel) and $w_{jk} < -\max_j |w_{jk}|$ (middle panel) are drawn. The right panel features a polar plot with all the $\mathbf{b}_k$, colored according to the weights and arranged by the orientation and spatial frequency of the linear feature. Colormap as in Fig. 4.7.

3) but are insensitive to the spatial frequency content of the image. Such encoding properties have been observed in V4 neurons, some of which are narrowly tuned for orientation, while others encode frequency information (David et al., 2006). Other neurons in the model indicate contrast in spatial frequencies across image locations (4), signaling a boundary of textures characterized by their statistical properties. Studies of texture boundary coding in visual cortex have been limited to simple synthetic stimuli (Lamme, 1995; Lee et al., 1998; Nothdurft et al., 2000; Rossi et al., 2001; Song and Baker, 2007); these results suggest ways to use more complex textures, defined in a statistical framework, to analyze neural responses. (An alternative visualization of the same twelve units is shown in Fig. 4.10, which draws individual weights to the underlying

linear features $\mathbf{b}_k$, as in Fig. 4.7a.

A more detailed comparison of model units to known properties of cortical cells follows in the next chapter, where we specifically examine the model representations as they relate to coding in V1, V2, and V4.

### 4.3.3 Effect of latent variable prior

As noted above, the symmetric prior on the covariance coefficients relies on a significant assumption about the symmetry of correlational patterns in natural images. A model that does not make this assumption and uses only positive latent variables is also easier to relate to neural activity (which of course does not include negative firing rates); for example, it would allow a more meaningful discussion of the roles played by image features that excite or suppress the cell.

We evaluated the effect of the symmetric prior assumption by training two different models: one that used a symmetric Laplacian prior $p(\mathbf{y})$ (SP) and another that used the exponential with support only in $y_j \geq 0$ (PP). The models were adapted to 10×10 image patches; each employed 200 linear features $\mathbf{b}_k$; the dimensionality of the latent variables was 40 for the SP model and 40 or 80 for the PP model (we tried both parameterizations, since the PP model can only span half the latent variable space of the SP model).

Inference with the exponential prior can be implemented using several approaches. One method is simply to restrict the gradient updates to $\mathbf{y}$ to the positive-only quadrant of the space (by shifting any negative values $y_j$ to 0 at each iteration). Alternatively, we can employ a vector of helper variables $\boldsymbol{\alpha}$ with support in the full space of reals and then map them to a vector of positive values $\mathbf{y} = g(\boldsymbol{\alpha})$. For example, we can use a Gaussian prior on $\boldsymbol{\alpha}$ and $g(\alpha) = \alpha^2$. (This does not yield the exponential distribution for $\mathbf{y}$, but a chi-square distribution of order 1, which is also more sparse than Gaussian.) We then compute the gradient with respect to the helper variables, which ensures that $\mathbf{y} > 0$. In simulations the two methods yielded identical results.

Models trained with the positive-only latent variable prior (PP) produced significantly different results from those that employed a symmetric Laplacian prior (SP). One important difference is in the abilities of the two models to simultaneously encode luminance and contrast edges. For example, take a unit that codes local contrast, such as unit (6) in Fig. 4.10. When this unit is highly active, there is a large difference in the contrast between the lower and the upper halves of the image patch. In natural images, such contrast differential is also often associated with a change in luminance across the horizontal edge. The PP model (which also learns such a spatial contrast unit) is able to encode this statistical regularity by grouping another linear feature $\mathbf{b}_k$ — one that encodes the low frequency horizontal edge — together with the rest of the spatially localized, and mostly higher frequency, linear features. Thus, when its coefficient is on (and by definition positive), the contrast in the lower half of the image patch is increased, but so is the difference in luminance across the horizontal edge. The SP model, on the other hand, does not associate the low frequency edge with the spatial contrast unit, for if it did, it would also be encoding the converse statistical pattern — one when the contrast in the *upper* half of the image is increased while the luminance difference across the edge is *diminished*. This is not a typical pattern in natural images, and the model learns neither this pattern nor its converse.

Beside the difference in coding luminance edges, the population properties learned by the PP model are quite different (Fig. 4.11). The population includes a small number of spatial contrast units, with the rest of the components coding high-variance oriented structure flanked by weaker low-variance orthogonal image features (only the PP model with 40-dimensional latent variables is shown, the larger PP model gave similar results).

The differences in the learned code are significant and systematic, but several problems must be resolved before we can draw firm conclusions. The distribution of MAP estimates for a set of image patches contains a very large proportion of near-zero values, which could mean that the optimal coefficients $\mathbf{y}$ are mostly zero,

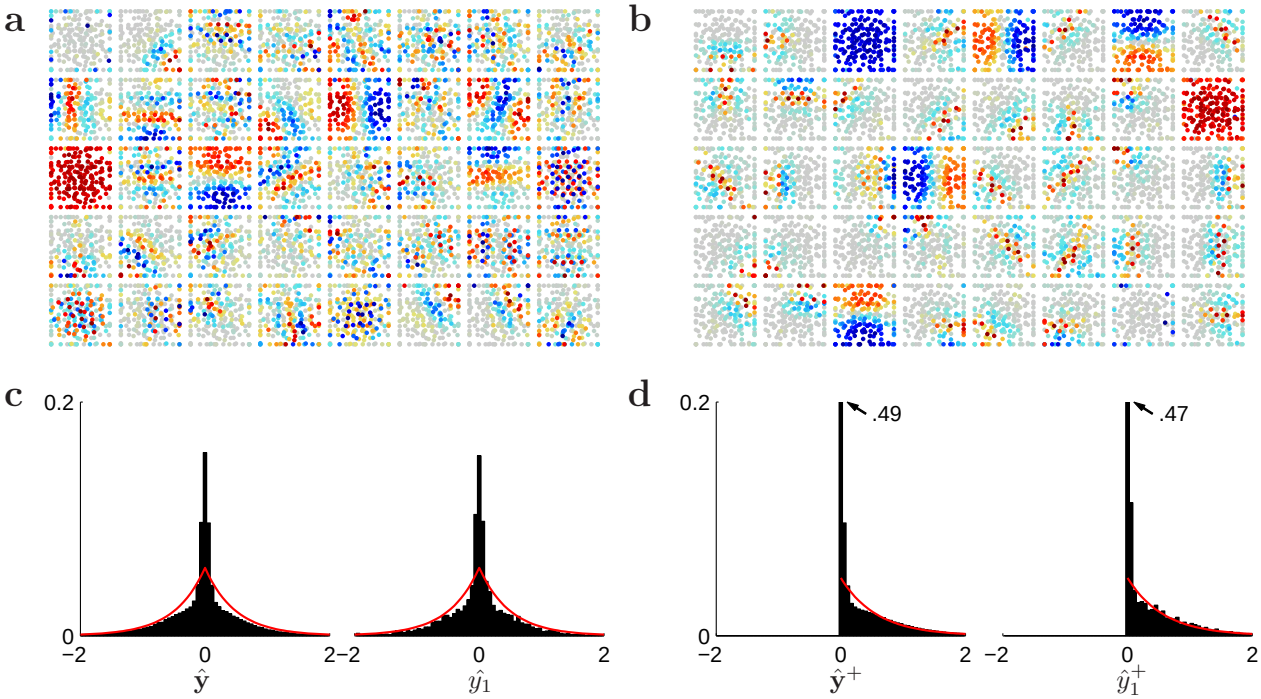


Figure 4.11: Representations of spatial structure learned by models with symmetric (***a***) and positive-only (***b***) priors are significantly different. We plot weights $w_{jk}$ according to the spatial position of the underlying linear features $\mathbf{b}_k$ in the image patch. The color indicates the sign and magnitude of the weight (saturated red ***c***, ***d***. Histograms of MAP values estimated for the two models in ***a*** and ***b*** reveal significant differences. The ordinate represents fraction of all values falling in each bin, but the peaks for positive-only distributions are cut off for clarity. Red lines show the Laplacian and exponential distributions of the same variance as the histograms. left panel: all coefficients; right panel: one typical coefficient.

or that the implementation of the inference procedure biased the estimates towards very sparse solutions. If the distribution of MAP estimates is in fact highly peaked at zero, the exponential prior might not be correct. Finally, it is unclear what effect replacing the posterior over the latent variables with its peak (MAP) has on the recovered parameters. In the next section we compare the performance of these two types of models using rough coding cost measures, but in order to reach definitive conclusions about which model more accurately represent image structure, new learning methods must be developed that do not rely on the MAP approximation.

### 4.3.4 Generalization and discrimination of image regions

Finally, we looked at the way the model uses the population of units to represent images. We computed "winner maps", as was done for the variance model, to examine how the model representation changes across the image. This was done by extracting every 20×20 image patch from a large image (using a sliding sampling window), computing the model representation for that patch, identifying the unit with the largest magnitude, and drawing these identities with unique colors. As Fig. 4.12b illustrates, these maps consisted of larger regions of uniform colors (meaning the same model unit was maximally active for the entire region) than the linear representation (not shown here, see Fig. 3.10).

If the model is able to generalize across the wide variability present in natural images, the image patches that were widely scattered in the original image space and could not be separated by simple linear codes should be tightly clustered in the space of the model's representation. This can be illustrated by projecting

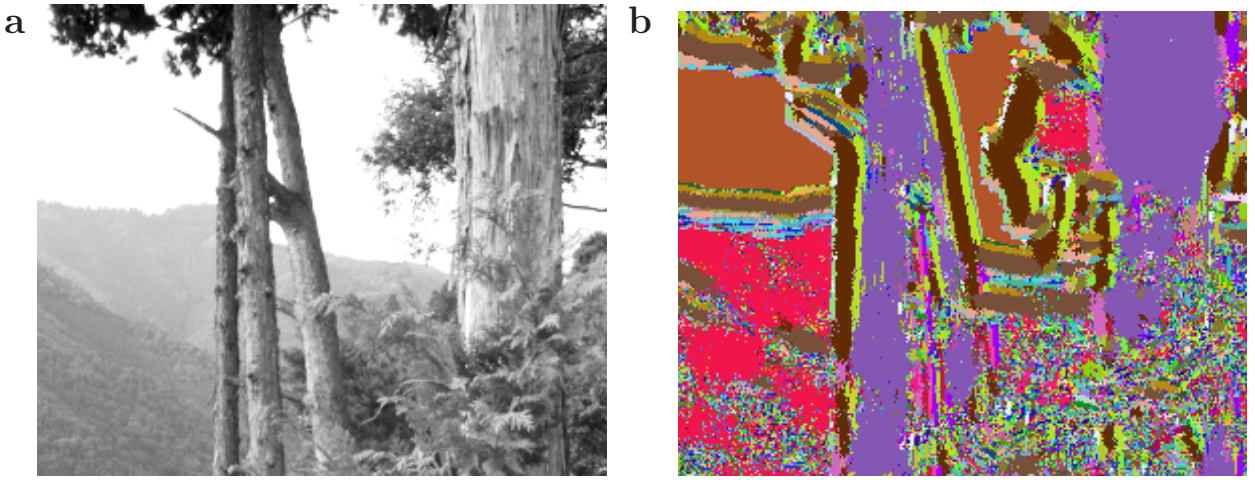


Figure 4.12: The activity of model units is more invariant across the image, as shown by this "winner map". The format of this figure is the same as in Fig. 3.10. The color of each pixel in $\boldsymbol{b}$ uniquely identifies the model unit $y_j$ that is maximally active ($\arg\max |\mathbf{y}|$) in the patch (extracted from the image in $\boldsymbol{a}$) centered at that pixel. The colors were randomly assigned and do not correspond to any organization of the model units.

into two dimensions (as was done with image space in Fig. 4.1) the 150-dimensional model representation of a collection of images. Fig. 4.13b shows the two best linear separating dimensions, in $\mathbf{y}$-space, for the four types of images (examples shown in Fig. 4.13c). These were computed using standard Linear Discriminant Analysis methods, assuming multivariate Gaussian distributions with equal covariances for each cluster and computing the generalized eigenvectors of $(\mathbf{S}_w^{-1}\mathbf{S}_b)$, where $\mathbf{S}_w$ is the within-cluster covariance and $\mathbf{S}_b$ is the between-cluster covariance.

A similar analysis performed in pixel space did not separate the clusters, though the set of points corresponding to the tree edge were somewhat separable (on average, these images contain a perceptible light-dark vertical edge). Projection of the set of $\mathbf{y}$ onto the first two principal components also gave fairly separable clusters, suggesting that the differences among these clusters are salient in the space of model representation. As hypothesized, by encoding image distributions rather than the precise feature content of each image, the model is able to encode perceptually similar images with similar representations and to separate distinct image types.

We can also examine the type of structure encoded by the model by sampling from the model using fixed covariance coefficients. The average of the MAP estimates for each image region gives the "center" distribution associated with each image type; we fix $\mathbf{y}$ to this value and sample image patches (Fig. 4.13d). The images drawn from the model have the average covariance structure of each image region — the "tree edge" patches have a difference in contrast across the edge, the bark images have predominantly vertical structure, and the hillside image patches are very low contrast — but many other cues and statistical regularities that define the regions are clearly not captured by the model.

## 4.4 Model comparison

How does this model compare to the linear models and the hierarchical model for variance discussed in the previous chapter? Do the hierarchical models significantly improve on the linear models? They learn interesting image structure and encode novel properties of the image, but it is also desirable to have a quantitative measure of the gain. This should also allow comparison to other hierarchical models for describing images, such as Markov Random Fields (MRF) and Gaussian Scale Mixtures (GSMs), though here we only

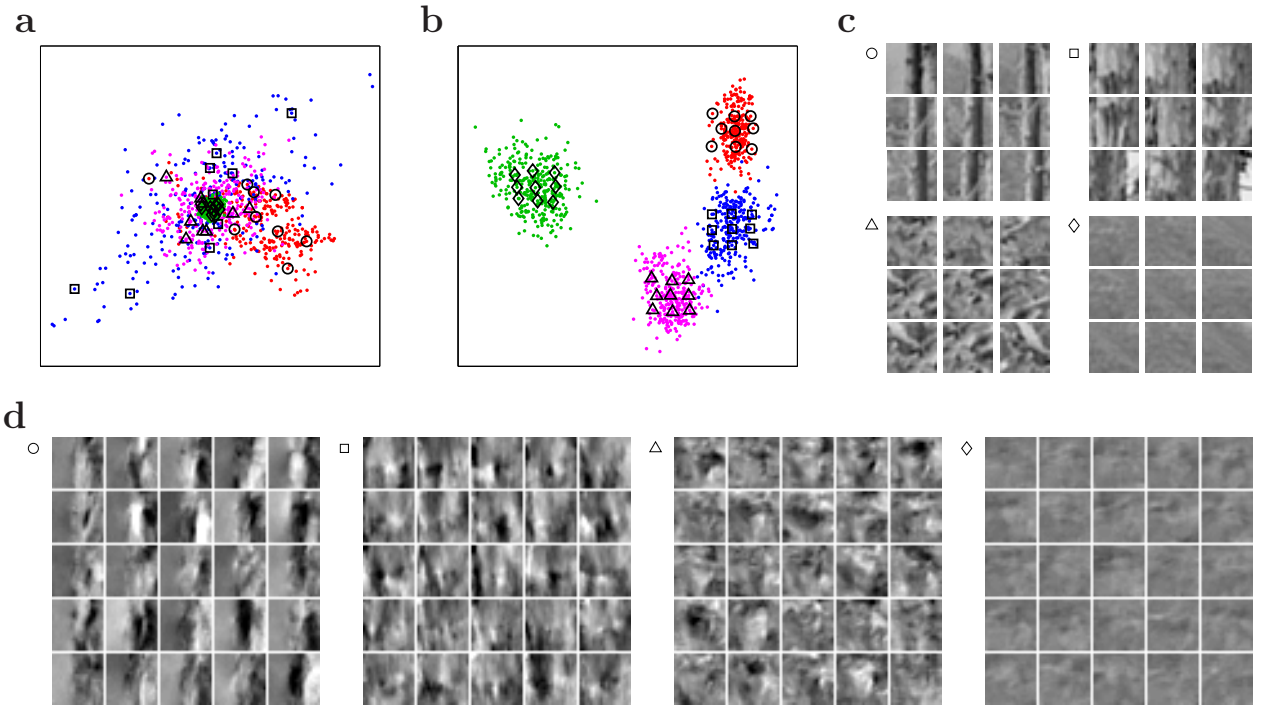


Figure 4.13: The model's representation is able to generalize across natural variability and discriminate between different image types. ***a***. Linear representations of images sampled from the four regions in figure 1a (colored dots) are highly overlapping (format as in Fig. 4.1). ***b***. A two-dimensional projection of the model's representation reveals a well-separated group of clusters. ***c***. Each 3×3 image group corresponds to the 3×3 array of symbols in ***b***, each of which is also plotted in ***a***. Despite the variability in the appearance of edges and textures, the model's representation of natural images generalizes within each region while still distinguishing among them. ***d***. Image patches sampled from the model's representation of one image from each type. Data were drawn from multivariate Gaussians $\mathcal{N}(\mathbf{0}, \mathbf{C}(\mathbf{y}_r))$, where $\mathbf{y}_r$ are the covariance coefficients for the top left image in each panel in ***c***.

analyze the relative performance of linear single-stage and the hierarchical models described above. Another important test for the developed models is what knowledge they impart on the study of neural processing in the cortex; this is addressed in more detail in the next chapter.

### 4.4.1 Natural image synthesis

One way to compare models is to visually examine the data generated by randomly sampling images from the model. As shown in section 2.2.2, linear models do not generate natural-looking images. How much better are the hierarchical models? Fig. 4.14 shows randomly sampled images as well as a set of natural images (this is the distribution we would like to capture). As above, the DC component has been subtracted from natural images and is not generated by the models.

The hierarchical model captures some of the inhomogeneous structure of natural images, generating regions of varying contrast, spatial frequency, and orientation, and some credible-looking textured patterns that resemble grass and bark. However, it fails to capture some other features of natural scenes such as elongated contours, phase-alignment of edges, and large scale structures. Note that image patches are small, and only so much structure is evident in such small windows onto natural scenes. It would be very interesting to extend this work to larger image patches and analyze the type of structure learned from these data. (Much of the interesting patterns discovered by the models trained on 20×20 image patches did not emerge when the models were trained on smaller images). However, 20×20 patches approach the limit of computationally tractable dimensionality using currently developed training methods.

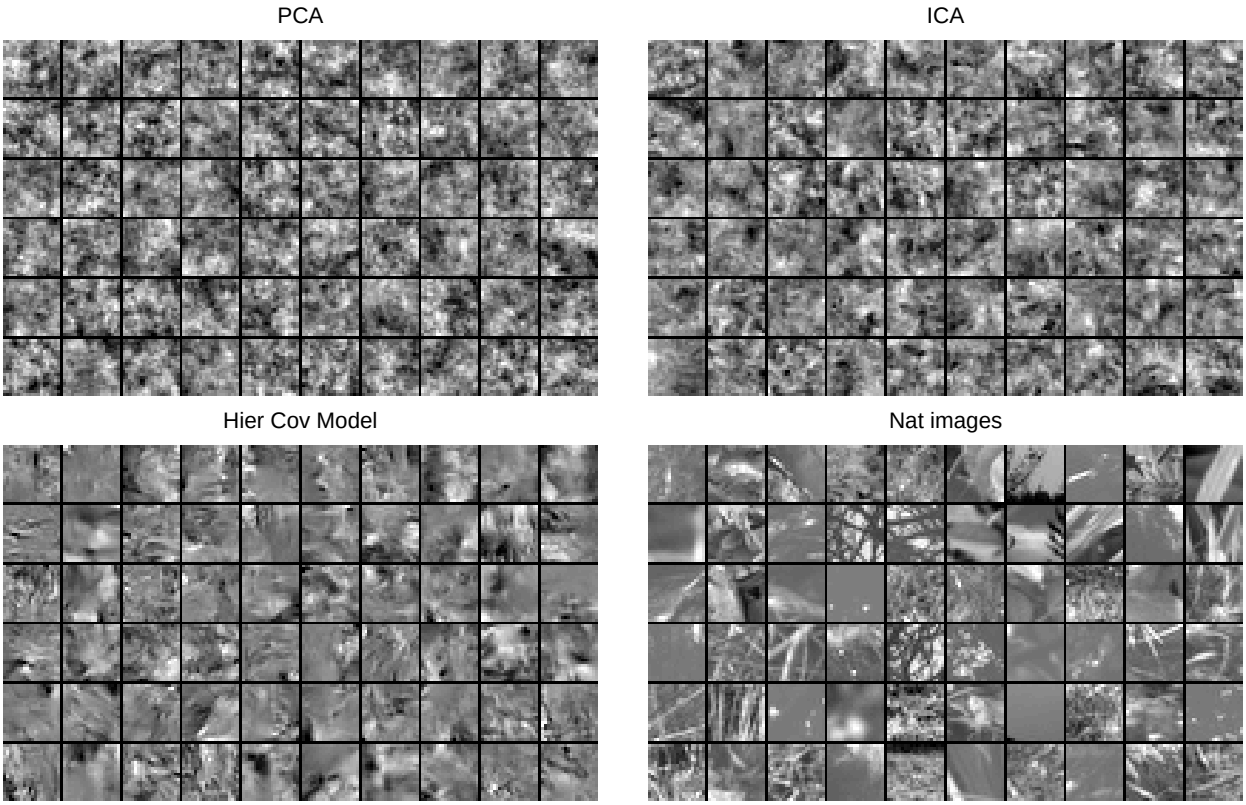


Figure 4.14: Image patches sampled from various generative models. The hierarchical model for non-stationary covariance (described in this chapter), as well as PCA and a complete, noise-free ICA model were used to generate 20×20 image patches. Example training natural images are shown for reference.

### 4.4.2 Comparing coding cost of natural images

Other ways to quantitatively compare different models is to measure the coding efficiency of a model's representation or to compute the likelihood of the data. The coding efficiency is a good measure for models that attempt to faithfully re-encode the data. In this case, the entropy of the code can be estimated (though accurate measurements can be difficult to obtain for high-dimensional codes); a low entropy code that preserves stimulus information is less redundant and indicates a model better fit to the data. However, this approach does not account for poor encoding of the data, as it assumes all the information about the stimulus is represented by the model, and then measures the efficiency of the model's code. Thus a model code can have low entropy but retain little information about the data, and this will not be reflected in the estimated coding cost. Most crucially, however, the hierarchical models described above do not encode the stimulus; for example, the covariance model only explicitly computes the *distribution* from which each data point is assumed to have been generated. Thus entropy measures will allow us to compare such models.

On the other hand, the likelihood of the data under the model can still be an accurate reflection of the model's ability to describe the data. Good models will assign high probability to regions of the space where most of the data lie and low probability to other parts of the space. Good allocation of probability mass can also be used to improve data quantization in lossy coding and incorporated into de-noising models based on statistical methods. Below we compare the hierarchical models described in this and the previous chapter, as well as standard linear statistical models (PCA and ICA).

**Data likelihood**

Let the true probability density (of the hold-out set) be $p(\mathbf{x})$, and the model density $q(\mathbf{x}|\theta)$ (note the change of notation for the rest of this section). We would like to evaluate the likelihood of the hold out set under the model's distribution and express it as an interpretable quantity. First, let us assume we can estimate the likelihood. How do we interpret it? We need to convert the likelihood density $q(\cdot)$ to a probability mass $Q(\cdot)$. To do this must specify a precision value $\delta$ with which we will bin the density function and obtain the probability value of a data sample lying within $\delta/2$ (or within the hyper-bin of length $\delta$),

$$Q(\mathbf{x}|\theta) = q(\mathbf{x}|\theta)\delta^D \tag{4.42}$$

$$\log_2 Q(\mathbf{x}|\theta) = \log_2 q(\mathbf{x}|\theta) + D\log_2(\delta) \tag{4.43}$$

This likelihood is related to bits through Shannon's theorem; it is the expectation of the entropy under the true distribution $P$,

$$\#\text{bits} \geq -E_P[\log_2 Q(\mathbf{x}|\theta)] \tag{4.44}$$

Another intuition is provided by considering the *perplexity* the model associates with the data: how many questions (bits) are necessary to establish that the data point lies within the bin of the specified size? This is defined as $PP = 2^{H(P,Q)}$, i.e. it is related to the cross-entropy between the model distribution $Q$ and the empirical distribution $P$ (the same quantity as above),

$$H(P,Q) = -E_p[\log_2 Q] = -\sum_n P(\mathbf{x}_n)\log_2 Q(\mathbf{x}_n|\theta) \tag{4.45}$$

$$= D_{KL}(P\|Q) + H(P) \tag{4.46}$$

The lower this number, the fewer the bits we must expend to explain data in the hold-out set. It also makes clear that if the KL divergence between the model density and the true density (the first term of Eqn. 4.46) can be made 0, the expected number of bits is bounded by the data entropy $H(P)$.

Once we select a discretization level $\delta$, we can convert the likelihood of the hold out set to bits. How do we obtain the data likelihood under the model $q(\mathbf{x}|\theta)$? With linear noiseless models based on PCA and ICA, the likelihood can be analytically computed (e.g. $q(\mathbf{x}|\mathbf{A}) = q_s(\mathbf{A}^{-1}\mathbf{x})/|\det \mathbf{A}|$). Models with latent variables, however, require marginalization over these variables that are typically intractable. Replacing the marginalization with the value at the MAP ($\hat{\mathbf{s}}$ or $\hat{\mathbf{v}}$) was good enough for learning, but is too gross an approximation for model comparison. Lewicki and Sejnowski (2000) describe a method that approximates the volume of the integral with a Gaussian distribution around the MAP estimate and uses numerical estimates of the Hessian. In our setting, a different approach based on kernel density estimates of the likelihood is somewhat more tractable and easy to compute.

**Kernel density estimation**

The kernel density estimate approximates the density function $q(\cdot)$ by placing kernels at samples drawn from the model; e.g. summing over $M$ kernels,

$$\hat{q}_h(\mathbf{x}) = \frac{1}{Nh^D}\sum_{n=1}^{N} K\left(\frac{\mathbf{x}_n - \mathbf{x}}{h}\right) \tag{4.47}$$

where $D$ is the dimensionality of the data and $h$ specifies the bandwidth (scale) of the kernel. We will call this the *data kernel* approximation. A standard choice is the isotropic multivariate Gaussian kernel $K(\mathbf{x}) = (2\pi)^{-D/2}\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right)$. Sampling in our models is computationally easy and very fast, making this approach attractive. However, this method does not scale well to large number of dimensions, where the curse of dimensionality makes it extremely difficult to thoroughly tile the data space.

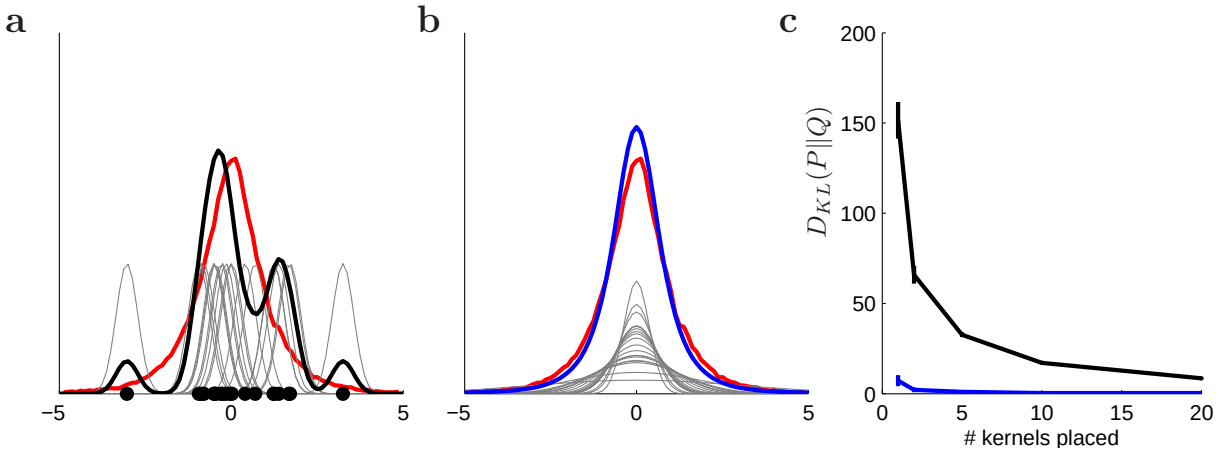


Figure 4.15: Density estimates for a hierarchical model obtained by placing kernels at each sampled data point (***a***, see Eqn. 4.47) or by summing distributions with different parameters (***b***, see Eqn. 4.48). The hierarchical model in this toy example is defined by $x \sim \mathcal{N}(0, e^v)$, where $v \sim \mathcal{N}(0, 1)$. Gray curves indicate individual densities (rescaled for clarity), black dots indicate locations of *data kernels*, the black and blue curves are the results of summing over the 20 kernels, and the red curves show the empirically-measured distribution after drawing 10,000 samples from the distribution. After placing 20 kernels, the *hyper-parameter kernel* estimate is much closer to the empirical distribution than the *data kernel* estimate. ***c***. The convergence of the two methods to the empirical distribution, given by the average Kullback-Leibler divergence between the kernel distribution $Q$ and the empirical distribution $P$, as the number of kernels employed increases. The plotted values are average KL divergence for 20 trials, and error bars indicate standard errors. Standard kernel density estimation converges much slower than that using *hyper-parameter* kernels. The difference is great even in the 1D toy example; in large number of dimensions it is much more significant.

Fortunately, for hierarchical models that have simple distributions when the latent variables are fixed (such as the models presented here), we do not have to approximate the model density $q$ with kernels placed at every sample. Because our models can be considered infinite mixtures of Gaussian distributions (or Laplacian for some forms of the variance component model) with different variances or covariances, we can approximate them with a finite sum of the component distributions whose hyper-parameters are sampled according to their priors (*hyper-parameter kernel approximation*),x

$$\hat{q}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} q(\mathbf{x}|\mathbf{y}_n) \tag{4.48}$$

where the set $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ represents instances of latent variables sampled according to model priors. For example, in the case of the non-stationary covariance model, we can sample sparse random vectors $\mathbf{y}$, construct covariance matrices using the learned parameters, $\mathbf{C}_n = \exp(\sum_j y_j \mathbf{A}_j)$ and plug in $q(\mathbf{x}|\mathbf{y}_n) = \mathcal{N}(\mathbf{0}, \mathbf{C}_n)$. This also means we do not have to select optimal bandwidths, and the above sum converges to the true distribution much faster than the standard kernel density estimate (Fig. 4.15). For the variance component model with Gaussian conditional distributions, we use $q(\mathbf{x}|\mathbf{v}_n) = \mathcal{N}(\mathbf{0}, \mathbf{A}[e^{\mathbf{B}\mathbf{v}}]\mathbf{A}^T)$ and for the Laplacian conditional form, when the model is noiseless and the basis complete, we can compute the scale parameters $\lambda_k = e^{\mathbf{B}\mathbf{v}}$, project $\mathbf{s} = \mathbf{W}\mathbf{x}$, and use $q(\mathbf{x}|\mathbf{v}_n) \propto \exp\left(-\sum \sqrt{2}|s_j|/\sqrt{\lambda_j}\right)$.

### Model comparison results

In order to make sure the models are not over-fitting the training data, we can hold out part of the image data and evaluate the model likelihood on this likelihood set. In practice, we find that because so many image patches are available for training (i.e. our data set is very large), the models tend not to overfit. Nevertheless, we will use hold-out sets.

We chose a value of 0.1 for $\delta$. At this level of quantization, pixel values with a standard normal distribution

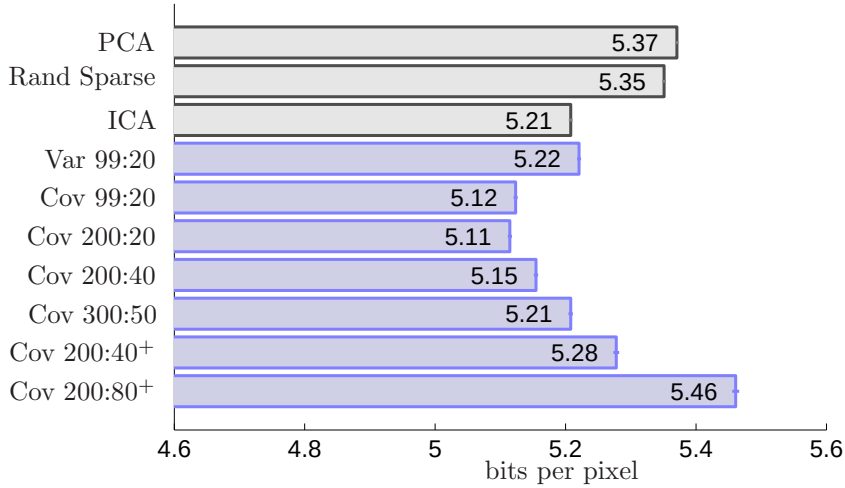


Figure 4.16: Coding cost of 10×10 image patches computed using model likelihood and kernel density estimates. We compared the variance component (noiseless, Laplacian conditional density) and covariance component models to standard linear statistical models (in gray: PCA, ICA). The last two covariance component models used positive-only higher-order coefficients; all the rest used Laplacian priors. Error bars indicate standard error of the mean.

are encoded at approximately 8 bits. We adapted several models on a training set of 10×10 image patches (approximately 38M unique patches) and evaluated coding efficiency using model likelihood on a held out set (10%)of images. The DC (mean luminance) of each image patch was removed during preprocessing (because none of these models were trained to account for the slightly different statistics of DC in image data). We estimated the coding cost for 50000 patches from the validation data set, using 200 kernels for the hierarchical models (we found this number sufficient to give consistent estimates). For the variance component model, we used the Laplacian distribution for the linear coefficient density $p(\mathbf{s}|\mathbf{v})$; for a model with only a complete set of linear features, the covariance model is essentially equivalent to the variance model with the Gaussian conditional distribution, so this allowed us to compare the two forms of conditional densities. Exact values of data likelihood were computed for PCA and ICA models. ICA basis functions were obtained using the natural gradient (Amari, 1999) and a Laplacian prior for the linear coefficients.

We also checked the efficacy of using the data kernel approximation with bandwidth ranging from 0.5 to 2.0. While this showed the same relative coding efficiencies as the results obtained with the hyper-parameter kernels, the likelihood estimates were much lower with this method, suggesting that kernels with small bandwidths did not sufficiently tile the data space and kernels with large bandwidths smoothed out important features of the likelihood.

Using the hyper-parameter kernels, we were able to obtain reliable estimates by sampling randomly the latent variables and building up distributions as described above. However, one issue required some correction — because of the constraints on the norm of the parameters (see the Methods section above), the latent variable priors did not always match the inferred MAP values. Specifically, the empirical distributions of latent variables often had larger or smaller variances than the priors (though the shapes of the distributions were consistent with the assumed sparse densities). When sampling these variables (e.g. $\mathbf{y}$, the covariance coefficients), empirically measured variances were used.

A comparison of bit-per-pixel coding costs is shown in Fig. 4.16. The "PCA" model describes the data distribution using only the covariance matrix. The "Rand Sparse" projected the whitened data onto a set of random orthogonal projections, and measured the likelihood of the resulting coefficients under the Laplacian distribution. The "ICA" model was obtained using the natural gradient algorithm for a noiseless, complete model (Amari, 1999). Variance component and covariance component models of different sizes were evaluated. The first number indicates the number of linear basis functions (in the variance component model) or the number of linear features $\mathbf{b}_k$ (in the covariance model); the second, the dimensionality of

the latent higher-order variables ($\mathbf{v}$ or $\mathbf{y}$). For the last two models, "Cov 200:40$^+$" and "Cov 200:80$^+$", the covariance coefficients were constrained to be all-positive (and model parameters adapted with this prior). Therefore the first of these contained the same number of covariance components as "Cov 200:40", while the second could, in principle, cover the same space of statistical structures.

Most of the hierarchical models yield higher data likelihoods than the single stage linear models such as PCA and ICA. In fact, the best model ("Cov 200:20") showed an improvement over ICA results that were on par with the that gained by going from the Gaussian (PCA) to a sparse (ICA) model of natural images. The variance component model with the Laplacian conditional density was slightly worse than ICA, and worse than the covariance model with the complete set of linear features ("Cov 99:20"). This suggests that mixtures of Laplacian distributions of different variances give too sparse marginal distributions for the linear coefficients. For the covariance component models, increasing the number of latent variables also resulted in lower likelihood. This could be because for such small images (10×10), there is only limited higher-order structure, and 20 covariance components sufficiently account for it. An over-complete model (with 200 linear features) did show an improvement in likelihood one with a complete number of $\mathbf{b}_k$s (99), but only when the number of latent variables was small.

The models with all positive coefficients were significantly worse that those that employed symmetric prior distributions. There are several possible reasons. First, the prior could be incorrect for the training data. Current inference methods only provide empirical distributions of MAP estimates, which are not the same as the posterior density, but as demonstrated in Fig. 4.11, these were very sparse for the positive-only models. (A test of a kernel density estimate using the empirical MAP distribution for $\mathbf{y}$ instead of the exponential prior to construct the density resulted in much lower coding cost, which matched the best values of other hierarchical models.)

We verified these results by computing the coding cost estimates on synthetic data generated by sampling from the PCA and ICA models. As expected, data sampled from the PCA model was best described by the PCA model, with the ICA and the hierarchical densities yielding much higher coding cost, and data drawn from the ICA model was best coded by the ICA model.

### 4.4.3 Image restoration: missing pixels

Another validation method afforded by these models is statistical image restoration. Models that better capture the statistics of natural images should be able to perform well in tasks like de-noising and filling in missing pixels. It can be argued that the main feature of hierarchical models presented here is encoding abstract properties of the image, rather than finding more efficient encoding for of their input, and that this function should be tested with other metrics (e.g. perceptual distortion). Nevertheless, the models are designed to accurately capture the statistics of the data, and should also fare well in low level tasks such as image restoration. Research into other metrics that more directly test the models' ability for abstraction, is left for future work.

Model based image restoration allows us to fill in unknown quantities with estimates provided by the models. For example, given an image $\mathbf{x}$ with a set of pixels missing ($\mathbf{x}_h$), we would like to fill them in with estimates that minimize the mean squared error (MMSE) $E[(\mathbf{x}_h - \hat{\mathbf{x}}_h)^2]$. The MMSE estimate is the expected value $\hat{\mathbf{x}}_h = E[\mathbf{x}_h]$. If the image model is Gaussian, i.e. the set of observed and deleted pixels is distributed as

$$\begin{bmatrix} \mathbf{x}_o \\ \mathbf{x}_h \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \qquad \text{where} \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_o \\ \boldsymbol{\mu}_h \end{bmatrix} \qquad \text{and} \qquad \mathbf{C} = \begin{bmatrix} \mathbf{C}_o & \mathbf{C}_{oh} \\ \mathbf{C}_{oh}^T & \mathbf{C}_h \end{bmatrix} \tag{4.49}$$

the distribution of the deleted pixels $\mathbf{x}_h$ is a Gaussian defined as

$$\mathbf{x}_h|\mathbf{x}_o \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\mathbf{C}}) \qquad \hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_h + \mathbf{C}_{oh}^T \mathbf{C}_o^{-1}(\mathbf{x}_o - \boldsymbol{\mu}_o) \qquad \hat{\mathbf{C}} = \mathbf{C}_h - \mathbf{C}_{oh}^T \mathbf{C}_0^{-1} \tag{4.50}$$

and the MMSE estimate is simply $\hat{\mathbf{x}}_h = \hat{\boldsymbol{\mu}}$.

For ICA and sparse coding models, the MMSE is more difficult to compute — the latent variable prior gives piece-wise constant log-probabilities and results in non-trivial integrals. On the other hand, we can easily obtain the MAP estimates of the basis function coefficients when some pixels are missing. In a noisy sparse coding model, we can set the noise parameter of the missing pixels to infinity, and then maximize the posterior of the coefficients (an approach taken in Lewicki and Olshausen, 1999). In square noiseless ICA, where every pixel configuration maps deterministically to a set of coefficients, we can directly climb the likelihood gradient (while keeping the observed pixels fixed). However, the MAP estimate of the latent variables does not give the MMSE estimate for the pixels. In practice it gives worse average error for small image patches than the PCA estimate (Eqn.4.50), and hence is not included in the comparison below. We face a similar difficulty using the variance component model with conditional Laplacian distribution, and this model is also not included.

In the hierarchical models, the expected value is computed by marginalizing over the latent variables, and this integral is also not tractable. There are two approaches: we can replace it with the MAP estimate of the latent variables, or approximate it numerically by sampling the latent variables. Both approximations allow for easy computation of the MMSE estimate for the pixel values. In the first case, one set of latent coefficients $\hat{\mathbf{y}}$ is computed, which yields a multivariate Gaussian distribution over images, with the MMSE estimate given as in Eqn. 4.50. A numerical approximation to an integral over different values $\mathbf{y}$ approximates the model density with a mixture of finite number of Gaussian distributions. In this case, the expected value $\hat{\mathbf{x}}_h$ is an average of the expected values under each of these Gaussian models, which again are computed as in the PCA solution above.

MAP estimate with respect to the observed pixels only (i.e. $\hat{\mathbf{y}}$ that maximizes $p(\mathbf{x}_o|\mathbf{y})$), is difficult to compute. Here we replace this with the posterior over all the pixels $p(\mathbf{x}|\mathbf{y})$, but first assign to the deleted pixels the MMSE values under the PCA model. The PCA model also describes the canonical distribution in the hierarchical model ($\mathbf{y} = 0$); thus using it to fill in the unknown dimensions should not bias the latent variables obtained by maximizing the posterior (though this is an approximation that does not guarantee the true MAP estimate). After $\hat{\mathbf{y}}$ is computed, we fill in the deleted pixels again, this time using the pixel covariance matrix given by the MAP estimate.

An additional technical issue is that the models are trained in the whitened data space (which included arbitrary rotations), and the pixel values were "deleted" in the original data space. Let the whitened data $\mathbf{x}'$ be related to the original images through the "un-whitening" matrix, $\mathbf{x} = \mathbf{U}\mathbf{x}'$. To compute the estimated pixel values in the original space, we transform the covariance matrix, given in the whitened space $\mathbf{C}' = f(\hat{\mathbf{y}})$ to covariance in the original space $\mathbf{C} = \mathbf{U}\mathbf{C}'\mathbf{U}^T$ and use this model for image restoration.

In practice, the numerical approximation to $E[\mathbf{x}_h]$ (using samples $\mathbf{y} \sim p(\mathbf{y})$) gave worse results than that obtained using the (approximate) MAP estimate $\hat{\mathbf{y}}$; we report the results of the latter method only. We ran several simulations, deleting 25%, 50%, 75%, and 90% of pixels in 10×10 image patches and testing the same 9 models evaluated in the coding cost analysis above. The analysis was performed on 5000 patches sampled randomly from a set of natural images. Results are plotted in Fig. 4.17. In the "easy" regime, where most pixels were not deleted, the hierarchical models did better than PCA, and unlike in the coding cost analysis, larger models (with more parameters) further improved reconstruction performance (except for models with positive-only coefficients). One possible explanation is that coding cost analysis used model priors for the latent variables (to construct the kernel density estimates), and these priors might not match the distributions of these variables. Here, MAP estimates of the latent variables were computed for each image patch, and very sparse coefficients $\mathbf{y}$ were used for subsequent image restoration.

With most of the pixels deleted, however, the hierarchical models performed worse than PCA. This could be caused by the approximations employed in the MMSE pixel estimates under the hierarchical models, or it could be related to more fundamental problems in using representations of higher-order image structure for estimation of precise pixel values. It is also unclear why the pixel restoration performance is so consistently different among different models when 90% of the pixels are deleted. There is no obvious trend (e.g. smaller hierarchical models do better than larger ones) and the differences among the average performance scores are significant when all 10,000 test images are evaluated.

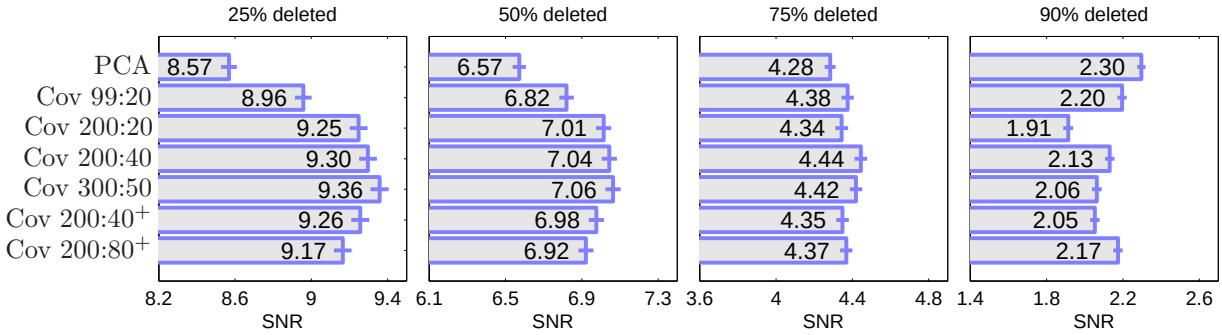


Figure 4.17: Results for image restoration (pixel fill-in) for different models. The plots show average SNR for the deleted pixel values, in dB, computed as $10\log_{10}(var(\mathbf{x}_h)/var(\mathbf{x}_h - \hat{\mathbf{x}}_h))$. The error bars give standard error of the mean over the 10000 test patches. The scale is the same in all panels.

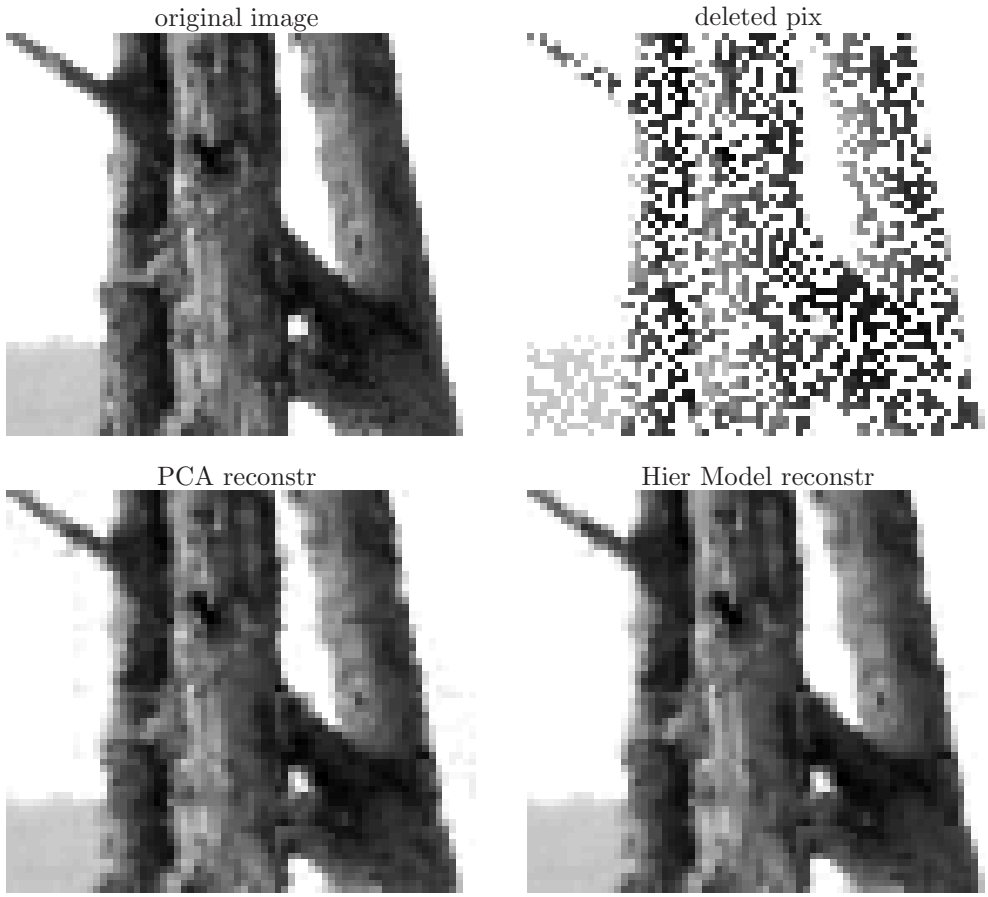


Figure 4.18: Missing pixel restoration on one part of a natural image. 50% of the pixels were deleted from the original image, which was subdivided into 10×10 patches and then filled in based on the Gaussian model and the hierarchical model (*Cov:300:50*). Restoration using the hierarchical model gives slightly sharper edges and more pronounced oriented textures, though the differences are hard to discern.

### 4.4.4 Summary

In this section we quantitatively compared several hierarchical models as well as standard linear generative models. We evaluated how well the model probability densities matched the data and found that some

hierarchical models provide a better fit. Although the improvement was not large (0.1 bits over ICA), it was comparable to the improvement in going from PCA to ICA. We found that larger models with a large number of parameters were generally worse than smaller models. We do not believe the models overfit (there is a very large number of training examples), but it is possible that the prior in the models is not correct. With increasing dimensionality of the latent variables, their distributions are typically more sparse, and density estimates obtained by sampling from Laplacian priors do not match the structure of the data. Also, patches that are only 10×10 pixels do not contain much long range structure, and we expect to see a bigger effect on large patches. However, training a large number of models on larger patches takes significant computational resources.

We also tested the performance of the hierarchical models on a task that requires a good model of the data — filling in pixels that have been deleted from natural images. Here too some hierarchical models performed better than the standard Gaussian model, although the benefits vanished when most of the pixel data were deleted. In principle, a better statistical model should still give an improved score on such task, and it is possible that our estimation method biased the results (we did not marginalize over the latent variables $\mathbf{y}$ or compute the optimal latent variables given only the observed pixels).

As Fig. 4.18 demonstrates, the differences in the reconstructed images are very slight, suggesting that this approach is not the application of the hierarchical models. When most pixels are deleted, model-based reconstruction also begins to "imaginatively" fill in pixels. While PCA gives blurry estimates, models like sparse coding will return "edgier" images, and the hierarchical models include even more structure: variations in contrast, dominant orientations, and spatial frequency changes. This suggests that better, or at least more revealing, tests of model performance can be obtained using other metrics besides the mean squared error (e.g. perceptual distortion metrics), or results computed on more structured data such as textures. A host of other applications are possible as well; for example, one can train classifiers on model representations of images rather than on raw pixel data, to see if the model captures fundamental properties of the data, and then compare performance to other supervised learning methods. This, however, is outside the scope of this dissertation.

## 4.5 Discussion

In this chapter we have developed an extension to the hierarchical variance model that can more directly capture patterns in the correlational structure of the data. This model also obviates the need to marginalize over the linear coefficients to compute the likelihood or the MAP estimates of the higher-order variables, and it provides a cleaner description of the canonical distribution (defaulting to the isotropic multivariate Gaussian even when the number of linear features is under- or over-complete).

Several issues remain unresolved. As discussed in section 4.3.3, the form of the prior on the latent variables can affect the estimated parameters. A symmetric prior makes certain assumptions that might not hold for the non-linear structure modeled here, and it is necessary to establish whether these assumptions are valid for natural images. This could also tell us what parameters are truly optimal and allow us to make predictions about coding in the visual system with more certainty. To test these ideas, however, different learning algorithm must be developed that does not rely on the MAP estimate. Sampling methods have been effectively employed for similar hierarchical modeling problems, and it should be possible to apply them to these models as well. An additional benefit of these techniques is that they allow us to estimate a posterior distribution over the parameters (as well as the latent variables), rather than computing only the maximum likelihood estimates. All the analysis performed on the current model can be reinforced by characterizing an entire set of parameter values, rather than a single estimate currently computed.

In this chapter I have shown that the proposed hierarchical models are better statistical models of natural images (though the benefits on low level tasks are limited). More central to the aims of this dissertation is the application of the developed techniques to the study of neural processing, including the analysis and

interpretation of neural activity, and predictions of neural response properties. The next chapter addresses these points directly.

# Chapter 5

# Theoretical predictions for cortical neural function

What new insights about processing in the visual cortex do the proposed models provide? In this chapter I first analyze the behavior of model units in response to stimuli typically used to characterize complex cells and show that many of these properties, as well as more subtle effects observed in V2, are exhibited by the model (Sec. 5.1). This suggests that at least some of the complex behaviors of cortical neurons can be derived directly from the statistical structure of natural scenes. I also draw parallels between learned model parameters and non-linear descriptions of receptive fields for V4 neurons. In section 5.2 I discuss the implications of hierarchical statistical modeling for interpretation of neural activity — if neural activation corresponds to states of latent variables in hierarchical statistical models, how do we interpret this code and how does this affect the questions we pose experimentally? Finally I show how model representations of images can be used to derive new descriptions of cortical neurons and make better predictions of neural responses to novel stimuli (Sec. 5.3). Because the covariance model generalizes the variance model and yields a more flexible description of image structure, it is this model's representations that are analyzed and compared to physiological results.

## 5.1 Comparison of model units to visual neurons

### 5.1.1 Classical properties of complex cells

The standard model for complex cells (Movshon et al., 1978; Heeger, 1992) was described in chapter 1. It consists of two linear filters (oriented, localized, band-pass functions 90° out of phase) whose output is squared and summed to give the neuron's response. The model successfully explains the selective response of complex cells to gratings of various orientations and spatial frequencies, and also accounts for their basic non-linear properties, including the strong response to edges of both polarities and phase invariance when presented with sinusoidal gratings. A large number of other non-linearities in the responses of these cells have been identified (for review see Albrecht et al., 2004; Carandini, 2004). A classic example is "cross-orientation suppression": when a grating at the preferred orientation is masked with an orthogonal grating (i.e. a second grating is linearly added to the preferred stimulus), the neuron's response is significantly suppressed (Morrone et al., 1982; Bonds, 1989). This effect is not predicted by the classical model that only includes excitatory image features. In addition, image structure in the region surrounding the neuron's receptive field (which according to the standard model should not affect response) modulates neural activity in a variety of ways. Typically, oriented structure extended into the surround suppresses the neuron, and

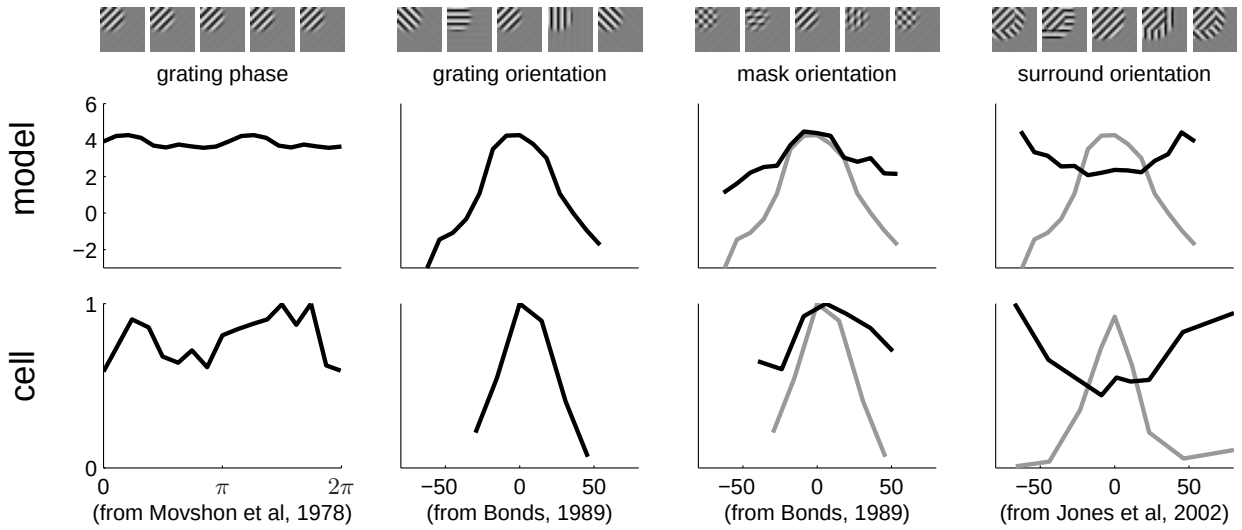


Figure 5.1: When presented with sinusoidal gratings, the model unit (previously analyzed in Fig. 4.7) replicates common aspects of responses in complex cells in cortical area V1. It is insensitive to the phase of the grating, but highly tuned to its orientation. Superimposition of a second grating of variable orientation on top of the preferred grating reduces the response, with maximal reduction when the superimposed grating is orthogonal. Suppression by an annulus grating is tuned, and maximal at the preferred orientation. All model neuron responses are plotted on the same scale (see first panel); cell firing rates were normalized to a maximum value of 1; peak orientation was shifted to $0^\circ$ for both model neuron and all cells.

this "surround suppression" is weaker when the surround contains gratings at other orientations (Jones et al., 2002; Cavanaugh et al., 2002).

A variety of models have been proposed that incorporate these effects, and some are quite simple and elegant (Heeger, 1992; Carandini et al., 1997; Cavanaugh et al., 2002). They account for many of the suppressive effects using inhibitory signals from other neurons, or a few unoriented suppressive fields. However, these results are obtained by starting with the observed neural behavior and then formulating parsimonious mathematical descriptions. What results is a mechanistic model that abstracts the observed phenomena, but it is fitted to data from specific neurons (often with few parameters). A much more difficult (and quite a different) problem is to make predictions based only on theoretical computational principles and then explain the function of the observed effects, i.e. in terms of their roles in achieving the computational goal. Only recently have statistical models begun to provide such functional explanations for non-linear properties of complex cells (Schwartz and Simoncelli, 2001).

The hierarchical models developed in this dissertation were not fitted to neural responses; nevertheless, the configuration of some parameters in the model suggests that many units encode image structure that is similar to that signaled by complex cells. In order to quantitatively evaluate this relationship, we compared the tuning properties of these units to well-known properties of complex cells described above. We trained the model on a large set of natural images, after which the parameters were fixed. Model response was computed to a set of gratings by estimating the MAP values $\hat{\mathbf{y}}$ for each stimulus. All stimuli were preprocessed in the same way as the training natural images (by subtracting the mean and whitening the images).

We identified the location, orientation, and spatial extent and frequency of a windowed sinusoidal grating that best activated the model unit (that is, the grating that yielded the most positive value of $\hat{y}_j$), and then varied each tested parameter to obtain the unit's tuning curves. Many units (40 out of 150) showed orientation tuning, phase invariance, and cross-orientation suppression. As an example, we show in detail the responses of one typical unit (Fig. 5.1, same unit as plotted in Fig. 4.7). This particular model neuron exhibited a variety of properties observed in complex cells in V1 and cells in V2, including phase invariance, orientation tuning, and complex suppressive effects.

Note that these results were obtained with no assumptions about the image structure encoded by visual neurons and without fitting a model to data from physiological experiments. Any correspondence emerged as a consequence of the computational goal of capturing image distributions and the statistical regularities present in natural images. Here I highlight the correspondence between one model unit and typical complex cells, but similar response properties are obtained for the population of units grouped into the same cluster in the population analysis in section 4.3.2.

### 5.1.2 Second-order receptive fields

Many properties of cortical cells are not well described by configurations of grating stimuli, and a large body of physiological, imaging, and psychophysical work has analyzed neural responses to more complex parametric stimuli. Particularly interesting among this research are studies that have explored the connection between cortical responses to simple parameterized stimuli and mid-level visual tasks such as texture perception and boundary detection (Leventhal et al., 1998; Lee et al., 1998; Nothdurft et al., 2000; Rossi et al., 2001; Landy and Oruc, 2002), figure-ground segregation (Lamme, 1995; Zhou et al., 2000), and illusory contour detection (Grosof et al., 1993; von der Heydt and Peterhans, 1989). Some of this work has characterized V1 and V2 neurons in a manner consistent with the representations learned by the hierarchical models. Specifically, "second-order" pattern processing, thought to underlie texture coding, natural contour detection, and other components of perceptual organization (for reviews, see Chubb et al., 2001; Baker and Mareschal, 2001), is described in terms image features closely related to encoding by units in the hierarchical models described above. In this section, I explore this connection in more detail.

In natural images, transitions between different surfaces, or an object and background, are often characterized not by changes in luminance, but by other changes: one dominant orientation is replaced by another, or the spatial frequency distribution varies across some region. Often a visually salient edge does not demarcate regions of different mean luminance, but the contrast (the amount of variability) on two sides of the edge is quite different. All of these image aspects can be characterized by second-order statistics — the variance and covariance patterns across space — and many units in the hierarchical models describe exactly such changes in local statistical structure.

As a concrete example, let us examine the set of model units that comprise the blue sub-tree of the clustering in Fig. 4.9. This large set, which makes up more than a third of the full population, encodes localized image contrast. I replot only this sub-population in Fig. 5.2. These units describe oscillating regions of high and low contrast, localized to one part of the image patch, and oriented in parallel bands. The spatial envelope of each unit is well described by a 2D Gabor function. Compared to optimal linear codes, which also resemble Gabor functions, these spatial covariance components are larger (they span a greater spatial extent of the patch), more multi-scale (there are many more large low-frequency units), and have more oscillations (larger number of positive and negative subunits). Within the oscillations of the spatial envelope, the weights to the linear features do not vary by orientation, frequency, or phase, meaning that these units are insensitive to these dimensions. As a population, this set of units encodes in a distributed manner the location of structure in the image and can describe non-luminance edges using a compact code.

The fact that the model learns a distinct set of such units means that this type of image structure is independent of other higher-order structure, such as orientation and spatial frequency. This leads to a novel set of predictions for optimal coding of second-order structure, in which a separate channel encodes changes in contrast (in a distributed code of oriented localized units) while other units convey information about other aspects of visual texture. While this encoding might not correspond directly to neuron types in V1 or higher areas, there are nevertheless interesting parallels between model predictions and experimental observations.

A number of physiological studies have analyzed cortical responses to second-order gratings to tease apart representation of orientation, contrast changes, and frequency. They rely on a limited, hand-constructed set of stimuli that cannot probe all these aspects of neural processing at once, but some findings are consistent across studies. As mentioned previously, a significant proportion of V1 and V2 (or equivalent cat areas

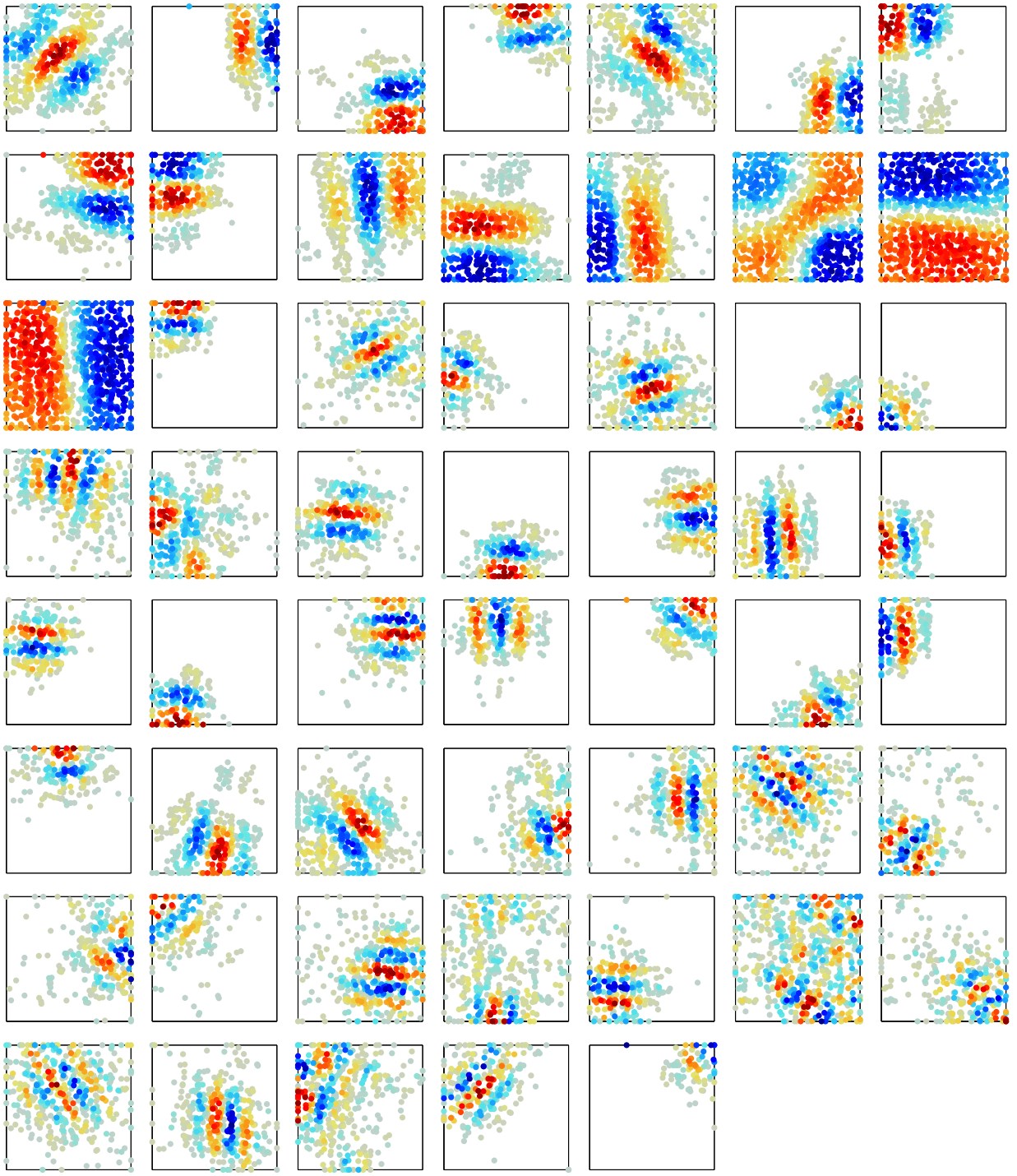

Figure 5.2: Location-only units in the covariance component model (all 54 units clustered in the right-most branch of the dendrogram in Fig. 4.9), replotted as in Fig. 4.7a, using dots to indicate centers of each image feature, instead of lines. The weights in all these units (the colors of the plotted points) are well explained by the spatial location of the underlying linear feature $\mathbf{b}_k$ (in each panel we omitted weights close to zero, i.e. those with magnitude less than 10% of the largest weight). The set of covariance components form a multi-scale, distributed representation of spatial contrast in the image patch.

17 and 18) neurons respond selectively to second-order structure (Zhou and Baker, 1994; Leventhal et al., 1998; Mareschal and Baker, 1998b). A larger proportion of neurons in V2 (or area 18) exhibit these non-linear properties (Leventhal et al., 1998; Zhou and Baker, 1996). Similar second-order stimulus sensitivity has been observed for motion stimuli (that include changes in correlations across time) in monkey area MT (Albright, 1992; O'Keefe and Movshon, 1998), but these results cannot be directly related to current models of static images. The extent of summation for second-order patterns is larger than that of luminance edges (Sukumar and Waugh, 2007), and neurons typically prefer lower spatial frequencies for contrast-defined gratings than for luminance gratings (Zhou and Baker, 1996; Mareschal and Baker, 1998b). There is also evidence that while neurons are highly sensitive to the orientation of contrast modulations, they are much more invariant to the orientation of image structure within each oscillation of the envelope (Leventhal et al., 1998; Mareschal and Baker, 1998a). Evidence from psychophysical studies supports the view that underlying neural mechanisms pool across local orientation information (McGraw et al., 1999).

For example, Mareschal and Baker (1998a) found that most neurons in cat area 18 respond strongly to luminance gratings of a particular orientation (their "preferred orientation", Fig. 5.3a). The non-linear properties of these cells were tested with a second-order gratings, constructed by multiplying a high frequency "carrier" sinusoidal grating with a low frequency "envelope" grating. When presented with second-order gratings with the preferred orientation carrier, and a varying orientation envelope, the neurons are again highly tuned for the orientation (here, of the envelope). The preferred envelope orientation coincides with the preferred luminance orientation (Fig. 5.3b). However, when the envelope orientation is fixed and the carrier varied, the neurons are much less sensitive to the parameter change (Fig. 5.3c), and the preferred carrier orientation does not correlate strongly with the preferred luminance or envelope orientations.

In order to quantitatively compare model responses to cortical neurons, we performed a "neurophysiological" analysis on one of the model units that encodes localized contrast, using a protocol similar to that used by Mareschal and Baker (1998a). We tested the first unit in Fig. 5.2 with second-order gratings. These stimuli were constructed by multiplying two 2-dimensional sinusoidal gratings, the carrier $C_{x,y}$ and the envelope $E_{x,y}$,

$$I_{x,y} = C_{x,y}(1 + E_{x,y})\,, \tag{5.1}$$

where the two sinusoidal gratings are parameterized with orientation and frequency (phase was fixed to 0),

$$C_{x,y} = \sin(2\pi f_c t) \qquad t = x\cos\theta_c + y\sin\theta_c \tag{5.2}$$

$$E_{x,y} = \sin(2\pi f_e u) \qquad u = x\cos\theta_e + y\sin\theta_e\,. \tag{5.3}$$

We first generated a set of luminance gratings (frequency 4pix/cyc) to test the unit's response to plain oriented images. For the envelope-varying stimuli, we used a fixed carrier grating ($f_c = 2.9$pix/cyc, $\theta_c = 60°$) and varied the orientation of the envelope ($f_e = 11$pix/cyc). For the carrier-varying stimuli, the envelope sinusoidal grating was fixed ($f_e = 11$pix/cyc, $\theta_e = 40°$) and the carrier rotated ($f_c = 2.9$pix/cyc). After generating these images, we preprocessed them as the training natural images: the mean luminance was removed from each patch, the mean contrast adjusted to match natural image data, and the dataset was whitened. We then computed the MAP values for the covariance components $\hat{\mathbf{y}}$.

The response of this particular unit ($y_{134}$) is shown in Fig.5.4. As expected, its response to luminance gratings was relatively weak and not tuned for a particular orientation (this is unlike the neurons probed with second-order gratings, which are quite selective for orientations of luminance gratings). However, this unit was highly tuned for the orientation of the grating envelope, which was consistent with the structure of its weights $w_{jk}$. When the envelope was fixed to the optimal orientation, the model unit was largely insensitive to the orientation of the carrier, "firing" strongly for all stimuli.

This response profile is consistent with some of the observed properties of early cortical neurons. The model is able to replicate localized, oriented tuning to contrast structure without making any prior assumptions about how such representations are organized. It reproduces the typical finding that cortical neurons are much less sensitive to the orientations within the carrier of second-order gratings, though it is perhaps more

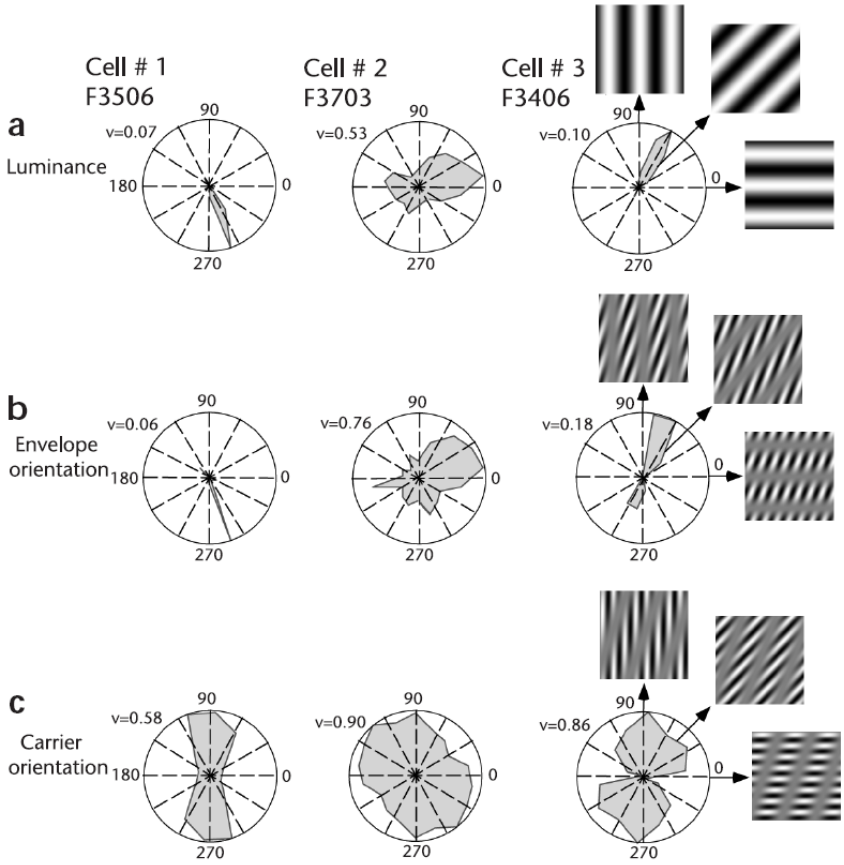


Figure 5.3: The response of three cat area 18 neurons to drifting luminance and second-order gratings (reproduced from Mareschal and Baker, 1998a). ***a***. These cells were fairly well tuned to the orientation of a luminance grating (stimuli shown on right). ***b***. Responses showed similar tuning to changes in envelope orientation (carrier held constant at preferred luminance grating orientation). ***c***. At the preferred envelope orientation, responses showed weaker tuning to the carrier orientation.

unequivocal about this property than the real neurons, which retain some selectivity to the carrier orientation. It also captures the larger spatial scale of second-order sensitivity, though here too, we have yet to establish a quantitative correspondence between the scales of integration over the visual field. One significant difference the model representations and V1/V2 neurons is that most neurons are also tuned for, and respond more strongly to luminance ratings (typically of the same orientation as contrast-modulated patterns), while the model units belonging to the group in Fig. 5.2 are insensitive to orientation or frequency of local image structure. It is possible that other units in the model that do not so obviously encode contrast modulations nevertheless respond second-order patterns while also retaining selectivity for oriented luminance gratings. On the other hand, it is more than likely that the model does not capture all of the wide range of properties of visual processing in V1 and V2, and must be further refined in areas where its predictions cannot be reconciled with neural data.

One benefit of the theoretical model is that it provides an account of how a population of such neurons encodes the structure in an image, something that is difficult to glean from isolated studies of single neurons or performance metrics in psychophysical tasks. Future analysis of interaction (competitive or otherwise) between individual units could provide new insights into the coding of complex edges and texture boundaries by populations of non-linear neurons.

Beside the population of units described in this section, the model employs a host of other types of units to describe the variation of image structure across visual space. A large number of units encode changes

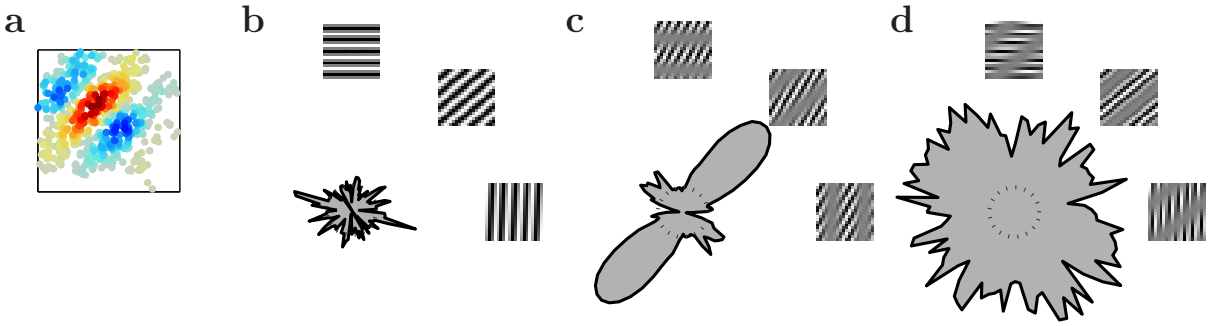


Figure 5.4: The response of the first unit in Fig. 5.2 to stimuli similar to those used by Mareschal and Baker (1998a). This unit encodes spatial contrast defined by several lobes oriented at approximately 40° (***a***). ***b***, ***c***, ***d***. Responses of this model unit to luminance and second-order gratings; the angle of the polar plot indicates the orientation of the grating, and the radius the MAP value $\hat{y}_j$ computed for that image patch. Example stimuli are shown at three points in the polar plot for each condition, as in Fig. 5.3. This model unit responds weakly to luminance gratings (***b***), is highly tuned for envelope orientation (***c***), and is fairly insensitive to carrier orientation (***d***). All polar plots are drawn at a fixed scale: the center corresponds to the minimum value of $\hat{y}_j \approx -0.55$ across all three conditions, the dashed line indicates $\hat{y}_j = 0$, and the peak response is $\hat{y}_j \approx 2.3$. Note that its responses are symmetric in the orientation plots because the gratings were static, unlike the drifting gratings used in the physiological experiments.

in orientation that might underlie coding of texture boundaries, and more detailed comparisons between model predictions and physiological and psychophysical findings could reveal more parallels. It would also be interesting to examine model encoding of illusory contours, long studied in V2 as precursor computation required for object representation (von der Heydt et al., 1984; von der Heydt and Peterhans, 1989). It has recently been shown that neurons sensitive to oriented contrast-modulated images exhibit similar tuning properties when probed with illusory contours, such as those resulting from phase shifts in the background grating (Song and Baker, 2007). These results were obtained with rather idiosyncratic stimuli, like broken lines and square gratings, and without a theoretical explanation they are difficult to reconcile with other non-linear effects in the neural response.

### 5.1.3 Relationship to spike triggered covariance

The results described above have relied on simple sets of parametrized stimuli, the selection of which was guided by the experiments' intuition about the response properties of neurons under study. An alternative approach, a family of methods called *spike triggered neural characterization*, uses randomly generated stimuli to map out neural response to arbitrary images. This method makes fewer assumptions about the computations performed by visual neurons, though it has its own set of limitations (especially in the number of images required to fit the model). When used to characterize non-linear aspects of the neural response, this method bears a close relationship to the covariance component model.

The general approach of spike triggered neural characterization aims to quantify neural response properties to a set of general stimuli while making few assumptions about the dimensions relevant for the response. This is done by presenting a set of random input patterns, and then studying how the statistics of those that elicit neural response (the spike triggered distribution) differ from those of the entire ensemble (for a review, see Schwartz et al., 2006). Neurons whose responses are fairly linear can be effectively characterized using the mean of the spike triggered distribution to obtain the *spike triggered average*,

$$STA = \frac{1}{N} \sum \mathbf{x}_n \,, \tag{5.4}$$

computed by summing over the stimuli that caused the neuron to spike. When analog values for the neural response $r(\mathbf{x}_n)$ are available (e.g. when internal currents can be measured, or neural responses over many

trials can be temporally binned and averaged), this estimate is computed as

$$STA = \frac{1}{\bar{r}} \sum r(\mathbf{x}_n)\mathbf{x}_n \,, \tag{5.5}$$

where the sum is now over all stimuli but is normalized by the average response rate $\bar{r}$ (Chichilnisky, 2001).

For non-linear neurons such as complex cells, the mean of the spike triggered distribution does not capture its statistical properties, just as a single linear filter is a poor predictor of neural response. However, the method can naturally be extended to characterize the covariance of the spike triggered distribution (de Ruyter van Steveninck and Bialek, 1988; Brenner et al., 2000), computed as

$$STC = \frac{1}{\bar{r}} \sum r(\mathbf{x}_n)\mathbf{x}_n\mathbf{x}_n^T \,. \tag{5.6}$$

Once this covariance is estimated, its spectral decomposition is analyzed to reveal directions of larger and smaller than average variation. If the (reference) distribution of all stimuli is normal, eigenvectors associated with positive eigenvalues indicate directions of larger than expected variation; these directions are deemed excitatory, since stimuli along them are likely to elicit response, even though they might be relatively rare in the entire ensemble. Conversely, negative eigenvalues indicate inhibitory directions; moving along them in stimulus space quickly reduces neural response.

In the model described above, the latent variables that represent higher-order visual neurons act to modify the covariance of the data. Associated with each model unit is a covariance pattern $\exp(\mathbf{A}_j)$ that is more pronounced the more active (positive) the unit is. Data sampled from the model when only this unit is active will have this covariance structure, and the spectrum of its covariance will reveal significant small and large eigenvalues for vectors such as those shown in Fig. 4.9.

If the distribution of image data perfectly matched the model distribution and we had access to the generating values $\mathbf{y}$, the STC of these activities would be proportional to the covariance given by the model parameters. However, this is turning the STC method around: real neurons do not generate visual patterns, and what we really want to compute is the distribution of spike-triggered data when the stimuli are random, e.g. in response to white noise. Therefore in order to draw direct comparisons to neural data, we should be using the response of the model (MAP estimates of $\mathbf{y}$? The expected $\mathbf{y}$ for a given stimulus?) to a random and controlled set of stimuli, as is done experimentally. The exact solution is difficult to obtain analytically, as there is no closed-form expression for the MAP or the expected value of $\mathbf{y}$. Another factor is that the model includes a sparse prior, so the inferred estimate of a unit's activity depends not only on the log-covariance component associated with this unit, but also on the activity of other model units that compete to represent the distribution with as sparse a vector as possible. These competitive effects can alter, even in a linear generative model, the statistics of spike-triggered stimulus distributions.

In practice, the space of covariance matrices is quite large and model units tend to encode fairly different covariance patterns. Also, MAP estimates $\hat{\mathbf{y}}$ are closely correlated to true generating values in toy experiments (as illustrated in Fig. 4.3) and empirical estimates of STC (using white noise images) yield structure nearly identical to the eigenvector analysis of $\mathbf{A}_j$'s (though STC is significantly noisier, since it requires a great deal of data to accurately estimate the full covariance matrix). This suggests that the link between STC neural characterization and the model's encoding is not so tenuous after all, and that the model can be a useful tool for the prediction and analysis of non-linear neural responses.

Unfortunately, the STC properties of higher-order neurons have not been studied in sufficient detail and few data are available. Complex cells in anesthetized cats, when analyzed using STC, exhibited a small number (typically 2) of localized, oriented, bandpass subunits, related to each other by a phase shift, that acted in an excitatory manner to increase cell firing (Touryan et al., 2005). These cells apparently had no linear receptive field components, and this analysis did not reveal any inhibitory dimensions in the input space.

Another study examined the responses of V1 neurons in anesthetized monkeys to spatiotemporal stimuli consisting of drifting vertical bars (Rust et al., 2005). The spatial component had only one dimension,

thus a direct comparison to our results is not possible. For these cells, however a large and varied number of image dimensions were important for predicting firing rate. Excitatory image features corresponded to localized drifting image structure, while inhibitory features typically had the same temporal frequency but opposite direction. These results cannot be directly compared to model predictions described above, but they is consistent with the classical account of cross-orientation suppression as well as the model's encoding of oriented structure.

It would be interesting to compare STC analysis of V2 and V4 neurons (especially ones that code for complex image structure) with the eigenvector analysis of model units that code similar structure. For example, it is unknown whether neurons tuned broadly for orientation or frequency (see below) are in fact sensitive to a large number of dimensions in input space, or only to a few image features whose spectrum is broad in these parameters. One difficulty is that these neurons often respond weakly to random unstructured images, and standard STC analysis using white noise simply does not drive the cells sufficiently. On the other hand, using natural images for STC estimation can lead to biased results — their distribution poorly understood, nor does it satisfy the conditions required for accurate STC estimation. One solution to these problems might be to devise a random but controlled stimulus set based on representations learned by the model; knowing the statistical structure of the presented stimuli could allow corrections to correlations present in the stimuli, and the presented images would have a more rich statistical structure than Gaussian correlated noise.

### 5.1.4 Spectral receptive fields

Another way to fit non-linear models of neural response is to transform the input space, passing pixel data through a non-linear transformation, and then estimate a linear model (i.e. a STA) in this space. This approach has been applied to describe the response of V4 neurons to sequences of natural images, using a Fourier representation of each image (David et al., 2006). The motivation is that, although cortical neurons encode localized image structure, pooling over different locations to compute a description of the cell's receptive field reduces the number of parameters in the estimated model and produces better predictions of V4 responses to natural images than other approaches. Separating or discarding phase also allows the fitted model to capture non-linear behavior of cortical neurons. In addition, some tuned responses to curvature and shape (Pasupathy and Connor, 2001) can also be related to encoding of spatial frequency and orientation structure (David et al., 2006).

To compute a neuron's *spectral receptive field* (or SRF), response to a set of rapidly flashed natural images is recorded; next, a Fourier transform is applied to the individual images and the phase component discarded; finally, a linear receptive field in the Fourier domain is computed by regressing the neuron's response on the spectral representation of the input images. A related technique was developed for visual areas 17 and 18 in the cat (thought to play similar roles to V1 and V2 in the monkey) by Nishimoto et al. (2006), who applied the Fourier transform to locally windowed image patches to produce a spatio-spectral representation of the receptive field.

Some of the units learned by the covariance component model show tuning that is broad (largely insensitive) to location and phase, but specific for orientation or spatial frequency, and we investigated whether these model neurons match the published descriptions of V4 neurons in the SRF analysis framework. Fig. 5.5 shows all the individual neurons described by David et al. (2006), and next to them the functionally closest model units learned in a single simulation using the covariance component model. We plotted the raw weights for three model units according to frequency and orientation. For these units these two dimensions describe the weights well, as is evident from the scatter plots (the positive and negative weights are well separated).

A significant number of V4 neurons show broad orientation tuning as in Fig. 5.5a; in the covariance component model, orientation-tuned neurons usually encode a single, rather narrow, range of orientations (Fig. 5.5b), though this peak in orientation tuning can be considered either excitatory or inhibitory (depending on the sign of the coefficient $y_j$). Some model units closely resemble V4 neurons that are more narrowly tuned for orientation, but broadly tuned for spatial frequency (Fig. 5.5c, d). The example unit even exhibits a striking

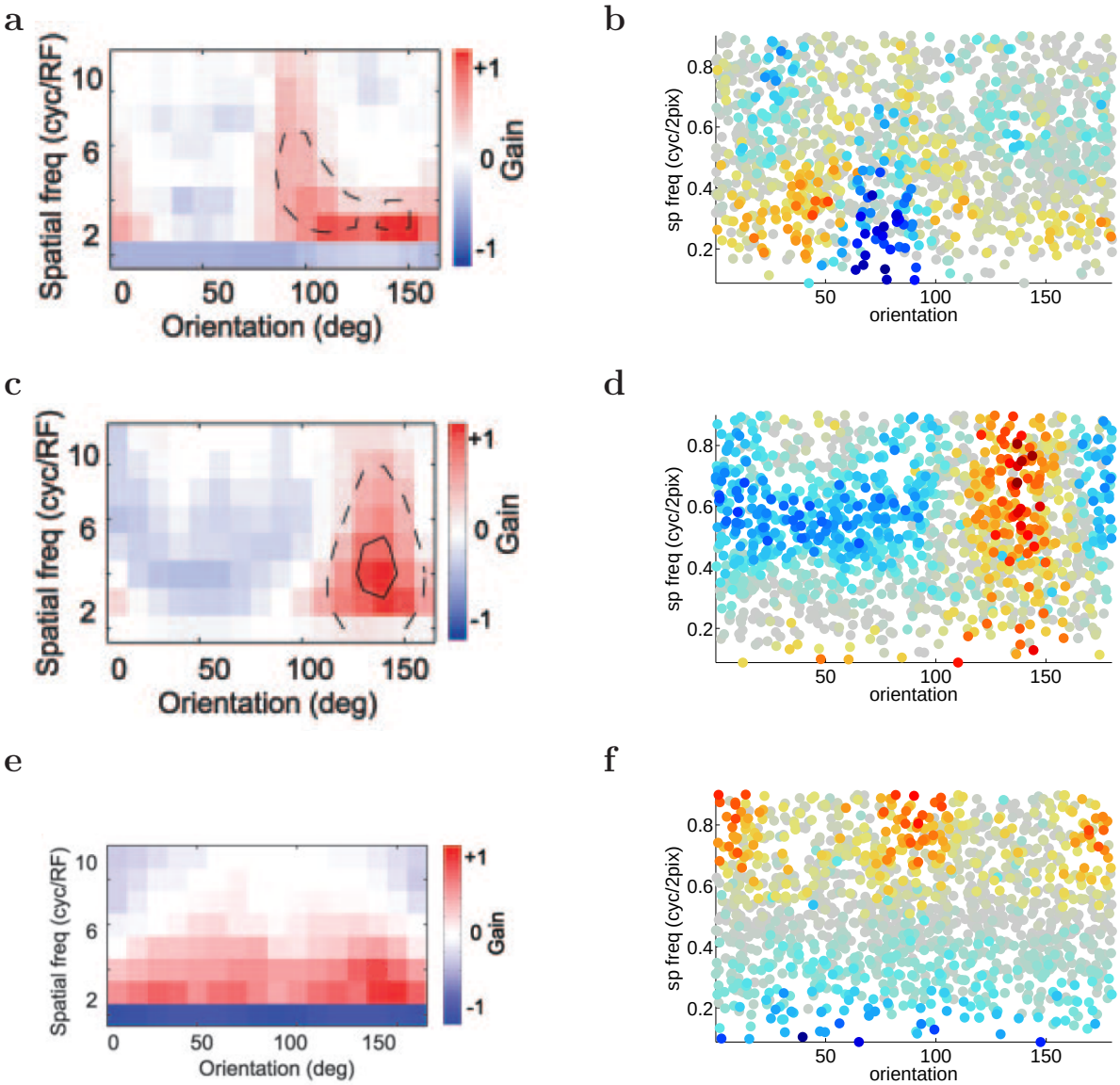


Figure 5.5: A comparison between spectral receptive fields (SRFs) of 3 V4 neurons and parameters of (spatially non-localized) covariance component model units. V4 SRFs (***a***, ***c***, ***e***) replotted from David et al. (2006). For each model unit (***b***, ***d***, ***f***), its weights $w_{jk}$ are placed on the spectral axes according to the orientation and frequency of the associated feature $\mathbf{b}_k$, and colored according to the value of the weight (each panel independently normalized, gray is zero, preferred orientation adjusted to match the V4 SRF.

similarity in the shape of the inhibitory regions in the frequency-orientation space. The model representation also independently encodes frequency structure in the image. Such a coding scheme is consistent with V4 neurons that are tuned broadly to orientation but that are highly sensitive to spatial frequency (Fig. 5.5e, f). Although the profile of spatial frequency tuning in the model unit does not match the band-pass tuning of the V4 cell in Fig. 5.5e, the plots illustrate that the model reproduces the basic structure of frequency-tuned V4 neurons, something that has not been previously derived.

Other experimental work, using classical parameterized stimuli such as gratings and bars, described neurons in earlier visual areas — V1 and V2 — that prefer image structure similar to that predicted by the model (von der Heydt et al., 1992). This set of experiments, though limited by the set of stimuli, specifically examined neural response to localized and non-localized stimuli, something that a spectral analysis (as in

David et al., 2006) cannot uncover. It was found that a small proportion of cortical neurons prefer gratings or other non-localized periodic patterns to bars and edges. These cells responded best to static (non-moving) flashed stimuli, and it was hypothesized that these periodic pattern detectors participate in coding textures. These observations are consistent with the non-localized orientation units learned by the model, but such effects can also be observed for the less localized contrast-coding units shown in Fig. 5.2.

## 5.2 Implications for neural coding

What new interpretations of neural activity do the hierarchical models developed in this dissertation provide? First we can consider the computations underlying the encoding process in these models. Because the models are defined in a generative framework, inference of the latent variables is a complex, non-linear operation. Currently, optimal solutions require iterative methods, but if the computation is approximated by a feed-forward series of steps (as in the previous chapter), it requires pooling information among different units in the model, both across and within the different stages of the hierarchy. Thus, feedback and lateral connections play an important role in deriving the model's representation. These processes are also involved in cortical visual processing, though their function is not well understood. Further analysis of individual computational steps in model inference might lead to new insights into their functional significance in real neural computation.

Another relevant issue is the meaning we attach to neural activity, as well its relationship to the stimulus. Latent variables in hierarchical models no longer have a direct relationship to the stimulus that characterizes linear models. In models that attempt to find an efficient code for images while preserving as much information as possible (e.g. in ICA or sparse coding), the activity of model units (and by analogy of visual neurons) re-encodes the stimulus, in the process removing the noise but doing no other information selection. In the hierarchical models such as those described above, higher-order variables encode abstract features of the stimuli without retaining all the information about precise pixel values, and specific neural codewords correspond to entire distributions of input data (Fig. 5.6). Thus, these models begin to implement some information selection, though the retained information only conveys the statistical context of the input — whether a data point is an outlier in the canonical distribution, and what kind of distribution best explains it. A similar interpretation applies to other hierarchical generative models (Gaussian mixture models, for example), but the models presented here employ distributed representations that are more consistent with our understanding of coding in the visual cortex.

Encoding distributions of inputs also implies that stimuli that maximally activate a particular model unit extend over large regions of input space, rather than being confined to a localized area. This allows the model to achieve generalization over inputs that come from the same distributions (as recognized by the model). If higher-order visual neurons employ a similar coding scheme, they should be invariant along specific directions in input space. This can be tested by investigating not only the optimal stimuli for driving cells in V2 and V4, but also characterizing dimensions of perturbation around the optimal stimulus along which the neural response remains particularly high, or falls off particularly quickly. Such changes around the "preferred" stimuli have traditionally been treated as modulating the neural response, but in this context they can be considered a central part of the coding properties.

The idea of encoding probability distributions over the input has been explored in other contexts. Within the Bayesian inference framework, it has been employed to deal with noise and uncertainty in the input (Anderson, 1994; Barber et al., 2003; Sahani and Dayan, 2003; Zemel et al., 1998). In these schemes, a neural population is hypothesized to encode, instead of a single estimate of a parameter of interest, an entire distribution function over its values, as a means of representing uncertainty and subsequently performing inference on desired quantities. These studies have described how specific algorithms and implementations of this approach can manipulate probability distributions, combine information encoded by different neurons, read out these distributions, and perform near-optimal inference consistent with behavioral performance when dealing with uncertainty in the stimulus (Zemel et al., 1998; Deneve et al., 2001; Ma et al., 2006;

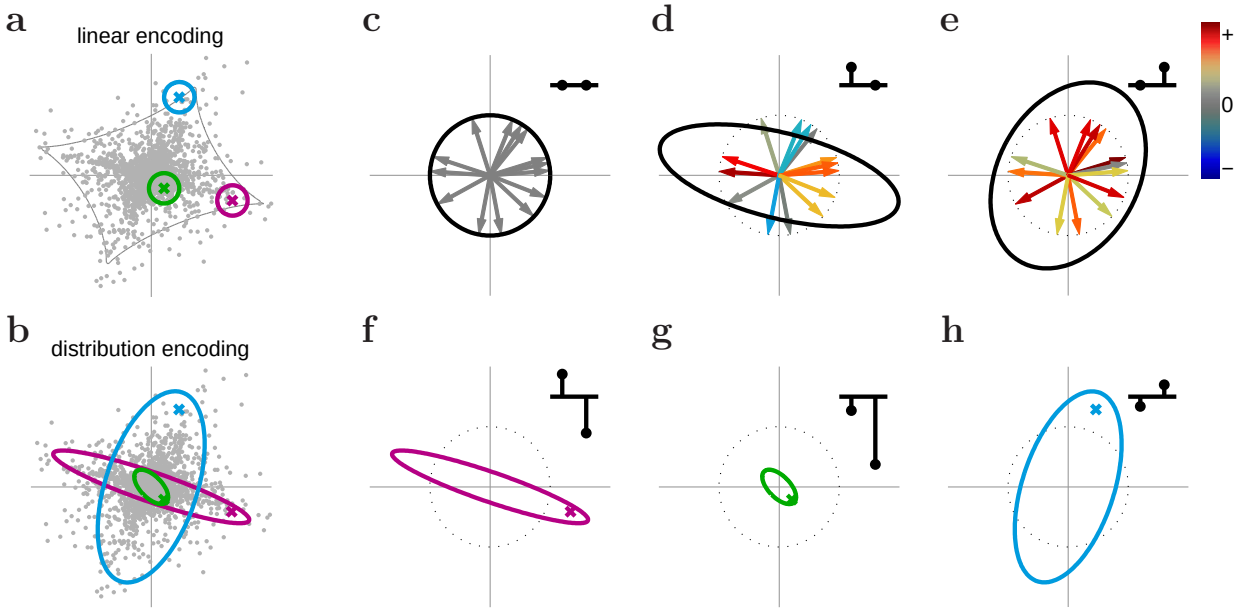


Figure 5.6: Encoding image distributions. The plots show schematics of ways models can represent high dimensional visual input (x and y axes). ***a***. In a linear coding scheme, the activity of model units encodes each local image patch (color x's) by representing its exact location in the input space (color circles). With no explicit model of local distributions, the representation generalizes according to the distance from the best stimulus for that neural code. This is true even if the image features are adapted to optimally represent the ensemble of natural images (ensemble distribution shown by gray points and contour). ***b***. In distribution encoding, the image patches are interpreted as instances from different distributions (color ellipses). The model generalizes because images from the similar distributions have similar representations, even though they may lie in very different parts of image space. ***c-e***. Image distribution encoding by the covariance component model. Linear vectors or features (arrows) represent directions in image space along which images typically exhibit identifiable changes in variance relative to a canonical distribution (indicated by the circle). ***c***. When there is no identifiable higher-order structure model units are inactive (indicated in the inset by the two dots on the zero line), resulting in the canonical distribution. ***d***, ***e***. Two model units encode local image distributions by morphing the canonical distribution to reflect local statistical structure. Increased firing of a unit (insets) indicates that along some directions there is greater or less than expected variation (red and blue arrows, see colorbar). The final effect is to describe specific deviations (black ellipses) from the canonical distribution (dotted circle). ***f-h***. Different graded activation of the population of units (here 2, insets) can describe changing local image distributions (ellipses) that best explain each visual input (x's).

Jazayeri and Movshon, 2006).

Although this work bears relevance to our approach and some of the developed algorithms could prove useful for extending the proposed models, it uses distribution encoding to solve a different problem — robust statistical estimation — and does not address the problems of abstraction and generalization. The goal of inference in the models described in this thesis, on the other hand, is to extract abstract properties of the stimulus and achieve invariance across a set of related stimuli, and the models rely on probability distributions as generalized notions of sets or grouping of data. A potentially useful approach would combine the probabilistic descriptions of uncertainty with the goals of invariance and abstraction as described here.

## 5.3 Neural response prediction

Arguably the most important test of a model's validity is its ability to predict the response of neurons to arbitrary new stimuli. Above, I showed that the responses of model units to sinusoidal gratings exhibit

non-linear effects observed in complex cells, and that the learned representations share complex properties of V2 and V4 neurons. Here I describe preliminary results that test how well the model's representation of image structure can predict responses of V2 neurons to natural images. In order to quantitatively evaluate this relationship, we fitted a regression model to predict the mean firing rate of a V2 neuron based on the model's representation of a set of images, and compared these predictions to other linear and non-linear regression techniques. Using the model to predict neural response allowed us to explain between 10% and 70% of variability in the neural response, yielding significant (if highly variable) improvement over the linear model, and matched the performance of best hand-crafted non-linear regression models of V2 activity (Willmore et al., 2007).

### 5.3.1 Methods

*Stimuli and neural responses.* Neural data were obtained from the laboratory of Jack Gallant, recorded in awake macaque monkeys in area V2. In these experiments, the animal held fixation while small circular patches of grayscale images (sampled randomly from outdoor scenes) were flashed at 60Hz in the receptive field of a V2 neuron, whose response (spike times) was simultaneously recorded. The stimulus sequence consisted of two sets of data: one sequence of 8000 unique image patches, as well as a shorter block of 600 images shown repeatedly (typically about 12 repetitions were presented to each cell). Responses of 12 V2 neurons were analyzed (for 3 cells, multiple trial data were not available). The neural response was binned at 16.5 msec (equal to one frame of the 60Hz stimulus "movie"). Spike counts in each bin of the single trial sequence were used for training the models; for the multiple trial sequences, spike counts were averaged to give the *mean firing rate*, and it was this quantity that we used to evaluate the prediction performance of the regression models. The number of training spikes per neuron ranged from 450 to about 9000 and the number of spikes collected for validation (in the multiple trial data) ranged from 340 to 9000. (See below for an evaluation of possible effects of training sample size on predictions.)

*Computing a regression model.* The regression model approximates the response variable $r$ (here, mean firing rate) by a linear combination of regressors weighted by regression coefficients $\beta_i$,

$$\hat{r} = \sum_i \beta_i x_i + \beta_0 \tag{5.7}$$

The regressors are the image data, represented in their original pixel space, or transformed through a linear (Gabor basis) or non-linear (model-based) representations.

Over-fitting is an enormous challenge when fitting these data — there are thousands of free parameters (as many as 11665 for some of the fitted models), and only 8000 training instances. As expected, evaluating performance on the testing set, without regularization or validation, is absolutely meaningless — the model can be made to predict the test set with almost arbitrary accuracy. We addressed this in several ways. First, all prediction results were calculated on the *testing* set (multiple trial image data) that was not used in any part of the regression procedure. Second, severe regularization was used when estimating regression coefficients on the training set, which prevented over-fitting the large set of parameters.

Many regularized regression methods constrain the set of solutions by imposing penalties on the magnitude of the regression coefficients (Seber, 1977; Tibshirani, 1996). We employed a more severe form of regularization, based on iterative parameter updates, in which the number of non-zero regression coefficients is smaller than in other methods such as the LASSO (Bühlmann and Yu, 2003, 2006; David et al., 2007). This leads to a somewhat improved interpretability of the results; more importantly, it has already been used to model these data with a different non-linear representation of images (Willmore et al., 2007), and we were interested in comparing the performance of the hierarchical models to this analysis.

The sparse regression method, called SparseL$_2$Boost, extends previous iterative methods for sparse regularization (see for example Friedman, 2001; Efron et al., 2004), to achieve a more sparse set of coefficients, but

remains computationally tractable for high-dimensional problems[1]. This is done by selecting, at each step of the fitting procedure, the coefficient that can best improve the estimate, and adjusting only this element of the regression model. This method effectively implements subset selection in a computationally feasible algorithm, and yields a more sparse set of regression coefficients.

The fitting algorithm proceeds as follows. On each iteration $t$, we compute the current prediction and the residual error,

$$\hat{\mathbf{r}}_t = \boldsymbol{\beta}_t^T \mathbf{x} \tag{5.8}$$

$$\mathbf{e}_t = \|\mathbf{r} - \hat{\mathbf{r}}_t\|^2 , \tag{5.9}$$

and find the regression coefficient that can best improve the current estimate, i.e. the dimension in the data $\mathbf{x}$ that is most correlated with the residual,

$$C_i = \text{corrcoef}(e_{i,t}, x_i) \tag{5.10}$$

$$k \leftarrow \arg\max_i |C_i| . \tag{5.11}$$

We then incrementally adjust the model by adding (or subtracting) a small weight $\epsilon$ to this coefficient,

$$\beta_{k,t+1} = \beta_{k,t} + \epsilon(\text{sgn}(C_k)) . \tag{5.12}$$

The incremental updates, in small fixed steps, guarantee improvement in the prediction error. In practice, we do not have to recompute the prediction $\mathbf{r}_t$ at each iteration, but only to incorporate the difference due to the updated coefficient $\beta_k$. This iterative approach also deals with the problem of correlated input data, which can bias regular linear regression estimates. Here, if the input dimensions are correlated, the updates to the regression coefficients can backtrack, but the procedure is nevertheless guaranteed to converge (Friedman, 2001). As mentioned above, this specific algorithm was chosen because it allowed us to benchmark against other models applied to the same data, but it would also be interesting to evaluate other regularization methods that might yield solutions that are more sparse (e.g. subset selection regression) or less sparse (e.g. standard lasso or ridge regression).

If run to completion, this "boosting" approach to regression will also over-fit the training data. To address this, we used early stopping, which makes the boosting procedure converge to $L_1$-norm regularized regression solutions (Zhang and Yu, 2005). This is implemented by computing updates to the regression weights $\boldsymbol{\beta}$ on one part of the training set (*fitting data*) while monitoring performance on a different, held-out part of the training set (*stopping data*). When performance on the stopping data set no longer improved, the fitting procedure was terminated. (We found this point by recording performance on the stopping dataset over a short history of updates, and when the slope of the errors over the interval passed 0, the coefficient updates were stopped.) We split the training data into five equal parts and ran five different fitting procedures, each time holding out one of the parts (as the stopping set) while fitting the coefficients on the other four parts. Fig. 5.7 illustrates the incremental updates to the set of the regression coefficients (boosting) as well the continuous validation steps that prevent over-fitting (early stopping).

The cross-validation trials yield five different estimates for the STRF; we analyzed them individually and also looked at the STRF obtained by averaging the regression coefficient vectors. In practice, the averaged STRF was less sparse but yielded slightly better prediction than the original STRFs (i.e. its correlation to the mean firing rate was slightly higher than the mean of the five individual correlation values).

*Response prediction measure.* Prediction performance was computed as percent of explainable variance accounted for by the regression outputs. Neurons are inherently noisy and their responses to repeated stimuli vary significantly. Our aim was to match the average response, thus capturing the variation that can be attributed solely to the stimulus. Therefore we computed the correlation coefficient between the regression predictions and *the mean firing rate* for each time bin in the sequence of presented images.

[1]I thank Michael Wu for pointing me to this method.

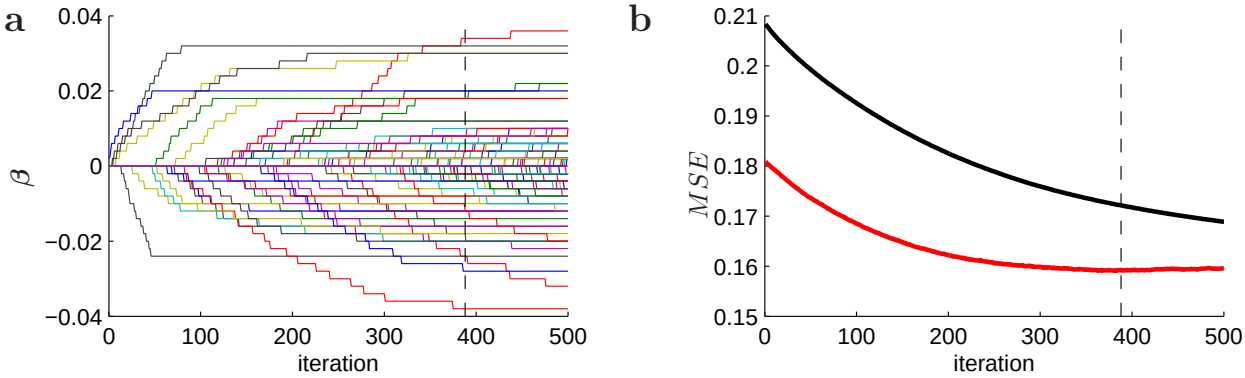


Figure 5.7: An illustration of boosting for regression with early stopping. ***a***. Evolution of regression coefficients during the regression procedure. At each iteration, one of the coefficients is chosen to be adjusted by a discrete step. ***b***. Mean squared error between the prediction and the response for the fitting dataset (black) and the stopping dataset (red). Once performance on the validation no longer improves, the procedure is terminated (dashed line), and the coefficients at that point are retained.

*Model-based regression.* As inputs to the regression fitting procedure, we used linear representation of images (both pixels and outputs of Gabor filters) as well as non-linear representations computed using the covariance component model. The model parameters were fixed after training on a large set of 20×20 image patches (this set was different from the images employed in the physiological experiments and on which predictions of neural activity were computed). In order to obtain the model's representation of the image sequence presented to the neurons, the 8600 image stimuli were cropped to include only the center square of the presented image, resized to 20×20 pixels, and the the MAP estimates $\hat{\mathbf{y}}$ for these images were computed. No temporal information was used during training of the models or during the encoding of the test images (i.e. for each time bin, the vector $\hat{\mathbf{y}}$ was computed independently).

The distribution of each covariance coefficient was first normalized by rescaling to unit variance on the training images. Because the model's responses are symmetric around zero in image space (it is sensitive to the covariance structure, but not the direction of a vector in pixel space), and because many V2 neurons had linear receptive field components, we also included raw pixels or the output of a Gabor basis in the regression analysis.

Because visual neurons respond to the flashed images with variable delay, and their response often integrates information over an extended period of time, for each response bin we collected regression variables (pixels, covariance component coefficients, etc.) from 8 time intervals — the bin when a spike occurred and the 7 preceding time bins. For example, in the linear regression model, the total dimensionality of the regression space was $400 \times 8 = 3200$, plus one bias term. All 3201 input variables were considered when trying to predict each spike. This STRF spanned a 133 msec time window.

The regression coefficients were estimated using the iterative boosting with early stopping procedure, as described above. The fixed increment $\epsilon$ was set to 0.01 of the variance of the regressors. The regression yields a linear mapping from the input space to the response of a V2 neuron. Although the resulting relationship is non-linear in image space, I refer to the fitted models as the neuron's *spatio-temporal receptive fields* (STRFs), following convention in physiological modeling literature (Jones and Palmer, 1987; DeAngelis et al., 1995; Ringach, 2002; David and Gallant, 2005). However, model-based STRFs describe the neuron's sensitivity to image structure through the lens of the model, and its positive and negative components correspond to excitatory and inhibitory aspects of images that might span whole subspaces of image space.

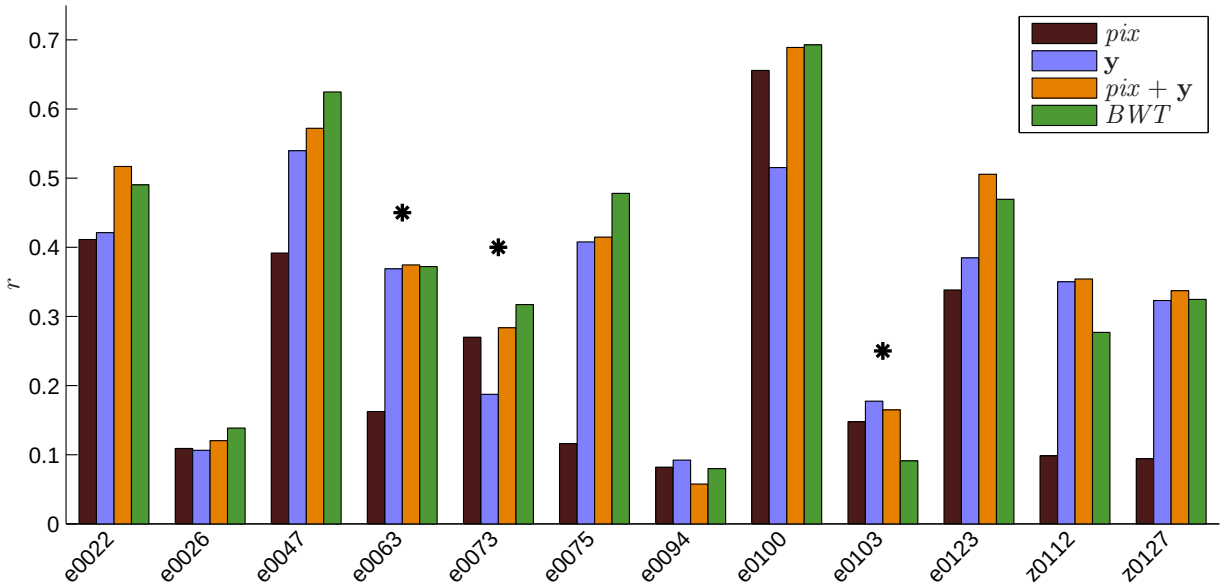


Figure 5.8: Prediction results, measured in percent explainable variance ($r$), for responses of 12 V2 neurons to natural images. For three cells (marked with asterisks) multiple trial data were not available and correlations to single trial binned spike counts are reported. Four regression models are compared: pixel space ($pix$), covariance component coefficients ($\mathbf{y}$), pixels and coefficients ($pix + \mathbf{y}$), and the Berkeley Wavelet Transform ($BWT$). Predictions from the hierarchical model ($pix + \mathbf{y}$) were on par with those of $BWT$, and much better than the linear regression model in pixel space. See text for details.

### 5.3.2 Results: predictive power

Results based on covariance component coefficients were compared to a strictly linear regression model (neural activity regressed against raw pixel values of input images) and a model based on the Berkeley Wavelet Transform (BWT), a phase-separated wavelet transform that has been previously used to analyze the same experimental data. To compute the BWT representation of each image, 27×27 image patches were projected onto a complete orthogonal multi-scale basis that resembles a Haar transform (Willmore et al., 2007), positive and negative coefficients were separated (effectively doubling the representation), and independent regression coefficients were estimated for the positive and the negative coefficients. The best models were able to account for 10% to 70% of explainable variance in activity of V2 neurons (Fig. 5.8). For three cells, repeated trial data were not available, and results on single trial training data set are shown. The estimated STRFs for these cells are computed just as for other cells, but the neural response prediction numbers are not reliable — on one hand, they are artificially high because they are evaluated on data used to fit the models; on the other hand, the correlation coefficients are estimated for single trial spike data (and not the mean firing rate across multiple trials) and this does not account for the inherent variability of the neural response.

For strictly linear models, we found no difference in performance between regression in the space of pixels versus outputs of Gabor filters. Incorporating the non-linear image representations as regressors always improved performance, but the improvement varied considerably from cell to cell. Although all cells were located in V2, at least one cell in our sample was explained relatively well by the linear model ($r = 0.66$) with only a small increase in performance afforded by the non-linear models ($r = 0.69$). This cell behaved much like a V1 simple cell — it had an oriented and localized linear receptive field which alone accounted for most of its response variability. As expected, using only the covariance model coefficients (without the linear component) resulted in a poor prediction for this cell.

Several neurons were not well predicted by any of the models (e.g. e0026, e0094). The causes could be rooted in a variety of experimental or theoretical problems. Fig. 5.9 shows that the number of spikes available for

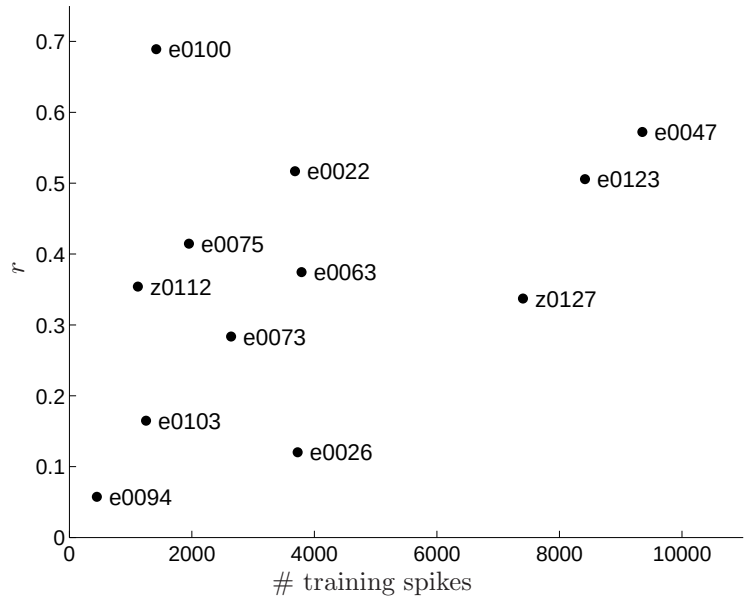


Figure 5.9: Effect of training sample size on prediction quality. The scatter plot shows the prediction performance using model-based regression (orange bars in Fig. 5.8) as a function of the number of spikes used for training.

training the models had a significant effect. All poorly predicted neurons ($r < 0.30$) were not very responsive, with cell e0094 firing only 450 spikes in response to the 8000 flashed images. On the other hand, responses of some other cells were predicted much better even though they also had few training spikes (e.g. e0100, e0022). The fact that the simple-cell-like neuron (e0100) could be explained well using only a few training spikes suggests that the small number of spikes is sufficient when the model matches the neuron well; thus, it is possible that the models we used are simply poor matches to the more complex neurons in our sample. Of course, it is possible that the poorly predicted neurons are less reliable (e.g. their response to the same images is more variable, or their behavior underwent a change from the training sequence to the testing sequence). Another possibility is that these neurons fall into the class of functions represented by the models, but these are not sparse and covary with a large number of model variables, a pattern which cannot be captured when severe regularization is applied to limited data.

For the majority of cells, model-based predictions showed significant improvement over the linear models, and are on par with the BWT results. The best model-based predictions were obtained using a covariance component model trained on 20×20 patches with 1000 linear features $\mathbf{b}_k$ and 200 latent variables $y_j$. For most neurons, it was also necessary to include a linear component (pixels or Gabor function coefficients) in the regression, which suggests that many V2 cells have significant linear response components.

How many variables must be included in the regression for optimal prediction? The regression method favors very sparse solutions, and we found that for most cells, only a small fraction of model coefficients and linear dimensions were included in the regression estimate (Fig. 5.10). Most STRFs had significant non-zero coefficients in 2-3 time frames approximately 50 msec prior to the spike time. It was not the case, however, that neural responses could be predicted from the activity of individual model coefficients. When the final prediction was poor, the model included very few coefficients (light-colored bars in Fig. 5.10), suggesting that the early stopping criteria were effective at preventing over-fitting and the construction of poor over-parameterized models.

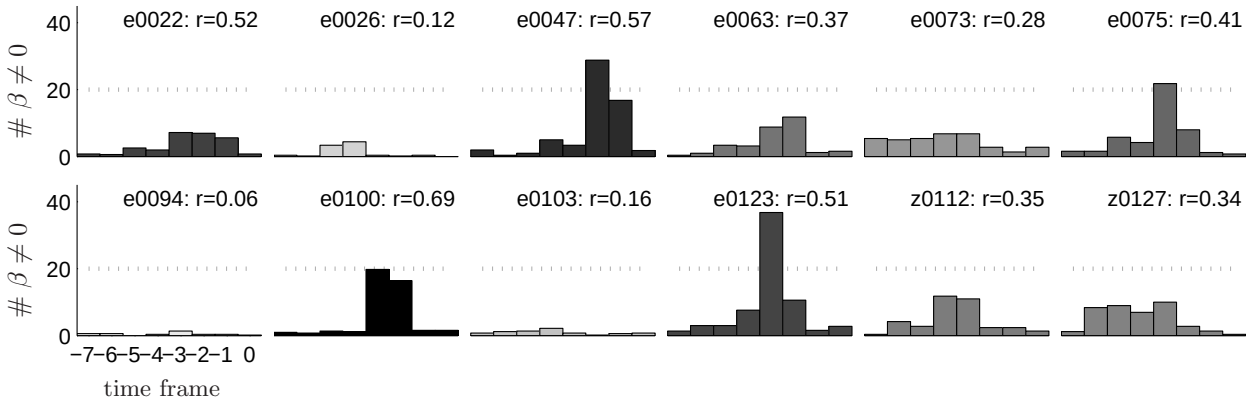


Figure 5.10: The average number of non-zero coefficients (out of 600) in each time slice of the model-based STRF for 12 V2 neurons. Each panel represents a different cell, the abscissa the time frames preceding a spike (which occurs in the last frame, 0), and the ordinate the number of non-zero coefficients used in that time frame. Each point is the average obtained from 5 cross-validation regression fits. Bars shaded to indicate quality of prediction (dark shades mean better performance). Most fits require only a small fraction of coefficients, typically localized to 2-3 time frames.

### 5.3.3 Results: model-based STRFs

The estimated STRF contains a set of coefficients that weight the contribution of each input dimension for neural response prediction. Fig. 5.11 shows, for three neurons with response properties we found interesting, and whose activity could be predicted with some accuracy, time slices of the STRFs and their projections back into image space. Fig. 5.11a and Fig. 5.11c show the raw regression coefficients; these are averages, and hence not as sparse as the STRFs of the individual cross-validation trials). The scales of these two sets of variables cannot be compared; they encode different aspects of the image data, with the first set representing a linear direction in data space, and the second changes in covariance associated with activation of model coefficients. Future work can address this by quantifying the relationship between the linear and non-linear components of the STRF and the relative contribution of different image features to the neural response (see for example Pillow and Simoncelli, 2006); here we analyze these two components separately.

The linear components of the STRF describe the receptive field of each neuron (Fig. 5.11b). Regression in the space of Gabor functions and in the space of pixels produced similar results. We chose to plot the pixel regression coefficients in the figure because these clearly show the effect of the sparseness constraint in the regression procedure and illustrate the pitfalls of assigning meaning to the precise values of regression coefficients. For example, while it is obvious that cell e0047 responds to vertical image structure, most likely it is not sensitive only to the isolated regions of light and dark pixels as suggested by the recovered STRF. The stark spatial receptive field results from the limited availability in the training data and should be construed only as a noisy estimate. Projecting from the space of Gabor functions produces a spatially smooth linear receptive field (constructed out of a small number of Gabor functions instead of a few pixels), but it only imposes a different set of assumptions about its shape. A similar caveat applies to the interpretation of the non-linear component of the STRF; unless a given model predicts the neural response with very high accuracy, its description of image structure represented by the neuron must be interpreted with caution.

To visualize the image structure corresponding to the *non-linear* part of the STRF, we used the regression coefficients to weight a linear combination of covariance components. The regression procedure modeled the activity of each neuron as a linear combination of the covariance coefficients in the hierarchical model; we used this weighting to combine the model units' own description of image structure. For each neuron, in each time bin $t$ the set of regression weights $\beta_{jt}$ combine the model weights

$$w_{kt} = \sum_j \beta_{jt} w_{jk} \tag{5.13}$$

to give the neuron's (non-linear) receptive field in terms of projections onto the image features $\mathbf{b}_k$. We can

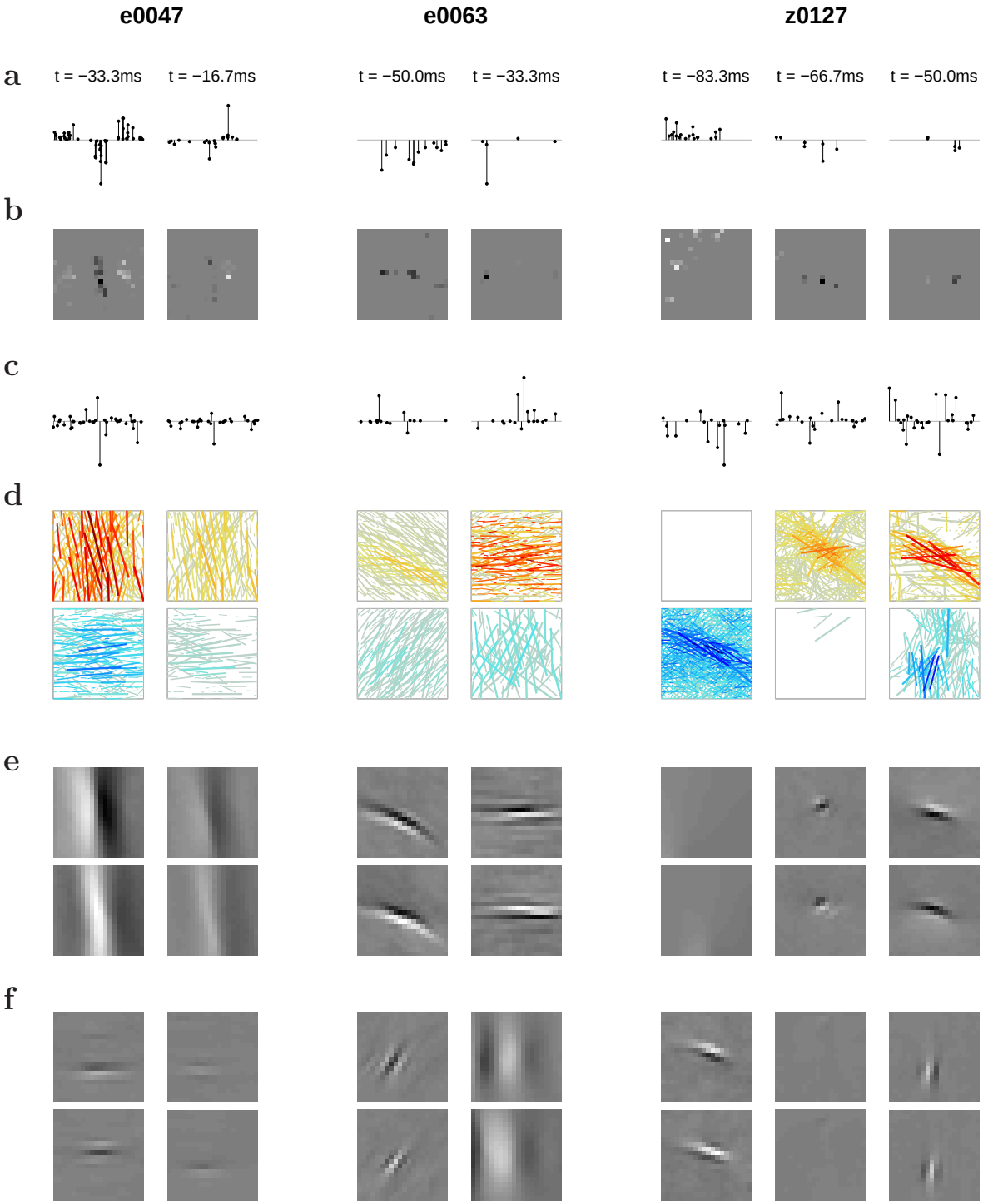


Figure 5.11: Example model-based STRFs for three V2 neurons. Only STRFs for time frames with significant non-zero weights are shown. These STRFs are averages computed over five cross-validation trials. Each column shows one time frame of a neuron's STRF, with the time of stimulus shown at top ($0ms$ = spike time). ***a***. Raw weights ($\beta$'s) for the linear component of the STRF (regressed directly on pixels, ordinate scale stretched to fit axes individually for each neuron). ***b***. The linear weights drawn as image patches. ***c***. Raw regression weights for the covariance model component of the STRF (scale unrelated to that in ***a***). ***d***. STRF-based covariance as line plots that show separately the excitatory features $\mathbf{b}_k$ in red and the inhibitory in blue (again, weights close to zero have been omitted). ***e***. The two most excitatory (second-order) image features for each neuron. ***f***. The two most inhibitory features for each neuron. (See text for more details.)

plot these just as we plotted the weights for each model unit (Fig. 5.11d).

We can also use the regression coefficients to compute the covariance matrix associated with these weights (again a different matrix for each time bin $t$),

$$\mathbf{C}^t = \exp\left(\sum_j \beta_{jt}\mathbf{A}_j\right) . \tag{5.14}$$

This characterizes the image distribution encoded in the vector $\mathbf{y}$ that best correlates with neural activity (Fig. 5.11c) at different delays preceding a spike. There is a linear relationship between the eigen-structure of the data covariance matrix and the vector $\mathbf{y}$: rescaling $\mathbf{y}$ only enhances the non-isotropic shape of the Gaussian distribution, while the negative vector $(-\mathbf{y})$ encodes the converse correlational pattern. Therefore, covariance matrices $\mathbf{C}^t$ describe image distributions that maximally activate the fitted neuron. When $\boldsymbol{\beta} = 0$ these distributions are no different from the global distribution of the data, and our model-derived spike triggered covariance analysis reveals no interesting structure. If, on the other hand, the fitted models include regression weights to covariance components, we can analyze the spectral decomposition of the resulting covariance matrix to reveal most activating and most inhibitory stimulus dimensions. For the three neurons analyzed in Fig. 5.11, we show the two most activating image features (eigenvectors corresponding to the largest eigenvalues, panel e) and the two most suppressive features (smallest eigenvalues, panel f).

As discussed above, regressing covariance component coefficients against neural activity identifies the covariance structure of excitatory stimuli and thus effectively performs STC analysis on the neuron. It is important to note that the resulting covariance matrices are restricted to the space spanned by the model (here, that means 200 free parameters, much smaller than the full 80200-dimensional space), and are regularized to be sparse in model coefficient space. Constraining solutions to be sparse in coefficient space does not mean they will be sparse in image space — some model neurons are sensitive to a large number of data dimensions, and we can see that a relatively sparse vector of regression coefficients in Fig. 5.11c still gives a large set of non-zero weights to image features in Fig. 5.11d, and the number of eigenvectors with significant eigenvalues is also quite large (not shown). (It would be interesting to perform a similar regression and analysis for experiments where the number of image dimensions have been estimated directly in image space using traditional STC and compare the number of relevant image/model subspaces, as well the performance of predictions derived in image space vs. those based on model encoding of image data. This data set, however, includes far too few training samples for this to be feasible.

For the three cells shown in Fig. 5.11, the plotted weights and the covariance matrix eigenvectors reveal somewhat localized, highly oriented image structure. All these cells exhibit cross-orientation suppression — image structure at one orientation excites them while orthogonal features suppress the cell. Cell e0047 has a vertically oriented linear receptive field, is also driven by second-order (correlational) structure of lower spatial frequencies but similar orientation, and is inhibited by horizontal structure containing higher spatial frequencies. Cell z0127 exhibits space-time inseparable properties, with fast excitation by horizontal image structure, fast inhibition by vertical image features, and delayed inhibition at a different orientation (for each cell in Fig. 5.11, left columns show preceding image structure that is slower to affect the neural response). I stress that these results are preliminary, and conclusions about neural properties such as subtle shifts in orientation or frequency preferences must be validated to ensure that regularization in the space of model representations does not produce these effects as artifacts.

Of the nine neurons whose responses could be predicted with some accuracy ($r > 0.30$), all but one were highly tuned for oriented structure, and most of these had STRFs localized to a small part of the image patch. For two cells, the STRFs included oriented regions of excitatory and inhibitory components, but the orientations of the features that contribute to each subfield varied widely. This is consistent with the type of contrast-envelope Gabor representations learned by the model (Fig. 5.2) and with the carrier-orientation insensitive cells revealed by experiments with second-order gratings (Mareschal and Baker, 1998a; Song and Baker, 2007).

Visual inspection of the model-based STRFs suggested that several different types of neurons were present in the population. However, we only analyzed a small sample of cells, and a larger set of neurons is necessary before we can provide a quantitative description of different types of neurons. Analysis of a larger population will also allow us to address other questions regarding the correspondence the model and the data. Some of the relevant variables are the spatial extent of pooling across visual features in individual STRFs, the balance of excitation and inhibition, the prevalence of frequency-tuned and orientation-tuned units (such as those described in section 5.1.4).

### 5.3.4 Validation using synthetic spike data

A number of factors could bear responsibility for the limited success of these methods to predict the responses of V2 neurons. Inherent neural variability or experimental problems can thwart the best models. The regression models are also operating at (or possibly past) the limit of available data, and we wanted to test the ability of the regression method to recover the underlying STRF. To test this we generated synthetic spike data using known STRFs and attempted to recover the true parameters with the same methods applied to neural data. We ran three experiments: in the first, the model-based STRF computed for cell e0047 (Fig. 5.11, left panels) was the underlying function generating spikes; in the second a random but equally sparse STRF was generating by permuting the elements of the e0047 STRF and used to generate spike; in the final experiment, a random non-sparse STRF was constructed by drawing elements from independent standard normal distributions. Note that this STRF is quite different from the sparse regression solutions in the first two experiments. The STRF spans 8 time frames, and while the recovered STRFs for V2 neurons were mostly zero outside one or two time frames, this set of $\beta$s is non-sparse throughout its spatio-temporal extent.

In each case, to produce the synthetic spike data set, the STRF was convolved with the image data represented as a vector of pixel intensities and model coefficients $(\mathbf{x}, \mathbf{y})$, and then thresholded, scaled, and quantized so that the output matched the total number and distribution of spikes for cell e0047. The aim was to test the ability of the regression algorithm to recover STRFs and to examine the effect of regularization on the estimated parameters. Because of response thresholding (number of spikes must not be negative), even exact recovery of the STRF will not give perfect predictions (for these experiments the ceiling was around 0.95). Expansive non-linearities such as the exponential or sigmoidal functions could fix this, but these were not employed in the neural prediction methods, nor were they incorporated here. Also, neural variability might be better captured with a stochastic output model, such as the Linear-Nonlinear-Poisson (Chichilnisky, 2001; Simoncelli et al., 2004), instead of the fixed quantization function. For a preliminary analysis, however, we relied on a fairly simple generating algorithm outlined above.

In experiment 1, the STRF was easily recovered (0.95 correlation between the value of the generating STRF and the average recovered STRF) and 90% of variance in the synthetic spike data was accounted for by the model ($r = 0.95$). Experiment 2 produced similar results, suggesting that a sparse STRF is recoverable even when its elements are random and not structured as they are in the original e0047 STRF. The non-sparse random STRF in experiment 3 produced entirely different results. The responses (i.e. synthetic spikes withheld for validation) were predicted fairly well: 81% variance accounted for, a much better prediction performance than on the real neurons in the previous section. However, the recovered STRF was very unlike the true generating one (0.09 correlation). The estimated STRF was very sparse (on average 87% of the coefficients were zero), and even the non-zero entries were weakly correlated ($r = 0.48$) with the corresponding elements in the true STRF. Although the estimated STRF was quite different from the generating values, the estimates were very robust: the same coefficients were consistently selected on different cross-validation runs (near 80% overlap of non-zero coefficients across pairs of trials) and assigned the same values (98% correlation between STRFs from different cross-validation trials).

The fact that a very different STRF can still predict activity well implies that the input data are redundant — there are multiple ways to construct the decent predictions, though some ways are consistently better as sparse solutions. It is possible that these different STRFs encode similar image structure, even though their

component coefficients are very different. In this experiment, this was not the case, because the original STRF was not at all sparse, and its projection into image space was quite different than the estimated sparse STRFs.

Clearly, regularization biases the STRFs towards sparse solutions. One factor that affects this is the amount of training data; regularization is designed to penalize complexity in the face of limited data, therefore we might expect the estimate to improve as we use more spikes for training. We recomputed the STRF estimates, as above, using 36000 training samples (rather than 8000). This led to an improvement in the recovery of the generating STRF, with correlation between STRF elements increasing from 0.09 to 0.20. This increase is consistent with the convergence rate for such regularized regression estimates, which typically converge as $O(\sqrt{N})$ in data sample size (Zhang and Yu, 2005). It would also be instructive to compare these results to performance of other less constraining (and more standard) regularization schemes, such as ridge regression or the lasso. Although their estimates might be less interpretable, it is possible that they would improve prediction, or simply confirm current findings. Here, we have restricted the analysis to a single estimation method in order to compare model-based prediction to other published techniques (e.g. Willmore et al., 2007).

These results suggest that sparse regularized regression effectively deals with overfitting and finds good response predictors. If the underlying function we are estimating is a sparse linear weighting, it is accurately recovered; otherwise the results is very different because regularization introduces a strong bias. This has two implications. First, even if a neuron is well predicted by the model, the description of the neuron obtained from this analysis can be radically different from the optimal (even if we restrict this description to lie in the space of fitted models). Second, it is possible to obtain very robust estimates of the STRF across different cross-validation trials without recovering the generating STRF (though the estimates approach true values as more data are used).

It is important to note that the experiments with synthetic data generated much better predictions than when the regression analysis was applied to real V2 neurons. One of the causes might be the inherent variability of neurons. Although prediction was evaluated on multiple trial data, which should average out internal neural noise, the number of trials (and number of spikes collected) was limited, and other factors, including experimental conditions, could vary during recording. Another possibility is that neural response is not well described using the model's representation. The models reduce the search space of functions for describing neural processing, but that space might not correspond to the dimensions encoded by the neurons. That means that the model representation does not linearize the relationship between image structure and neural response and cannot be fit with a linear regression model.

Nevertheless, this preliminary analysis demonstrates the potential of using hierarchical models, adapted to the statistics of natural scenes, to constrain the modeling and analysis of neural data. As with other methods, this approach imposes its own set of assumptions about the features of natural images encoded by cortical neurons, but here the possible space of fitted models is constrained by the statistical regularities in the data and is therefore more general than hand constructed models.

## 5.4 Discussion

In this chapter I have shown that the activity of units in the model reproduces a number of properties of cells in V1 and V2, and that characterizations of V4 neurons in the spectral domain are also consistent with a select set of model units. This is significant, since the models developed in this dissertation make no assumptions about the employed representations, nor do they presuppose the ultimate goals of the early visual system (e.g. invariant object recognition).

To the extent that the theoretical predictions match physiological findings, this also has significant implications for mechanisms underlying the observed effects. Non-linear properties that are predicted by the model are derived from the statistics of visual input; this suggests that higher level cognitive signals (e.g.

relating to the perception of shapes or objects) are not necessary to produce these effects. As examples, cross-orientation suppression and the selectivity for oriented regions of contrast-modulation follow directly from image statistics. Such a code is simply a compact way to represent a variety of natural image distributions, it can be derived without supervised "top-down" signals, and it might be implemented in the cortex in local mechanisms that do not rely on extensive feedback of information. These conclusions are not very surprising for the effects analyzed here, but it is likely that a number of other phenomena that are typically linked to higher level processing, such as illusory contour perception and figure-ground identification, can also be explained in terms of their relationship to statistical regularities in natural scenes.

We have established correspondence between model responses and some physiological results, but many other neural properties have not been tested, and might not be predicted by the model. Certainly, further analysis will reveal a large number of neural properties that are not consistent with model predictions, and even some predictions made by the current models are not realistic for cortical representations. For example, both the variance and the covariance component models encode the global contrast of the input image with a single (DC variance) unit. This arises from the statistical structure of the data — variance tends to be correlated across all the pixels in the image. Consequently all the other units in the models are invariant to global contrast levels, since this "dimension" in the variance structure is best represented independently by the DC unit. It is unlikely that the visual system employs this coding scheme; contrast normalization occurs in stages, in individual cells in the early visual system (the retina, the lateral geniculate nucleus) as well as in cortical neurons. This discrepancy could be explained by constraints inherent in the biological system (e.g. it might not be feasible to integrate contrast information across large portions of the visual field) or by other organizing computational principles. We do not expect a single model to explain the large number of poorly understood response properties of cortical neurons. Nevertheless, the models make interesting predictions and suggest new ways to test them by analyzing neural responses to natural images and constructing test stimuli constrained by model parameters.

This work is one of the first attempts to explain a wide range of phenomena on a functional, rather than a mechanistic, level (other hierarchical statistical models, e.g. Hinton et al., 2006, might be leveraged for a similar analysis, but this has not yet been done). Other models for processing in V2 and V4 have been proposed (Riesenhuber and Poggio, 1999; Cadieu et al., 2007), but these are designed to solve specific supervised learning tasks (object recognition). Furthermore, they implement computations assumed *a priori* to be performed by cortical neurons (template matching and the maximum operation) and their parameters are chosen manually for best performance. These models set out to solve a different problem and therefore cannot be compared to the proposed models, whose properties are derived from statistical regularities of the data ensemble.

Another relevant questions is: which cortical area do the proposed models represent? After all, the covariance model takes as input the raw pixel data and forms representations of complex image structure that are compared to V1, V2, and V4 neurons. The answer is not clear, in part because distinctions between neurons in different cortical areas are not well established, and it is also likely that these areas contain heterogeneous populations of cells that convey different aspects of visual information. The model yields a diverse population of unit types, of which a large subset shares characteristics with V1 complex cells (and no units encode linear features as would simple cells), while others seem closer to the less localized and more invariant responses of V2 and V4. Here, too, progress in experimental and theoretical research will have to proceed hand in hand; this work produces several new quantitative methods of neural analysis, and new physiological observations will have to guide the refinement of the model.

A related issue is how to map the computations in the model to neural circuits that implement them. Information traveling through the visual system passes through half a dozen synapses before reaching neurons in extrastriate visual areas. The proposed models, on the other hand, leap from data to response in a single transformation, though that computation is implemented with an iterative procedure. The feed-forward approximations explored in the previous chapter suggested how the inference in the model can be computed by projecting the image onto a small number of linear features and weighting the resulting magnitudes. Incorporating constraints inherent to the biological system should also bring model predictions closer to real data. Breaking down model computation by stages that could correspond to the visual pathway could also

elucidate the functional roles of cells along the processing hierarchy.

# Chapter 6

# Conclusion

## 6.1 Summary of contributions

In this dissertation we have developed a family of flexible parametric hierarchical models that can describe non-trivial statistical patterns in high-dimensional data, while making few assumptions about their underlying structure. The models are general and can be applied to any high-dimensional signal with sufficiently rich statistical structure. We also derived algorithms for maximum likelihood estimation of model parameters, as well as exact iterative and fast approximate inference methods for latent variable estimation.

Trained on natural images, the models account for dependencies observed in linear models. Drawing samples from the models is computationally easy, and produces image patches that look more "natural" than images sampled from PCA or ICA models. We also showed that the models provide better density estimates for image data and therefore can be used for a variety of statistical image restoration tasks. Model representations capture more abstract properties of the image, and are thus more stable over larger regions of the image. These results suggest that, at least to some extent, invariant representations can be derived using only the statistical regularities of the data, without making assumptions about its temporal coherence. Model representations are able to distinguish image regions with perceptually different structure that are not easily separable using linear projections, and they are able to generalize across individual instances in each class. Thus they might be expected to improve performance in image processing tasks such as clustering or segmentation. This direction, however, was not the focus of this thesis and we did not pursue it further.

The models developed here are not specifically tailored for representing images and can be applied to any continuous high-dimensional data. Image analysis is particularly revealing because of our extensive experience with visual data (our primary source of sensory information); we can use our intuitions to gauge the performance of the models and interpret their representations of the data. Many other types of data contain rich hierarchical structure. Obvious examples include sensory signals generated by interactions of objects in the physical world (e.g. sounds), as well as other processes unrelated to biological sensing (e.g. financial data). In many of these applications, variability of the signal carries important information (amplitude in sound, volatility in stock markets) and changes are coordinated across many dimensions, thus we expect the models to be useful in their analysis. Our experiments in section 3.4 revealed structure in speech data - patterns across frequency and time, as well as coordinated changes in both dimensions (e.g. frequency sweeps). The analysis was restricted to very short sound fragments (16 msec) and could not reveal structure at the level of phonemes. As with the analysis of natural images, we expect that increasing the dimensionality would significantly increase the complexity of the learned higher-order patterns. Another important direction is to link these results to representations formed in the primary auditory cortex. This part of the cortex is less well understood than its visual counterpart (V1), so theoretical predictions are especially important. Our initial results suggest that the approach is tractable in data other than natural images, and warrants further

development in the future.

Unlike many other hierarchical models typically used for image analysis, such as Markov Random Fields, mixtures of Gaussians, or wavelet tree cascades, the proposed models use flexible distributed architectures. This makes fewer assumptions about the underlying relationships between elements in images; more importantly, it allows us to directly map model representations to neural populations, which also employ distributed codes. We analyzed model responses to stimuli used in classical physiological experiments and showed that model units exhibit properties of simple and complex cells, such as phase invariance, cross-orientation suppression, and tuned surround suppression. Because model response properties were derived from the statistics of natural scenes, these results provide novel functional interpretation for classical physiological findings. Non-linear behaviors of simple and complex cells, such as cross-orientation inhibition and surround suppression, traditionally viewed either in terms of mechanistic processes or perceptual effects (pop-out, sharpening of orientation tuning for better discrimination), can be viewed as playing a role in optimal coding of image distributions.

This work also illustrated that other cortical effects, previously unexplained, can also be derived from natural scene statistics. Selectivity for second-order patterns — spatial modulation in contrast or orientation — has been reported in V1 and V2 neurons, and model units exhibit these properties as well. The models make specific predictions about the non-linear receptive fields underlying these effects: in the model, the encoded contrast structure is localized, confined to oriented spatial lobes, and invariant to orientation within the spatial subunits. These predictions have not been directly tested and would require stimuli designed specifically around them (these properties cannot be identified with first- or second-order sinusoidal gratings), and the models can be used to design such experiments.

Some units in the model exhibit complex properties observed in V4 neurons that encode curved shapes, spiral image structure, or global image properties such as orientation or spatial frequency. To our knowledge, the proposed models are the first to produce such behaviors. Of course, our understanding of these neurons is far from complete, and the experimental descriptions might not accurately capture the cells' functional roles in processing natural images. Nevertheless, the models make specific predictions about how this type of image structure is represented, which can be directly tested in physiological experiments. For example, when a candidate spiral-tuned neuron is identified (as in Gallant et al., 1993), it can be probed with a battery of images constrained by model parameters corresponding to the similarly selective model unit.

We also demonstrated how model representations of natural images can be used to describe responses of cortical neurons to a set of images, derive model-constrained spatio-temporal receptive fields of these neurons, and predict activity to novel stimuli. We related this approach to regularization of receptive field estimates, a necessary step for fitting complex models with limited experimental data. Estimating neural receptive fields using constraints derived from natural scene statistics is a fairly new approach, and although our experiments did not yield significant improvements over other methods, further work in this direction might prove fruitful.

Another novel aspect of this work is that it provides a different functional view of neural codes in higher cortical areas. As earlier models, the proposed hierarchical models attempt to capture the statistical structure of their input. However, rather than seeking the most efficient representation that retains all the information in the image, these models have a different explicit goal. The latent variables, which we map to activities of cortical neurons, represent entire distributions of images, and are recruited only to indicate a statistically "salient" deviation in the input from the canonical distribution defined by the statistics of the entire image ensemble. Thus, not all the information is preserved in this higher-order code; nor are visual features simply pooled to achieve a specific invariance. Instead, the models implement a form of information selection that is based on abstraction and generalization, rather than behavioral goals or computational constraints. This is a somewhat trivial observation, as any hierarchical model (e.g. k-means clustering) contains variables that form a more abstracted description of image structure, but it is typically not applied to the analysis of neural function.

## 6.2 Future directions

This work naturally suggests a number of future directions to pursue, both in terms of improving models of natural scenes and applying current results to the study of the visual system. The analysis of model representation would be greatly aided by more robust and principled learning algorithms. For example, current methods rely on the MAP approximation to the marginalization over the posterior of the latent variables, which introduces degeneracies (parameters can grow without bound) and provides no guarantees about the optimality of the derived solutions. Alternative techniques, e.g. using Markov chain Monte Carlo methods, can alleviate these problems and also give full distributions over parameters, as well as latent variables, something that can be very useful in interpreting model predictions.

As discussed in section 4.2.2, approximations to model inference using closed-form expressions can be useful for mapping model computations to neural circuits and in applications of hierarchical models in on-line scenarios where speed of computation is essential. We have investigated some approximation schemes, but the results presented here are preliminary and do not address some important issues, such as the theoretical analysis of error in the approximations and fast computation of sparse representations.

One obvious limitation of the models proposed here is that, once the higher-order variables are fixed, they encode Gaussian (or Laplacian) distributions. This means that these higher-order codes only generalize across image distributions described by multi-variate Gaussian or factorial Laplacian densities; and their description of image structure essentially captures only second-order statistics. Texture and edges, which are not well described by their second-order statistics, require more powerful conditional distributions. Sampling from the model reveals these limitations; while the generated images are more heterogeneous, edges are not very well defined (because important phase information is lost in the representation). This also limits the extent of invariance of model representations to physical transformations such as translation and rotation, which alter image data in highly non-linear (but predictable) ways.

A large part of this dissertation is devoted to the comparison of the hierarchical models to processing in the visual cortex. However, our contributions stop short of developing a full program of experimental research to test these predictions. Such a project would require more extensive and quantitative analysis of the properties of individual model units and the full population, and characterization of responses to a variety of synthetic and natural stimuli. Another task is to extend the neural prediction analysis of section 5.3 to a larger set of V2 (and V4) neurons, examine model-based STRFs in more detail, and describe the population properties of cells in these areas. All of these are worthwhile directions (which could also benefit from a greater input from experimentalists), but they are beyond the scope of current work.

It would also be very informative to investigate model encoding of other stimuli used to study V2 and V4, such as polar and hyperbolic patterns (Gallant et al., 1996), curved shapes (Pasupathy and Connor, 2001), and a miscellany of other image sets (Hegdé and Van Essen, 2003). Neural behaviors that have been tied to higher-order perceptual tasks, such as computation of illusory contours, figure-ground segmentation, border ownership, or shape from shading, would be particularly interesting subjects of comparison. There is some evidence that these computations are done locally in V1 or V2 (Pillow and Rubin, 2002), but mechanisms of computation for many tasks are unknown, and there is considerable debate whether they arise purely from bottom-up processes or are aided by feedback signals from areas like IT (Kleffner and Ramachandran, 1992; Lee et al., 2002; Mamassian et al., 2003). To the extent that models developed here exhibit these behaviors, they would support the argument that some of these mechanisms are based on early visual processing and underlying neural behaviors result from specific adaptations to the statistical structure of the input.

# Appendix A

# Details of hierarchical variance modeling

## A.1 Derivation of gradients

Maximum likelihood estimates of model parameters maximize

$$L = \log p(\mathbf{x}|\mathbf{A}, \mathbf{B}) \tag{A.1}$$

$$= \log \int_{\mathbf{v}} \int_{\mathbf{s}} p(\mathbf{x}|\mathbf{A}, \mathbf{s}) p(\mathbf{s}|\mathbf{B}, \mathbf{v}) p(\mathbf{v}) d\mathbf{v} d\mathbf{s} \,. \tag{A.2}$$

The integrals in this function are not tractable, and their computation is replaced by evaluating this expression at the MAP values of the unknown variables.

In the noiseless and complete (square $\mathbf{A}$) case for the first stage of the model, the first term collapses to $\delta(\mathbf{x} - \mathbf{A}\mathbf{s})$, the linear coefficients are computed as $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$, and the linear basis functions are optimized as in ICA (Amari, 1999),

$$\frac{\partial L}{\partial \mathbf{W}} = \left(\mathbf{I} - \phi'(\mathbf{s})\mathbf{s}^T\right) \mathbf{W} \,, \tag{A.3}$$

where $\mathbf{W} = \mathbf{A}^{-1}$. In standard ICA, $\phi'(\mathbf{s})$ contained the first derivatives of the prior distribution $p(\mathbf{s})$, but here the prior is replaced by the conditional distribution $p(\mathbf{s}|\mathbf{B}, \hat{\mathbf{v}})$. The estimate $\hat{\mathbf{v}}$ is the value that maximizes the posterior $p(\mathbf{v}|\mathbf{B}, \mathbf{s})$ given a set of linear coefficients $\mathbf{s}$. In this case we interleave the optimization of $\hat{\mathbf{v}}$ and the linear basis functions $\mathbf{A}$.

In the case of a noisy lower stage, the noise is modeled as i.i.d. Gaussian with variance $\sigma_\epsilon^2$, and the linear coefficients must be inferred and are estimated as in Sparse Coding (Olshausen and Field, 1996), and the sparse prior is again replaced by a function conditional on the higher-order variables $\mathbf{v}$,

$$\frac{\partial L}{\partial s_j} = \frac{1}{\sigma_\epsilon^2} \mathbf{A}_j^T (\mathbf{x} - \mathbf{A}\mathbf{s}) + \phi'(s_j) \tag{A.4}$$

and the gradient for the linear basis functions is similarly computed

$$\frac{\partial L}{\partial \mathbf{A}_j} = \frac{1}{\sigma_\epsilon^2} (\mathbf{x} - \mathbf{A}\mathbf{s}) \mathbf{s}^T \,. \tag{A.5}$$

Once the linear coefficient estimates are obtained (or in parallel with this computation), the estimates for $\hat{\mathbf{v}}$ are computed by the following the gradient

For the most general case, we can use a generalized Gaussian distribution both for the conditional distribution $p(\mathbf{s}|\mathbf{B},\mathbf{v})$ and the prior $p(\mathbf{v})$. This distribution is defined as

$$p(x) = \frac{q}{2\lambda\Gamma(1/q)} \exp\left(-\left|\frac{x}{\lambda}\right|^q\right), \tag{A.6}$$

where $q$ parameterizes the distribution's sparseness and $\lambda$ is a scale variable related to the variance through $\lambda = \sqrt{\sigma^2\Gamma(1/q)/\Gamma(3/q)} = c(q)\sigma$ (Choi et al., 2000). When $q = 2$, this results in the Gaussian distribution, while $q = 1$ yields the Laplacian.

If we model the variance of linear coefficients as $\sigma_i^2 = e^{[\mathbf{Bv}]_i}$, and assume variance of 1 for all $\mathbf{v}$s, the log-likelihood is

$$L \propto \sum_{i=1}^{N}\left(-\log\lambda_i - \left|\frac{s_i}{\lambda_i}\right|^{q_s}\right) - \sum_{j=1}^{M}\frac{|v_j|^{q_v}}{c(q_v)} \tag{A.7}$$

$$\propto \sum_{i=1}^{N}\left(-\frac{[\mathbf{Bv}]_i}{2} - \frac{|s_i|^{q_s}}{c(q_s)e^{q_s[\mathbf{Bv}]_i/2}}\right) - \sum_{j=1}^{M}\frac{|v_j|^{q_v}}{c(q_v)}, \tag{A.8}$$

The derivative w.r.t. latent variable $v_j$ is

$$\frac{\partial L}{\partial v_j} = \frac{1}{2}\sum_i\left(-B_{ij} + q_s B_{ij}\frac{|s_i|^{q_s}}{c(q_s)e^{q_s[\mathbf{Bv}]_i/2}}\right) - q_v \mathrm{sign}(v_j)\frac{|v_j|^{q_v-1}}{c(q_v)}. \tag{A.9}$$

The gradient ascent procedure was sensitive to inital conditions and in some cases did not converge to a solution. We tried several alternatives, including a closed-form approximation to the MAP estimate. Ultimately, the most effective learning method was to adjust the step size $\epsilon$ by the stochastic estimate of the Hessian over each batch of data (LeCun et al., 1998):

$$\eta_j = \frac{\epsilon}{\langle\frac{\partial^2 L}{\partial v_j^2}\rangle + \mu}, \tag{A.10}$$

where $\mu$ is a small constant that improves stability when the second derivative is very small. We used the diagonal approximation to the Hessian (i.e. we considered only the terms $\partial^2 L/\partial v_j^2$. The second derivative for a data sample is given by

$$\frac{\partial^2 L}{\partial v_j^2} = -\sum_{i=1}^{N} q_s^2 B_{ij}^2 \left|\frac{s_i}{ce^{[\mathbf{Bv}]_i}}\right|^{q_s} - q_v(q_v-1)\frac{|v_j|^{q_v-2}}{c^{q_v}}. \tag{A.11}$$

The density component matrix $\mathbf{B}$ was estimated by maximizing the cost function in Eqn. A.8 with values of $\mathbf{s}$ and $\mathbf{v}$ fixed to the MAP estimates. The general form of the gradient is

$$\frac{\partial L}{\partial B_{ij}} = \frac{1}{2}\left(-v_j + v_j q_s \left|\frac{s_i}{c(q_s)e^{[\mathbf{Bv}]_i/2}}\right|^{q_s}\right). \tag{A.12}$$

For a Laplacian conditional distribution ($(c(q_s) = 1/\sqrt{2})$ we get

$$\frac{\partial L}{\partial B_{ij}} = \frac{1}{2}\left(-v_j + v_j\frac{\sqrt{2}|s_i|}{e^{[\mathbf{Bv}]_i/2}}\right) \tag{A.13}$$

and for a Gaussian ($c(q_s) = \sqrt{2}$),

$$\frac{\partial L}{\partial B_{ij}} = \frac{1}{2}\left(-v_j + v_j\frac{s_i^2}{e^{[\mathbf{Bv}]_i}}\right). \tag{A.14}$$

# Appendix B

# Details of covariance modeling

## B.1 Derivation of approximate gradients

*Series expansion.* In order to obtain the MAP estimates $\hat{\mathbf{y}}$, the matrix exponential of $\log \mathbf{C}$ must be computed at each iteration for each data sample. We can use the series expansion

$$\exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{A}^k \tag{B.1}$$

to speed up the computation of individual gradients, replacing the matrix exponential with terms in the expansion up to a limited order.

Another useful relation between a matrix $\mathbf{A}$ and its exponentiated form $\mathbf{C} = \exp(\mathbf{A})$ is based on the relationship between their eigenvalues, $\lambda_i^C = \exp(\lambda_i^A)$. Using the fact that the determinant is the product of the eigenvalues of a matrix and the trace their sum, we obtain

$$\log\det(\mathbf{C}) = \log \prod \lambda_i^C = \sum \log(\lambda_i^C) = \sum \lambda_i^A = \mathrm{Tr}(\mathbf{A})\,. \tag{B.2}$$

The log-likelihood is therefore defined as

$$L = -\frac{1}{2}\log\det(\mathbf{C}) - \frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} \tag{B.3}$$

$$= -\frac{1}{2}\mathrm{Tr}(\mathbf{A}) - \frac{1}{2}\mathbf{x}^T e^{-\mathbf{A}}\mathbf{x} \tag{B.4}$$

$$\hat{L} = -\frac{1}{2}\mathrm{Tr}(\mathbf{A}) - \frac{1}{2}\mathbf{x}^T\left(\mathbf{I} - \mathbf{A} + \frac{1}{2}\mathbf{A}\mathbf{A} - \frac{1}{6}\mathbf{A}\mathbf{A}\mathbf{A} + \dots\right)\mathbf{x} \tag{B.5}$$

$$= -\frac{1}{2}\mathrm{Tr}(\mathbf{A}) - \frac{1}{2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{A}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{A}\mathbf{x} - \frac{1}{6}\mathbf{x}^T\mathbf{A}\mathbf{A}\mathbf{A}\mathbf{x} + \dots\right) \tag{B.6}$$

where $\lambda_{\mathbf{C}}$ are the eigenvalues of $\mathbf{C}$.

Let $\mathbf{Z}$ be a matrix containing the vector product $\mathbf{W}\mathbf{y}$ on the diagonal (i.e. $z_{kk} = \sum_j w_{jk} y_j \; z_{kl} = 0, \forall k \neq l$), and let the matrix $\mathbf{B}$ collect all the feature vectors $\mathbf{b}_k$. Then

$$\mathbf{A} = \sum_j y_j \mathbf{A}_j = \sum_j y_j \left(\sum_k w_{jk}\mathbf{b}_k\mathbf{b}_k^T\right) = \sum_k \left(\sum_j y_j w_{jk}\right)\mathbf{b}_k\mathbf{b}_k^T = \mathbf{B}\mathbf{Z}\mathbf{B}^T \tag{B.7}$$

and thus

$$-\frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} = -\frac{1}{2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \frac{1}{6}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} + \ldots\right) . \quad \text{(B.8)}$$

In this expression, $\mathbf{Z}$ is the only variable that varies with $\mathbf{y}$ and $w_{jk}$. This form also lends itself to a fast and loop-free computation of the likelihood for a block of input data (see below).

*Inference gradient.* The gradient for estimating MAP values $\hat{\mathbf{y}}$ is derived from the series expasion above.

The following will be useful,

$$\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}\left(\sum_j y_j\mathbf{A}_j\right) = \sum y_j\mathrm{Tr}(\mathbf{A}_j) \quad \text{(B.9)}$$

$$\mathrm{Tr}(\mathbf{A}_j) = \mathrm{Tr}\left(\sum_j w_{jk}\mathbf{b}_k\mathbf{b}_k^T\right) = \sum_j w_{jk}\mathrm{Tr}\left(\mathbf{b}_k\mathbf{b}_k^T\right) = \sum_j w_{jk} . \quad \text{(B.10)}$$

The last step only holds when $\|\mathbf{b}_k\| = 1$ and thus the trace of $\mathbf{b}_k\mathbf{b}_k^T = 1$. When we take the derivative of the trace w.r.t. the vectors $\mathbf{b}_k$, we do not make this assumption because an unconstrained gradient step could produce $\|\mathbf{b}_k\| \neq 1$.

Using the result above and the expansion of $\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x}$,

$$\frac{\partial\hat{L}}{\partial y_j} = -\frac{1}{2}\frac{\partial}{\partial y_j}\mathrm{Tr}(\mathbf{A}) - \frac{1}{2}\frac{\partial}{\partial y_j}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} \quad \text{(B.11)}$$

$$= -\frac{1}{2}\frac{\partial}{\partial y_j}\mathrm{Tr}\left(\sum y_j\mathbf{A}_j\right) + \frac{\partial\mathbf{Z}}{\partial y_j}\frac{\partial}{\partial\mathbf{Z}}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x} \quad \text{(B.12)}$$

$$= -\frac{1}{2}\sum_k w_{jk} + \frac{1}{2}\mathbf{w}_j^T\frac{\partial\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x}}{\partial\mathbf{Z}} \quad \text{(B.13)}$$

$$= -\frac{1}{2}\mathbf{w}_j^T\left(1 - \frac{\partial}{\partial\mathbf{Z}}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} + \frac{1}{2}\frac{\partial}{\partial\mathbf{Z}}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \ldots\right) \quad \text{(B.14)}$$

$$= -\frac{1}{2}\mathbf{w}_j^T\left(1 - \left(\mathbf{B}^T\mathbf{x}\right)^2 + \frac{1}{2}\left(\mathbf{B}^T\mathbf{x}\right)\odot\left(\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x}\right) + \frac{1}{2}\left(\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x}\right)\odot\left(\mathbf{B}^T\mathbf{x}\right) - \ldots\right) \quad \text{(B.15)}$$

$$= \frac{1}{2}\mathbf{w}_j^T\left(-1 + \left(\mathbf{B}^T\mathbf{x}\right)^2 - \left(\mathbf{B}^T\mathbf{x}\right)\odot\left(\mathbf{B}^T\mathbf{A}\mathbf{x}\right) + \frac{1}{3}\left(\mathbf{B}^T\mathbf{x}\right)\odot\left(\mathbf{B}^T\mathbf{A}\mathbf{A}\mathbf{x}\right) + \frac{1}{6}\left(\mathbf{B}^T\mathbf{A}\mathbf{x}\right)^2 - \ldots\right) , \quad \text{(B.16)}$$

where $\mathbf{w}_j$ is the column vector $(w_{j1}, w_{j2}, \ldots)^T$. Note that we omitted the data index $n$; several quantities actually change with each data point, and should in fact be written with it: $\mathbf{x}_n$, $\mathbf{y}_n$, $(\mathbf{C})_n$, and $(\mathbf{A})_n$.

We can speed up the implementation of this gradient (and other expressions, such as the likelihood) in MATLAB by avoiding the computation of the matrix $(\mathbf{A})_n$ for each data sample, since all we really need are the matrix-vector products $\mathbf{B}^T(\mathbf{A})_n\mathbf{x}_n$, $\mathbf{B}^T(\mathbf{A})_n(\mathbf{A})_n\mathbf{x}_n$, and so on. For example,

$$\mathbf{B}^T(\mathbf{A})_n\mathbf{x}_n = \mathbf{B}^T\mathbf{B}\left[\mathbf{Z}_n\mathbf{B}^T\mathbf{x}_n\right] = \mathbf{B}^T\mathbf{B}\left[(\mathbf{W}\mathbf{y}_n)\odot\left(\mathbf{B}^T\mathbf{x}_n\right)\right] . \quad \text{(B.17)}$$

Here the data index $n$ indicates which quantities vary sample to sample. The last term uses element-wise multiplication between two vectors, which can be performed for a whole block of data (and these vectors can also be computed block-wise). This allows us to avoid loops over the data samples, which can be quite large.

*Parameter estimation.* We use the same series expansion to derive the gradients for parameter updates,

$$\frac{\partial \hat{L}}{\partial \mathbf{b}_k} = -\frac{1}{2}\frac{\partial}{\partial \mathbf{b}_k}\left[\sum_j \mathrm{Tr}\left(y_j \sum_k w_{jk}\mathbf{b}_k\mathbf{b}_k^T\right) + \left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \dots\right)\right] \tag{B.18}$$

$$= -\frac{1}{2}\sum_j y_j w_{jk}\frac{\partial}{\partial \mathbf{b}_k}\mathrm{Tr}\left(\mathbf{b}_k\mathbf{b}_k^T\right) - \frac{1}{2}\frac{\partial}{\partial \mathbf{b}_k}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \dots\right) \tag{B.19}$$

$$= -\Big(\sum_j y_j w_{jk}\Big)\mathbf{b}_k + \left(\frac{1}{2}\frac{\partial}{\partial \mathbf{b}_k}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} + \frac{1}{2}\frac{\partial}{\partial \mathbf{b}_k}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \dots\right) \tag{B.20}$$

$$= -\Big(\sum_j y_j w_{jk}\Big)\mathbf{b}_k + \left(\Big(\sum_j y_j w_{jk}\Big)\mathbf{x}\mathbf{x}^T\mathbf{b}_k + \frac{1}{2}\Big(\sum_j y_j w_{jk}\Big)(\mathbf{x}\mathbf{x}^T\mathbf{A}\mathbf{b}_k + \mathbf{A}\mathbf{x}\mathbf{x}^T\mathbf{b}_k) - \dots\right) \tag{B.21}$$

$$= \Big(\sum_j w_{jk}y_j\Big)\left(-\mathbf{I} + \mathbf{x}\mathbf{x}^T - \frac{1}{2}\left(\mathbf{x}\mathbf{x}^T\mathbf{A} + \mathbf{A}\mathbf{x}\mathbf{x}^T\right) + \frac{1}{6}\left(\mathbf{x}\mathbf{x}^T\mathbf{A}\mathbf{A} + \mathbf{A}\mathbf{A}\mathbf{x}\mathbf{x}^T + \mathbf{A}\mathbf{x}\mathbf{x}^T\mathbf{A}\right) + \dots\right)\mathbf{b}_k \tag{B.22}$$

For the gradient w.r.t. $w_{jk}$, we first note that for the matrix $\mathbf{Z}$ as defined above (a diagonal matrix with entries $z_{kk} = sum_j y_j w_{jk}$),

$$\frac{\partial}{\partial w_{jk}}\mathbf{a}^T\mathbf{Z}\mathbf{b} = \frac{\partial}{\partial w_{jk}}\sum_k a_k\Big(\sum_j w_{jk}y_j\Big)b_k \tag{B.23}$$

$$= y_j a_k b_k\,. \tag{B.24}$$

The gradient is given as

$$\frac{\partial \hat{L}}{\partial w_{jk}} = -\frac{1}{2}\frac{\partial}{\partial w_{jk}}\left[\sum_j \mathrm{Tr}\Big(y_j \sum_k w_{jk}\mathbf{b}_k\mathbf{b}_k^T\Big) + \left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \dots\right)\right] \tag{B.25}$$

$$= -\frac{1}{2}\sum_j y_j \sum_k \frac{\partial}{\partial w_{jk}}w_{jk}\mathrm{Tr}\Big(\mathbf{b}_k\mathbf{b}_k^T\Big) - \frac{1}{2}\frac{\partial}{\partial w_{jk}}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{B}\mathbf{Z}\mathbf{B}^T\mathbf{x} - \dots\right) \tag{B.26}$$

$$= -\frac{1}{2}y_j - \frac{1}{2}\left(-y_j[\mathbf{B}^T\mathbf{x}]_k[\mathbf{B}^T\mathbf{x}]_k + y_j\frac{1}{2}[\mathbf{B}^T\mathbf{x}]_k[\mathbf{B}^T\mathbf{Z}\mathbf{B}\mathbf{B}^T\mathbf{x}]_k + y_j\frac{1}{2}[\mathbf{B}^T\mathbf{Z}\mathbf{B}\mathbf{B}^T\mathbf{x}]_k[\mathbf{B}^T\mathbf{x}]_k - \dots\right) \tag{B.27}$$

$$= \frac{1}{2}\left(-1 + \left(\mathbf{b}_k^T\mathbf{x}\right)^2 - \left(\mathbf{b}_k^T\mathbf{x}\right)\left(\mathbf{b}_k^T\mathbf{A}\mathbf{x}\right) + \frac{1}{3}\left(\mathbf{b}_k^T\mathbf{x}\right)\left(\mathbf{b}_k^T\mathbf{A}\mathbf{A}\mathbf{x}\right) + \frac{1}{6}\left(\mathbf{b}_k^T\mathbf{A}\mathbf{x}\right)^2 - \dots\right)y_j \tag{B.28}$$

The same reformulation trick (Eqn. B.17) can be applied to speed up the MATLAB implementation.

# Bibliography


Abdallah, S. A. and Plumbley, M. D. (2001). If the independent components of natural images are edges, what are the independent components of natural sounds? In *Proc. 3rd Intl. Conf. on Independent Component Analysis and Signal Separation, ICA2001*, pages 534–539, San Diego.

Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *J Opt Soc Am A*, 2(2):284–299.

Albrecht, D., Geisler, W. S., and Crane, A. (2004). Primary visual cortex: Linear and nonlinear properties. In Chalupa, L. and Werner, J., editors, *The Visual Neurosciences, Vol. 1*, pages 747–764. MIT Press, Cambridge.

Albright, T. D. (1992). Form-cue invariant motion processing in primate visual cortex. *Science*, 255(5048):1141–1143.

Amari, S. (1999). Natural gradient learning for over- and under-complete bases In ICA. *Neural Computation*, 11(8):1875–1883.

Anderson, C. (1994). Basic elements of biological computational systems. *International Journal of Modern Physics C*, 5(2):135–137.

Andrews, D. F. and Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society B*, 36(1):99–102.

Asai, M., McAleer, M., and Yu, J. (2006). Multivariate stochastic volatility: A review. *Econometric Reviews*, 25(2-3):145–175.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychol Rev*, 61(3):183–193.

Baddeley, R. (1996). Visual perception. an efficient code in V1? *Nature*, 381(6583):560–561.

Baker, C. L. J. and Mareschal, I. (2001). Processing of second-order stimuli in the visual cortex. *Prog Brain Res*, 134:171–191.

Barber, M. J., Clark, J. W., and Anderson, C. H. (2003). Neural representation of probabilistic information. *Neural Comput.*, 15(8):1843–1864.

Barlow, H. B. (1961). Possible principles underlying the transformation of sensory messages. In Rosenbluth, W. A., editor, *Sensory Communication*, pages 217–234. MIT Press, Cambridge.

Barnard, J., McCulloch, R., and Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, 10(4):1281–1311.

Bell, A. J. and Sejnowski, T. J. (1995). An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Bell, A. J. and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Res*, 37(23):3327–3338.

Boik, R. J. (2002). Spectral models for covariance matrices. *Biometrika*, 89(1):159–182.

Bonds, A. B. (1989). Role of inhibition in the specification of orientation selectivity of cells in the cat striate cortex. *Vis Neurosci*, 2(1):41–55.

Bonin, V., Mante, V., and Carandini, M. (2005). The suppressive field of neurons in lateral geniculate nucleus. *J Neurosci*, 25(47):10844–10856.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. (2000). Adaptive rescaling maximizes information transmission. *Neuron*, 26(3):695–702.

Buccigrossi, R. W. and Simoncelli, E. P. (1999). Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12):1688–1701.

Bühlmann, P. and Yu, B. (2003). Boosting with the L2 loss: Regression and classification. *J. Amer. Statist. Assoc.*, 98:324–340.

Bühlmann, P. and Yu, B. (2006). Sparse boosting. *Journal of Machine Learning Research*, 7:1001–1024.

Cadieu, C., Kouh, M., Pasupathy, A., Connor, C., Riesenhuber, M., and Poggio, T. (2007). A Model of V4 shape selectivity and invariance. *J Neurophysiol*, In the press.

Carandini, M. (2004). Receptive fields and suppressive fields in the early visual system. In *The Cognitive Neurosciences*. MIT press.

Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., Gallant, J. L., and Rust, N. C. (2005). Do we know what the early visual system does? *J Neurosci*, 25(46):10577–10597.

Carandini, M., Heeger, D. J., and Movshon, J. A. (1997). Linearity and normalization in simple cells of the macaque primary visual cortex. *J Neurosci*, 17(21):8621–8644.

Cavanaugh, J. R., Bair, W., and Movshon, J. A. (2002). Nature and interaction of signals from the receptive field center and surround in macaque V1 neurons. *J Neurophysiol*, 88(5):2530–2546.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159.

Chichilnisky, E. J. (2001). A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12(2):199–213.

Chiu, T. Y. M., Leonard, T., and Tsui, K.-W. (1996). The matrix-logarithmic covariance model. *Journal fo the American Statistical Association*, 91(433):198–210.

Choi, S., Cichocki, A., and Amari, S.-I. (2000). Flexible independent component analysis. *Journal of VLSI Signal Processing*, 26(1-2):25–38.

Chubb, C., Olzak, L., and Derrington, A. (2001). Second-order processes in vision: introduction. *J. Opt. Soc. Am. A*, 18(9):2175–2178.

Cohen, L. (1989). Time–frequency distributions – A review. *Proc. IEEE*, 77:941–981.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314.

Dan, Y., Atick, J. J., and Reid, R. C. (1996). Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *J Neurosci*, 16(10):3351–3362.

Daniels, M. J. and Kass, R. E. (1999). Nonconjugate Bayesian estimation of covariance matrices in hierarchical models. *Journal of the American Statistical Association*, 94:1254–1263.

Daugman, J. G. (1989). Entropy reduction and decorrelation in visual coding by oriented neural receptive-fields. *IEEE Trans. Bio. Eng.*, 36(1):107–114.

David, S. V. and Gallant, J. L. (2005). Predicting neuronal responses during natural vision. *Network: Computation in Neural Systems*, 16(2-3):239–260.

David, S. V., Hayden, B. Y., and Gallant, J. L. (2006). Spectral receptive field properties explain shape selectivity in area V4. *J Neurophysiology*, 96(6):3492–505.

David, S. V., Mesgarani, N., and Shamma, S. A. (2007). Estimating sparse spectro-temporal receptive fields with natural stimuli. *Network: Computation in Neural Systems*, 18(3):191–212.

Davies, M. and Mitianoudis, N. (2004). A sparse mixture model for overcomplete ICA. *IEEE Proceedings on Vision Image and Signal Processing*, 151(1):35–43.

de Boer, E. and Kuyper, P. (1968). Triggered correlation. *IEEE Transactions on Bio-Medical Engineering*, 15:169–179.

de Ruyter van Steveninck, R. and Bialek, W. (1988). Real-time performance of a movement sensitive neuron in the blowfly visual system: Coding and information transfer in short spike sequences. *Proc R Soc London Ser B*, 234:379–414.

De Valois, R. L., Albrecht, D. G., and Thorell, L. G. (1982). Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559.

DeAngelis, G. C., Ohzawa, I., and Freeman, R. D. (1995). Receptive-field dynamics in the central visual pathways. *Trends in Neuroscience*, 18(10).

Dempster, A. (1972). Covariance selection. *Biometrics*, 28:157–175.

Deneve, S., Latham, P. E., and Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nat Neurosci*, 4(8):826–831.

Doi, E., Inui, T., Lee, T.-W., Wachtler, T., and Sejnowski, T. J. (2003). Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation*, 15:397–417.

Dong, D. W. and Atick, J. J. (1995). Statistics of natural time-varying images. *Network: Computation in Neural Systems*, 6(3):345–358.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression (with discussion). *Annals of Statistics*, 32(2):407–409.

Evans, I. G. (1965). Bayesian estimation of parameters of a multivariate normal distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, 27(2):279–283.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex*, 1(1):1–47.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am A*, 4(12):2379–2394.

Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4):559–601.

Foldiak, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Gallant, J. L., Braun, J., and Van Essen, D. C. (1993). Selectivity for polar, hyperbolic, and Cartesian gratings in macaque visual cortex. *Science*, 259:100–103.

Gallant, J. L., Connor, C. E., Rakshit, S., Lewis, J. W., and Van Essen, D. C. (1996). Neural responses to polar, hyperbolic, and Cartesian gratings in area V4 of the macaque monkey. *J Neurophysiol*, 76(4):2718–2739.

Girolami, M. (2001). A variational method for learning sparse and overcomplete representations. *Neural Computation*, 13:2517–2532.

Grosof, D. H., Shapley, R. M., and Hawken, M. J. (1993). Macaque V1 neurons can signal 'illusory' contours. *Nature*, 365(6446):550–552.

Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. *Vis Neurosci*, 9(2):181–197.

Heeger, D. J., Simoncelli, E. P., and Movshon, J. A. (1996). Computational models of cortical visual processing. *Proc Natl Acad Sci U S A*, 93(2):623–627.

Hegdé, J. and Van Essen, D. C. (2000). Selectivity for complex shapes in primate visual area V2. *J Neurosci*, 20(5):RC61:1–6.

Hegdé, J. and Van Essen, D. C. (2003). Strategies of shape representation in macaque visual area V2. *Vis Neurosci*, 20(3):313–328.

Hegdé, J. and Van Essen, D. C. (2007). A comparative study of shape representation in macaque visual areas V2 and V4. *Cereb Cortex*, 17(5):1100–1116.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput*, 18(7):1527–1554.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 160:106–154.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *J Physiol*, 195(1):215–243.

Hurri, J. and Hyvärinen, A. (2003). Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3):663–691.

Hyvärinen, A. (1999). Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, 11:1739–1768.

Hyvärinen, A. and Hoyer, P. (2000). Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12:1705–1720.

Hyvärinen, A. and Hoyer, P. O. (2001). A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41:2413–2423.

Hyvärinen, A., Hoyer, P. O., and Inki, M. (2001). Topographic independent component analysis. *Neural Computation*, 13:1527–1558.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430.

Inki, M. and Hyvärinen, A. (2001). Two methods for estimating overcomplete independent component bases. In *International Workshop on Independent Component Analysis and Blind Signal Separation*.

Ito, M. and Komatsu, H. (2004). Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. *J Neurosci*, 24(13):3313–3324.

Jazayeri, M. and Movshon, J. A. (2006). Optimal representation of sensory information by neural populations. *Nat Neurosci*, 9(5):690–696.

Jones, H. E., Wang, W., and Sillito, A. M. (2002). Spatial organization and magnitude of orientation contrast interactions in primate V1. *J Neurophysiol*, 88(5):2796–2808.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258.

Karklin, Y. and Lewicki, M. (2003). Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14:483–499.

Karklin, Y. and Lewicki, M. (2005). A hierarchical Bayesian model for learning non-linear statistical regularities in non-stationary natural signals. *Neural Computation*, 17:397–423.

Karklin, Y. and Lewicki, M. S. (2006). Is early vision optimized for extracting higher-order dependencies? In *Advances in Neural Information Processing Systems 18*. MIT Press.

Kayser, C., Einhäuser, W., Dümmer, O., König, P., and Körding, K. (2001). Extracting slow subspaces from natural videos leads to complex cells. *Artificial Neural Networks*, 2130:1075–1080.

Kleffner, D. A. and Ramachandran, V. S. (1992). On the perception of shape from shading. *Percept Psychophys*, 52(1):18–36.

Knierim, J. J. and van Essen, D. C. (1992). Neuronal responses to static texture patterns in area V1 of the alert macaque monkey. *J Neurophysiol*, 67(4):961–980.

Lamme, V. A. (1995). The neurophysiology of figure-ground segregation in primary visual cortex. *J Neurosci*, 15(2):1605–1615.

Landy, M. S. and Oruc, I. (2002). Properties of second-order spatial frequency channels. *Vision Res*, 42(19):2311–2329.

LeCun, Y., Bottou, L., Orr, G., and Muller, K. (1998). Efficient backprop. In Orr, G. and K., M., editors, *Neural Networks: Tricks of the trade*. Springer.

Lee, T. S., Mumford, D., Romero, R., and Lamme, V. A. (1998). The role of the primary visual cortex in higher level vision. *Vision Res*, 38(15-16):2429–2454.

Lee, T. S., Yang, C. F., Romero, R. D., and Mumford, D. (2002). Neural activity in early visual cortex reflects behavioral experience and higher-order perceptual saliency. *Nat Neurosci*, 5(6):589–597.

Lee, T.-W. and Lewicki, M. S. (2000). The generalized Gaussian mixture model using ICA. In *International Workshop on Independent Component Analysis*, pages 239–244.

Lennie, P. (1998). Single units and visual cortical organization. *Perception*, 27(8):889–935.

Leonard, T. and Hsu, J. S. J. (1992). Bayesian inference for a covariance marix. *The Annals of Statisics*, 20(4):1669–1696.

Lesica, N. A. and Stanley, G. B. (2004). Encoding of natural scene movies by tonic and burst spikes in the lateral geniculate nucleus. *J Neurosci*, 24(47):10731–10740.

Leventhal, A. G., Wang, Y., Schmolesky, M. T., and Zhou, Y. (1998). Neural correlates of boundary perception. *Vis Neurosci*, 15(6):1107–1118.

Levitt, J. B., Kiper, D. C., and Movshon, J. A. (1994). Receptive fields and functional architecture of macaque V2. *J Neurophysiol*, 71(6):2517–2542.

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363.

Lewicki, M. S. and Olshausen, B. A. (1999). A probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America A*, 16(7):1587–1601.

Lewicki, M. S. and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Computation*, 12(2):337–365.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11):1432–1438.

Mamassian, P., Jentzsch, I., Bacon, B. A., and Schweinberger, S. R. (2003). Neural correlates of shape from shading. *Neuroreport*, 14(7):971–975.

Mareschal, I. and Baker, C. L. J. (1998a). A cortical locus for the processing of contrast-defined contours. *Nat Neurosci*, 1(2):150–154.

Mareschal, I. and Baker, C. L. J. (1998b). Temporal and spatial response to second-order stimuli in cat area 18. *J Neurophysiol*, 80(6):2811–2823.

Maunsell, J. H. and Newsome, W. T. (1987). Visual processing in monkey extrastriate cortex. *Annu Rev Neurosci*, 10:363–401.

McGraw, P. V., Levi, D. M., and Whitaker, D. (1999). Spatial characteristics of the second-order visual pathway revealed by positional adaptation. *Nat Neurosci*, 2(5):479–484. Clinical Trial.

Mechler, F. and Ringach, D. L. (2002). On the classification of simple and complex cells. *Vision Res*, 42(8):1017–1033.

Morrone, M. C., Burr, D. C., and Maffei, L. (1982). Functional implications of cross-orientation inhibition of cortical visual cells. I. Neurophysiological evidence. *Proc R Soc Lond B Biol Sci*, 216(1204):335–354.

Movshon, J. A., Thompson, I. D., and Tolhurst, D. J. (1978). Receptive field organization of complex cells in the cat's striate cortex. *J Physiol*, 283:79–99.

Najfeld, I. and Havel, T. F. (1995). Derivatives of the matrix exponential and their computation. *Adv. Appl. Math.*, 16(3):321–375.

Nishimoto, S., Ishida, T., and Ohzawa, I. (2006). Receptive field properties of neurons in the early visual cortex revealed by local spectral reverse correlation. *J Neurosci*, 26(12):3269–3280.

Nothdurft, H. C., Gallant, J. L., and Van Essen, D. C. (2000). Response profiles to texture border patterns in area V1. *Vis Neurosci*, 17(3):421–436.

O'Keefe, L. P. and Movshon, J. A. (1998). Processing of first- and second-order motion signals by neurons in area MT of the macaque monkey. *Vis Neurosci*, 15(2):305–317.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23).

Olshausen, B. A. and Field, D. J. (2005). How close are we to understanding V1? *Neural Comput*, 17(8):1665–1699. Comparative Study.

O'Neill, J. C. (1999). DiscreteTFDs time-frequency analysis software. `http://tfd.sourceforge.net/`.

Paninski, L. (2003). Convergence properties of some spike-triggered analysis techniques. *Network: Computation in Neural Systems*, 14:437–464.

Park, H.-J. and Lee, T.-W. (2004). A hierarchical ICA method for unsupervised learning of nonlinear dependencies in natural images. In *International Conference on Independent Component Analysis and Blind Signal Separation*.

Pasupathy, A. and Connor, C. E. (2001). Shape representation in area V4: position-specific tuning for boundary conformation. *J Neurophysiol*, 86(5):2505–2519.

Petersen, K. B. and Pedersen, M. S. (2007). The matrix cookbook. Version 20070905.

Pillow, J. and Rubin, N. (2002). Perceptual completion across the vertical meridian and the role of early visual cortex. *Neuron*, 33(5):805–813.

Pillow, J. W. and Simoncelli, E. P. (2006). Dimensionality reduction in neural models: An information-theoretic generalization of spike-triggered average and covariance analysis. *Journal of Vision*, 6(4):414–428.

Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. (2001). Adaptive Wiener denoising using a Gaussian scale mixture model in the wavelet domain. In *Proc 8th IEEE Int'l Conf on Image Proc*, volume II, pages 37–40, Thessaloniki, Greece. IEEE Computer Society.

Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. (2003). Image denoising using Gaussian scale mixtures in the wavelet domain. *IEEE Transactions on Image Processing*, 12:1338–1351.

Pourahmadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: uncontrasined parameterisation. *Biometrika*, 86(3):677–690.

Pourahmadi, M. (2000). Maximum likelihood estimation of generalized linear models for multivariate normal covariance matrix. *Biometrika*, 87:425–435.

Pourahmadi, M. (2004). Simultaneous modelling of covariance matrices: GLM, Bayesian and nonparametric perspectives. *submitted*.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine learning*.

Reinagel, P. and Reid, R. C. (2000). Temporal coding of visual information in the thalamus. *J Neurosci*, 20(14):5392–5400.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025.

Ringach, D. L. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J Neurophysiol*, 88(1):455–463.

Romberg, J., Choi, H., and Baraniuk, R. (2001). Bayesian tree-structured image modeling using wavelet domain Hidden Markov models. *IEEE Transactions on Image Processing*, 10(7).

Rossi, A. F., Desimone, R., and Ungerleider, L. G. (2001). Contextual modulation in primary visual cortex of macaques. *J Neurosci*, 21(5):1698–1709.

Ruderman, D. R. and Bialek, W. (1994). Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–818.

Rust, N. C. and Movshon, J. A. (2005). In praise of artifice. *Nat Neurosci*, 8(12):1647–1650.

Rust, N. C., Schwartz, O., Movshon, J. A., and Simoncelli, E. P. (2005). Spatiotemporal elements of macaque V1 receptive fields. *Neuron*, 46(6):945–956.

Sahani, M. and Dayan, P. (2003). Doubly distributional population codes: simultaneous representation of uncertainty and multiplicity. *Neural Comput*, 15(10):2255–2279.

Schwartz, O., Pillow, J. W., Rust, N. C., and Simoncelli, E. P. (2006). Spike-triggered neural characterization. *J Vision*, 6(4):484–507.

Schwartz, O. and Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4:819–825.

Seber, G. (1977). *Linear Regression Analysis*. J. Wiley.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois.

Sillito, A. M. (1975). The contribution of inhibitory mechanisms to the receptive field properties of neurones in the striate cortex of the cat. *J Physiol*, 250(2):305–329.

Simoncelli, E. and Olshausen, B. (2001). Natural image statistics and neural representation. *Ann. Rev. Neurosci.*, 24:1193–1216.

Simoncelli, E. P. (1997). Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA.

Simoncelli, E. P., Pillow, J., Paninski, L., and Schwartz, O. (2004). Characterization of neural responses with stochastic stimuli. In Gazzaniga, M., editor, *The Cognitive Neurosciences, III*, pages 327–338. MIT Press.

Smith, M. and Kohn, R. (2002). Parsimonious covariance matric estimation for longitudinal data. *Journal of American Statistical Association*, 97:1141–1153.

Smyth, D., Willmore, B., Baker, G. E., Thompson, I. D., and Tolhurst, D. J. (2003). The receptive-field organization of simple cells in primary visual cortex of ferrets under natural scene stimulation. *The Journal of Neuroscience*, 23(11):4746–4759.

Song, Y. and Baker, C. L. J. (2007). Neuronal response to texture- and contrast-defined boundaries in early visual cortex. *Vis Neurosci*, 24(1):65–77.

Sukumar, S. and Waugh, S. J. (2007). Separate first- and second-order processing is supported by spatial summation estimates at the fovea and eccentrically. *Vision Res*, 47(5):581–596.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B*, 58:267–288.

Tolhurst, D. J., Tadmor, Y., and Chao, T. (1992). Amplitude spectra of natural images. *Ophthalmic Physiol Opt*, 12(2):229–232.

Touryan, J., Felsen, G., and Dan, Y. (2005). Spatial structure of complex cell receptive fields measured with natural images. *Neuron*, 45(5):781–791. Comparative Study.

Valpola, H., Harva, M., and Karhunen, J. (2004). Hierarchical models of variance sources. *Signal Processing*, 84(2):267–282.

van Hateren, J. H. (1997). Processing of natural time series of intensities by the visual system of the blowfly. *Vision Res*, 37(23):3407–3416.

van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society, London B*, 265:359–366.

von der Heydt, R. and Peterhans, E. (1989). Mechanisms of contour perception in monkey visual cortex. I. Lines of pattern discontinuity. *J Neurosci*, 9(5):1731–1748.

von der Heydt, R., Peterhans, E., and Baumgartner, G. (1984). Illusory contours and cortical neuron responses. *Science*, 224(4654):1260–1262.

von der Heydt, R., Peterhans, E., and Dürsteler, M. R. (1992). Periodic-pattern-selective cells in monkey visual cortex. *J Neurosci*, 12(4):1416–1434.

Wainwright, M. J., Simoncelli, E. P., and Willsky, A. S. (2001). Random cascades on wavelet trees and their use in analyzing and modeling natural images. *Applied Computational and Harmonic Analysis*, 11:89–123.

Wax, M., Shan, T.-J., and Kailath, T. (1984). Spatio-temporal spectral analysis by eigenstructure methods. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 32.

Willmore, B., Prenger, R. J., and Gallant, J. L. (2007). Neural representation of natural images in visual area V2. *submitted*.

Wiskott, L. and Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770.

Wong, F., Carter, C. K., and Kohn, R. (2003). Efficient estimation of covariance selection models. *Biometrika*, 90(4):809–830.

Yang, R. and Berger, J. O. (1994). Estimation of a covariance matrix using the reference prior. *Annals of Statitics*, 22:1195–1211.

Yen, S.-C., Baker, J., and Gray, C. M. (2007). Heterogeneity in the responses of adjacent neurons to natural stimuli in cat striate cortex. *J Neurophysiol*, 97(2):1326–1341.

Zemel, R. S., Dayan, P., and Pouget, A. (1998). Probabilistic interpretation of population codes. *Neural Comput*, 10(2):403–430.

Zetzsche, C. (1990). Sparse coding: the link between low level vision and associative memory. In Eckmiller, R., Hartmann, G., and Hauske, G., editors, *Parallel Processing in Neural Systems and Computers*, pages 273–276, Amsterdam (North-Holland). Elsevier Science.

Zetzsche, C. and Röhrbein, F. (2001). Nonlinear and extra-classical receptive field properties and the statistics of natural scenes. *Network: Computation in Neural Systems*, 12(3):331–350.

Zhang, T. and Yu, B. (2005). Boosting with early stopping: convergence and consistency. *Annals of Statistics*, 33:1538–1579.

Zhou, H., Friedman, H. S., and von der Heydt, R. (2000). Coding of border ownership in monkey visual cortex. *J Neurosci*, 20(17):6594–6611.

Zhou, Y. X. and Baker, C. L. J. (1994). Envelope-responsive neurons in areas 17 and 18 of cat. *J Neurophysiol*, 72(5):2134–2150.

Zhou, Y. X. and Baker, C. L. J. (1996). Spatial properties of envelope-responsive cells in area 17 and 18 neurons of the cat. *J Neurophysiol*, 75(3):1038–1050.