Continuous basis pursuit and its applications

Chaitanya Ekanadham

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

New York University

September 2012

Eero P. Simoncelli

Daniel Tranchina

To my family, mentors, and teachers

Acknowledgments

There are many people to whom I owe the opportunity to write this thesis: Eero Simoncelli and Daniel Tranchina for their guidance, support, and advice throughout the years; Eero for his generosity and for including me in his lab; all the current and former members of the Lab for Computational Vision who provided a relaxed, fun, and supportive environment in which to be a graduate student; my fellow classmates at Courant, for all their support especially in the early stages of the program; Yan Karklin, who collaborated with me on the hierarchical spike coding work; Liam Paninski and Sinan Gunturk, for always willing to be sounding boards; Jose Acosta and Tamar Arnon, for making sure I stayed on track and did not miss any deadlines; Rob Young, for always assisting with technical issues.

I'd also like to thank several mentors and teachers who I've had the fortune of learning from in the past: Peter Knopf, Andrew Ng , Nils Nilsson, Susan Holmes, Gautam Iyer, and Daphne Koller. Their enthusiasm and passion for science, mathematics, and engineering, are the primary reason I chose to pursue a graduate career.

Finally, I thank my parents, sister, and fiancè, for their unwavering support, selflessness, and love: I could not have done this without them.

Abstract

Transformation-invariance is a major source of nonlinear structure in many real signal ensembles. To capture this structure, we develop a methodology for decomposing a signal into a sparse linear combination of *continuously transformed* features. The central idea is to approximate the manifold(s) of transformed features(s) by linearly combining interpolation functions using constrained coefficients that can be recovered via convex programming. The advantage of this approach over traditional sparse coding methods is threefold: (1) it is built upon a more accurate probabilistic source model for transformation-invariant ensembles, (2) it uses a more efficient dictionary, and (3) both structural and transformational information can be extracted separately from the representation via well-defined mappings. The method can be used with any linear interpolator, and includes basis pursuit denoising as a special case corresponding to nearest-neighbor interpolation. We propose a novel polar interpolation method with which our method significantly outperforms basis pursuit on a sparse deconvolution task. In addition, our method outperforms the state-of-the-art in identifying neural action potentials from voltage recordings on multiple simulated and real data sets. The advantage of our method is primarily due to its supe-

rior handling of near-synchronous action potentials, which overlap in the trace and are not recoverable by standard spike sorting methods. Finally, we develop a hierarchical formulation in which successive layers encode more complex features and their associated transformation parameters. A two-layer time- and frequency-shiftable representation is learned from speech data. The second layer encoding compactly represents sounds in terms of acoustic features such as harmonic stacks, sweeps, and ramps in time-frequency space. Despite its compactness, synthesis reveals that it is a faithful representation of the original sound and yields significant improvement over wavelet thresholding techniques on an acoustic denoising task. These two applications demonstrate the advantage of representations which separate content and transformation, and our proposed methodology provides an effective tool for computing such a representation.

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

Many signal ensembles encountered in the real world are high-dimensional, but exhibit low-dimensional structure. For example, natural images (represented by pixel values) and sounds (represented by sound pressure samples) are extremely high-dimensional objects, but exhibit striking regularities that set them apart from "randomly" generated signals of the same dimensionality. Similarly, signals recorded by medical imaging, radar, sonar, and seismological devices have very few underlying degrees of freedom compared to the dimensionality with which they are typically represented upon acquisition. Representing signals with respect to their "intrinsic dimensions" in a compact and interpretable form is essential for many applications in signal processing such as compression, denoising, detection, classification, and source separation.

Transformations are a major source of low-dimensional, nonlinear structure. For example, simply translating a sampled univariate signal can trace out a highly complicated and nonlinear manifold in the sample space [130], as illustrated in Figure 1.1. Many real signal ensembles

inherit this nonlinear structure since they are invariant to several transformations. Image ensembles are often invariant to spatial translation, dilation, and rotation. Sound ensembles are often invariant to translation in the temporal and frequency domains. Many tasks require invariance to these transformations (e.g., object recognition, speaker identification), while others require precise measurement of transformation amounts (e.g., pose estimation, visuomotor planning). Complex tasks, such as facial recognition, can require invariance (e.g., pose invariance) and precision (e.g., spatial relationship of eyes, nose, mouth) with respect to different signal properties. To accommodate this dual requirement, it is advantageous for a representation to factor signals into "what" and "where" information specifying the structural and transformational content of a signal, respectively [133, 141, 56, 7, 14]. This factorization is computationally challenging because of the nonlinearity introduced by transformations.

There is a wide body of literature focusing on the extraction of "what" information. The vast majority of models assume (either explicitly or implicitly) that the coefficients of signals with respect to a particular linear basis, or "dictionary," are independently and identically distributed. For example, wavelet coefficients of images have often been modeled by a factorial, "sparse" (or heavy-tailed) distribution, for which one can posit an analytic form [131, 86, 68]. Such a distribution has also been imposed on coefficients with respect to a *learned* dictionary that is optimized for a particular signal ensemble [23, 25, 101, 5, 65, 69, 135, 2, 40, 136, 75, 84]. Several of these employ "over-complete" representations, whose dimensionality is larger

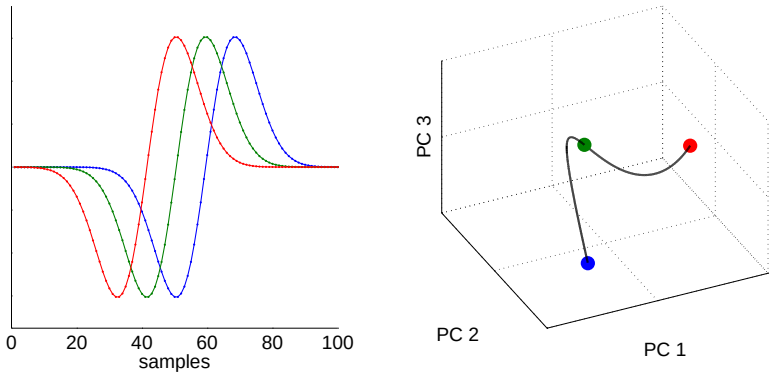


Figure 1.1: Simple translation traces out highly nonlinear manifolds in the signal space. **Left:** 3 translated versions of the waveform $f(x) = xe^{-\frac{x^2}{2}}$, each represented by a 100-dimensional vector of samples. **Right:** Manifold traced out by continuously translating the waveform between the red and blue curves on the left, plotted with respect to the 3 leading principal components computed over 1000 intermediate translates (accounting for 99.97% of the variance). Colored dots represent the corresponding time-shifts on the left.

than that of the signal space. Over-complete representations can capture complex dependencies [82], and their use has been made computationally tractable by recent theoretical [35, 143, 17, 36, 43] and algorithmic [27, 24, 62, 39, 47, 8, 42, 19, 30, 28, 34] advances. Over-complete representations have been employed successfully for compression, denoising, and source separation [41, 29, 45, 108].

Despite their success, these methods fail to separate "where" information from "what" information. Instead, they construct (or learn) dictionaries that implicitly reflect the transformation(s) present in the

signal ensemble. For example, dictionaries used for image processing, whether hand-constructed [12, 1, 132, 85, 86, 133, 46, 137, 109] or learned [101, 5, 120, 57, 67], are often composed of prototypical kernels replicated at discretely sampled spatial scales, positions, and orientations. Acoustic representations often use a dictionary of bandpass filters replicated at discretely sampled center times and center frequencies [103, 69, 55, 135, 136, 57, 77]. There are three major disadvantages of employing such "transformation-invariant" dictionaries in conjunction with a sparse factorial model on their coefficients. First, the sparse factorial assumption is systematically violated as a result of discretizing the transformations: when the amount of transformation in the signal lies in between the grid-points corresponding to the dictionary elements, the coefficients exhibit non-sparse and dependent activity. Second, precise "where" information cannot be cleanly extracted from the coefficients due to blocking artifacts and discontinuities [133, 22]. In other words, the coefficients vary non-intuitively and discontinuously as transformations are applied. Third, these dictionaries can be highly inefficient, since one needs to finely sample the transformation, and are often ill-suited for sparse recovery methods that are used in practice.

A separate line of work has focused on modeling the effect of transformations. Several of these efforts attempt to identify signal properties that are invariant to natural transformations, thus discarding "where" information altogether [48, 151, 150]. Others have tried to explicitly model transformations from before-and-after pairs of transformed signals [114, 130, 94, 92, 93], but lack a probabilistic model of "what" information.

Our work is close in spirit to the work of [141, 56, 7, 14], which employ models that represent "what" and "where" information using separate sets of variables. However, these efforts focus on adapting the dictionary to a particular ensemble, and lack efficient and reliable computational methods for inferring "what" /"where" representations. Developing such a method is the primary focus of Chapter 2.

Thesis organization In Chapter 2 we motivate sparse representations, review their usage in the literature, and describe in detail the currently available computational methods for inferring them. We then introduce *continuous basis pursuit* (CBP) as an inference method that combines the modeling advantages of sparse representations with the ability to explicitly model known transformations, and present empirical evidence of its advantage. In Chapter 3, we apply our methodology to the problem of neural action potential identification ("spike sorting"). We review existing solutions, present our approach using CBP, and then compare its performance with existing standards on both simulated and real extracellular voltage recordings. In Chapter 4, we develop an approach for hierarchically processing "what" /"where" representations. We review previous efforts to hierarchically model structure in sounds. We then fit a two-layer model to speech data, and analyze its properties. In Chapter 5 we summarize our contributions and conclude.

Chapter 2

Continuous basis pursuit

2.1 Sparse representations

2.1.1 Motivation and formulation

Sparse representations assume that the observed signal $\vec{x}$ is a noisy linear combination of a few elements from a dictionary of "atomic" features:

$$\vec{x} = \mathbf{D}\vec{\alpha} + \vec{\epsilon} \tag{2.1}$$

The dictionary $\mathbf{D}$ is a matrix whose columns are the individual features, and $\vec{\alpha}$ is a "sparse" vector of coefficients, of which only a few are non-zero. The second term $\vec{\epsilon}$ represents additive noise. The coefficient vector $\vec{\alpha}$ is often referred to as a "sparse code" of the signal $\vec{x}$.

This sparse coding model has arisen in many contexts. In classical regression, for instance, sparsity serves as a criteria for "model selection" when it is known that only a small subset of the predictor variables actually influence the observed variable [60]. In image processing, it is well known that wavelet coefficients of images have sparse distributions

[86, 131, 41]. This property provides an obvious advantage when applied to image compression [12], since only the indices and values of nonzero wavelet coefficients need to be transmitted. The sparsity property of wavelet coefficients has also been used for image denoising, since noise is often non-sparse in the wavelet domain and can thus be more easily distinguished from the clean image. By combining a Gaussian noise probability model $P_{\text{noise}}(\vec{\epsilon})$ (with variance σ^2) with a sparse probability model for the coefficients $P_{\text{prior}}(\vec{\alpha})$, one can use Bayes rule to estimate the *most probable* wavelet coefficients given the noisy image (also known as a *maximum-a-posteriori*, or MAP estimator):

$$\begin{aligned}
\vec{\alpha}_{\text{MAP}}(\vec{x}) &= \arg\max_{\vec{\alpha}} P(\vec{\alpha}|\vec{x};\mathbf{D}) \\
&= \arg\max_{\vec{\alpha}} P_{\text{noise}}(\vec{x}-\mathbf{D}\vec{\alpha})P_{\text{prior}}(\vec{\alpha}) \\
&= \arg\min_{\vec{\alpha}} -\log P_{\text{noise}}(\vec{x}-\mathbf{D}\vec{\alpha}) - \log P_{\text{prior}}(\vec{\alpha}) \\
&= \arg\min_{\vec{\alpha}} \frac{1}{2\sigma^2}\|\vec{x}-\mathbf{D}\vec{\alpha}\|_2^2 - \log P_{\text{prior}}(\vec{\alpha}) \qquad (2.2)
\end{aligned}$$

Once the wavelet coefficients are estimated in this way, the denoised image is obtained simply by applying the wavelet transform:

$$\vec{x}_{\text{denoised}} = \mathbf{D}\vec{\alpha}_{\text{MAP}}(\vec{x}) \qquad (2.3)$$

Orthogonal dictionaries If the wavelet dictionary is orthogonal, then we have $\|\vec{x}-\mathbf{D}\vec{\alpha}\|_2^2 = \|\vec{z}-\vec{\alpha}\|_2^2$ where $\vec{z} = \mathbf{D}^T\vec{x}$. If in addition the prior probability model is factorial (i.e. $P_{\text{prior}}(\vec{\alpha}) = \prod_i P_{\text{prior}}(\alpha_i)$), then the optimization of Eq. 2.2 decouples and can be solved by separately

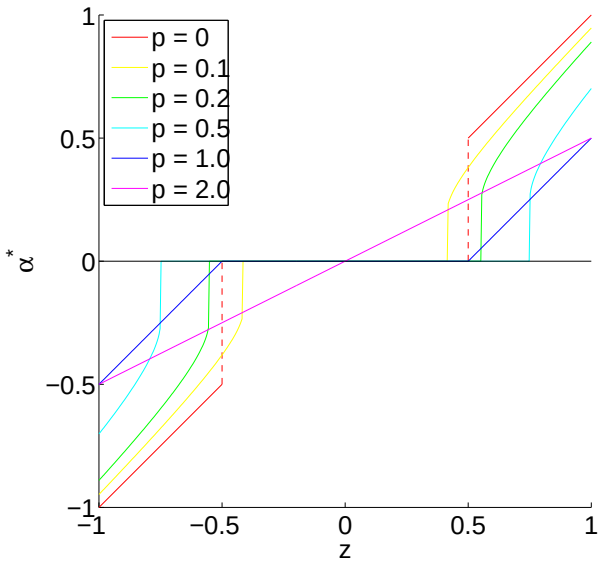


Figure 2.1: "Coring" operators which yield MAP estimators for coefficients, assuming a prior probability model $P(\vec{\alpha}) \propto e^{-\frac{1}{2}\sum_i |\alpha_i|^p}$ for six values of p.

optimizing each coefficient:

$$\alpha_i = \arg\min_{\alpha} \frac{1}{2\sigma^2}(z_i - \alpha)^2 - \log P_{\text{prior}}(\alpha) \qquad \forall i \tag{2.4}$$

For sparsity-promoting prior probability models, the solution of Eq. 2.4 is the result of a so-called "shrinkage" or "coring" operation on the wavelet coefficient $z_i = (\mathbf{D}^T\vec{x})_i$ of the noisy image. Figure 2.1 plots this operation in the case that the prior probability model is a generalized Gaussian with different exponents. This shrinkage approach serves as the basis for many successful image and acoustic denoising algorithms [86, 109, 41].

Redundant (over-complete) dictionaries Redundant or "over-complete" dictionaries have also been used for image and sound compression and denoising. In this case, the dictionary transform is non-invertible, so there are infinitely many coefficient vectors that will reconstruct the signal up to a given accuracy. The MAP optimization of Eq. 2.2 uses the sparse prior as a criteria for selecting among these. Over-complete dictionaries have several advantages over traditional orthogonal dictionaries. First, they yield sparser representations, since they can model a wider range of structure in the signal ensemble. For example, sounds have been decomposed using the union of a Fourier-basis (modeling harmonic sounds) and a Dirac-basis (modeling onsets and attacks) [45]. Second, they are more stable to noise and deformations: orthogonal representations change erratically as translation or noise is applied due to blocking artifacts caused by the diadic spacing of the wavelets [46, 22, 133, 82]. Third, over-complete dictionaries can linearize complex dependencies in the signal ensemble, potentially capturing nonlinear structure that complete dictionaries cannot [82]. Sparse, over-complete representations using both hand-chosen (e.g., tight wavelet frames) and learned dictionaries have proven successful in a variety of signal processing applications such as compression, denoising/enhancement, inpainting, and source separation (see [41] and [108] for a review of image and acoustic applications, respectively).

Learned dictionaries Dictionaries adapted to natural signal ensembles (with a sparsity prior on the coefficients), whether complete or over complete, have exhibited many desirable properties. Independent com-

ponent analysis (ICA) learns a complete dictionary, assuming a non-Gaussian factorial prior on the coefficients. This technique has been often used for blind source separation of mixed audio signals [25]. When adapted to natural image patches, both the ICA and an over-complete dictionary learn localized, oriented, gabor-like filters akin to the receptive fields of simple cells in primary visual cortex [101, 5]. Analogously, dictionaries adapted to natural sounds yield time- and frequency-localized gammatone-like filters resembling the receptive fields of cochlear cells [136]. In addition to their relationship to neural response properties, these representations have also proven useful in various coding and denoising applications [80, 81, 152, 136].

Inference with non-orthogonal dictionaries The non-orthogonality of over-complete dictionaries prevents the optimization of Eq. 2.2 from decoupling, and so more sophisticated methods must be employed to jointly solve for the coefficients. The common prior distribution that is (implicitly or explicitly) used is $P_{\text{prior}}(\vec{\alpha}) \propto e^{-\lambda\|\vec{\alpha}\|_0}$ where $\|\cdot\|_0$ is the L_0 "pseudonorm" which counts the number of non-zero elements. The resulting MAP optimization becomes:

$$\vec{\alpha}_{\text{MAP}}(\vec{x}) = \arg\min_{\vec{\alpha}} \frac{1}{2\sigma^2}\|\vec{x} - \mathbf{D}\vec{\alpha}\|_2^2 + \lambda\|\vec{\alpha}\|_0 \tag{2.5}$$

If one has a bound on the noise energy $M = \|\epsilon\|_2^2$, then the dual formulation is often used:

$$\vec{\alpha}_{\text{MAP}}(\vec{x}) = \arg\min_{\vec{\alpha}:\|\vec{x}-\mathbf{D}\vec{\alpha}\|_2^2 \leq M} \|\vec{\alpha}\|_0 \tag{2.6}$$

Unfortunately, solving either Eq. 2.5 or Eq. 2.6 exactly is NP-Hard [96, 31] since one must search all possible subsets of nonzero coefficients of $\vec{\alpha}$.

Techniques to approximate the solution can be split into greedy methods and convex relaxation methods, discussed in Sections 2.1.2 and 2.1.3, respectively.

2.1.2 Greedy methods

Greedy methods attempt to solve Eq. 2.5 by successively adding nonzero coefficients to $\vec{\alpha}$ until the objective Eq. 2.5 can no longer decrease. The first such methods date back to variable selection methods used in classical statistics [50]. More recent greedy methods are exemplified by the well-known matching pursuit algorithm [88], summarized in Algorithm 1. The basic procedure is to select, at each iteration, the dictionary element that is most correlated with the residual portion of the signal that is currently unexplained. If it is possible to decrease the objective in Eq. 2.5 by changing this element's coefficient, then the coefficient is updated and the procedure is repeated until no further decrease is possible (or, in the dual formulation, until the residual squared norm goes below M). The algorithm can be viewed as a generalization of the well-known matched-filtering procedure for identifying a template within a signal [99].

There are several variations of matching pursuit that have been shown to enjoy better convergence properties, usually at the cost of computational complexity. For example in *orthogonal matching pursuit* (OMP) [102], all non-zero coefficients are re-optimized (without sparsity penalization) every time the support of $\vec{\alpha}$ is augmented. The optimal coefficients on a restricted support $\mathcal{I}$ can be computed in closed form using

Algorithm 1 Matching pursuit algorithm $MP(\mathbf{D}, \vec{x}, \lambda))$

$\vec{\alpha} \leftarrow \vec{0}$ {initialize coefficient vector}

$df \leftarrow \infty$ {initialize change in residual error}

while $df > \lambda$ **do**

$i^* \leftarrow \arg\max_i(\mathbf{D}^T(\vec{x} - \mathbf{D}\vec{\alpha}))_i$ {find best coefficient}

$df \leftarrow \frac{(\mathbf{D}^T(\vec{x}-\mathbf{D}\vec{\alpha}))_{i^*}^2}{2\sigma^2(\mathbf{D}^T\mathbf{D})_{i^*i^*}}$ {compute decrease in residual error}

if $df > \lambda$ **then**

$\alpha_{i^*} \leftarrow \frac{(\mathbf{D}^T(\vec{x}-\mathbf{D}\vec{\alpha}))_{i^*}}{(\mathbf{D}^T\mathbf{D})_{i^*i^*}}$ {update coefficient}

end if

end while

the Moore-Penrose pseudoinverse:

$$\alpha_{\mathcal{I}} \leftarrow (\mathbf{D}_{\mathcal{I}}^T\mathbf{D}_{\mathcal{I}})^{-1}\mathbf{D}_{\mathcal{I}}^T\vec{x} \tag{2.7}$$

$$\text{where } \mathcal{I} = \{i : \alpha_i \neq 0\}$$

Adding this step ensures that at each iteration, the best coefficient α_{i^*} is chosen to explain a portion of the residual that is *orthogonal* to the subspace spanned by the dictionary elements that are already being used. This provides an advantage particularly with correlated dictionaries, since it discourages choosing new dictionary elements that can be explained in terms of ones that are already used. *Stagewise orthogonal matching pursuit* (STOMP) generalizes OMP by allowing *multiple* coefficients to be added to the support at each iteration. The coefficients are selected by applying a carefully chosen threshold to the back-projected threshold $(\mathbf{D}^T(\vec{x}-\mathbf{D}\vec{\alpha}))$ based on a signal detection analysis with respect to the noise [34]. The authors show that this method performs well when the dictionary $\mathbf{D}$ is approximately Gaussian-distributed. The work of

[30, 97] propose different procedures for selecting which coefficients to update within this framework, and conjugate gradient updates are used in [30] instead of directly solving Eq. 2.7 to speed up computations.

Greedy algorithms have the advantage of being fast, intuitive, and easy to implement. However, if several dictionary elements are very similar in shape (relative to the noise), it necessarily becomes difficult to select which subset best explains the observed signal. In these cases, greedy methods often yield solutions that are quite different from the true solution of Eq. 2.5. Indeed, theoretical results which guarantee the accuracy of greedy approximations depend crucially on the so-called "coherence" of $\mathbf{D}$, which bounds the cross-correlations between dictionary elements [143]:

$$\mu(\mathbf{D}) = \max_{i \neq j} \frac{|(\mathbf{D}^T\mathbf{D})_{ij}|}{\sqrt{(\mathbf{D}^T\mathbf{D})_{ii}(\mathbf{D}^T\mathbf{D})_{jj}}} \tag{2.8}$$

Behavior with transformation-invariant dictionaries

This issue of coherent dictionaries is especially problematic for dictionaries containing the same kernel(s) replicated at multiple transformation parameters. A fine sampling of the transformation parameter is needed for good signal reconstruction, but this results in a very coherent dictionary that is ill-suited for greedy methods. As an example, consider an observation consisting of multiple instances of a single kernel, shifted to different times. A greedy algorithm usually recovers isolated instances correctly, but often fails to recover multiple instances occurring at nearby times. As Fig. 2.2 illustrates, the reason is because a greedy procedure would select a time-shift in the middle of these, since this will be more

correlated with the signal than any of the correct time-shifts in isolation. As a result, the procedure gets stuck in a local minimum. This problem is inherent to the greediness of the algorithm, and exists regardless of the dictionary spacing or the value of λ.

2.1.3 Convex relaxation methods

The other class of approximate solutions for Eq. 2.5 arises by replacing the non-convex L_0 pseudonorm with the convex L_1 norm. The resulting objective is:

$$\vec{\alpha}(\vec{x}) = \arg\min_{\vec{\alpha}} \frac{1}{2\sigma^2}\|\vec{x} - \mathbf{D}\vec{\alpha}\|_2^2 + \lambda\|\vec{\alpha}\|_1 \tag{2.9}$$

This approach has frequently been used in statistics to regularize model parameters (i.e. effectively do model selection) in the presence of noise (the *LASSO* [142]) and in signal processing to recover sparse solutions with respect to over-complete wavelet dictionaries (*basis pursuit denoising* [20]). The objective in Eq. 2.9 is convex and thus has a unique global minimum that can be obtained via several standard optimization techniques, such as interior point methods [10]. The parameter λ in Eq. 2.9 need not be the same as in Eq. 2.5, since it penalizes a different functional of the coefficients. It is unclear how to determine the value of λ that will yield the most accurate approximation to the solution of Eq. 2.5. As a result, λ is typically chosen by cross-validation, or the L_1 relaxation is applied to the dual formulation of Eq. 2.6, thus avoiding the task of choosing a value for λ:

$$\vec{\alpha}(\vec{x}) = \arg\min_{\vec{\alpha}:\|\vec{x}-\mathbf{D}\vec{\alpha}\|_2^2 \leq M} \|\vec{\alpha}\|_1 \tag{2.10}$$

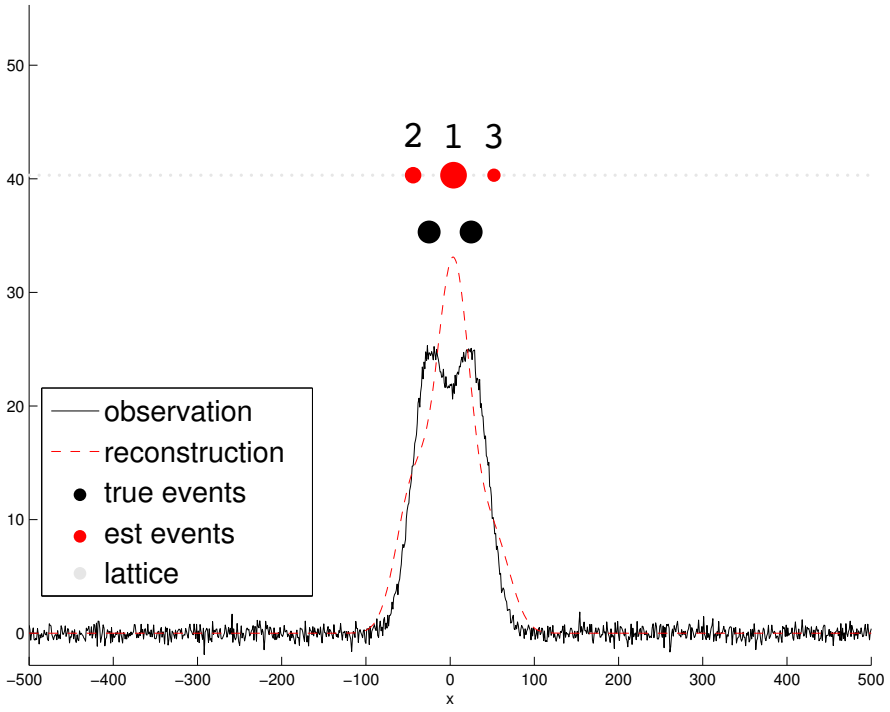


Figure 2.2: Example of how matching pursuit fails to resolve nearby events when using a dictionary of translated copies of a Gaussian kernel $f(x) \propto e^{\frac{-x^2}{2}}$ with $\lambda = 100$ and $\sigma = 1$. The estimated and true events are indicated by the red and black circles, respectively, with size indicating amplitude. The gray circles behind the estimated events indicate the time-shifts associated with each dictionary element. The numbers above the estimated events indicate the order in which non-zero coefficients were chosen in Algorithm 1.

Iterative thresholding methods

Recently, several fast "iterative thresholding" procedures [27, 24, 62, 39, 47, 8] have been proposed for solving Eq. 2.9 that resemble the greedy iterative procedures described in Section 2.1.2 (see [42] for a review). These can be viewed as an adaptation of the approach described in Section 2.1.1 to handle non-orthogonal dictionaries [42]. Recall that in the orthogonal case, the optimization decouples and the solution is obtained by applying a "soft threshold" to $\vec{z} = \mathbf{D}^T\vec{x}$ (shown by the blue line corresponding to $p = 1$ in Fig. 2.1):

$$\alpha_i = \begin{cases} 0 & \text{if } |z_i| < \lambda \\ z_i - \lambda & \text{if } z_i > \lambda \\ z_i + \lambda & \text{if } z_i < -\lambda \end{cases} \tag{2.11}$$

In the non-orthogonal case, we can turn the objective of Eq. 2.9 into one that decouples by adding two terms:

$$Q(\vec{\alpha}; \vec{\alpha}_0) = \frac{1}{2}\|\vec{x} - \mathbf{D}\vec{\alpha}\|_2^2 + \lambda\|\vec{\alpha}\|_1 + \frac{c}{2}\|\vec{\alpha} - \vec{\alpha}_0\|_2^2 - \frac{1}{2}\|\mathbf{D}\vec{\alpha} - \mathbf{D}\vec{\alpha}_0\|_2^2 \tag{2.12}$$

$$= -\vec{\alpha}^T\left[\mathbf{D}^T(\vec{x} - \mathbf{D}\vec{\alpha}_0)\right] + \lambda\|\vec{\alpha}\|_1 + \frac{c}{2}\|\vec{\alpha}\|_2^2 + \text{ const}$$

Notice that the sum of the two added terms in Eq. 2.12 is always positive for sufficiently large c. Furthermore, the value and gradient of each added term vanishes when $\vec{\alpha} = \vec{\alpha}_0$. As a result, $Q(\vec{\alpha}; \vec{\alpha}_0)$ is (1) an upper bound of the objective of Eq. 2.9, (2) equal to Eq. 2.9 at $\vec{\alpha} = \vec{\alpha}_0$, (3) tangent to Eq. 2.9 at $\vec{\alpha} = \vec{\alpha}_0$. Furthermore, $Q(\vec{\alpha}; \vec{\alpha}_0)$ is convex for sufficiently large

c. Therefore we can iteratively update the coefficient estimate:

$$\vec{\alpha}^{(t+1)} \leftarrow \arg\min_{\vec{\alpha}} Q(\vec{\alpha}; \vec{\alpha}^{(t)}) \tag{2.13}$$

This is an example of the Expectation-Maximization algorithm in machine learning [32], or majorization-minimization/bound optimization in the optimization literature [46]. It is well known that this procedure necessarily converges to the global minimum of Eq. 2.9. The "surrogate function" $Q(\vec{\alpha}; \vec{\alpha}_0)$ decouples and so the minimizer can be found by separately optimizing each coefficient. Setting the derivative to 0 gives:

$$\frac{\partial Q}{\partial \alpha_i} = \alpha_i - \frac{1}{c}(\mathbf{D}^T(\vec{x} - \mathbf{D}\vec{\alpha}_0))_i + \frac{\lambda}{c}\text{sign}(\alpha_i) = 0 \quad \forall i \tag{2.14}$$

yielding the solution:

$$\alpha_i = \begin{cases} (z_i - \frac{\lambda}{c}) & \text{if } z_i > 0 \\ 0 & \text{if } z_i = 0 \\ (z_i + \frac{\lambda}{c}) & \text{if } z_i < 0 \end{cases} \quad \forall i \tag{2.15}$$

$$\text{where } \vec{z} = \vec{\alpha}_0 + \frac{1}{c}\mathbf{D}^T(\vec{x} - \mathbf{D}\vec{\alpha}_0)$$

Another related class of methods for solving Eq. 2.9 employs iterative reweighted least square (IRLS) techniques where the L_1 term is approximated by a weighted L_2 term, with the weights being derived from the previous coefficients in the previous iteration [53, 19, 28]. These algorithms also amount to iteratively applying a point-wise nonlinear "shrinkage" function as in Eq. 2.15 [42].

Theoretical guarantees

Several recent theoretical results [18, 16, 17, 36, 39] provide sufficient conditions for the accuracy of the solution of Eq. 2.9, as an approxima-

tion of Eq. 2.5. These conditions rely on a "restricted isometry property" (RIP) of the dictionary $\mathbf{D}$, requiring all small subsets of dictionary elements to be "nearly" orthogonal systems. Mathematically, we define the RIP constant, δ_K, to be the minimal value satisfying:

$$(1-\delta_K)\|\vec{x}\|_2^2 \leq \|\mathbf{D}\vec{x}\|_2^2 \leq (1+\delta_K)\|\vec{x}\|_2^2 \quad \forall \vec{x} \text{ s.t. } \|\vec{x}\|_0 = K \tag{2.16}$$

Sufficient conditions in the literature typically impose bounds on sums of different RIP constants. For example, the condition of [16] requires:

$$\delta_K + \delta_{2K} + \delta_{3K} < 1 \tag{2.17}$$

in the noiseless case, where K is the number of nonzero elements in the true sparse coefficient vector. The condition of [18] requires:

$$\delta_{3K} + 3\delta_{4K} < 2 \tag{2.18}$$

for stable recovery in the noisy case.

Behavior with transformation-invariant dictionaries

Although the RIP is in general a weaker condition than the coherence bounds required by greedy methods (Eq. 2.8), it still does not apply to most transformation-invariant dictionaries. A fine sampling of the transformation parameter causes neighboring dictionary elements to be very correlated, thus violating the RIP (even δ_2 will have a large value). Figure 2.3 demonstrates the failure of the L_1 approximation with a simple translation-invariant dictionary containing three shifted versions of a single waveform (Fig. 2.3(a)). The observed signal is simply this waveform at an intermediate time-shift, with added noise (superimposed in gray).

Figure 2.3(c) illustrates the optimization problem of Eq. 2.9. Notice that family of solutions, parametrized by λ (red path), are not sparse in the L_0 sense (i.e., they do not intersect either of the two axes) until λ is so large that the signal reconstruction is quite poor. This is the case regardless of how small Δ is, and is due both to the failure of the L_1 norm to approximate the L_0 pseudonorm and to the inability of the discrete model to account for continuous event times.

2.2 Bilinear models which account for transformation

In [141], a bilinear generative model was proposed for images, in which "content" and "style" information (corresponding to what we refer to as "what" and "where" information) is captured by two sets of variables $\{\alpha_c\}_c$ and $\{\beta_s\}_s$, respectively. Mathematically this is expressed as:

$$\vec{x} = \sum_{s,c} \alpha_c \vec{d}_{cs} \beta_s + \vec{\epsilon} \tag{2.19}$$

The intuitive appeal of the bilinear approach is that continuously transforming an image's "style" can be modeled by multiplicatively modulating the coefficients, rather than varying additive coefficients. In [141], it was shown that learning such a model for digits (with different fonts), or face images (with different poses) produces a representation suitable for classification into discrete content and style groups. This model was extended in [56] to include a sparse factorial prior distribution on both content and style coefficients. The kernels, when adapted to natural images patches, resembled the Gabor-like filters learned in [101] but with a

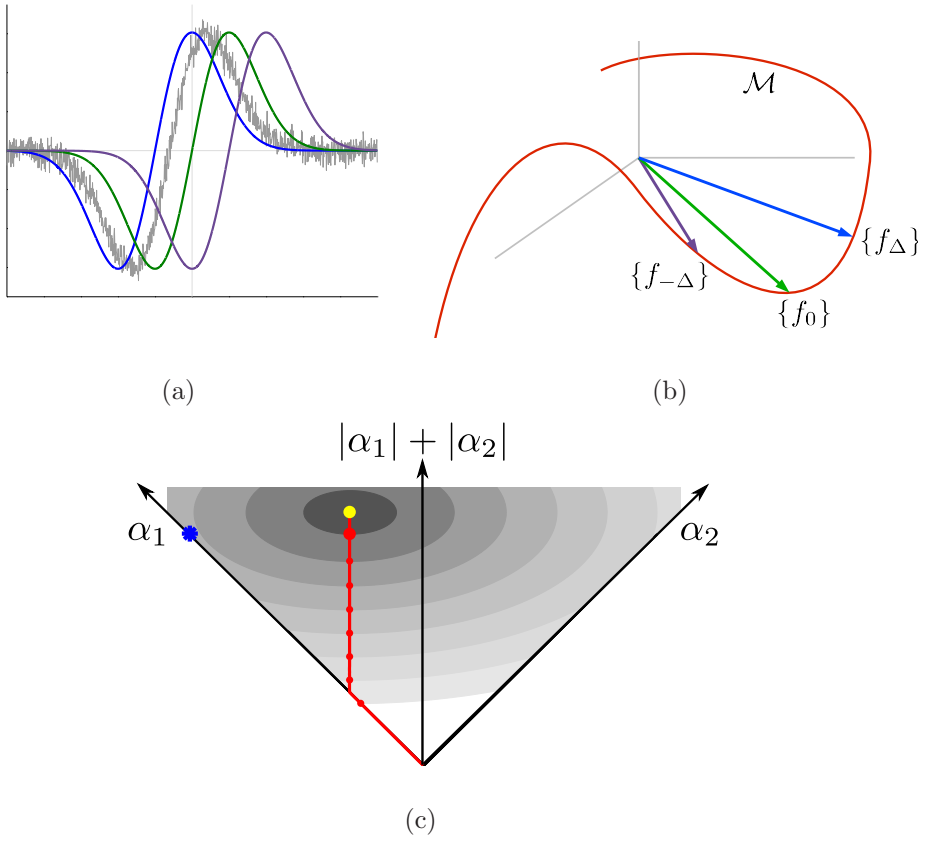


Figure 2.3: Illustration of sparse recovery via an L_1 relaxation using a convolutional dictionary. (a) Dictionary containing the waveform $f(t) \propto te^{-\frac{t^2}{2}}$ at 3 shifts $(-\Delta, 0, \Delta)$. The noisy observation, $x(t) = f(t + 0.65\Delta)$, is superimposed in gray. (b) Dictionary in the underlying vector space. The three points lie on the translation-invariant manifold $\mathcal{M}_{f,T}$ corresponding to the waveform. (c) Solution of Eq. 2.9 in the space of the first two coefficients (α_1, α_2). A solution occurs when a level curve of the quadratic term (shaded ovals) is tangent to a level curve of the L_1 term (horizontal lines in the rotated axes). The red line plots the solution, as a function of λ, from $\lambda = 0$ (yellow dot) to very large values (origin). Red dots demarcate equal increments of λ. The blue star marks the true L_0-minimizing solution.

natural grouping ($\left[\vec{d_{cs}}\right]_s$ for each c) corresponding to transformed copies of a filter with the same spatial orientation and scale.

The bilinear model suffers from several disadvantages. First, it is unclear that the model will automatically learn to separate content and style information into the two variable sets. Indeed, this separation is enforced in the learning algorithms proposed by [141] and [56] by incorporating prior knowledge of the content and style category for each data sample. This is somewhat remedied in [7], which learns a bilinear model in a totally unsupervised manner by uses temporal stability in natural videos as an indicator of content versus style information (building on the temporal stability criteria used in [48, 151, 150]). However, this study focused on connections to neuroscience rather than on the advantages of the computational representation itself. Our method, on the other hand, is an inference method that assumes known transformation *types* across the ensemble but unknown *amounts* in any given signal. Second, the mapping from the inferred style coefficients to transformation parameters is not well-defined. While it is clear that they co-vary continuously with transformations such as translation [56, 7], one cannot explicitly recover the transformation amount. Third, inference in these bilinear models is inherently non-convex (due to the "chicken-and-egg" problem of content and style variables) and inefficient when combined with sparsity constraints. Learning also requires very complicated procedures [56, 7].

2.3 Motivation for CBP

Recall that conventional methods for approximating the sparse linear inverse solution of Eq. 2.5 yield suboptimal solutions when used with "transformation-sample" dictionaries (i.e. dictionaries constructed by replicating a set of prototypical kernels with evenly sampled transformation amounts). The reason is the fundamental tradeoff between fine sampling of the transformation amounts (which is required for good signal reconstruction), and the conditions required for these approximations to be accurate. Greedy algorithms are susceptible to suboptimal local minima (Fig. 2.2) when dictionary elements are highly correlated, as is the case when the transformations are finely sampled. Convex relaxation methods, on the other hand, employ the L_1 norm which fails to approximate the L_0 measure with correlated dictionaries (Fig. 2.3).

We now introduce a novel method, called *continuous basis pursuit* (CBP), for the sparse recovery problem that addresses the limitations of conventional methods in the transformation-invariant context. The key is to change the dictionary into a form that is more amenable for sparse coefficient recovery (rather than changing the sparse recovery algorithm itself), while keeping the space spanned by the dictionary elements essentially the same. We motivate the new form of dictionary by formulating a simple source model for the 1D translation-invariant case in Section 2.4. By adding new variables to parametrize the transformations via a predefined interpolation mapping, we are able to recover these quantities by solving a constrained convex optimization problem, and then applying the inverse map to the solution. We develop two versions of this ap-

proach in Sections 2.5 and 2.6, and present a general form in Section 2.7. In Section 2.8, we present empirical evidence of our method's advantage over conventional methods on a sparse deconvolution task.

2.4 Source model formulation

We consider the simple case in which we observe a noisy linear combination of a small number of time-shifted versions of a single known waveform $f(t)$. The function $f(t-\tau)$ is abbreviated by $f_\tau(t)$ or simply f_τ. We formulate the problem in terms of the original variables which we would like to infer, namely the real-valued amplitudes $\{a_j\}$ and time-shifts $\{\tau_j\}$ that comprise the observed signal:

$$x(t) = \sum_{j=1}^{N} a_j f_{\tau_j}(t) + \epsilon(t), \tag{2.20}$$

where $\epsilon(t)$ represents a noise process. In the rest of the section, we assume (without loss of generality) that $\|f(t)\|_2 = 1$, and that the amplitudes, $\{a_j\}$, are nonnegative. The goal is to find event amplitudes $\{a_j\}$ and time-shifts $\{\tau_j\}$ that minimize a tradeoff between signal reconstruction and the number of events:

$$\min_{N,\{a_j\},\{\tau_j\}} \frac{1}{2\sigma^2}\|y(t) - \sum_{j=1}^{N} a_j f_{\tau_j}(t)\|_2^2 + \lambda N \tag{2.21}$$

Solving Eq. 2.21 can also be interpreted as performing MAP estimation of $\{a_j\}, \{\tau_j\}$, assuming Gaussian white noise (with variance σ^2) and a Poisson process prior on the $\{\tau_j\}$. One can also incorporate a term corresponding to the prior probability of the amplitudes $\{a_j\}$, if known. Solving Eq. (2.21) directly is intractable, due to the discrete nature of

N and the nonlinear embedding of the τ_j's within the argument of the waveform $f(\cdot)$. It is thus desirable to find alternative formulations that (i) still approximate the signal distribution well, (ii) have parameters that can be tractably estimated, and (iii) have an intuitive mapping back to the original variables of interest $(N, \{a_j\}, \{\tau_j\})$.

Recall that the standard approach is to construct a dictionary $\mathbf{D}_\Delta$ of time-shifted copies of $f(t)$ with a spacing Δ and then solve the familiar sparse inverse problem (Eq. 2.5 of Section 2.1):

$$\min_{\vec{\alpha}} \frac{1}{2\sigma^2} \|x(t) - (\mathbf{D}_\Delta \vec{\alpha})(t)\|_2^2 + \lambda \|\vec{\alpha}\|_0 \tag{2.22}$$

$$\text{where } (\mathbf{D}_\Delta \vec{\alpha})(t) := \sum_{i=1}^{N} \alpha_i f_{i\Delta}(t). \tag{2.23}$$

The coefficient α_i represents events occurring between $i\Delta - \frac{\Delta}{2}$ and $i\Delta + \frac{\Delta}{2}$ for $i = 1, ..., N$ (where $N = \lceil T/\Delta \rceil$). The dictionary $\mathbf{D}_\Delta$ can be viewed as a uniform sampling of the nonlinear translation manifold given by:

$$\mathcal{M}_{f,T} := \{a f_\tau : a \geq 0, \tau \in [0, T]\} \tag{2.24}$$

The span of the dictionary elements provides a linear subspace approximation of this manifold, as illustrated in Fig. 2.3(b). However, the representation of a single element of the manifold will typically be approximated by a superposition of two or more elements from the dictionary $\mathbf{D}_\Delta$. This was in fact the case in the simple example illustrated in Fig. 2.3. We can remedy this by augmenting the dictionary to include *interpolation* functions, that allow better approximation of the continuously shifted waveforms. We describe two specific examples of this method, and then provide a general form.

2.5 CBP with Taylor interpolation

If $f(t)$ is sufficiently smooth, one can approximate local shifts of $f(t)$ by linearly combining $f(t)$ and its derivative via a first-order Taylor expansion:

$$f_\tau = f_0 - \tau f_0' + O(\tau^2) \tag{2.25}$$

This motivates a dictionary consisting of the original shifted waveforms, $\{f_{i\Delta}\}$, and their derivatives, $\{f'_{i\Delta}\}$. Since the Taylor expansion is only valid locally, we must choose the spacing, Δ to be the largest spacing that provides a desired approximation accuracy δ:

$$\Delta := \max\{\Delta' : \max_{|\tau|<\frac{\Delta'}{2}} \|f_\tau - (f_0 - \tau f_0')\|_2 \leq \delta\} \tag{2.26}$$

The value of δ can be set according to the observed noise level: when there is higher noise, we can allow for more approximation error since this error can be attributed to noise. We can then approximate the manifold of scaled and time-shifted waveforms using *constrained* linear combinations of dictionary elements:

$$\mathcal{M}_{f,T} \approx \left\{ \alpha f_{i\Delta} + d f'_{i\Delta} : |d| \leq \frac{\Delta}{2}\alpha \right\} \tag{2.27}$$

There is a one-to-one correspondence between sums of waveforms on the manifold $\mathcal{M}_{f,T}$ and their respective approximations with this dictionary:

$$\sum_i \alpha_i f_{i\Delta} + d_i f'_{i\Delta} = \sum_i \alpha_i (f_{i\Delta} + \frac{d_i}{\alpha_i} f'_{i\Delta}) \approx \sum_i \alpha_i f_{(i\Delta - d_i/\alpha_i)}. \tag{2.28}$$

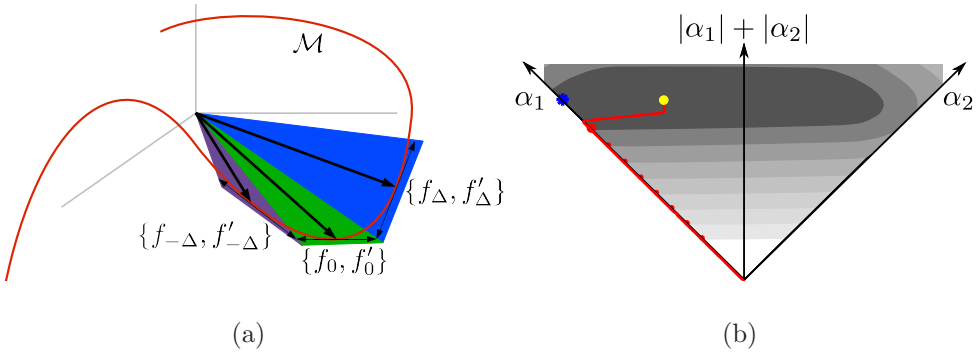
This correspondence holds as long as $|d_i/\alpha_i| \neq \frac{\Delta}{2}$ (equality corresponds to the situation where the the waveform is shifted exactly halfway in between two lattice points, and can thus be equally well represented by the

basis function and associated derivative on either side). The inequality constraint on d in Eq. 2.27 is essential, and serves two purposes. First, it ensures that the Taylor approximation is only used when it is accurate, so that only time-shifted and scaled waveforms are used in reconstructing the signal. Second, it discourages neighboring pairs $(f_{i\Delta}, f'_{i\Delta})$ and $(f_{(i+1)\Delta}, f'_{(i+1)\Delta})$ from explaining the same event. The inference problem can now be cast as a *constrained* convex optimization:

$$\begin{aligned} \min_{\vec{\alpha},\vec{d}} & \frac{1}{2\sigma^2}\|y(t) - (\mathbf{D}_\Delta\vec{\alpha})(t) - (\mathbf{D'}_\Delta\vec{d})(t)\|_2^2 + \lambda\|\vec{\alpha}\|_1 \\ \text{s.t.} \quad & |d_i| \leq \frac{\Delta}{2}\alpha_i \text{ for } i = 1, ..., N \end{aligned} \tag{2.29}$$

where the dictionary $\mathbf{D}_\Delta$ is defined as in Eq. (2.23), and $\mathbf{D'}_\Delta$ is a dictionary of time-shifted waveform derivatives $\{f'_{i\Delta}\}$. The dictionary and associated coefficient constraints are illustrated in Fig. 2.4(a), showing that the manifold is now approximated by constrained triangular regions, providing a better tiling than in Fig. 2.3(b). This local linearization of the transformation manifold is used in the tangent prop method of [130] in the context of distance learning and classification. Eq. (2.28) provides an explicit mapping from appropriately constrained coefficients to event amplitudes and time-shifts. Figure 2.4(b) illustrates this objective function for the same single-waveform example described previously. The shaded regions are the level sets of the L_2 term of Eq. (2.29) visualized in the (α_1, α_2)-plane by minimizing over the derivative coefficients (d_1, d_2) subject to the constraints. Note that unlike the corresponding BP level sets shown in Fig. 2.3(c), these are no longer elliptical, and that they allow sparse solutions (i.e., points on the α_1-axis) with low reconstruction error. As a result, the solution of Eq. (2.29) is not only sparse

in the L_0 sense, but also provides a good reconstruction of the signal for appropriately chosen λ.



(a) (b)

Figure 2.4: (a) Continuous basis pursuit with first-order Taylor interpolator (CBP-T), as specified by Eq. (2.29). Each pair of functions, $(f_{i\Delta}, f'_{i\Delta})$, with properly constrained coefficients, represents a triangular region of the space (shaded regions). (b) CBP with Taylor interpolation applied to the same illustrative example described in Fig. 2.3(c). Shaded regions denote the level curves of the quadratic term in Eq. 2.29 in the space of two amplitude variables (α_1, α_2), minimized over derivative variables (d_1, d_2) that satisfy the inequality constraints.

2.6 CBP with polar interpolation

2.6.1 The polar interpolator

Although the Taylor series provides the most intuitive and well-known method of approximating time-shifts, we have developed an alternative interpolator that is much more accurate for a wide class of waveforms.

The solution is motivated by the observation that the manifold $\mathcal{M}_{f,T}$ of time-shifted waveforms must lie on the surface of a hypersphere (translation preserves the L_2-norm, barring border effects) in the function space underlying $f(t)$. Furthermore, this manifold must have a constant curvature (by symmetry). This leads to the notion that it might be well-approximated by a circular arc. As such, we approximate a segment of the manifold, $\{f_\tau : |\tau| \leq \frac{\Delta}{2}\}$, by the unique circular arc that circumscribes the three functions $\{f_{-\Delta/2}, f_0, f_{\Delta/2}\}$, as illustrated in Fig. 2.5(a). The resulting interpolator is an example of a trigonometric spline [122], in which three basis functions $\{c(t), u(t), v(t)\}$ are linearly combined using trigonometric coefficients to approximate intermediate translates of $f(t)$:

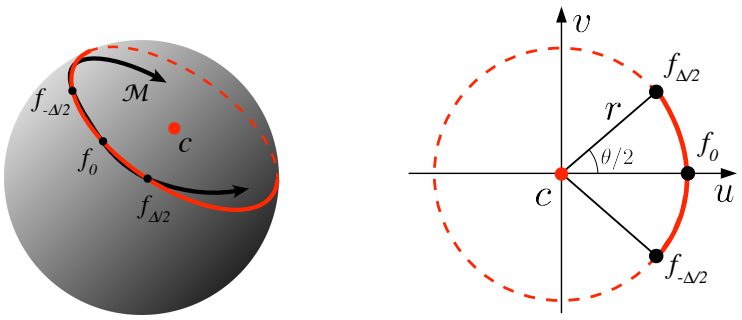
$$f_\tau(t) \approx \begin{pmatrix} 1 \\ r\cos(\frac{\tau}{\Delta}\theta) \\ r\sin(\frac{\tau}{\Delta}\theta) \end{pmatrix}^T \begin{pmatrix} c(t) \\ u(t) \\ v(t) \end{pmatrix} \quad \text{for } |\tau| < \frac{\Delta}{2} \tag{2.30}$$

The constants r and θ are the radius and subtended angle of the circumscribing arc, respectively (see Fig. 2.5(b)). These constants, along with the three basis functions, can be computed in closed form for a given waveform, and the main properties of the polar interpolator can be summarized as follows (see Appendix A for details):

1. The polar approximation of Eq. 2.30 is *exact* when $f(t)$ is sinusoidal, regardless of Δ.

2. The approximation accuracy degrades as the bandwidth of $f(t)$ increases.

3. The basis $\{c(t), u(t), v(t)\}$ is an orthogonal system.



(a) (b)

Figure 2.5: Illustration of the polar interpolator. (a) The manifold of time shifts of $f(t)$ (black line) lies on the surface of a hypersphere. We approximate a segment of this manifold, for time shifts $\tau \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$, with a portion of a circle (red), with center defined by $c(t)$.

2.6.2 Optimization formulation

We now approximate the translational manifold $\mathcal{M}_{f,T}$ using a dictionary of time-shifted copies of the functions used to represent the polar interpolation, $\{c_{i\Delta}, u_{i\Delta}, v_{i\Delta}\}$, together with constraints on their coefficients:

$$\mathcal{M}_{f,T} \approx \left\{ \begin{array}{lll} \alpha c_{i\Delta} & & \beta^2 + \gamma^2 = \alpha^2 r^2, \\ + \beta u_{i\Delta} & : & 0 \leq \alpha r \cos(\frac{\theta}{2}) \leq \beta \\ + \gamma v_{i\Delta} & & i = 1, ...N \end{array} \right\} \tag{2.31}$$

The constraints on the coefficients (α, β, γ) ensure that the linear combination approximates a scaled translate of $f(t)$. Notice that $\beta^2 + \gamma^2 = \alpha^2 r^2$ is a non-convex constraint. In order to maintain tractability we relax to

the convex hull computed from the constraints in Eq. 2.31:

$$\mathcal{M}_{f,T} \approx \left\{ \begin{array}{lll} \alpha c_{i\Delta} & & \sqrt{\beta^2+\gamma^2} \leq \alpha r, \\ + \beta u_{i\Delta} & : & \alpha r \cos\left(\frac{\theta}{2}\right) \leq \beta \\ + \gamma v_{i\Delta} & & i = 1, ...N \end{array} \right\} \tag{2.32}$$

As with the Taylor approximation, we have a one-to-one correspondence between event amplitudes/time-shifts and the constrained coefficients:

$$\sum_i \alpha_i c_{i\Delta} + \beta_i u_{i\Delta} + \gamma_i v_{i\Delta} \approx \sum_i \alpha_i f_{(i\Delta - \frac{\Delta}{\theta}\tan^{-1}(\gamma_i/\beta_i))} \tag{2.33}$$

as long as $\frac{\gamma_i}{\beta_i} \neq \tan\left(\frac{\theta}{2}\right)$ for all i (note that the inequality constraints ensure that $|\tan^{-1}\left(\frac{\gamma_i}{\beta_i}\right)| \leq \frac{\theta}{2}$). The inference problem again boils down to minimizing a constrained convex objective function:

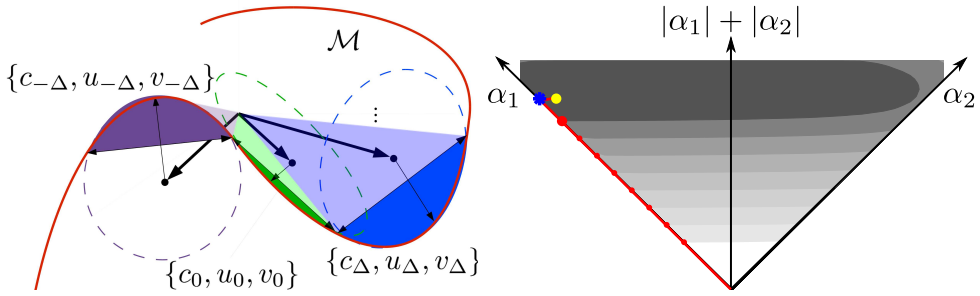
$$\begin{aligned} \min_{\vec{\alpha},\vec{\beta},\vec{\gamma}} \ & \frac{1}{2\sigma^2}\left\| y(t) - (\mathbf{C}_\Delta \vec{\alpha})(t) - (\mathbf{U}_\Delta \vec{\beta})(t) - (\mathbf{V}_\Delta \vec{\gamma})(t) \right\|_2^2 + \lambda \|\vec{\alpha}\|_1 \\ \text{s.t.} \quad & \left\{ \begin{array}{l} \sqrt{\beta_i^2+\gamma_i^2} \leq \alpha_i r, \\ \alpha_i r \cos\left(\frac{\theta}{2}\right) \leq \beta_i \end{array} \right\} \text{ for } i = 1, ...N \end{aligned} \tag{2.34}$$

where $\mathbf{C}_\Delta, \mathbf{U}_\Delta, \mathbf{V}_\Delta$ are dictionaries containing Δ-shifted copies of $c(t)$,$u(t)$, and $v(t)$, respectively. Equation (2.34) is an example of a "second-order cone program" for which efficient solvers exist [10]. After the optimum values for $\{\vec{\alpha}, \vec{\beta}, \vec{\gamma}\}$ are obtained, time-shifts and amplitudes can be inferred by first projecting the solution back to the original (non-convex) constraint set of Eq. 2.31:

$$(\alpha_i, \beta_i, \gamma_i) \leftarrow (\alpha_i, \frac{\beta_i \alpha_i r}{\sqrt{\beta_i^2+\gamma_i^2}}, \frac{\gamma_i \alpha_i r}{\sqrt{\beta_i^2+\gamma_i^2}}) \tag{2.35}$$

(corresponding to a radial projection in Fig. 2.5(b)) and then using Eq. (2.33) to solve for the event times.

Figure 2.6(a) illustrates that the polar interpolator yields a piece-wise circular approximation of the manifold. Figure 2.6(b) illustrates the optimization of Eq. (2.34) for the simple example described in the previous section. Notice that the solution corresponding to $\lambda = 0$ (yellow dot) is substantially sparser relative to both the CBP-T and BP solutions, and that the solution becomes L_0 sparse if λ is increased by just a small amount, giving up very little reconstruction accuracy.



(a) (b)

Figure 2.6: (a) Continuous basis pursuit with polar interpolation (CBP-P), as specified by Eq. (2.34). Each triplet of functions, $(c_{i\Delta}, u_{i\Delta}, v_{i\Delta})$, represents a section of a cone (see Fig. 2.5(b) for parametrization) (b) CBP with polar interpolation applied to the same illustrative example described in Figs. 2.3(c) and 2.4(b). Shaded regions denote the level curves of the quadratic term in Eq. 2.34 in the space of two amplitude variables (α_1, α_2), minimized over all auxiliary variables $(\beta_1, \beta_2, \gamma_1, \gamma_2)$ that satisfy the inequality constraints.

2.7 General interpolation

We can generalize the CBP approach to use any linear interpolation scheme. Suppose we have a set of basis functions $\{\phi_n(t)\}_1^m$ and a corresponding interpolation map $\Psi : \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right] \to \mathbb{R}^m$ such that local shifts can be approximated as:

$$f_\tau \approx \sum_{n=1}^{m} \Psi_n(\tau)\phi_n, \quad |\tau| \leq \frac{\Delta}{2}. \tag{2.36}$$

(e.g., Eq. (2.25) and Eq. (2.30)). Let S be the set of all nonnegative scalings of the image of $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ under the interpolator:

$$S = \{a\vec{x} : a \geq 0 \ , \ \vec{x} \in \text{Range}(\Psi)\}.$$

In general, S may not be convex (as in the polar case), and we denote its convex hull by $\mathcal{H}$. If we have a set of coefficients $\vec{\alpha} \in \mathbb{R}^{N\times m}$ where each block $\vec{\alpha}_i := [\alpha_{i1}, ..., \alpha_{im}]$ is in $\mathcal{H}$, then the signal given by:

$$\sum_{i=1}^{N}\sum_{n=1}^{m} \alpha_{in}\phi_n(t - i\Delta) \tag{2.37}$$

approximates a sum of scaled and translated waveforms. If Ψ is invertible, the amplitudes and shifts are obtained as follows:

$$a_i \leftarrow a \quad \text{s.t.} \quad \frac{P_S(\vec{\alpha}_i)}{a} \in \text{Range}(\Psi) \tag{2.38}$$

$$\tau_i \leftarrow i\Delta + \Psi^{-1}(P_S(\vec{\alpha}_i)/a) \tag{2.39}$$

where $P_S(.)$ projects points in $\mathcal{H}$ onto S. Note that in this general form, the L_2 norm of (the projection of) each group $\vec{\alpha}_i$ governs the amplitude of the corresponding time-shifted waveform.[1] Finally, we can obtain the

[1] Our specific examples used the amplitude of a single coefficient as opposed to the group L_2 norm. However, the constraints in these examples make the two formula-

coefficients by solving:

$$\min_{\vec{\alpha}} \frac{1}{2\sigma^2} \|y(t) - \mathbf{D}_\Delta \vec{\alpha})(t)\|_2^2 + \lambda \sum_{i=1}^{N} \|\vec{\alpha}_i\|_2 \qquad (2.40)$$

$$\text{s.t.} \quad \vec{\alpha}_i \in \mathcal{H} \quad \text{for} \quad i = 1, ..., N$$

where the linear operator $\mathbf{D}_\Delta$ is defined as:

$$(\mathbf{D}_\Delta \vec{x})(t) := \sum_{i=1}^{N} \sum_{n=1}^{m} \alpha_{in} \phi_n(t - i\Delta)$$

Equation (2.40) can be solved efficiently using standard convex optimization methods (e.g., interior point methods [10]). It is similar to the objective functions used to recover so-called "block-sparse" signals (e.g., [65, 43]), but includes auxiliary constraints on the coefficients to ensure that only signals close to span($\mathcal{M}_{f,T}$) are represented. Table 2.1 summarizes the Taylor and polar interpolation examples within this general framework, along with the case of nearest-neighbor interpolation (which corresponds to standard BP described in Section 2.1.3).

Property	BP	CBP-T	CBP-P
$\{\phi_n(t)\}$	$[f(t)]$	$[f(t)\ ,\ f'(t)]$	$[c(t)\ ,\ u(t)\ ,\ v(t)]$
$\vec{\Psi}(\tau)$	1	$[1\ ,\ \tau]^T$	$[1\ ,\ r\cos(\theta\frac{2\tau}{\Delta})\ ,\ r\sin(\theta\frac{2\tau}{\Delta})]^T$
S	$\{\alpha_1 \geq 0\}$	$\{\lvert\alpha_2\rvert \leq \alpha_1 \frac{\Delta}{2}\}$	$\{\alpha_2^2 + \alpha_3^2 = r^2\alpha_1^2\ ,\ 0 \leq r\alpha_1\cos(\theta) \leq \alpha_2\}$
$\mathcal{H}$	$\{\alpha_1 \geq 0\}$	$\{\lvert\alpha_2\rvert \leq \alpha_1 \frac{\Delta}{2}\}$	$\{\sqrt{\alpha_2^2 + \alpha_3^2} \leq r\alpha_1\ ,\ r\alpha_1\cos(\theta) \leq \alpha_2\}$
$P_S(\vec{\alpha})$	$\vec{\alpha}$	$\vec{\alpha}$	$[\alpha_1\ ,\ r\alpha_1\frac{\alpha_2}{\sqrt{\alpha_2^2+\alpha_3^2}}\ ,\ r\alpha_1\frac{\alpha_3}{\sqrt{\alpha_2^2+\alpha_3^2}}]^T$

Table 2.1: Components of the BP, CBP-T and CBP-P methods.

tions equivalent. For the Taylor interpolator, $\alpha_i^2 + d_i^2 \approx \alpha_i^2$. For the polar interpolator, $c_i^2 + u_i^2 + v_i^2 \approx (1 + r^2)c_i^2$.

The quality of the solution relies on (1) accuracy of the interpolator, (2) the convex approximation $\mathcal{H} \approx S$, and (3) the ability of the block-L_1 based penalty term in Eq. (2.40) to achieve L_0-sparse solutions that reconstruct the signal accurately. The first two of these are relatively straightforward, since they depend solely on the properties of the interpolator (see Fig. A.3). The last is difficult to predict, even for the simple examples illustrated in Figs. 2.3(c),2.4(b),2.6(b). The level sets of the L_2 term can have a complicated form when taking the constraints into account, and it is not clear a priori whether this will facilitate or hinder the L_1 term in achieving sparse solutions.

The theoretical results described in Section 2.1.3 suggest that if dictionary correlations in the CBP dictionaries are less than those in the BP dictionary, then a sparse solution should be able to be recovered via an L_1-based optimization such as Eq. 2.9. Intuitively, the correlations in the CBP dictionaries are decreased for three reasons: (1) there are no correlations within the interpolation groups (they are orthogonal systems in both the Taylor and polar cases), (2) the groups are able to be spaced further apart along the manifold (i.e. larger Δ), and (3) constraints further reduce the set of possible coefficient combinations). The next section provides empirical results which clearly indicate that solving Eq. (2.40) with Taylor and polar interpolators yields substantially sparser solutions than those achieved with standard BP.

2.8 Empirical results

2.8.1 Single feature

We evaluate our method on data simulated according to the generative model of Eq. (2.20). Event amplitudes were drawn uniformly from the interval $[0.5, 1.5]$. We used a single template waveform $f(t) \propto te^{-\alpha t^2}$ (normalized, so that $\|f\|_2 = 1$), for which the interpolator performances are plotted in Fig. A.3. We compared solutions of Eqs. (2.9), (2.29), and (2.34). Amplitudes were constrained to be nonnegative (this is already assumed for the CBP methods, and amounts to an additional linear inequality constraint for BP). Each method has two free parameters: Δ controls the spacing of the basis, and λ controls the tradeoff between reconstruction error and sparsity. We varied these parameters systematically and measured performance in terms of two quantities (corresponding to the two terms in Eq. 2.21): (1) signal reconstruction error (which decreases as λ or Δ decreases), and (2) sparsity of the estimated event amplitudes, which increases as λ increases. The former is simply the first term in the objective function (for all three methods). For the latter, to ensure numerical stability, we used the L_p norm with $p = 0.1$ (results were stable with respect to the choice of p, as long as $p << 1$ and p was not below the numerical precision of the optimizations). Computations were performed numerically, by sampling the functions $f(t)$ and $y(t)$ at a constant spacing that was finer than any Δ used. We used the convex solver package `CVX` [54] to obtain numerical solutions.

A small temporal window of the events recovered by the three methods is provided in Fig. 2.7. The three plots show the estimated event times and amplitudes for BP, CBP-T, and CBP-P (upward stems) com-

pared to the true event times/amplitudes (downward stems). The figure demonstrates that CBP, equipped with either Taylor or polar interpolators, is able to recover the event train more accurately, and with a larger spacing between basis functions (indicated by the tick marks on the x-axis). As predicted by the reasoning laid out in Fig. 2.3(c), basis pursuit tends to split events across two or more adjacent low-amplitude coefficients, thus producing less sparse solutions and making it hard to infer the number of events and their respective amplitudes and times. Sparsity can be improved by increasing λ, but at the expense of a substantial increase in approximation error.

Figure 2.8 illustrates the tradeoff between sparsity and approximation error for each of the methods. Each panel corresponds to a different noise level. The (x, y) coordinates of single point represent the reconstruction error and sparsity of the solution (with color indicating the method) for a single (Δ, λ) combination, averaged over 500 trials. The solid curves are the (numerically computed) convex hulls of all points obtained for each method, and clearly indicate the tradeoff between the two types of error. We can see that the performance of BP is strictly dominated by that of CBP-T: For every BP solution, there is a CBP-T solution that has lower values for both error types. Similarly, CBP-T is strictly dominated by CBP-P, which can be seen to come close to the error values of the ground truth answer (which is indicated by a black dot). Note that the reconstruction error of the true solution and of all methods is bounded below by the variance of the noise that lies outside of the subspace spanned by the set of shifted copies of the waveform. We performed a signal detection analysis of the performance of these

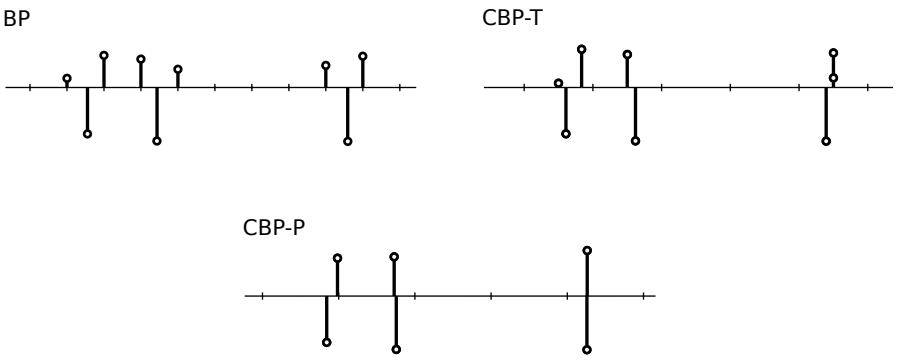


Figure 2.7: Sparse signal recovery example. The source was a sparse set of 3 event times/amplitudes, represented by the downward stems in all three plots (horizontal displacement indicates time, height indicates amplitude). These were then convolved with a waveform $f(t) \propto te^{\gamma t^2}$ and Gaussian noise was added with standard deviation $\|f\|_\infty/12$. Upward stems on the three plots show the source recovered by BP (Eq. (2.9)), CBP-T (Eq. (2.29)), and CBP-P (Eq. (2.34)), respectively. For each method, the values of Δ and λ were chosen to minimize the sum of sparsity and reconstruction error (large dots in Fig. 2.8). Horizontal displacements indicate event times determined by the interpolation coefficients via Eq. (2.38), while stem height indicates amplitude. Amplitudes less than 0.01 were eliminated for clarity. Ticks denote the location of the basis functions corresponding to each upward-pointing stem.

methods, classifying identification errors as misses and false positives. We match an estimated event with a true event if the estimated location is within 3 samples of the true event time, the estimated amplitude is within a threshold $\frac{1}{\sqrt{12}}$ of the true amplitude (one standard deviation of the amplitude distribution Unif ($[0.5, 1.5]$)), and no other estimated event has been matched to the true event. We found that results were rela-

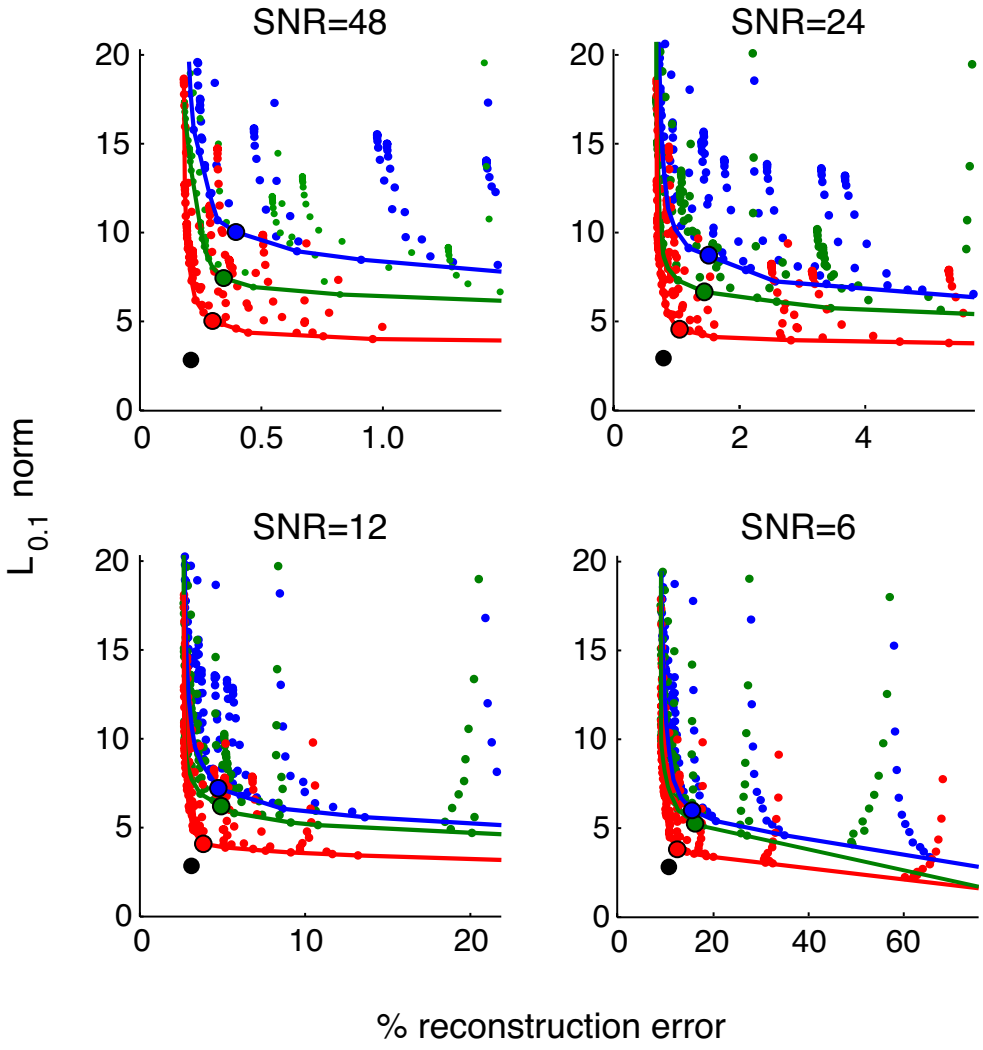


Figure 2.8: Error plots for 4 noise levels (SNR is defined as $\|f\|_\infty/\sigma$). Each graph shows the tradeoff between the average reconstruction error and sparsity (measured as average $L_{0.1}$ norm of estimated amplitudes). Each point represents the error values for one of the methods, applied with a particular setting of (Δ, λ), averaged over 500 trials. Colors indicate the method used (blue:BP,green:CBP-T, red:CBP-P). Bold lines denote the convex hulls of all points for each method. The large dots indicate the "best" solution, as measured by Euclidean distance from the correct solution (indicated by black dots).

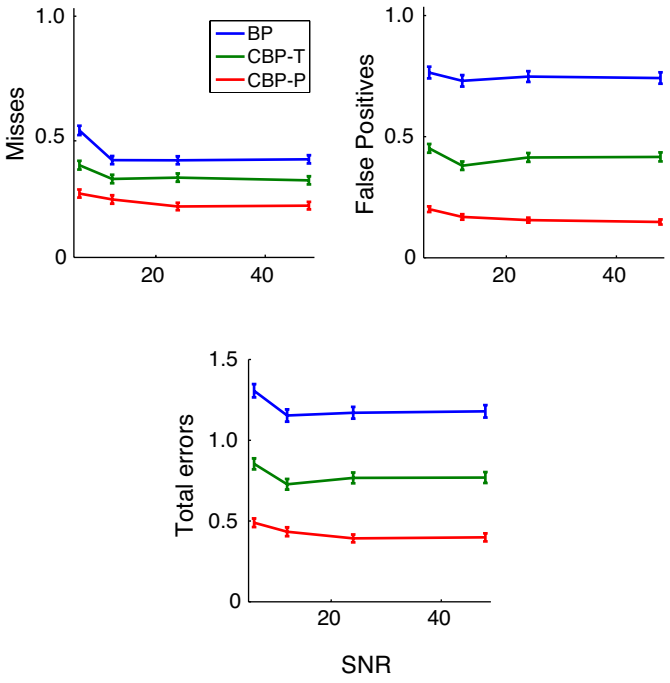


Figure 2.9: Signal detection analysis of solutions at three SNR levels. (see text). Misses, false positives, and total errors (sum of misses and false positives), as a fraction of the mean number of events, were computed over 500 trials for each method, and for each SNR (defined as $\|f\|_\infty/\sigma$).

tively stable with respect to the threshold choices. For each method and noise level we chose the (λ, Δ) combination yielding a solution closest to ground truth (corresponding to the large dots in Fig. 2.8). Fig. 2.9 shows the errors as a function of the noise level. We see that performance of all methods is surprisingly stable across SNR levels. We also see that BP performance is dominated at all noise levels by CBP-T, which has fewer misses as well as fewer false positives, and CBP-T is similarly dominated by CBP-P.

Finally, Fig. 2.10 shows the distribution of the nonzero amplitudes es-

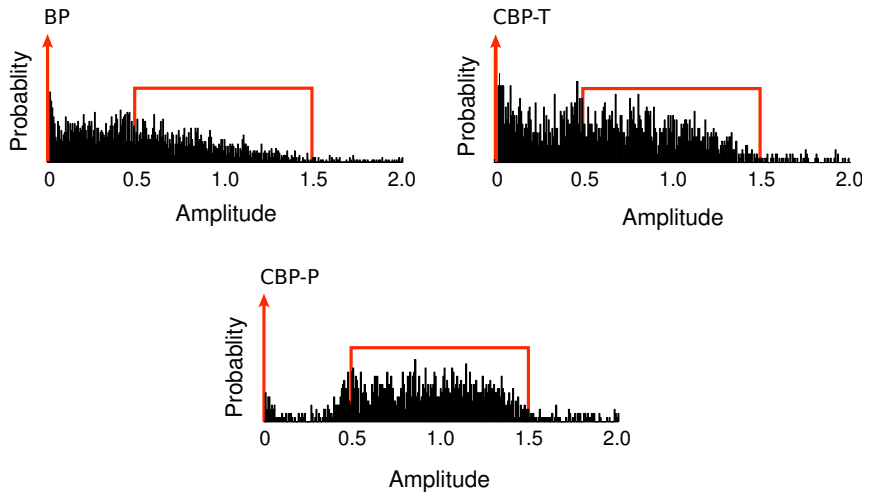


Figure 2.10: Histograms of the estimated amplitudes for BP, CBP-T, and CBP-P, respectively. All methods were constrained to estimate only nonnegative amplitudes, but no upper bound was imposed. The true distribution from which amplitudes were generated is indicated in red.

timated by each algorithm, compared with the true uniform distribution from which the amplitudes were generated. We see that CBP-P produces amplitude distributions that are far better-matched to the correct distribution of amplitudes.

2.8.2 Multiple features

For M sources, the source model becomes:

$$x(t) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} a_{ij} \left(f_i\right)_{\tau_{ij}} (t) + \epsilon(t), \tag{2.41}$$

All the methods we described can be extended to this case by taking as a dictionary the union of dictionaries associated with each individual waveform.

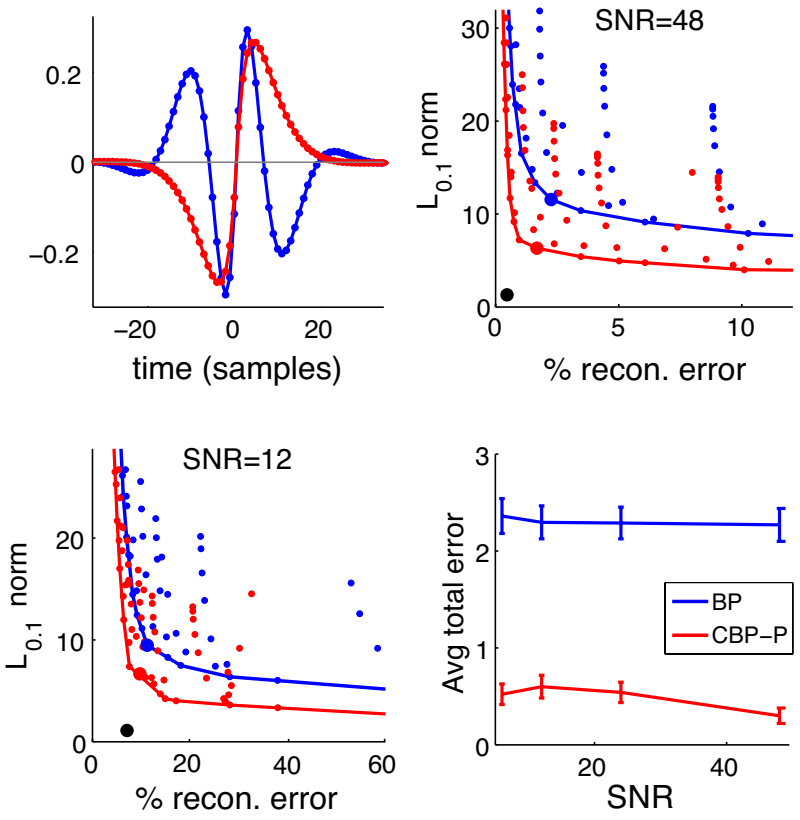


Figure 2.11: Sparse signal decomposition with two waveforms. Upper-left: Gammatone waveforms, $f_i(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi\omega_i t)$ for $i = 1, 2$. Upper-right and lower-left: Sparsity and reconstruction errors for BP (blue) and CBP-P (red), as in Fig. 2.8, with SNR's of 48 and 12, respectively (again, SNR is defined as $\|f\|_\infty/\sigma$). Lower right: total number of misses and false positives (with same thresholds as in bottom plot of Fig. 2.9) for each method.

We performed a final set of experiments using two "gammatone" features (shown in upper left of Fig. 2.11), which are commonly used in audio processing [103]. To generate a large amount of overlap in the data, event times were sampled from two correlated Poisson processes with the same marginal rate λ_0 and a correlation of $\rho = 0.5$. These were generated by independently sampling 2 Poisson process with rate $\lambda_0(1-\rho)$ and then superimposing a randomly jittered "common" Poisson process with rate $\lambda_0\rho$. As before, event amplitudes were drawn uniformly from the interval $[0.5, 1.5]$. We compared the performance of BP and CBP-P, in both cases using dictionaries formed from the union of dictionaries for each template with a common spacing, Δ, for both templates. In general, the spacing could be chosen differently for each waveform, providing more flexibility at the expense of additional parameters that must be calibrated. The upper right and lower left plots in Fig. 2.11 show the error tradeoff for different settings of (Δ, λ) at SNR levels of 48 and 12, respectively (the results were qualitatively unchanged for SNR values of 24 and 6). The lower right plot in Fig. 2.11 plots the total number of event identification errors (misses plus false positives) for each method as a function of SNR, at each method's optimal (Δ, λ) setting.

2.9 Summary and discussion

We have introduced a novel methodology that combines the advantages of sparse representations with those of modeling transformations. Our approach relies on a probabilistic source model in which features undergo transformations (of a known type, but unknown amount) before linearly

combining to form the observed signal, and can be interpreted as an approximate inference method for recovering the feature amplitudes and transformation amounts. Traditional methods construct a dictionary by discretely sampling the transformation manifold(s) and employ greedy or L_1-based recovery methods to solve its coefficients. These methods are limited by the tradeoff between the discretization error and the accuracy of the recovery methods. Our method addresses this limitation by employing a linear interpolation dictionary, with appropriately constrained coefficients, to represent the transformation manifold(s). The method can be seen as a continuous form of the well-known basis pursuit method, and we thus have dubbed it *continuous basis pursuit*. We showed, using both simple illustrative examples and large-scale simulations, that our method approximates the sparse linear inverse solution much more accurately (and across a wide range of noise levels) than basis pursuit when using simple first-order (Taylor) and second-order (polar) interpolation schemes. The resulting representations affords substantially better identification of events (fewer misses and false positives), and yields amplitudes whose distribution is well-matched to the source model. We conclude that this methodology provides a powerful and tractable tool for modeling and decomposing sparse translation-invariant signals.

There have been other attempts to solve the arrival-time recovery problem of Eq. 2.21 in addition to the conventional greedy and L_1-based sparse recovery methods described in Section 2.1. The field of array signal processing deals with direction-of-arrival (DOA) and time-delay estimation (see [70] for a review). However, these methods typically rely on the known geometry of the sensor array, whereas we address the prob-

lem in which we observe a sum of convolutions with arbitrarily-shaped kernels. In addition, several of these methods rely on spectral methods, taking advantage of the Fourier representation of translation, and thus do not generalize to other transformations. In [147], a general sampling theory was developed for a wide class of non-band-limited signals which includes streams of Dirac pulses. However, they focus on proving theoretical results when the convolution kernel is of a known analytic form (e.g., Gaussian or sinc function) and where the number of pulses is known.

Our method can be extended in various ways. We believe our method can be employed with transformations other than translation, such as dilation/frequency-modulation for acoustic signals (see Chapter 4), or rotation/dilation for images (e.g., [105]). Both the Taylor and polar basis constructions can be extended to account for multiple transformations. In the Taylor case, one only needs to add waveform derivatives with respect to each transformation and corresponding linear inequality constraints for their coefficients. In the polar case, one can model the (renormalized) transformation manifold locally as a 2D patch on the surface of a sphere, ellipsoid, or torus (instead of a 1D arc) which can be parametrized with two angles. In general, the primary hurdles for such extensions are to specify (1) the form of the linear interpolation (for joint variables, this might be done separably, or using a multi-dimensional interpolator), (2) the constraints on coefficients (and a convex relaxation of these constraints), and (3) a means of inverting the interpolator so as to obtain transformation parameters from recovered coefficients. Another natural extension is to use CBP in the context of learning optimal fea-

tures for decomposing an ensemble of signals, as has been previously done with BP (e.g., [120, 101, 135, 5, 7]), which we attempt to do in Chapter 3. Finally, one could learn a representation of the transformations present in the data instead of assuming a set of known transformations (e.g., [65, 56, 14]).

Several questions remain regarding the theoretical reasons underlying our approach's advantage. We know that the first-order Taylor interpolation accuracy depends on the curvature of the waveform. However, an analogous condition for measuring polar interpolation accuracy is lacking, although we expect there to be a Nyquist-like criteria relating bandwidth to accuracy (see Appendix A). In addition, it is unknown whether one can bound the difference between the solutions of Eq. 2.40 and Eq. 2.21. Most bounds on the approximation error of sparse linear inverse solutions rely on the coherence of the dictionary, which is generally lower for the cases we explored. However, the presence of the coefficient constraints prevents a straightforward application of previously used proof techniques to obtain bounds.

Another issue is the proper resolution of two or more events occurring within a time interval of size Δ, the spacing of the dictionary. Since we optimize with respect to a linear model with convex constraints, any nonnegative combination of events occurring within Δ of each other can be encoded in the coefficients corresponding to that bin. However, it is unclear how to resolve the individual events from these coefficients. Therefore, Δ must be chosen carefully, taking into account the event time statistics.

Nevertheless, we see our method as a significant step toward separat-

ing content from transformation in signals: one set of coefficients varies continuously in a known way as transformations are applied (in small amounts) to the signal, while the other set remains relatively invariant to transformation. Although we have described the use of an interpolator basis in the context of L_1-based recovery methods, we believe the same representations can be used to improve other recovery methods (e.g., greedy or iterative thresholding methods as described in Section 2.1.2-2.1.3). Furthermore, our polar approximation of a translational manifold can provide a substrate for new forms of sampling (e.g., [144, 147, 145]). By introducing a basis and constraints that explicitly model the local geometry of the manifold on which the signals lie, we expect our method to offer substantial improvements in many of the applications for which sparse signal decomposition has been found useful.

Chapter 3

Application to neural spike identification

In this chapter, we apply the continuous basis pursuit (CBP) method of Chapter 2 to the problem of identifying neural action potentials in extracellular recorded voltage data. This identification process, called "spike sorting," is a critical step in analyzing much of neural data, and is a computationally challenging problem for which several algorithms have been proposed. In Section 3.1, we describe the spike sorting problem in detail, review various existing solutions and discuss their advantages and disadvantages.

In Section 3.1.1, we propose a generative probabilistic model for the observed voltage trace and recast the spike sorting problem as a *maximum-a-posteriori* estimation of the spike times and amplitudes given the observed trace. Under the proposed model, the inference problem is very similar to the amplitude and time-shift recovery problem of Eq. 2.21 in Chapter 2, assuming we know the spike waveforms. We

therefore propose to solve the spike sorting problem using CBP (to infer spike times) within a dictionary learning scheme.

In Section 3.2, we apply our CBP-based method to four simulated data sets (single-channel) that have previously been used for evaluating spike sorting methods ([112, 149]), as well as two independent tetrode data sets described in [59] and [148] for which ground truth is available via simultaneous intracellular recordings. For all data sets, we compare our methods error rate (misses, false positives) with that of a standard procedure. On the simulated data of [112], we also compare our results with those reported by them. In all cases, our method outperforms the clustering-based methods. Using a technique introduced by [59], we also show that in almost all cases, our method outperforms the *best possible* clustering-based method that uses elliptical boundaries to classify spikes. The primary reason is the proper resolution of overlapping spikes, which clustering-based methods systematically fail to handle. Our CBP-based spike sorting solution offers several advantages over current spike sorters: (1) it is theoretically grounded in a probabilistic source model, and relies on fewer parameters, (2) it is able to accommodate non-Gaussian noise distributions, which frequently arise in experimental settings [118, 129, 44], (3) it handles real-valued spike times and is not susceptible to alignment and windowing artifacts, (4) it properly handles near-synchronous spikes, (5) it is highly automated (except for choosing the number of cells), and (6) its computational cost scales well for multiple electrodes (described in Section 3.2.4).

3.1 Background and previous work

The problem of detection, time-estimation, and cell classification of neural action potentials from extracellular electrode measurements is fundamental to experimental neuroscience. Electrode(s) are embedded in neural tissue, and a voltage trace is recorded as a function of time. When a neuron in the vicinity of the electrode fires an action potential, a stereotypical waveform is superimposed onto the recorded voltage [79, 119, 148]. The shape of this waveform depends on the cell's morphology and position, as well as the filtering properties of the medium and the electrode(s). The "spike sorting" problem consists of detecting the occurrence of these individual waveforms and estimate their corresponding arrival times.

Despite the ubiquity and succinct formulation of the problem, there is no *de facto* standard for spike sorting. Many experimentalists manually position a single electrode and define threshold triggers to identify the spikes of individual cells [116]. However, this becomes substantially more difficult when recording from several cells simultaneously, and is infeasible for multi-electrode arrays. Computer-assisted solutions have converged on a general methodology that we will refer to as "clustering," consisting of three steps [79], illustrated in Fig. 3.1: (1) detection of temporal segments of the voltage trace that are likely to contain spikes, (2) estimation of a set of features for each segment, and (3) classification of the segments according to these features. A variety of methods exist for solving each step (e.g., (1) thresholding based on absolute value [100] , squared values [117], Teager energy [21], or other nonlinear operators

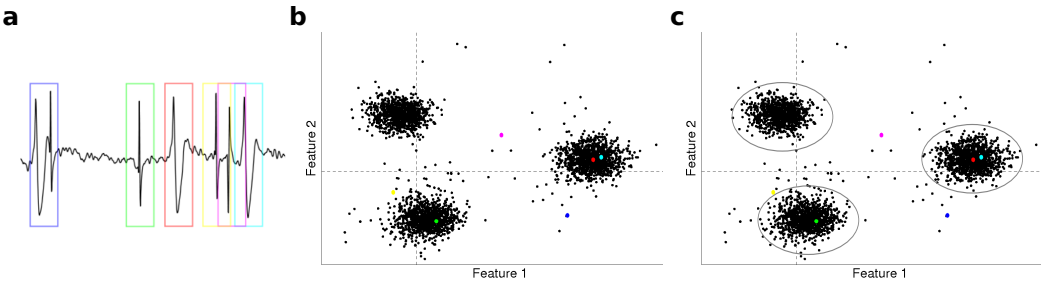


Figure 3.1: Schematic of 3-step procedure common to most current spike sorting methods. (**a**) Thresholding/windowing. Times of voltage peaks are estimated by crossings of a chosen threshold. The temporal segments of the voltage trace that lie within a fixed-duration window around each peak (colored rectangles) are gathered. (**b**) Feature estimation. Feature values are determined for each segment. As a typical example, we plot the projection of each segment onto the first two principal components of the full set of segments. Colored points in this plot correspond to the windowed segments of corresponding color in (**a**). (**c**) Classification. Segments are grouped within the feature space, typically using an automatic clustering method such as K-means or estimation of a Gaussian mixture model.

[115], (2) features such as peak-to-peak width/amplitude, projections onto principal components [79], or wavelet coefficients [112, 71], and (3) classification methods such as K-means [79], mixture models [118, 129], or superparamagnetic methods [112]). Although methods exist for solving each of the three steps in isolation, it is unclear how to tie the sequential application of these steps directly to the optimization of a single objective, making it difficult to state the assumptions and operating conditions needed for success. Since each successive step does not take into

account errors introduced by previous steps, errors tend to accumulate. In addition, many of these methods require human supervision (especially for the classification step), which is not only costly, but generally inaccurate [59] and highly variable [154]. The lack of a standard automated methodology makes it difficult to compare results of scientific studies.

Most importantly, the conventional three-step procedure mishandles overlapping spikes. If two or more cells fire near-synchronously, their respective waveforms are superimposed in the voltage trace, creating a shape that differs from either spike in isolation [78, 119, 148, 106]. If the waveform shapes partially cancel, the initial detection stage may miss the spikes altogether. Even if the segment is detected, its appearance will depend on the time delay between the two spikes [106]. If this is significantly different from that of either spike in isolation, it will be misidentified as a fictitious third cell or discarded as an outlier, as illustrated by the spikes outlined in blue, yellow, and pink in Fig. 3.1.

Failure to resolve overlapping spikes can have serious consequences: Even basic measurements, such as mean firing rates and cross-correlations, can be heavily biased due to spike sorting artifacts [4, 104, 106]. Properly handling this bias is crucial when studying a region where there is a high level of synchronous activity or when the study itself focuses on the correlation of firing patterns [89, 33, 121, 128, 107]. Such studies are more frequent with the advent of multi-electrode array recordings, which allow the simultaneous recording of large populations of neurons [91, 52, 11, 107, 127].

There have been several proposed methods to augment the cluster-

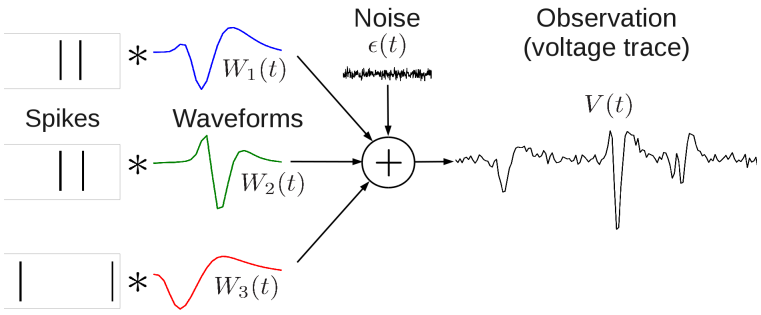


Figure 3.2: Illustration of the measurement model. Each cell generates a voltage trace containing time-shifted copies of its spike waveform, and the observed voltage trace is assumed to be a sum of these and a noise variable.

ing approach to account for overlapping spikes [3, 78, 125, 156, 146, 111, 106]). However, these methods generally rely on brute-force examination of all combinations of spike waveforms at all time separations (impractical for simultaneous recordings of many cells), or use of "greedy algorithms that iteratively subtract the waveform of the best-fitting cell until the residual amplitude is within the range expected for noise. A notable exception is the family of ICA-based spike sorting methods [138, 139, 49], which bear some resemblance to our approach, but have not been developed or implemented in the context of a unified probabilistic model for the voltage measurements, and have not been extensively tested and compared to traditional clustering methods.

3.1.1 CBP approach

Our method is derived from a simple generative model for the observed voltage trace ([118, 106]), as illustrated in Fig. 3.2. A spike from the

nth neuron, occurring at time $\{\tau_{ni}\}$, is assumed to produce a temporally localized waveform $a_{ni}W_n(t-\tau_{ni})$, where $W_n(t)$ has unit norm, and a_{ni} represents the spike amplitude. These time-shifted and scaled waveforms are then added together with noise to form the electrode voltage trace:

$$V(t) = \sum_{n=1}^{N} \sum_{i=1}^{C_n} a_{ni} W_n(t-\tau_{ni}) + \epsilon(t) \tag{3.1}$$

In the case of multi-electrode recordings, $V(t)$ and $W_n(t)$ are vector-valued with as many dimensions as electrodes, but for notational convenience, the derivation below is written for the scalar case. The distribution of the noise, $\epsilon(t)$, is assumed to be log-concave, which leads to a tractable optimization algorithm, while allowing for Gaussian or more heavy-tailed distributions (e.g., Laplacian or power-law) that arise in many experimental settings [118, 129, 44]. For the data sets analyzed in this thesis, we found that a Gaussian distribution performed well, so we restrict ourselves to that case from here on. Our approach can also handle correlated noise, but to simplify the derivation, we assume any such correlations have been removed through a pre-processing step (see Section B.0.8). Given these assumptions, the spike sorting problem is to recover $\{W_n(t)\}_{n=1}^{N}, \{\tau_{ni}\}_{i=1}^{C_n}, \{a_{ni}\}_{i=1}^{C_n}$ given only $V(t)$.

3.1.2 Maximum-a-posteriori estimation.

Note that Eq. 3.1 is identical to Eq. 2.21, with the only difference being that waveform $\{W_n(t)\}$ shapes are unknown and must be optimized along with the spike times/amplitudes. Therefore, under the Gaussian noise

assumption, the *maximum-a-posteriori* (MAP) objective is:

$$\underset{\{a_{ni}\},\{\tau_{ni}\},\{W_n(t)\}}{\arg\min} \frac{1}{2}\|V(t) - \sum_{n,i} a_{ni} W_n(t - \tau_{ni})\|_2^2 \tag{3.2}$$

$$- \log P(\{a_{ni}\}, \{\tau_{ni}\}) \tag{3.3}$$

Note that the waveforms are treated as model parameters since we do not model them with a prior probability distribution, although such *a priori* information could easily be incorporated if it was available. This optimization problem partitions naturally into two subproblems: solving for the waveform shapes, $\{W_n(t)\}$, and solving for the spike amplitudes and times, $\{a_{ni}, \tau_{ni}\}$. Thus, we use a "coordinate descent algorithm, alternating between solving for each of these two subsets of parameters while holding the other subset fixed. Such methods have often been used when adapting over-complete dictionaries to signal ensembles [101, 82, 40, 84].

Initialization of waveform shapes. We initialized the waveforms $\{W_n(t)\}$ using K-means clustering (see Section B.0.7). Note that unlike clustering algorithms for spike sorting, we do *not* use the cluster assignments to identify spikes. Rather, we use the cluster *centroids* to initialize the waveform shapes. These estimates are typically quite accurate, as long as each cell produces a substantial number of isolated spikes. As such, we find in practice that the coordinate descent iterations (alternating between solving for waveforms and spike times/amplitudes) offers only minor improvement (see Fig. 3.5(c)).

Solving spikes given waveforms: CBP Given the waveforms, solving directly for the spike times/amplitudes is exactly the sparse recovery problem for which the CBP method (Chapter 2) was designed. Adopting this approach yields the following objective:

$$\min_{\vec{\alpha}\in\mathcal{H},\{W_n(t)\}} \frac{1}{2\sigma^2}\|V(t) - (\mathbf{D}_{W,\Delta}\vec{\alpha})(t)\|_2^2 - \sum_i \log P_A(\alpha_{i1}) \tag{3.4}$$

where $\mathbf{D}_{W,\Delta}$ is a dictionary constructed from the waveform shapes $\{W_n(t)\}$ using a polar interpolator and replicating them with spacing Δ, and $\mathcal{H}$ is the second-order cone constraint set described in Section 2.6.1 of Chapter 2. For spike sorting, the desired prior is strictly binary, since we want to force amplitudes to be either 0 (no spike) or 1 (spike). Unfortunately, this binary distribution, like the L_0 prior of Eq. 2.5, is discontinuous and makes the problem intractable. In Chapter 2, the L_0 penalty was relaxed to an L_1 penalty. However, we found that for this application it was better to adopt a prior of the form

$$P_A(a) \propto (\eta + a)^{-p} \Rightarrow -\log P_A(a) = p\log(\eta + a) + \text{const} \tag{3.5}$$

The values of p and η were set to 10 and 10^{-16}, respectively. This prior encourages much sparser solutions than the L_1 penalty, at the cost of losing log-concavity. However, we can adopt an iterative reweighting L_1 minimization scheme [13] to approximate the solution of Eq. 3.4. The idea is to solve a series of convex optimization problems, where in each iteration the log prior term in Eq. 3.4 is approximated with its first-order

Taylor expansion about the previously estimated amplitudes:

$$p\log(\eta+\alpha) \approx p\log(\eta+\alpha_{ni1}^{(k)}) \tag{3.6}$$

$$+ p\left[\frac{d}{dx}\log(\eta+\alpha)\right]_{\alpha=\alpha_{ni}^{(k)}}\left(\alpha-\alpha_{ni}^{(k)})\right) \tag{3.7}$$

$$= \left(\frac{p}{\eta+\alpha_{ni}^{(k)}}\right)\alpha + \text{const} \tag{3.8}$$

Thus, we can solve for $\vec{\alpha}$ by initializing weights $\lambda_{ni}^{(0)} = \lambda_0$ and then iteratively optimizing:

$$\vec{\alpha}^{(t+1)} \leftarrow \arg\min_{\vec{\alpha}\in\mathcal{H}} \frac{1}{2}\|V(t) - \mathbf{D}_{W,\Delta}\vec{\alpha}\|_2^2 + \sum_{n,i}\lambda_{ni}^{(k)}\alpha_{ni1} \tag{3.9}$$

$$\lambda_{ni}^{(k+1)} \leftarrow \frac{1}{\eta+\alpha_{ni1}^{(k+1)}} \qquad \forall n,i \tag{3.10}$$

until convergence. Each iteration amounts amounts to solving a weighted version of the original CBP problem of Eq. 2.40 which is still convex and can be solved efficiently. Note that under this reweighting scheme, small weights induce high weights in the next iteration, and are therefore pushed to zero after a very small number of iterations.

To improve computational efficiency, the voltage trace was partitioned into non-overlapping excerpts separated by intervals of silence, and each excerpt was processed independently. Silences were defined as intervals with duration longer than half the minimal waveform duration (approximately 2ms) in which the voltage trace did not exceed the threshold in Eq. (B.0.7). Each excerpt was tested for whether it could be explained with a single isolated waveform placed at any time. If the energy of the residual obtained by subtracting the optimally placed waveform was less than a fixed percentile p of a chi-squared distribution (the distribution of

the noise energy), then this single-waveform explanation for the interval was accepted ($p = 99.999$, chosen based on the spike waveform amplitudes relative to the noise level). Otherwise, the interval was processed according to the procedure described above.

Thresholding of spike amplitudes. Since the spike amplitudes inferred from the solution of Eq. 3.4 can take on any nonnegative value, a threshold must be used for final spike identification. This threshold value determines the tradeoff between missed spikes and false positives, which is a choice best left up to the investigator, who can assess the relative costs of the two types of error with regards to the scientific goals of the experiment. For the purposes of providing a simple automated choice, however, we compute a smoothed estimate (Gaussian kernel density estimator [9]) of the spike amplitude density and identify the largest value at which the density has a local minimum. The red vertical lines in Fig. 3.7 indicate this automatically computed threshold. Note that if there are multiple cells with similar-shaped waveforms but different amplitudes, the spike amplitude distribution will be multimodal, and multiple thresholds should be chosen.

Solving waveform shapes given spikes Once we solve the spike times and amplitudes, we can optionally go back and solve for the optimal waveform shapes (or take a gradient step with respect to the quadratic data reconstruction term). For most of our data except in one case, this was typically unnecessary since the initial waveforms were sufficiently accurate. Waveform shapes were updated according to a simple gradient

step, followed by a renormalization in order to prevent the redundant tradeoff between waveform and spike amplitudes. The dictionary $\mathbf{D}_{W,\Delta}$ can be expressed as a linear function of the waveforms:

$$\begin{aligned}\mathbf{D}_{W,\Delta} &= \left(\mathbf{D}_{W_1,\Delta} \,\middle|\, ... \,\middle|\, \mathbf{D}_{W_N,\Delta} \right) \\ &= \left(\text{conv}_\Delta(\mathbf{P_1}\vec{w}_1) \,\middle|\, ... \,\middle|\, \text{conv}_\Delta(\mathbf{P_N}\vec{w}_N) \right)\end{aligned} \tag{3.11}$$

where $\vec{w}_n$ is the n'th waveform (represented as a vector of samples), and the matrices $\mathbf{P_n}$ convert them into the $\vec{c}, \vec{u}, \vec{v}$ representation (see Eq. A.4). The operator conv_Δ produces a convolutional matrix by replicating each column of its input at a spacing Δ. We can then express the signal reconstruction, $\mathbf{D}_W\vec{\alpha}$, as a linear function of the vectorized waveforms:

$$\begin{aligned}\mathbf{D}_W\vec{\alpha} &= \left(\text{conv}_\Delta(\mathbf{P_1}\vec{w}_1) \,\middle|\, ... \,\middle|\, \text{conv}_\Delta(\mathbf{P_N}\vec{w}_N) \right) \begin{pmatrix} \vec{\alpha}_1 \\ ... \\ \vec{\alpha}_N \end{pmatrix} \\ &= \begin{pmatrix} \overline{\text{conv}_\Delta}(\vec{\alpha}_1)\mathbf{P_1} & 0 & 0 \\ 0 & ... & 0 \\ 0 & 0 & \overline{\text{conv}_\Delta}(\vec{\alpha}_N)\mathbf{P_N} \end{pmatrix} \begin{pmatrix} \vec{w}_1 \\ ... \\ \vec{w}_N \end{pmatrix} \\ &= \mathbf{S}\vec{w}\end{aligned} \tag{3.12}$$

Here, $\overline{\text{conv}_\Delta}$ denotes reverse convolution. Ignoring terms that are constant with respect to the waveform shapes, we can express the objective as:

$$f(\vec{w}) = \frac{1}{2\sigma^2}\|\vec{v} - \mathbf{S}\vec{w}\|_2^2 \tag{3.13}$$

where $\vec{v}$ is the voltage trace sampled at the same resolution as the waveforms. Since this is simply a quadratic function of the waveform shapes,

we can easily apply a gradient update (with appropriate renormalization) of the following form:

$$K_n \leftarrow \|\vec{w}_n\|_2 \tag{3.14}$$

$$\vec{w} \leftarrow \vec{w} + \eta \frac{\mathbf{S}^T(\vec{v} - \mathbf{S}\vec{w})}{\sigma^2} \tag{3.15}$$

$$\vec{w}_n \leftarrow \frac{\vec{w}_n}{\|\vec{w}_n\|_2} K_n \tag{3.16}$$

Note that the $\mathbf{P_n}$ are computed involving the radius and angular constants (r, θ) associated with each waveform shape $\vec{w}_n$ (Appendix A). Therefore, $\mathbf{S}\vec{w}$ may deviate from a sum of polar-interpolated waveforms. However, since we make only small changes in the waveform shape, recomputing the radius and angular constants on each iteration, we assume that this deviation is negligible.

3.2 Results

3.2.1 Simulated data

We first apply our method to four simulated data sets [112], each containing spiking activity from three neurons with background noise at four different levels. Waveforms (shown in Fig. 3.3(a-d)) were taken from real recordings, and noise was constructed to reflect realistic background activity. Excerpts of voltage traces for two different noise levels are shown in Fig. 3.3(e-f).

The top row of Fig. 3.4 compares the spikes sorted by our method to those arising from standard clustering for one of the data sets (corresponding waveforms shown in Fig. 3.3(a)), plotted in the space of the

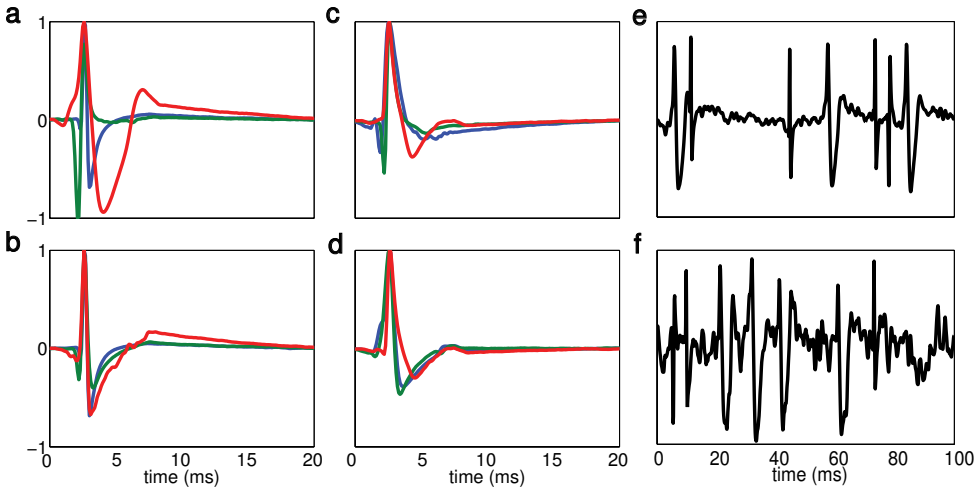


Figure 3.3: Examples of simulated electrode data [112]. (**a-d**) Each panel shows spike waveforms of three distinct cells, used to generate the simulated data sets. **(e-f)** Example simulated voltage trace using waveforms in (**a**) for noise levels $\sigma = 0.1$ and $\sigma = 0.2$, respectively.

first two principal components. Although clustering correctly identifies the majority of spikes, it misses a substantial subset that are distant from the cluster centers. Our method correctly recovers nearly all of these missed spikes. The inset graphs of Fig. 3.4(b) demonstrate that the corresponding voltage snippets do indeed contain superpositions of multiple spikes.

Figure 3.5(a) compares the total number of errors of our method with three other methods: (1) standard clustering using PCA and K-means (Section B.0.7); (2) superparamagnetic clustering [112]; (3) the *best ellipsoid error rate* (BEER) measure [59]. Note that the BEER is not an actual spike sorting method – its parameters are adjusted to optimize performance on data for which the true spikes are known –

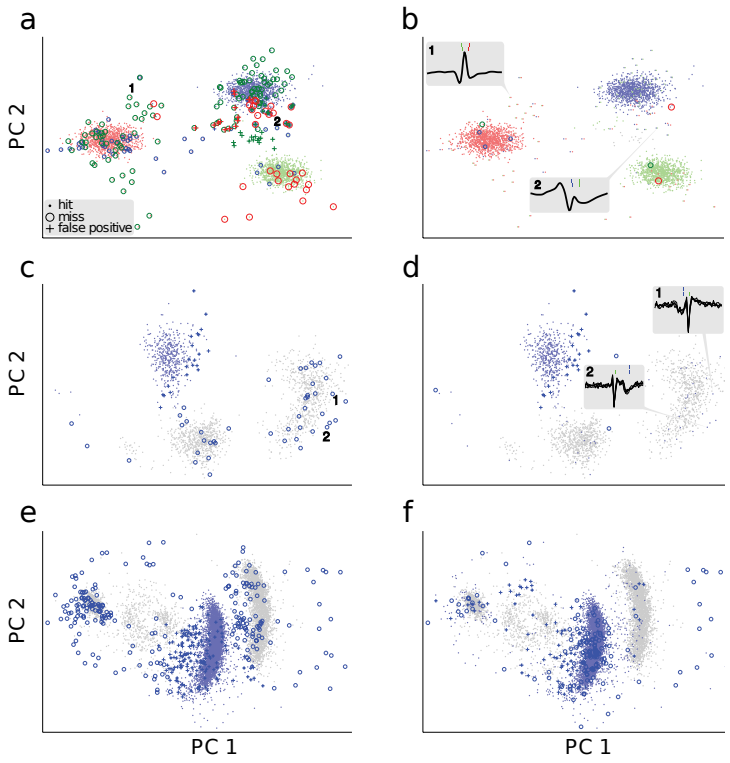


Figure 3.4: Spike sorting results for simulated data [112] (top row) using waveforms in Fig. 3.3(a), and tetrode data [59, 148] (bottom and middle rows). **(a,c,e):** Spike identification by standard clustering (Section B.0.7). Each marker represents the projection of a voltage segment onto the leading 2 principal components. Points, circles, and crosses represent hits, missed spikes, and false positives, respectively. Color indicates cell identity. Gray points (bottom two rows) correspond to segments of real data for which no ground truth is available. **(b,d,f):** Spike identification by our method, represented in the same space as **(a,c,e)**. Insets show example voltage segments containing overlapping spikes (corresponding to numbered points in **(a,c)**) in the time domain. Colored vertical lines in the insets represent the occurrence times of ground truth spikes (top row) and spikes estimated by our method (bottom row).

but serves instead as a bound on the class of clustering-based methods that use elliptical boundaries (see Section B.0.9 for details). Our method substantially outperforms the three other methods under all conditions, except for the highest noise level of the fourth data set (for which BEER has the best performance).

3.2.2 Tetrode data from rat hippocampus (Harris, 2000)

We also applied our method to a portion of publicly-available data, recorded from CA1 in anesthetized rat hippocampus [59]. The data include simultaneous recordings from an extracellular tetrode, and an intracellular electrode that was used to obtain the actual spike times (so-called ground truth) for a single cell. Figure 3.6 shows an excerpt of the tetrode recording, with the scaled intracellular trace superimposed in gray. There are two prominent waveforms appearing in the recording, the smaller of which corresponds to the intracellularly recorded cell.

Figure 3.4(c) illustrates the performance for standard clustering (Section B.0.7). Notice that, unlike the simulated data of Fig. 3.4(a), there are many segments that presumably contain spikes but for which no ground truth is available (gray dots). For the intracellularly recorded cell, the majority of isolated spikes are correctly identified (blue dots), but a substantial number of other spikes are missed (blue circles) because they are far from the cluster center in the feature space. The majority of these missed spikes are recovered by our method, as illustrated in Fig. 3.4(d). The insets of Fig. 3.4(d) demonstrate that, as with the sim-

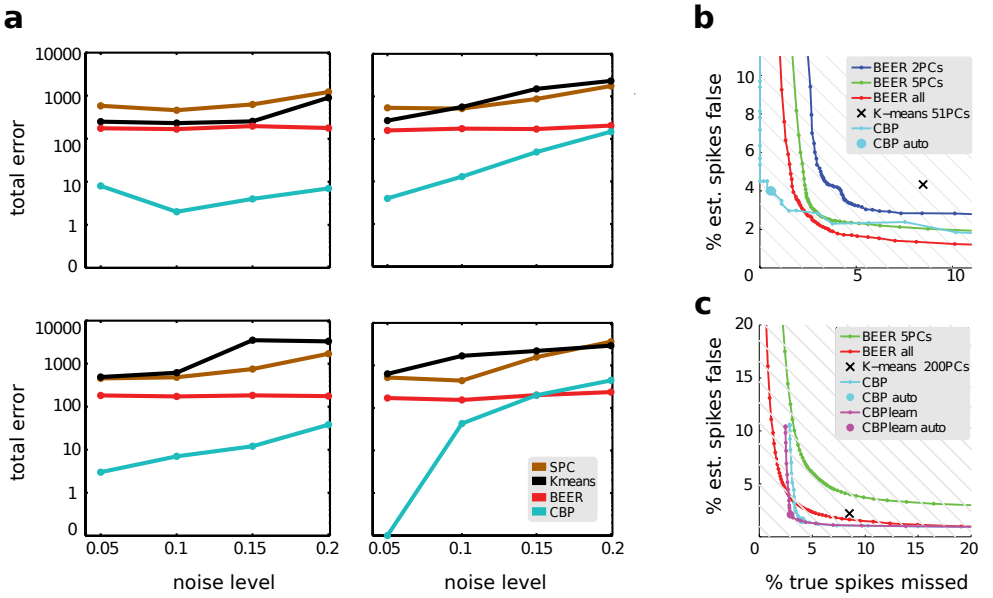


Figure 3.5: Spike sorting performance comparison. (**a**) Each panel shows the total sorting errors (as a function of noise level) for each of the 4 simulated data sets [112] incurred by standard clustering (black, Section B.0.7), superparamagnetic clustering [112] (brown), BEER (red, Section B.0.9), and our method (cyan, Section 3.1.1). For all four examples, a fixed threshold of 0.5 was used to identify spikes in our method. (**b,c**) Tradeoff between "false positive and "miss errors on each of the tetrode data sets [59, 148], respectively, as the assignment probability threshold is varied for the BEER (blue/green/red curves) and as the spike coefficient threshold is varied for our method (cyan curve, magenta curve with waveform learning). In (**b**), waveform learning did not significantly improve performance with CBP, and so the corresponding curve is not shown. Large points indicate automatically chosen thresholds (see Section 3.1.1). The black X indicates the performance of standard clustering (Section B.0.7). Diagonal gray lines indicate contours of constant total error.

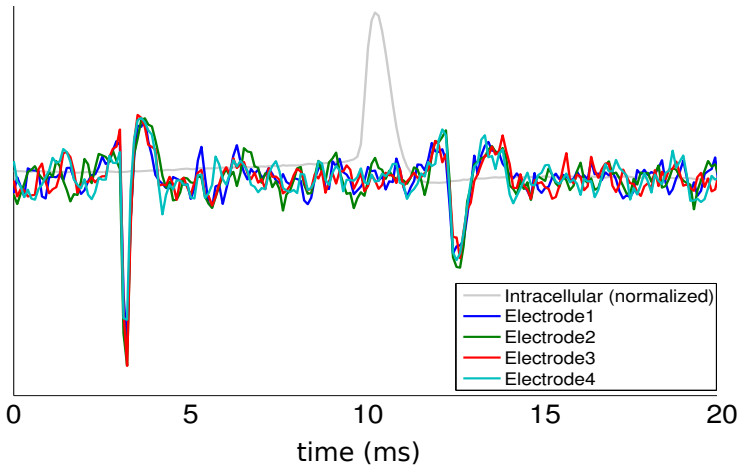


Figure 3.6: Example portion of tetrode data [59]. The four filtered extracellular electrode responses (colored traces) indicate two prominent spikes of different shape. Also shown is the intracellular response of a single cell (gray trace, vertically re-scaled for visualization) for a single cell, corresponding to the second of the extracellular events. The intracellular spikes are easily and unambiguously identifiable, and typically precede the extracellular spike waveform by approximately 2 ms.

ulated data, these missed spikes typically overlap with the waveform of another spike.

Figure 3.7 shows the distribution of spike amplitudes for three waveforms obtained by our method. The second waveform corresponds to the ground truth cell, and the red line indicates an automatically chosen threshold, based on the procedure described in Section 3.1.1. Notice that all waveforms also have a significant amount of low-amplitude activity, unlike the simulated case. The isolation of a group of spike amplitudes relative to noise and the amplitudes of other cells with similar wave-

form shapes provides an informal indication that the activity originates from a single cell. For example the higher-amplitude modes in the first two distributions are clearly isolated and most likely correspond to the activity of two distinct cells. The lower-amplitude modes in these two distributions are likely due to background spikes. On the other hand, the two modes of the third distribution are not well separated, and any choice of threshold is likely to result in a substantial number of errors. As in any signal detection problem, a criterion (threshold) can be used to finally decide whether a given portion of the voltage trace contains a spike or not. Changes in the choice of threshold will trade off the number of false positives and misses. A simple automatic procedure can be used to select a threshold, but the correct choice ultimately depends on the costs of the two types of error, which can only be specified by the investigator.

Figure 3.5(b) shows the tradeoff between misses and false positives incurred by clustering, the BEER measure, and our method. The error curves for the BEER measure were computed by varying the threshold on the class assignment probabilities computed with a quadratic classifier (see Section B.0.9), while the error curve for our method is formed by varying the spike amplitude threshold.

3.2.3 Tetrode data in locust (Wehr, 1999)

We also applied our method to data, recorded from locust *in vivo* [148]. We applied the same analysis as in Section 3.2.2. Figs 3.4(e-f) visualize the spike sorting results for standard clustering and our method, while

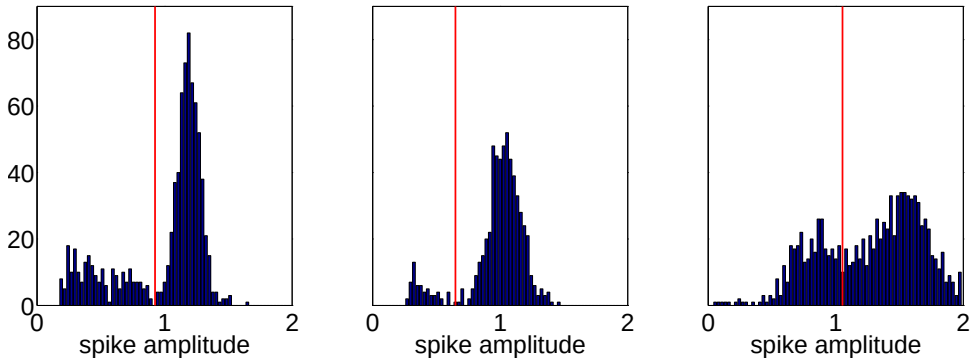


Figure 3.7: Histograms of spike amplitudes for the three neurons corresponding to the waveforms estimated by our method. The middle plot corresponds to the cell for which ground truth is available via intracellular recordings. The red vertical line indicates an automatic threshold for identifying spikes, based on the procedure described in Section 3.1.1, and was used to obtain the error rates corresponding to the large cyan dot in Fig. 3.5(b).

Fig. 3.5(c) compares our methods performance with that of standard clustering and the BEER. The ground truth cell is not as well isolated as in the rat data set, resulting in a higher error rate for all methods. Despite this, our method again outperforms both clustering and the BEER measure, assuming equal weighting of misses and false positives. However, Fig. 3.4(f) indicates that our method misses some spikes that are correctly identified by clustering. We attribute this to the properties of the particular waveforms in this data set. Specifically, the voltage traces appear to contain spike waveforms of two neurons with very similar shapes, but different amplitudes. Our model allows spikes of any positive amplitude, but the objective function (Eq. 3.4) imposes a penalty

that increases with amplitude. Therefore, our method prefers to explain a small-amplitude event in the voltage trace as a small-amplitude spike of a large waveform, rather than a large-amplitude spike of a similarly-shaped small waveform. In the example of Fig. 3.4(f), the two neurons have slightly different waveform shapes, and the number of misses can be reduced slightly by iterating between refining the waveform shapes and estimating the spikes (see Section 3.1.1), as indicated by the magenta line in Fig. 3.5(c).

3.2.4 Algorithm complexity

The optimization of Eq. 3.4 was implemented using the `CVX` package [54], a reasonably efficient and highly accurate package for convex optimization. We examined the computational costs of our algorithm as a function of three parameters: (1) the number of time samples in the voltage trace, (2) the number of distinct waveforms (neurons), and (3) the number of channels (electrodes). In practice, we can split the voltage trace $V(t)$ whenever there is a period without any spiking activity, and process the portions between these silences independently. As such, the first parameter is specified not in terms of the experiment length, but rather in terms of the typical duration between silences, which depends only on the firing rates. Figure 3.8 shows the execution time for the algorithm as a function of each of these parameters, while keeping the other two fixed at typical values. The computation time grows approximately quadratically with the number of cells, and linearly with the temporal duration and number of electrodes. The last of these implies that the

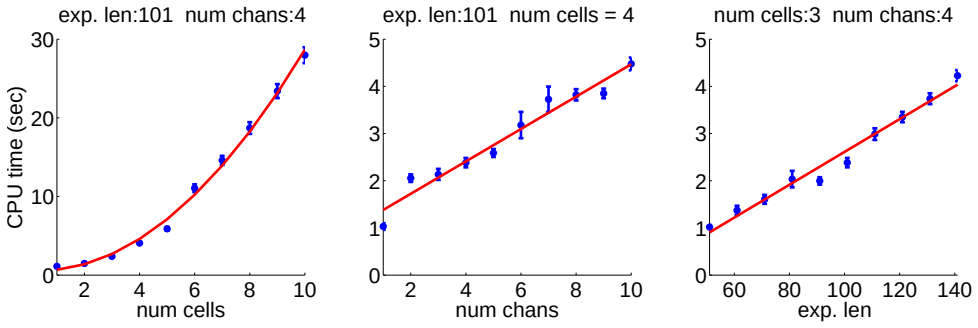


Figure 3.8: The effect on computation time of three variables: the number of cells C, the number of electrodes E and the number of voltage samples T. Computation time is plotted as a function of **a** C with $T = 100, E = 4$, **b** E with $T = 100, C = 3$, and **c** T with $C = 3, E = 4$. Blue points represent mean computation time taken over 25 trials, and the error bars indicate standard error. The red line indicates a quadratic fit (**a**) or a linear fit (**b,c**).

algorithm scales efficiently for multi-electrode arrays.

3.3 Summary and discussion

Starting from a simple probabilistic model for extracellular voltage measurements, we have developed a unified sparse estimation methodology for spike sorting. We have shown that, on simulated and real data sets taken from recent literature, this method is much more accurate than clustering-based methods for resolving overlapping spikes (and equally accurate when resolving isolated spikes). By comparing performance with the BEER bound, we have also shown that our method outperforms the entire class of clustering methods which use elliptical boundaries,

which encompasses the vast majority of spike sorting methods currently in use. Finally, our method is well-suited for sorting multiple cells and/or multiple electrodes, as it is reasonably efficient and requires no human intervention except for selecting the number of waveforms. Previous methods have probabilistically modeled the number of cells ([78, 153]), and such techniques could be incorporated into our method. However, choosing the number of cells, as with the spike amplitude thresholds, is a decision that must weight the different types of error, and so is perhaps best left to the investigator.

Despite the strong performance of the method, we see several opportunities for improvement. First, the current algorithm assumes that all spike waveforms can be differentiated by their *shape*, and cannot handle multiple spike waveforms that have the same shape (see Fig. 3.4(f), and section 3.2.3). A more general solution should separate waveforms according to both their shape *and* amplitude similarity, by including a step that is able to identify and partition multimodal amplitude distributions. Second, the method also assumes that waveform shapes remain constant throughout a recording session. However, it is well-known that tissue relaxation, electrode drift, and bursting activity, amongst other factors, can cause the waveform amplitude or shape associated with a single cell to change over time. This can be partially overcome by re-estimating the waveform shapes in consecutive chunks of the data [49]. A more unified solution requires incorporation of the drift in shape or amplitude into the probabilistic model. For example, the waveform shapes could be estimated as a function of time by modeling the dynamics as a stationary process (e.g., using Markov chain methods [110, 15]). Finally,

our method was implemented as a "batch algorithm, operating simultaneously on a full data set. It should be possible to develop an on-line version, that can operate on the voltage measurements as they arrive.

Although the probabilistic framework underlying this work has been described in a number of previous publications [118, 107, 106], very few spike sorting methods make direct use of it, primarily because of the difficulty of solving for spike times with respect to the linear time-shift model. Instead, most spike sorting methods are implemented as a sequence of procedural steps, each relying on additional free parameters and/or substantial human supervision, and each introducing additional sources of error. By overcoming the technical challenges associated with spike inference, our results demonstrate the potential advantages of a unified probabilistic framework, providing a base on which future spike sorting methods may be built, and facilitating the objective comparison of their performance.

Chapter 4

Hierarchical spike coding of sounds

4.1 Review of hierarchical signal modeling

[1] Recall that the underlying motivation for the continuous basis pursuit method was to "factor out" the nonlinear structure in the data distribution that is due to transformation-invariance. The method proposed in Chapter 2 decomposes a signal in terms of a set of atomic feature instances and associated transformation parameters (e.g., time-shifts). In many complex signal ensembles, however, these linear decompositions will still possess a high degree of nonlinear structure.

It is well-known, for instance, that the components of image and sound representations (e.g., sparse coding models like those described in Chapter 2) exhibit complex and nonlinear dependencies, despite the frequent *a priori* assumptions that these components are statistically in-

[1] The work described in this chapter was done in collaboration with Yan Karklin.

dependent. These dependencies can originate from the statistical structure inherent to the signal distribution and/or the dependencies between the dictionary elements (e.g., filter outputs can be correlated even when applied to unstructured data). These dependencies often involve higher moments, thus being present even if the components appear statistically uncorrelated. For example, the variance of an oriented Gabor filter output across natural image patches can change as a function of the output of an adjacent filter output (although the mean is constantly 0) [123]. An analogous observation has been made for the output of Gammatone filterbanks applied to natural sounds [124, 123]. Statistical dependencies within local neighborhoods of wavelet coefficients of natural images (using an over-complete multiscale oriented wavelet dictionary) have been modeled using bivariate Gaussians [126] and Gaussian scale mixtures [109]. Variance modulation across these neighborhoods was itself modeled as a sparse combination of "density components" in a two-layer model of natural images [66]. In [83], it was shown that a nonlinear radial operator applied to filter outputs can bring the representations closer to the factorial representations with which they are modeled.

There have been numerous attempts to successively decompose statistical dependencies in the data through hierarchical processing. When employing (sparse or non-sparse) linear models, it is important to note that a naive "stacking" of such models will not yield any additional expressive power over a single-layer model (at least in a generative sense). Therefore, some nonlinearity must be introduced to extend sparse linear models hierarchically in a non-trivial manner. Multilayer neural networks, which successively apply a linear transform followed by a point-

wise sigmoidal nonlinearity [74, 38], have long been utilized toward this end. However, such networks contain many free parameters and are difficult to train because of local minima. A recent surge of effort has focused on deep belief networks (DBN's) which recast the operations used by traditional neural networks within a probabilistically sound framework which supports efficient learning and inference [63, 6]. Such models provide representations that have achieved state-of-the-art performance on digit recognition [64] and motion recognition [140]. Convolutional deep belief networks [73, 77, 67, 113, 76] replicate weights across many spatial locations and have the advantage of being able to represent larger-scale images with the same number of learned parameters. Several models incorporate max-pooling or divisive normalization operations at each layer, which significantly improves classification performance [76] and has been shown to further reduce higher-order statistical dependencies ([123, 83]).

None of these models, however, explicitly factor "what" and "where" information in a probabilistically sound manner. As discussed in Chapter 1, making the features convolutional does not rectify this problem, since the corresponding source model is inaccurate. Systematic dependencies will be introduced due to the convolutional structure of each layer. These dependencies will be difficult to distinguish from the dependencies inherent in the data. Secondly, "where" information is successively discarded as the data travels up the hierarchy, due to the pooling and normalization operations that are performed at each layer (which are usually hard-coded into the architecture). As a result, these models typically perform well on, and are almost always tested on, tasks such as object classification which require a high degree of invariance. Since

the relationship between the model parameters at the highest layers and encapsulated structure in the signal domain is poorly understood, it is unclear whether these same models will generalize to other tasks which require more precise "where" information.

In this chapter, we introduce a hierarchical model in which each layer encodes the signal using a sparse, spike-based representation. Each spike has an associated kernel (or feature) which induces a pattern of spiking activity in the layer below (or a continuous pattern if the layer below is the signal itself), an amplitude, and a set of transformation parameters associated with it (e.g., spatio-temporal offset, dilation/frequency, etc.). The model bears many resemblances to convolutional deep belief networks [73, 76], but with different nonlinearities and noise models at each stage. Another key difference is that the structure of spiking activity within each layer is modeled as coming from two sources: (1) a hierarchical component coming from the spike representation in the layer above, which captures coarser-scale, non-stationary structure inherent to the data, and (2) a recurrent component which captures fine-scale stationary structure that presumably arises from the inherent structure in the layer's dictionary. Both the hierarchical and recurrent components multiplicatively modulate the probability of spiking within the layer.

We develop our approach in the context of modeling sound signals, which serve as an ideal testbed for this type of modeling since they exhibit hierarchical structure at multiple scales. A segment of recorded speech, for example, can be decomposed as a sequence of words occurring at specific times. Each word can be decomposed into phonemes, which can be further decomposed into simpler acoustic events and so

on. Other sounds used for communication, such as music and animal vocalizations, can also be characterized as a sequence of acoustic events that have precise timing relationships. These timing relationships can carry important information regarding source identity. On the other hand, tasks such as speech recognition or speaker verification require invariance to transformations such as time expansion and pitch shifting. These transformations can be global (e.g., changing the speaker entirely) or local (e.g., pitch-shifting one word or syllable). An auditory representation that precisely captures time/frequency relationships, but can easily be made invariant to such transformations on both local and global scales would thus provide a useful substrate for many applications.

In Section 4.2, we review previous work on auditory representations, focusing on hierarchical and sparse modeling efforts. In Section 4.3 we formulate our hierarchical model mathematically and develop learning and inference algorithms. In Section 4.4, we apply this approach to speech data and analyze the learned representation. The second-layer representations learned by the model encode complex acoustic events that are shiftable in both time and frequency. It is much more compact than the first-layer representation, which is itself a compact description of the sound pressure waveform. We show that using a very sparse hierarchical code, the model can generate sounds that retain much of the acoustic information and approximate well the original sound. Finally, we demonstrate that the model performs well on a denoising task, particularly when the noise is structured, suggesting that the higher-order representation provides a useful statistical description of speech data.

4.2 Review of auditory representations

There is a vast literature on computational representations of acoustic signals dating back several decades. The majority of these rely on signal decompositions which measure how much energy the signal carries as a function of time and frequency. This can be implemented in various ways, such as by applying a short-time Fourier transform (STFT), or a bank of bandpass filters that tile the frequency axis [26]. These decompositions also approximate, to a large degree, the initial processing of sound pressure in the cochlea [155]. Several properties of the time-frequency representation of a sound are known to carry meaningful information. Many natural sounds are well-described as combinations of onsets and tonal sounds [103], which appear as temporally localized but broadband energy and temporally sustained but narrowband energy, respectively. For speech signals, the position of "formants", or energy peaks along the frequency axis, carry significant information about vowel identity [72]. Mel-frequency cepstral coefficients, which are often used in automatic speech recognition [55], build upon a time-frequency representation by taking the log of energies along the frequency axis according to a special "Mel" scale which emphasizes lower frequencies over higher frequencies, and applying a discrete cosine transform (DCT). Linear predictive coefficients (LPC's) decompose a time-frequency representation of a sound according to its formants and are used for sending speech signals across a telephone network [55]. Not until recently, however, have there been efforts to *automatically* learn features, in sounds in an unsupervised manner, that carry meaningful information

Recent efforts to learn auditory representations in an unsupervised setting, analogous to image representations, have focused on sparsely encoding sounds with a superposition of waveforms chosen to capture the structure inherent in sound ensembles. Dictionaries that have been chosen by hand often contain time- and frequency-localized kernels [55, 29, 45, 108]. There have also been attempts to adapt a dictionary to a sound ensemble. For example, Klein et al [69] learned a set of time-frequency kernels to represent spectrograms of speech signals and showed that the resulting kernels were localized and bore resemblance to auditory receptive fields. Lee et al [77] trained a two-layer deep belief network to learn a representation of spectrogram patches and used it for several auditory classification tasks (phone/gender/speaker classification). In [58] and [61], a sparse representation was learned on top of DFT coefficients and a contrast-normalized time-frequency representation, respectively, to perform music genre classification. In [95], a hidden Markov model (HMM) was used in which deep belief networks modeled the generative relationship between hidden states and a spectrogram representation of speech.

These examples have several limitations. First, they operate on spectrograms (rather than the original sound waveforms), which suffer from a tradeoff between temporal and spectral resolution. Since spectrograms rely on a short-time Fourier transform, smaller time bins impose lower limits on the highest measurable frequency. Spectrogram representations typically have 25ms bins with 10ms overlap between consecutive bins, thus discarding any timing information at finer scales [135, 98], which can carry meaningful acoustic information [134, 103, 98]. Spectrograms

and other block-based representations (similar to those applied to images) are also susceptible to blocking artifacts –precisely-timed acoustic events can appear across multiple blocks, and events can appear at different temporal offsets relative to the block, making their identification and representation difficult [135]. Second, the features in these methods are frequency-specific, and thus need to be replicated at many different frequency offsets to accommodate pitch shifts that occur in natural sounds.

4.3 Hierarchical spike code (HSC) model

We build our hierarchical model on top of the "spikegram" representation of [135], in which a sound is encoded using a sparse linear combination of time-shifted kernels $\phi_f(t)$:

$$y_t = \sum_{\tau,f} \alpha_{\tau,f} \phi_f(t-\tau) + \epsilon_t \tag{4.1}$$

where ϵ_t denotes Gaussian white noise and the coefficients $\alpha_{\tau,f}$ are mostly zero. The kernels $\phi_f(t)$ are usually localized in both time and frequency. As in [135], we choose to use gammatone functions with varying center frequencies indexed by f:

$$\phi_f(t) = at^{n-1}e^{-2\pi b_f t}\cos(2\pi f t) \tag{4.2}$$

The constants b_f controlling the temporal extent (or equivalently, inverse bandwidth) of the kernels were chosen to be proportional to the center frequency, i.e. $b_f = \beta f$. The constant a was chosen to make $\|\phi_f(t)\|_2 = 1$. In [136], it was shown that when a dictionary is adapted to combination

of natural sounds, animal vocalizations, and speech, the learned filters are well-fit by gammatone functions of this form, at various center frequencies. Gammatone filterbanks have also frequently been used for the initial processing of sound pressure [155, 103]. In order to encode the signal, a sparse set of coefficients $\alpha_{\tau,f}$ is estimated using a sparse decomposition method such as continuous basis pursuit or a traditional method (Chapter 2). In this work, we chose to use the greedy matching pursuit method [88], for the sake of computational efficiency and ease of use, and also because it was used in [135, 136] to generate spikegrams. The resulting spikegram, shown in Fig. 4.1, offers an efficient representation of sounds [136] that avoids the blocking artifacts and time-frequency tradeoffs associated with more traditional spectrogram representations. Spikes placed at precise locations in time and frequency reveal acoustic features, harmonic structures, as well as slow modulations in the sound envelope.

We aim to model the statistical regularities present in these spikegram representations. Fig. 4.1 illustrates that the spikegram exhibits clear statistical structure, both at coarse (Fig. 4.1(b-c)) and at fine temporal scales (Fig. 4.1(e-f)). Non-stationarity at the coarse scale is likely caused by higher-order acoustic events, such as phoneme utterances, that span a much larger time-frequency range than the individual gammatone kernels. On the other hand, the fine-scale correlations are due to some combination of the (stationary) correlations inherent in the gammatone filterbank, as well as precise (nonstationary) temporal structure present in speech. Indeed, spikegrams computed using the same set of kernels for white noise show similar (but not identical) autocorrelation structure

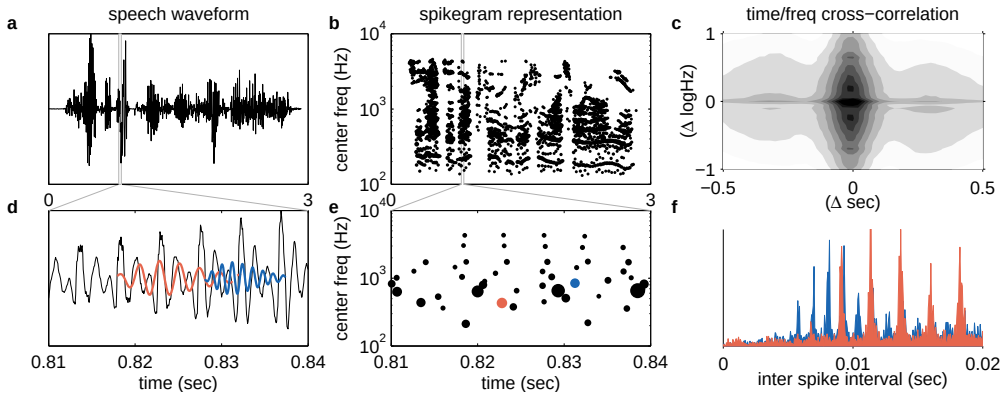


Figure 4.1: Coarse (top row) and fine (bottom row) scale structure in spikegram encodings of speech. (**a**) The sound pressure waveform of a spoken sentence, and (**b**) the corresponding spikegram. Each spike (dot) has an associated time (abscissa) and center frequency (ordinate) as well as an amplitude (dot size). (**c**) Cross-correlation function for a spikegram ensemble reveals correlations across large time/frequency scales. (**d**) For illustration, two gammatones (matching spikes in **e**) are shown at the location and scale specified by the spikes. (**e**) Spike timing exhibits strong regularities at a fine scale. (**f**) Histograms of inter-spike-intervals for two frequency channels corresponding to the colored spikes in **e** reveal strong temporal dependencies.

on a fine scale as in Fig. 4.1(e-f) (not shown).

We introduce a hierarchical spike code (HSC) model, illustrated in Fig. 4.2, to capture the structure in the spikegrams on both coarse and fine scales. We introduce a second layer of unobserved spikes, assumed to have a Poisson process prior, which are convolved with a set of time-frequency kernels ("rate kernels") to modulate the log firing rate of the first-layer spikes on a coarse scale. This log firing rate is also recurrently

modulated on a fine scale by the convolution of local spike history in the first layer at neighboring frequencies with a different set of time-frequency kernels ("coupling kernels"). The coupling kernels for each center frequency need not be identical (although only one such kernel is shown in Fig. 4.2 for simplicity). Similar to the log rate, the mean log *amplitudes* of the first-layer spikes are also modulated by the same second-layer spikes through convolution with a separate set of time-frequency kernels ("amplitude kernels", not shown), but without any recurrent contribution. The model parameters are therefore comprised of the rate, coupling, and amplitude kernels, as well as bias values for the rates and amplitudes corresponding to each first-layer frequency channel. The model can be summarized mathematically using the notation in Table 4.1 with the following equations (note that $*$ denotes convolution):

$$P(S^{(1)}_{t,f} \neq 0) = \Delta_t \Delta_f e^{R_{t,f}} \tag{4.3}$$

$$\log(S^{(1)}_{t,f}) \mid S^{(1)}_{t,f} \neq 0 \sim \mathcal{N}\left(A_{t,f}, \sigma^2\right) \tag{4.4}$$

$$\text{where } R_{t,f} = b^r_f + (K^c * \mathbf{1}_{S^{(1)}})_{t,f} + \sum_i \left[(K^r_i * S^{(2)}_i)_{t,f}\right] \tag{4.5}$$

$$A_{t,f} = b^a_f + \sum_i \left[(K^a_i * S^{(2)}_i)_{t,f}\right] \tag{4.6}$$

4.3.1 Learning

The log joint probability of the observed spikegram data and unobserved second-layer spikes can be expressed as a function of the model param-

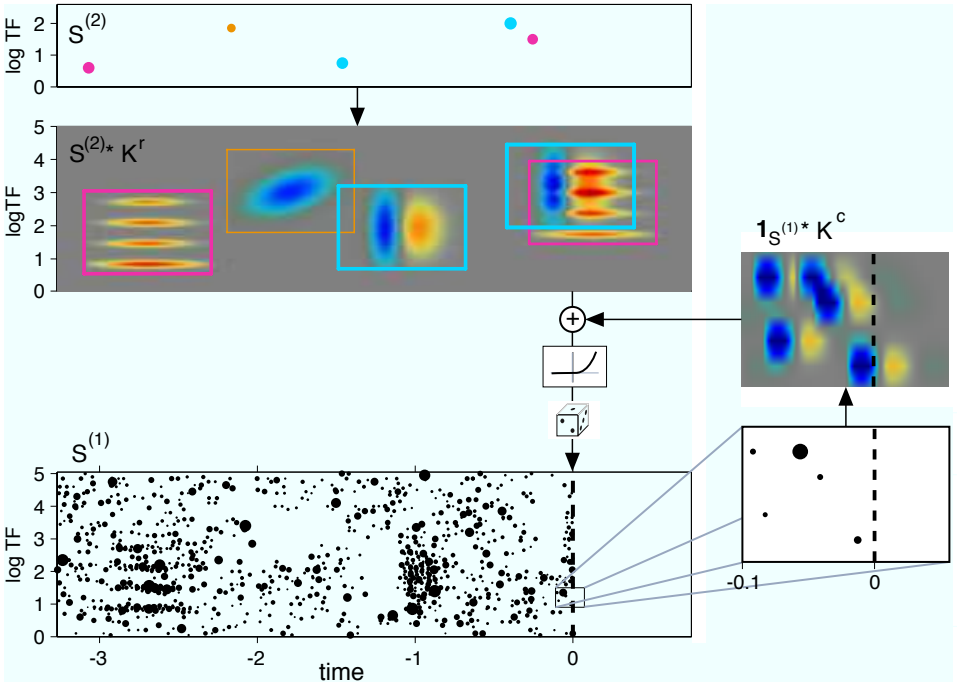


Figure 4.2: Illustration of the hierarchical spike code model. Second-layer spikes $S^{(2)}$ associated with 3 features (indicated by color) are sampled in time and frequency according to a Poisson process, with exponentially-distributed amplitudes (indicated by dot size). These are convolved with corresponding rate kernels K^r (outlined in colored rectangles), summed together, and passed through an exponential nonlinearity to drive the instantaneous firing rate of the first-layer spikes on a coarse scale. The first-layer spike rate is also modulated on a fine scale by a recurrent component that convolves spike history (at all frequencies) with coupling kernels K^c. At a given time step (indicated by the dashed line), the instantaneous firing rate of $S^{(1)}$ depends on both hierarchical and recurrent terms.

$\vec{b}^r$	Rate bias vector
$\vec{b}^a$	Amplitude bias vector
K^r	Rate kernels
K^a	Amplitude kernels
K^c	Coupling kernels
Θ	All model parameters $(\vec{b}^r, \vec{b}^a, K^r, K^c, K^a)$
$S^{(1)}$	Observed first layer spikes
$S^{(2)}$	Unobserved second layer spikes
λ	firing rate of second layer spikes
σ^2	variance of the log amplitudes of spikes
$\mathbf{1}_x$	indicator function: 1 if $x \neq 0$, 0 otherwise

Table 4.1: Notation

eters Θ and the unobserved second-layer spikes $S^{(2)}$:

$$\begin{aligned}
\mathcal{L}(\Theta, S^{(2)}) &= \log P(S^{(1)}, S^{(2)}; \Theta, \lambda) \\
&= \log P(S^{(1)}|S^{(2)}; \Theta) + \log P(S^{(2)}; \lambda) \\
&= \sum_{(t,f)\in S^{(1)}} R_{t,f} - \sum_{t,f} e^{R_{t,f}} \Delta_t \Delta_f \\
&\quad - \frac{1}{2\sigma^2} \sum_{(t,f)\in S^{(1)}} \left(\log S^{(1)}_{t,f} - A_{t,f}\right)^2 \\
&\quad + \log\left(\lambda \Delta_t \Delta_f\right) \|S^{(2)}\|_0 + \text{const}
\end{aligned} \tag{4.7}$$

Maximizing the data likelihood with respect to Θ requires integrating $\mathcal{L}$ over all possible second-layer representations $S^{(2)}$, which is computationally intractable. Instead, we choose to approximate the optimal Θ by maximizing $\mathcal{L}$ jointly over Θ and $S^{(2)}$. This can be interpreted as

performing expectation-maximization (EM) [32], where at each iteration we approximate the posterior distribution over $S^{(2)}$ by a point mass at its maximum. If $S^{(2)}$ is known, then the model falls within the well-known class of generalized linear models (GLMs) [90]: the log firing rate of $S^{(1)}$ is a linear function of the parameters Θ. As a result, Eq. 4.7 is convex in Θ. By symmetry, if Θ is known then Eq. 4.7 is convex in $S^{(2)}$ except for the L_0 penalization term. Motivated by these facts, we adopt a coordinate-descent approach by alternating between the following steps:

$$S^{(2)} \leftarrow \arg\max_{S^{(2)}} \mathcal{L}(\Theta, S^{(2)}) \tag{4.8}$$

$$\Theta \leftarrow \Theta + \eta \nabla_{\Theta} \mathcal{L}(\Theta, S^{(2)}) \tag{4.9}$$

We chose to adapt the model parameters according to a slow learning rate η, rather than optimizing them completely at each iteration, in order to avoid local minima occurring near the initialization point. Section 4.3.2 describes a method for approximate inference of the second-layer spikes (solving Eq. 4.8). To apply Eq. 4.9, we simply compute the gradients with respect to the model parameters:

$$\frac{\partial \mathcal{L}}{\partial b^r_f} = (\#\ 1' \text{ spikes in channel } f) - \sum_t e^{R_{t,f}} \Delta_t \Delta_f \tag{4.10}$$

$$\frac{\partial \mathcal{L}}{\partial b^a_f} = \sum_t \left(\log S^{(1)}_{t,f} - A_{t,f} \right) \tag{4.11}$$

$$\frac{\partial \mathcal{L}}{\partial K^r_{\tau,\zeta,i}} = \sum_{(t,f) \in S^{(1)}} S^{(2)}_i(t-\tau, f-\zeta) - \sum_{t,f} e^{R_{t,f}} S^{(2)}_{t-\tau,f-\zeta,i} \Delta_t \Delta_f \tag{4.12}$$

$$\frac{\partial \mathcal{L}}{\partial K^c_{\tau,f,f'}} = \sum_{t \in S^{(1)}_f} \mathbf{1}_{S^{(1)}_{t-\tau,f'}} - \sum_t e^{R_{t,f}} \mathbf{1}_{S^{(1)}_{t-\tau,f'}} \Delta_t \Delta_f \tag{4.13}$$

4.3.2 Inference

Inference of the second-layer spikes $S^{(2)}$ (Eq. 4.8) involves maximizing the tradeoff between the GLM likelihood term, which we denote by $\tilde{\mathcal{L}}$:

$$\tilde{\mathcal{L}}(\Theta, S^{(2)}) = \sum_{(t,f)\in S^{(1)}} R_{t,f} - \sum_{t,f} e^{R_{t,f}} \Delta_t \Delta_f - \frac{1}{2\sigma^2} \sum_{(t,f)\in S^{(1)}} \left(\log S^{(1)}_{t,f} - A_{t,f} \right)^2 \tag{4.14}$$

and the term containing the L_0 norm which penalizes the number of spikes. This is similar to the sparse linear inverse problem (Eq. 2.5 in Chapter 2) which CBP is meant to solve, except that the loss function is no longer a quadratic least-squares term. As a result, solving Eq. 4.8 exactly is NP-hard, and we are again confronted with the choice of greedy or convex relaxation approximations. In principle, since the rate and amplitude kernels are shiftable in time and frequency, an approximation scheme which takes these into account, such as CBP, can be adapted for the non-quadratic loss function. However, for computational speed and convenience, we adopt a variant of the matching pursuit algorithm [88] (Algorithm 2) that is adapted to the non-quadratic loss function. While this algorithm is known to produce suboptimal solutions when there is significant overlap of kernels (see Section 2.1.2), it is possible these errors average out and do not bias the estimation of the kernels themselves, which is our primary goal in this chapter. As with traditional matching pursuit (Algorithm 1), a single coefficient is chosen and updated on each iteration. However, the optimal coefficient is chosen based on a second-order expansion of the loss function $\tilde{\mathcal{L}}$ about the current estimates of

Algorithm 2 Greedy algorithm for solving Eq. 4.8

$S^{(2)} \leftarrow \vec{0}$

$\Delta\tilde{\mathcal{L}} \leftarrow \infty$

while $\Delta\tilde{\mathcal{L}} > -\log(\Delta_t \Delta_f)$ **do**

$i^* \leftarrow \arg\max_i - \left(\frac{\partial \tilde{\mathcal{L}}}{\partial S^{(2)}_{\tau,\zeta,i}}\right)^2 / \frac{\partial^2 \tilde{\mathcal{L}}}{\partial {S^{(2)}_{\tau,\zeta,i}}^2}$ {use quadratic approx. of $\tilde{\mathcal{L}}$}

$\delta^* \leftarrow \arg\max_\delta \tilde{\mathcal{L}}(\Theta, S^{(2)} + \delta \vec{e}_{i^*}) - \tilde{\mathcal{L}}(\Theta, S^{(2)})$ {do a line search}

$\Delta\tilde{\mathcal{L}} \leftarrow \tilde{\mathcal{L}}(\Theta, S^{(2)} + \delta^* \vec{e}_{i^*}) - \tilde{\mathcal{L}}(\Theta, S^{(2)})$ {compute maximal loss decrease}

if $\Delta\tilde{\mathcal{L}} > -\log(\Delta_t \Delta_f)$ **then**

$S^{(2)}_{i^*} \leftarrow S^{(2)}_{i^*} + \delta$ {update coefficient}

end if

end while

$(\Theta, S^{(2)})$:

$$\tilde{\mathcal{L}}(\cdot) \approx \langle \vec{\nabla} \tilde{\mathcal{L}}\big|_{(\Theta, S^{(2)})}, \cdot \rangle + \frac{1}{2} \langle \cdot, \nabla^{\mathbf{2}} \tilde{\mathcal{L}}\big|_{(\Theta, S^{(2)})} \cdot \rangle + \text{const} \tag{4.15}$$

Under this quadratic approximation, the maximal decrease resulting from changing a coefficient $S^{(2)}\tau, \zeta, i$ can be computed in closed form:

$$- \left(\frac{\partial \tilde{\mathcal{L}}}{\partial S^{(2)}_{\tau,\zeta,i}}\right)^2 / \frac{\partial^2 \tilde{\mathcal{L}}}{\partial {S^{(2)}_{\tau,\zeta,i}}^2} \quad \text{where}$$

$$\frac{\partial \tilde{\mathcal{L}}}{\partial S^{(2)}_{\tau,\zeta,i}} = \sum_{(t,f) \in S^{(1)}} \sum_{t,f} K^r_{t-\tau, f-\zeta, i} e^{R_{t,f}} \Delta_t \Delta_f + \frac{1}{\sigma^2} \sum_{(t,f) \in S^{(1)}} K^a_{t-\tau, f-\zeta, i} \left(\log S^{(1)}_{t,f} - A_{t,f}\right)$$

$$\frac{\partial^2 \tilde{\mathcal{L}}}{\partial {S^{(2)}_{\tau,\zeta,i}}^2} = -\sum_{t,f} \left(K^r_{t-\tau,f-\zeta,i}\right)^2 e^{R_{t,f}} \Delta_t \Delta_f + \frac{1}{\sigma^2} \sum_{(t,f)\in S^{(1)}} \left(K^a_{t-\tau,f-\zeta,i}\right)^2 \tag{4.16}$$

Once a coefficient is chosen in this way, a numerical line search is performed to compute the step size that will maximize the original GLM likelihood $\tilde{\mathcal{L}}$. If the maximal improvement in $\tilde{\mathcal{L}}$ is larger than the L_0 penalty incurred by adding another nonzero coefficient, then this coefficient is updated and the procedure is repeated until no further improvement is possible.

4.4 Results when applied to speech

4.4.1 Learned model

We applied the model to the TIMIT speech corpus [51]. First, we obtained spikegrams by encoding sounds to 20dB precision. This was done by using the matching pursuit algorithm (Algorithm 1) using a set of 200 gammatone filters with center frequencies spaced evenly on a logarithmic scale (see [135] for details). For each audio sample, this gave us a spikegram with fine time and frequency resolution (6.25×10^{-5}s and 3.8×10^{-2} octaves, respectively). We trained a model with 20 rate and 20 amplitude kernels, with frequency resolution equivalent to that of the spikegram and time resolution of 20ms. Rate and amplitude kernels extended over 400ms×3.8 octaves (spanning 20 time and 100 frequency bins). Coupling kernels were defined independently for each frequency channel; they extended over 20ms and 2.7 octaves around the chan-

nels center frequency (within 35 channels on either side) with the same time/frequency resolution of the spikegram. All parameters Θ were initialized randomly, and were learned according to Eq. 4.8-4.9. The first

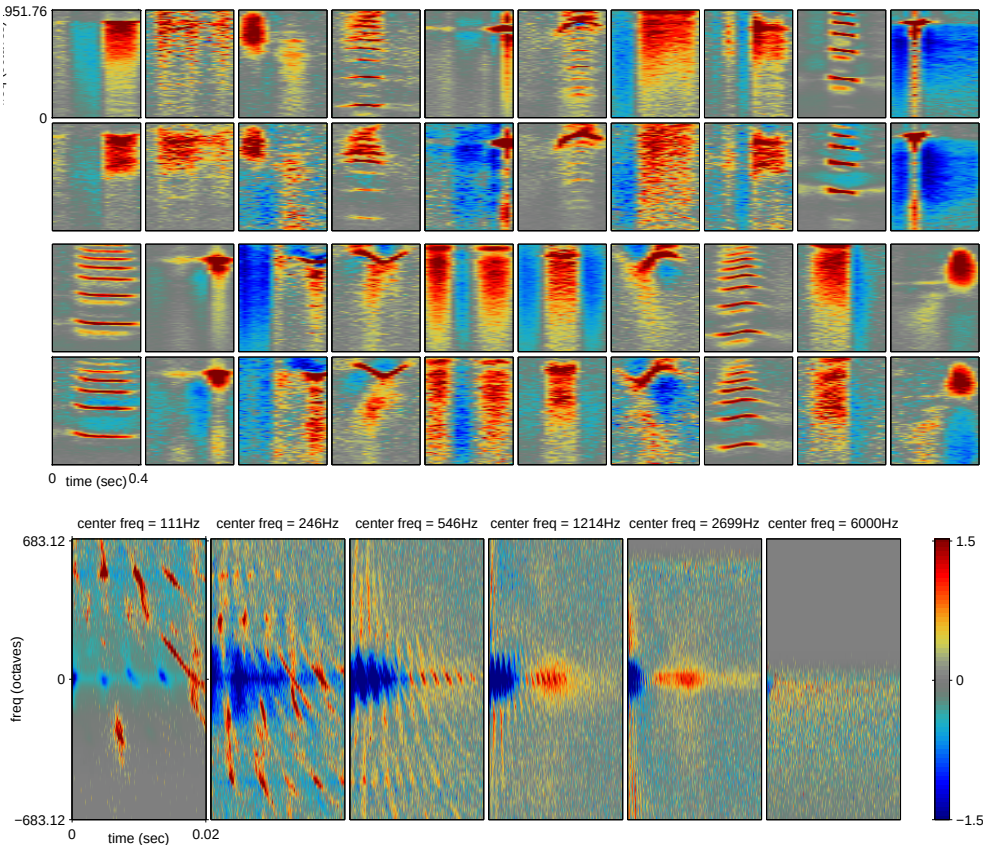


Figure 4.3: Example model kernels learned on the TIMIT dataset. (a) Twenty rate kernels (1st and 3rd rows), with their corresponding amplitude kernels (2nd and 4th rows). Colormaps are individually rescaled. (b) Six coupling kernels (colormap scaling indicated by colorbar).

two rows of Fig. 4.3 display two sets of learned rate kernels (top) and corresponding amplitude kernels (bottom). Among the patterns learned by the rate kernels are harmonic stacks of different durations and pitch

shifts (e.g., kernels 4, 9, 11, 18), ramps in frequency (kernels 1, 7, 15, 16), sharp temporal onsets and offsets (kernels 7, 13, 19), and acoustic features localized in time and frequency (kernels 5, 10, 12, 20). Sounds synthesized by turning on single features is available in supplementary materials. The corresponding amplitude kernels contain patterns highly correlated with the rate kernels, suggesting a strong dependence in the spikegram between spike rate and magnitude.

The coupling kernels are displayed in the bottom of Fig. 4.3. Note that the temporal span of each coupling kernel is equivalent to one bin in the rate/amplitude kernels. For most frequency channels, the coupling kernels are strongly negative at times immediately following the spike and at adjacent frequencies, representing "refractory" periods observed in the spikegrams. Positive peaks in the coupling kernels encode precise alignment of spikes across time and frequency. The coupling patterns that were learned reflect to a large extent the cross-correlational structure between the gammatone kernels, which explains their oscillatory behavior.

4.4.2 Analysis of second layer code

The learned kernels combine in various ways to represent complex acoustic events. For example, Fig. 4.4 illustrates how features can combine to represent two different phone pairs. Vowel phones are approximated by a harmonic stack (outlined in yellow), which can be used for multiple vowels, together with a ramp in frequency (outlined in orange and blue). Because the rate kernels drive the log of the firing rate, their superpo-

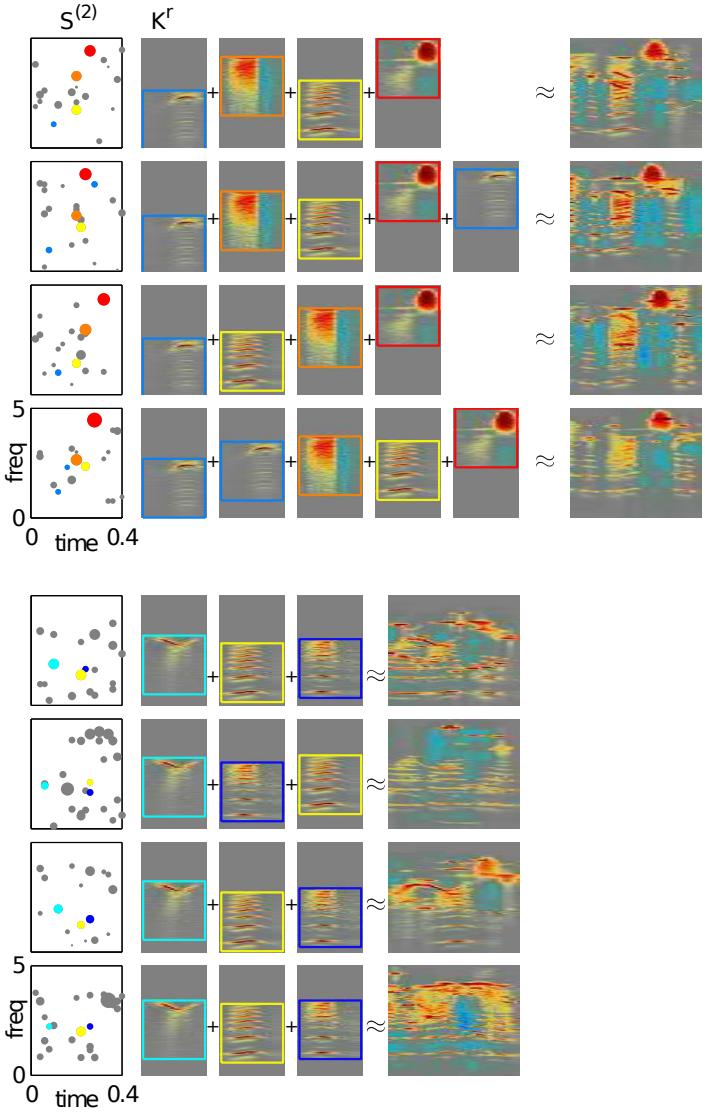


Figure 4.4: Model representation of phone pairs *aa+r* (top) and *ao+l* (bottom), as uttered by four speakers (four rows: two male, two female). Each panel shows inferred second-layer spikes (the spikes whose kernels are most correlated with the occurrence of each phone pair are drawn in color), the corresponding rate kernels, and the encoded log firing rate centered on the phone utterance.

sition results in a multiplicative modulation of the intensities at each level of the harmonic stack. This multiplicative modulation effects the locations of the spectral peaks, or "formants," which are known to be crucial in determining vowel identity.

The 'r' consonant following 'aa' in the first example is characterized by a high concentration of energy at the high frequencies, and is largely accounted for by the kernel outlined in red. The 'l' consonant following 'ao' contains a pitch modulation which is largely accounted for by the v-shaped feature (outlined in cyan). Translating the kernels in log-frequency allows the same set of fundamental features to participate in a range of acoustic events: the same vocalizations at different pitches are often represented by the same set of features. In Fig. 4.4, the same set of kernels are used in a similar configuration across different speakers and genders. It should be noted that the "where" information is not discarded by the second-layer code: this information is encoded in the times and frequencies of the second-layer spikes, which encodes data-specific absolute time and frequency, together with the rate/amplitude kernels, which encode *relative* time-frequency structure that has been abstracted from the entire data ensemble. By a simple binning of spikes at the second layer, this representation can easily be made invariant to temporal and pitch/frequency modulations.

4.4.3 Synthesis

One can further understand the acoustic information that is captured by a set of second-layer spikes by sampling a spikegram according to the

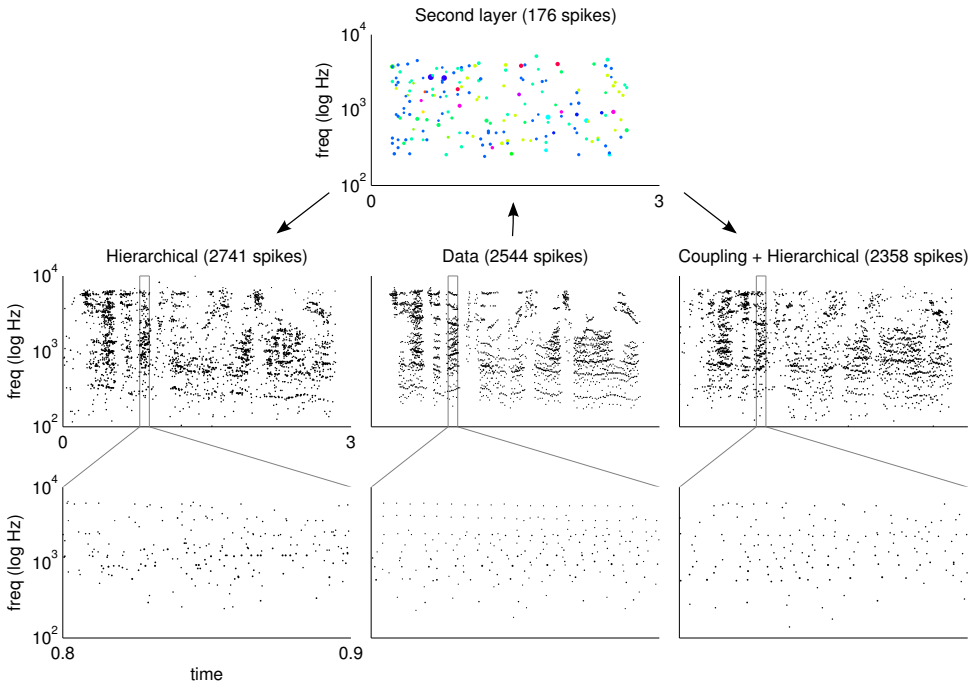


Figure 4.5: Synthesis from inferred second-layer spikes. The second-layer spike representation (middle top) is inferred from a first-layer encoding of the sentence displayed in Fig. 4.1 (middle bottom). Left: first-layer spikes generated using only the hierarchical model component; Right: first-layer spikes generating using hierarchical and coupling kernels. Synthesized waveforms are included in the supplementary materials.

generative model. We took the second-layer encoding of a single sentence from the TIMIT speech corpus [51] (Fig. 4.5 middle) and sampled two spikegrams: one with only the hierarchical component (left), and one that also included both hierarchical and coupling components (right). At a coarse scale the two samples closely resemble the spikegram of the original sound (shown in Fig. 4.1b). However, at the fine time scale, only the spikegram sampled with coupling contains the regularities observed

noise level (SNR)	Wiener	wavelet thresh	MP	HSC
-10dB	-7.00	2.41	2.26	**2.50**
-5dB	0.00	4.93	4.79	**5.01**
0dB	5.49	7.94	7.71	**7.99**
5dB	7.84	11.15	11.01	**11.33**
10dB	10.31	14.64	14.49	**14.83**

Table 4.2: Denoising accuracy (dB SNR) for speech corrupted with white noise.

in speech data. Sounds were also generated from these spikegram samples by superimposing gammatone kernels as in [135]. Despite the fact that the second order representation contains over 15 times less spikes as the first-layer spikegrams, the synthesized sounds are of reasonable quality and the addition of the coupling filters provides a noticeable improvement (audio examples in supplementary materials).

4.4.4 Denoising

Although the model parameters have been adapted to the data ensemble, obtaining an estimate of the likelihood of the data ensemble under the model is difficult, as it requires integrating over unobserved variables ($S^{(2)}$). Instead, we can use performance on unsupervised signal processing tasks, such as denoising, to validate the model and compare it to other methods that explicitly or implicitly represent data density.

In the noiseless case, a spikegram is obtained by running matching pursuit (Algorithm 1) until the residual norm falls below a threshold

noise level (SNR)	Wiener	wavelet thresh	MP	HSC
-10dB	-8.68	-8.73	-5.12	**-4.37**
-5dB	-3.09	-3.63	-0.96	**-0.38**
0dB	1.90	1.23	2.97	**3.30**
5dB	6.37	6.06	7.11	**7.40**
10dB	9.68	11.28	11.58	**11.88**

Table 4.3: Denoising accuracy (dB SNR) for speech corrupted with sparse temporally modulated noise.

(20dB); in the presence of noise, this encoding process can be formulated as a denoising operation (see Section 2.1), terminated when the improvement in the log-likelihood (variance of the residual divided by the variance of the noise) is less than the cost of adding a spike (the negative log-probability of spiking). We incorporate the HSC model directly into this denoising algorithm by replacing the fixed probability of spiking with the rate inferred by the model. Since neither the first- nor second-layer spike code for the noisy signal is known, we obtain the inferred rate by maximizing the posterior given the noisy waveform, $P(S^{(1)}, S^{(2)}|x)$. The denoised waveform is obtained by reconstructing from the resulting first-layer spikes.

To the extent that the parameters learned by HSC reflect statistical properties of the signal, incorporating the more sophisticated spikegram prior into a denoising algorithm should allow us to better distinguish signal from noise. We tested this by denoising speech waveforms (held out during model training) that have been corrupted by additive white

Gaussian noise. We compared the models performance to that of the matching pursuit encoding (sparse signal representation without a hierarchical model), as well as to two standard denoising methods, Wiener filtering and wavelet-threshold denoising (implemented with MATLABs `wden` function, using symlets, SURE estimator for soft threshold selection; other parameters optimized for performance on the training data set) [87].

Model-based denoising is able to outperform standard methods, as well as matching pursuit denoising (Table 4.2). Although the performance gains are modest, the fact that a generative model, which is not optimized for the task or trained on noisy data, can match the performance of adaptive algorithms like wavelet filtering denoising, suggests that the model is indeed capturing meaningful structure in the signal distribution.

To test more rigorously the benefit of a structured prior, we evaluated denoising performance on signals corrupted with nonstationary noise whose power is correlated over time. This is a more challenging task, but it is also more relevant to real-world applications, where sources of noise are often non-stationary. Algorithms that incorporate specific (but often incorrect) noise models (e.g., Wiener filtering) tend to perform poorly in this setting.

We generated sparse temporally modulated noise by scaling Gaussian white noise with a temporally smooth envelope (given as a convolution of a Gaussian function with st. dev. of 0.02s with a Poisson process with rate $16s^{-1}$). All methods fare worse on this task. Again, the hierarchical

model outperforms other methods (Table 4.3), but here the improvement in performance is larger, especially at high noise regimes when the model prior plays a greater role. Note also that the reconstruction SNR does not fully convey the manner in which different algorithms handle noise. Perceptually, the sounds denoised by the model appear to be more similar to the original (audio examples in supplementary materials).

4.5 Summary and discussion

We developed a general methodology for learning hierarchical representations with multiple "spike code" layers. We showed that a two-layer model adapted to speech data captures complex acoustic structure. Our work builds on top of the spikegram representation of [135], and makes a number of novel contributions. Unlike previous work [61, 58, 69, 77], the learned kernels are shiftable in both time *and* log-frequency, which enables the model to learn time and frequency-relative patterns and use a small number of kernels efficiently to represent a wide variety of sound features. In addition, the model describes acoustic structure on multiple scales (via a hierarchical component and a recurrent component), which capture fundamentally different kinds of statistical regularities.

Technical contributions in developing this model include methods for learning and performing approximate inference in a generalized linear model in which some of the inputs are unobserved and sparse (in this case, the second-layer spikes). The computational framework developed here is general, and may have wide applications in modeling sparse data with partially observed variables. Because the model is nonlinear, multi-

stage cascades could lead to substantially more powerful models.

A related approach is the work of [14], in which natural movies are factored into temporally persistent variables representing form, and temporally dynamic variables representing motion. These groups of variables are then separately processed into second-layer representations to model intermediate form and motion structure. The model is similar in spirit to the approach we have taken, but with many differences. First, the motion variables only model the temporal derivatives, and therefore any information about absolute position is lost. As a result, the representation does not reflect a full generative model of the signal. Second, the model was trained on video "patches" and so was not convolutional in space or time, although the phase variables presumably modeled some of the transformation structure. On the other hand, the learned phase variables were able to capture more complex transformations than simple spatio-temporal shifting, such as spatial rotation [14].

Applying the model to complex natural sounds (speech), we demonstrated that it can learn non-trivial features, and we have showed how these features can be composed to form basic acoustic units. We also showed a simple application to denoising, demonstrating improved performance to wavelet thresholding. The framework provides a general methodology for learning higher-order features of sounds, and we expect that it will prove useful in representing other structured sounds such as music, animal vocalizations, or ambient natural sounds.

Chapter 5

Conclusion

We believe this thesis contributes to multiple bodies of literature, both conceptually and methodologically. A large body of work has employed the notion of sparse representations when modeling real signal ensembles such as sounds and images, motivated in part by empirical observations (e.g., heavy-tailed distributions of wavelet coefficients of images) and also by the fact that sparse over-complete representations can capture rich structure . However, this notion has also been incorrectly used to handle transformation-invariance. When transformations are present in the ensemble, the dictionary is no longer a countable union of individual features, but rather a countable union of smooth manifolds corresponding to each feature. Rather than imposing a sparse factorial prior distribution on a discrete lattice lying on the transformational manifold(s), we operate under a source model which (approximately) imposes a sparse prior on the manifolds themselves, resulting in a much more accurate model for transformation-invariant signals.

From a methodological point of view, we provide an efficient algo-

rithm to recover sparse representations as well as transformational information. When the signal ensemble is invariant to continuous transformations, this method offers significant advantage over conventional sparse recovery methods in terms of accuracy, sparsity, and possibly efficiency.

Nevertheless, there are still many challenges that lie ahead. For example, our approach only deals with the case when there is a single known transformations. As discussed at the end of Chapter 2, the extension to multiple transformations is straightforward with Taylor interpolation, although there are still some outstanding issues when using polar interpolation. The types of transformations are typically known when dealing with physical signals. However, there are other forms of input (i.e. the signals after several stages of processing), for which these transformations may be unclear. A more complete characterization of signal distributions could parametrize content, the transformation operator itself, and transformation amount(s) present in the signal. One potentially interesting avenue to explore to this end is to combine CBP with manifold learning techniques.

Another open question deals with learning transformation-invariant dictionaries. Does accounting for transformation-invariance simply result in a more efficient learned dictionary (avoiding learning transformed copies of the same kernel), or does it change the fundamental structure of the learned dictionary? Does the choice of inference method used within a dictionary learning scheme (greedy methods, BP, CBP) have a significant effect on the resulting learned dictionary?

Finally, although we have formulated a hierarchical model for processing transformation and content information, it still relied on a con-

volutional spike representation of the signal at each layer. This choice was made more for the sake of computational tractability. However, a more direct approach to modeling "where" information could operate on the continuous-valued transformation parameters that are computed by methods like CBP.

We hope that future work will benefit from the CBP approach, both in the context of building more accurate and powerful statistical models of transformation-invariant ensembles, and also as a tool for analyzing electrophysiological, seismological, radar, sonar, and imaging signals, along with sounds and images.

Appendix A

Polar interpolator details

The polar interpolator approximates intermediate timeshifts by an arc circumscribing three time-shifted version of a waveform $f(t)$: $\{f_{-\Delta/2}, f_0, f_{\Delta/2}\}$ via the equation:

$$f_\tau(t) \approx \begin{pmatrix} 1 \\ r\cos\left(\frac{\tau\theta}{\Delta}\right) \\ r\sin\left(\frac{\tau\theta}{\Delta}\right) \end{pmatrix}^T \begin{pmatrix} c(t) \\ u(t) \\ v(t) \end{pmatrix} \quad \text{for } |\tau| < \frac{\Delta}{2} \tag{A.1}$$

In this section we discuss in detail the derivation of the quantities in Eq. A and derive the interpolator for some illustrative example of $f(t)$. We use the following notation:

$$\langle f_1, f_2 \rangle = \int_{-T}^{T} f_1(t) f_2(t) dt$$

$$\|f\|_2^2 = \int_{-T}^{T} (f(t))^2 \, dt$$

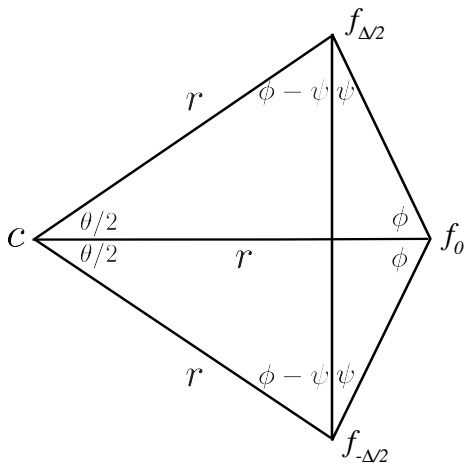


Figure A.1: Geometric relationship between angles in the 2D plane containing the 3 time shifts, explaining the derivation of Eq. A.2-A.3.

A.0.1 Derivation of r and θ

As illustrated in Fig. A.1, the constants r and θ can be derived using simple rules from 2D geometry. The closed form expressions are:

$$\begin{aligned} \theta &= 2(\pi - 2\phi) = 2(\pi - 2(\frac{\pi - 2\psi}{2})) \\ &= 4\psi = 4\angle\left(f_0 - f_{\frac{\Delta}{2}}, f_{-\frac{\Delta}{2}} - f_{\frac{\Delta}{2}}\right) \qquad \text{(A.2)} \\ r &= \sqrt{\frac{\|f_{\frac{\Delta}{2}} - f_0\|_2^2}{2\left(1 - \cos\left(\frac{\theta}{2}\right)\right)}} \qquad \text{(A.3)} \end{aligned}$$

where $\angle(\cdot, \cdot)$ is the angle between two functions. Eq. A.3 follows from the cosine rule.

A.0.2 Derivation of $\{c, u, v\}$

The functions $\{c, u, v\}$ are obtained by solving a linear system of 3 equations corresponding to the reconstructions of $\{f_{-\frac{\Delta}{2}}, f_0, f_{\frac{\Delta}{2}}\}$:

$$\begin{pmatrix} f_{-\frac{\Delta}{2}} \\ f_0 \\ f_{\frac{\Delta}{2}} \end{pmatrix} = \begin{pmatrix} 1 & r\cos\left(\frac{\theta}{2}\right) & -r\sin\left(\frac{\theta}{2}\right) \\ 1 & r & 0 \\ 1 & r\cos\left(\frac{\theta}{2}\right) & r\sin\left(\frac{\theta}{2}\right) \end{pmatrix} \begin{pmatrix} c \\ u \\ v \end{pmatrix} \tag{A.4}$$

Inverting the matrix in Eq. A.4 gives the solution:

$$\begin{pmatrix} c \\ u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{-\cos\left(\frac{\theta}{2}\right)}{1-\cos\left(\frac{\theta}{2}\right)} & \frac{1}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} \\ \frac{-1}{2r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{1}{r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{-1}{2r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} \\ \frac{-1}{2r\sin\left(\frac{\theta}{2}\right)} & 0 & \frac{1}{2r\sin\left(\frac{\theta}{2}\right)} \end{pmatrix} \begin{pmatrix} f_{-\frac{\Delta}{2}} \\ f_0 \\ f_{\frac{\Delta}{2}} \end{pmatrix} \tag{A.5}$$

A.0.3 Orthogonality of $\{c, u, v\}$

Let $\alpha = \frac{1}{\left(1-\cos\left(\frac{\theta}{2}\right)\right)}$, $\beta = \frac{1}{2\sin\left(\frac{\theta}{2}\right)}$, and $\overline{f} = \frac{f_{\frac{\Delta}{2}} + f_{-\frac{\Delta}{2}}}{2}$ (i.e. the mean of the oppositely-shifted versions). Then from Eq. A.5, we can see that:

$$\begin{aligned} c &= \alpha\overline{f} + (1-\alpha)f_0 \\ u &= \frac{\alpha}{r}(f_0 - \overline{f}) \\ v &= \frac{\beta}{r}(f_{\frac{\Delta}{2}} - f_{-\frac{\Delta}{2}}) \end{aligned}$$

Since $\overline{f} \perp (f_{\frac{\Delta}{2}} - f_{-\frac{\Delta}{2}})$ and $f_0 \perp (f_{\frac{\Delta}{2}} - f_{-\frac{\Delta}{2}})$ (by symmetry), it follows that $c \perp v$ and $u \perp v$. Also, by construction of c we have that:

$$\begin{aligned} &\|f_{-\frac{\Delta}{2}} - c\|_2^2 = \|f_0 - c\|_2^2 = \|f_{\frac{\Delta}{2}} - c\|_2^2 = r^2 \\ &\Rightarrow \langle c, f_{-\frac{\Delta}{2}}\rangle = \langle c, f_0\rangle = \langle c, f_{\frac{\Delta}{2}}\rangle \\ &\Rightarrow \langle c, f_0 - \overline{f}\rangle = 0 \\ &\Rightarrow c \perp u \end{aligned}$$

A.0.4 Sinusoid example

The polar approximation of Eq. A can be more clearly understood by considering the simple example when $f(t) = \sin(\omega t)$. In this case we have:

$$\begin{aligned} f_{-\frac{\Delta}{2}} &= \sin\left(\omega t + \frac{\omega\Delta}{2}\right) \\ &= \sin(\omega t)\cos\left(\frac{\omega\Delta}{2}\right) + \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \\ f_0 &= \sin(\omega t) \\ f_{\frac{\Delta}{2}} &= \sin\left(\omega t - \frac{\omega\Delta}{2}\right) \\ &= \sin(\omega t)\cos\left(\frac{\omega\Delta}{2}\right) - \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \end{aligned} \tag{A.6}$$

These epressions follow from standard trigonometric identities. Using the two observations $\langle \sin(\omega t), \cos(\omega t)\rangle = 0$ and $\| \sin(\omega t) \|_2 = \| \cos(\omega t) \|_2$, we can use Eq. A.2 to derive the subtended angle θ:

$$\begin{aligned} \theta &= 4 \cos^{-1}\left(\frac{\langle f_0 - f_{\frac{\Delta}{2}}, f_{-\frac{\Delta}{2}} - f_{\frac{\Delta}{2}}\rangle}{\|f_0 - f_{\frac{\Delta}{2}}\|_2 \|f_{-\frac{\Delta}{2}} - f_{\frac{\Delta}{2}}\|_2}\right) \\ &= 4 \cos^{-1}\left(\frac{\langle \sin(\omega t)\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right) + \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right), -2\cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right)\rangle}{\| \sin(\omega t)\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right) + \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \|_2 \| 2\cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \|_2}\right) \\ &= 4 \cos^{-1}\left(\frac{\left(\sin\left(\frac{\omega\Delta}{2}\right) \| \cos(\omega t) \|_2\right)^2}{\| \sin(\omega t)\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right) + \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \|_2 \left(\sin\left(\frac{\omega\Delta}{2}\right) \| \cos(\omega t) \|_2\right)}\right) \\ &= 4 \cos^{-1}\left(\sqrt{\frac{\sin^2\left(\frac{\omega\Delta}{2}\right) \| \cos(\omega t) \|_2^2}{\left(\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)^2 + \sin^2\left(\frac{\omega\Delta}{2}\right)\right) \| \cos(\omega t) \|_2^2}}\right) \\ &= 4 \cos^{-1}\left(\frac{\sin\left(\frac{\omega\Delta}{2}\right)}{\sqrt{2\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)}}\right) \\ &= 4 \cos^{-1}\left(\frac{2\sin\left(\frac{\omega\Delta}{4}\right)\cos\left(\frac{\omega\Delta}{4}\right)}{2\sin\left(\frac{\omega\Delta}{4}\right)}\right) \\ &= \omega\Delta \end{aligned} \tag{A.7}$$

We can then obtain the values of $(c(t), u(t), v(t))$ via Eq. A.5:

$$\begin{aligned}
\begin{pmatrix} c \\ u \\ v \end{pmatrix} &= \begin{pmatrix} \frac{1}{2\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)} & \frac{-\cos\left(\frac{\omega\Delta}{2}\right)}{1-\cos\left(\frac{\omega\Delta}{2}\right)} & \frac{1}{2\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)} \\ \frac{-1}{2r\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)} & \frac{1}{r\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)} & \frac{-1}{2r\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right)} \\ \frac{-1}{2r\sin\left(\frac{\omega\Delta}{2}\right)} & 0 & \frac{1}{2r\sin\left(\frac{\omega\Delta}{2}\right)} \end{pmatrix} \\
&\quad \times \begin{pmatrix} \sin(\omega t)\cos\left(\frac{\omega\Delta}{2}\right) + \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \\ \sin(\omega t) \\ \sin(\omega t)\cos\left(\frac{\omega\Delta}{2}\right) - \cos(\omega t)\sin\left(\frac{\omega\Delta}{2}\right) \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ \frac{\sin(\omega t)}{r} \\ \frac{-\cos(\omega t)}{r} \end{pmatrix}
\end{aligned} \tag{A.8}$$

Therefore, the center, $c(t)$ of the circle used in the polar approximation is identically 0, and so the circle is a "great circle" on the hypersphere. Using Eq. A.4, the polar approximation in this example can be written as:

$$\begin{aligned}
\sin(\omega t - \omega\tau) &\approx \begin{pmatrix} 1 \\ r\cos(\omega\tau) \\ r\sin(\omega\tau) \end{pmatrix}^T \begin{pmatrix} 0 \\ \frac{\sin(\omega t)}{r} \\ -\frac{\cos(\omega t)}{r} \end{pmatrix} \\
&= \sin(\omega t)\cos(\omega\tau) - \cos(\omega t)\sin(\omega\tau) \text{ for } |\tau| < \frac{\pi}{2}
\end{aligned} \tag{A.9}$$

Eq. A.9 in facts holds with equality for all t, τ and is a well-known trigonometric identity. Therefore, the polar approximation is *exact* for sinusoidal waveforms for any spacing Δ, since the translational manifold really is a circle on the hypersphere whose center is the origin.

A.0.5 Fourier analysis

The sinusoid example motivates an analysis of the polar approximation in the Fourier domain. Combining Eq. A with Eq. A.4 gives an expression

for the interpolated time-shift as a linear function of the three timeshifts $\{f_{-\Delta/2}, f_0, f_{\Delta/2}\}$:

$$\begin{aligned} f_\tau &\approx \begin{pmatrix} 1 \\ r\cos\left(\frac{\tau\theta}{\Delta}\right) \\ r\sin\left(\frac{\tau\theta}{\Delta}\right) \end{pmatrix}^T \begin{pmatrix} \frac{1}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{-\cos\left(\frac{\theta}{2}\right)}{1-\cos\left(\frac{\theta}{2}\right)} & \frac{1}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} \\ \frac{-1}{2r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{1}{r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} & \frac{-1}{2r\left(1-\cos\left(\frac{\theta}{2}\right)\right)} \\ \frac{-1}{2r\sin\left(\frac{\theta}{2}\right)} & 0 & \frac{1}{2r\sin\left(\frac{\theta}{2}\right)} \end{pmatrix} \begin{pmatrix} f_{-\frac{\Delta}{2}} \\ f_0 \\ f_{\frac{\Delta}{2}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1-\cos\left(\frac{\tau\theta}{\Delta}\right)}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} - \frac{\sin\left(\frac{\tau\theta}{\Delta}\right)}{2\sin\left(\frac{\theta}{2}\right)} \\ \frac{\cos\left(\frac{\tau\theta}{\Delta}\right)-\cos\left(\frac{\theta}{2}\right)}{1-\cos\left(\frac{\theta}{2}\right)} \\ \frac{1-\cos\left(\frac{\tau\theta}{\Delta}\right)}{2\left(1-\cos\left(\frac{\theta}{2}\right)\right)} + \frac{\sin\left(\frac{\tau\theta}{\Delta}\right)}{2\sin\left(\frac{\theta}{2}\right)} \end{pmatrix}^T \begin{pmatrix} f_{-\frac{\Delta}{2}} \\ f_0 \\ f_{\frac{\Delta}{2}} \end{pmatrix} \quad \text{for } |\tau| < \frac{\Delta}{2} \end{aligned} \tag{A.9}$$

Taking the Fourier transform of both sides gives:

$$e^{-i\omega\tau}\hat{f}(\omega) \approx \begin{pmatrix} a(\tau) - b(\tau) \\ 1 - 2a(\tau) \\ a(\tau) + b(\tau) \end{pmatrix}^T \begin{pmatrix} e^{i\omega\frac{\Delta}{2}} \\ 1 \\ e^{-i\omega\frac{\Delta}{2}} \end{pmatrix} \hat{f}(\omega) \quad \text{for } |\tau| < \frac{\Delta}{2} \tag{A.10}$$

Dividing out $\hat{f}(\omega)$ and equating real and imaginary parts gives the following two approximations:

$$\cos(\omega\tau) \approx \frac{\cos\left(\frac{\tau\theta}{\Delta}\right)\left(1-\cos\left(\frac{\omega\Delta}{2}\right)\right) + \left(\cos\left(\frac{\omega\Delta}{2}\right) - \cos\left(\frac{\theta}{2}\right)\right)}{1-\cos\left(\frac{\theta}{2}\right)} \tag{A.11}$$

$$\sin(\omega\tau) \approx \frac{\sin\left(\frac{\tau\theta}{\Delta}\right)\sin\left(\frac{\omega\Delta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} \tag{A.12}$$

Notice that in the sinusoidal case $f(t) = \sin(\omega t)$, we have $\theta = \omega\Delta$ and the approximations in Eq. A.11-A.12 hold with equality. In the general case, however, θ is not strictly linear in Δ since it must integrate the translation-induced changes at all frequencies at which $f(t)$ has power. As a result, we expect that polar interpolation accuracy to degrade as the bandwidth of $f(t)$ increases. However, unlike classical interpolation that derives from the sampling theorem (i.e. interpolation using a sinc kernel), this degradation is graceful and the interpolation is still quite accurate for non-bandlimited

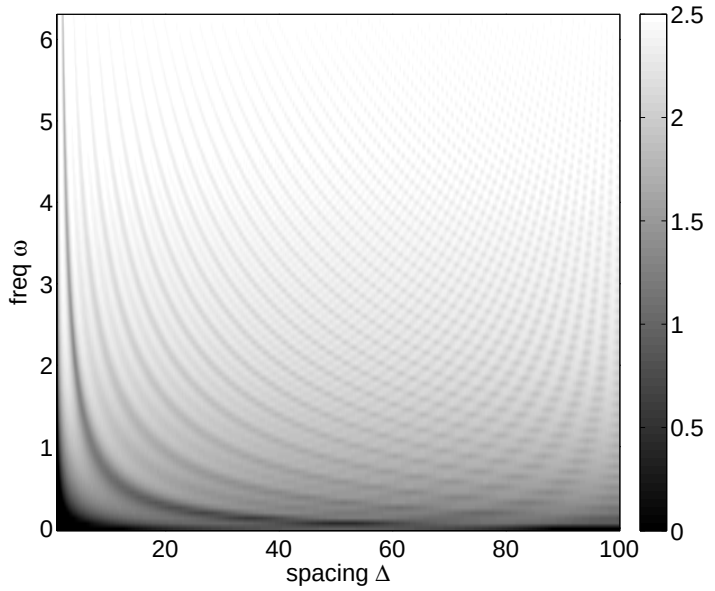


Figure A.2: Mean squared error of the polar interpolator approximation, Eq. A.10, with respect to spacing Δ and frequency ω for a waveform $f(t) \propto te^{-\alpha t^2}$.

functions. For example, Fig. A.2 plots the interpolation error for $f(t) \propto e^{-\alpha t^2}$, which is not bandlimited.

A.0.6 Interpolation comparison

Figure A.3 compares nearest neighbor (implicitly used in BP), first-order Taylor, and polar interpolation in terms of their accuracy in approximating time-shifts of a Gaussian derivative waveform, $f(t) \propto te^{-\alpha t^2}$. For reference, the second-order Taylor interpolator is also included. The polar interpolator is seen to be much more accurate than nearest-neighbor and 1st-order Taylor, and even surpasses 2nd-order Taylor by an order of magnitude (although they have the same asymptotic rate of convergence). This allows one to choose a

much larger Δ for a given desired accuracy.

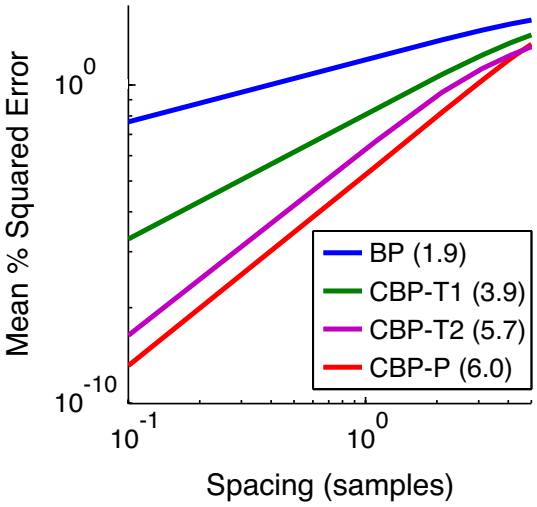


Figure A.3: Comparison of the nearest neighbor, first-order Taylor, and polar interpolators (as used in BP, CBP-T, and CBP-P, respectively) for a waveform $f(t) \propto te^{-\alpha t^2}$. Second-order Taylor interpolation are also shown. The estimated log-domain slopes (asymptotic rates of convergence) are indicated in parenthesis in the legend.

Appendix B

Spike sorting appendix

B.0.7 Clustering method

We implemented a conventional three-step clustering method [79], as illustrated in Fig. 3.1, and used it to obtain all performance benchmarks presented in Section 3.2.

Detection. To identify segments of the voltage trace containing spiking activity, we identify peaks in the voltage trace exceeding a threshold that is derived from an estimate of the noise level [59, 112, 37]:

$$T := 4\hat{\sigma} \qquad \hat{\sigma} := \frac{median(|V(t)|)}{0.6745} \tag{B.1}$$

Five millisecond windows, centered around these peaks, are identified as segments of spiking activity in the signal. The windowed segments are temporally upsampled by a factor of five using cubic spline interpolation, re-centered about the maximal value (across all electrodes) assuming zero padding on both ends, and then downsampled to the original rate.

Feature extraction. We then reduced the dimensionality of the data by transforming each of the segments into a low-dimensional feature space. Specifically, we formed a data vector for each segment (i.e., the voltage samples of all electrodes lying within each temporal window were concatenated into a single vector). We performed principal components analysis (PCA) on this set of vectors, selected the leading components that accounted for 90 percent of the variance over all the segments (e.g., see legends of Figs. 3.5(b-c)), and projected the contents of all windows onto these components. Figures 3.4(a,c,e) show the projections of these segments onto the first two principal components for three different data sets.

Clustering. The dimensionally-reduced feature vectors are then automatically grouped according to similarity. For our primary comparisons we used K-means clustering to accomplish this [38]. The number of clusters was manually adjusted to minimize the number of errors, and several random initializations were tried in order to get the optimal clustering assignment. For the simulated data set we also compare our results with those of a superparamagnetic clustering method (in the space of wavelet coefficients) [112]. For this method, we obtain a lower bound on the number of errors by adding three numbers reported in their paper: (1) the number of detection errors, (2) the number of classification errors, and (3) the number of detected voltage segments which contain two or more spikes (clustering must miss at least one spike from each of these snippets).

B.0.8 Preprocessing of real data

Filtering of raw voltage trace. All extracellular traces, both simulated and real, were highpass-filtered at 250Hz with a Butterworth filter of order 50.

No preprocessing (other than amplitude re-scaling) was done for intracellular traces. For each tetrode data set, a sustained period in which the recording was stable was selected for analysis. For one of the tetrode sets [59], a period of $0.8s$ containing anomalous bursting activity was removed from the analysis.

Ground truth spike identification from intracellular trace. For real data, ground truth spikes were inferred by identifying peaks in the intracellular traces that exceeded 4 standard deviations from the baseline. Since the intracellular traces have almost no noise, this simple procedure can reliably and accurately identify all spikes (see Fig. 3.6).

Noise covariance estimation. We assume that the full spatio-temporal noise covariance matrix is space-time separable, allowing us to first whiten each channel in time, and then whiten across channels. For each channel, the temporal covariance matrix was assumed to be Toeplitz (i.e. stationary noise), depending only on the noise autocovariance. The autocovariance was estimated from "noise" regions in the extracellular trace which did not exceed $2\hat{\sigma}$ for a period of 50 ms or more, where $\hat{\sigma}$ was computed as in Eq. B.0.7. A whitening filter was then computed by taking the central column of the inverse matrix-square-root of the temporal autocovariance matrix. The channel data was then convolved with this whitening filter. Once each channel was whitened in time, the spatial covariance matrix (across electrodes) was estimated from the same noise regions. Each time slice was then left-multiplied by the inverse matrix-square-root.

B.0.9 Evaluation

Counting misses and false positives. For evaluation, we matched spikes in the estimated spike train with spikes in the true spike train. A true

spike could be matched with an estimated spike of the same cell if it occurred within 4 milliseconds of the true spike time.

Best ellipsoidal error rate (BEER) bounds error of clustering-based methods. We computed the *best ellipsoidal error rate* (BEER) measure [59], which serves as an upper bound on the performance of any clustering-based spike sorting method that uses elliptical cluster boundaries. After thresholding and feature extraction, the windowed segments of the trace were labeled according to whether or not they contained a true spike. Half of this labeled data set was then used to train a support vector machine whose decision rule was a linear combination of all pairwise products of the features of each segment, and was thus capable of achieving any elliptical decision boundary. This decision boundary was then used to predict the occurrence of spikes in the segments in the remaining half of the labeled data, and the success or failure of these predictions then provided an estimate of the miss and false positive rate.

Bibliography

[1] E. H. Adelson, E. P. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. In *Proc SPIE Visual Communications and Image Processing II*, volume 845, pages 50–58, Cambridge, MA, October 1987.

[2] M. Aharon, M. Elad, and A. Bruckstein. The k-svd: An algorithm for designing of overcomplete dictionaries for sparse representation. *Structure*, 54(11):4311–4322, 2006.

[3] A. Atiya. Recognition of multiunit neural signals. *Biomedical Engineering, IEEE Transactions on*, 39(7):723 –729, july 1992.

[4] I. Bar-Gad, Y. Ritov, E. Vaadia, and H. Bergman. Failure in identification of overlapping spikes from multiple neuron activity causes artificial correlations. *Journal of Neuroscience Methods*, 107(12):1 – 13, 2001.

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

[6] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *In NIPS*. MIT Press, 2007.

[7] P. Berkes, R. E. Turner, and M. Sahani. A structured model of video reproduces primary visual cortical organisation. *PLoS Computational Biology*, 5(9), 2009.

[8] J. Bioucas-Dias and M. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans Image Processing*, 16(12):2992–3004, 2007.

[9] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[11] E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461, May 2004.

[12] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[13] E. J. C. Enhancing sparsity by reweighted 1 minimization. *J. Fourier Analysis and Applications*, pages 877–905, 2008.

[14] C. F. Cadieu and B. A. Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural Comput.*, 24(4):827–866, Apr. 2012.

[15] A. Calabrese and L. Paninski. Kalman filter mixture model for spike sorting of non-stationary data. *Journal of Neuroscience Methods*, 196(1):159 – 169, 2011.

[16] E. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203 – 4215, dec. 2005.

[17] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans IT*, 52(2):489–509, 2006.

[18] E. J. Cands, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[19] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3869 –3872, 31 2008-april 4 2008.

[20] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[21] J. H. Choi, H. K. Jung, and T. Kim. A new action potential detector using the mteo and its effects on spike sorting systems at low signal-to-noise ratios. *Biomedical Engineering, IEEE Transactions on*, 53(4):738 –746, april 2006.

[22] R. R. Coifman and D. Donoho. Translation-invariant de-noising. pages 125–150. Springer-Verlag, 1995.

[23] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[24] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[25] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. ¡ce:title¿Higher Order Statistics¡/ce:title¿.

[26] M. Cooke, S. Beet, and M. Crawford, editors. *Visual representations of speech signals*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[27] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. in Pure and Appl. Math*, 57:1413–1457, 2004.

[28] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Gntrk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[29] M. Davies and L. Daudet. Sparse audio representations using the mclt. *Signal Processing*, 86(3):457 – 470, 2006.

[30] M. E. Davies and T. Blumensath. Faster & greedier: algorithms for sparse reconstruction of large datasets. In *Proc. 3rd Int. Symp. Communications, Control and Signal Processing ISCCSP 2008*, pages 774–779, 2008.

[31] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13:57–98, 1997. 10.1007/BF02678430.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[33] S. Devries. Correlated firing in rabbit retinal ganglion cells. *Journal of Neurophysiology*, 81(2):908–920, 1999.

[34] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 58(2):1094 –1121, feb. 2012.

[35] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197–2202, 2003.

[36] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans Info Theory*, 52(1):6–18, 2006.

[37] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[38] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, Nov. 2001.

[39] M. Elad. Why simple shrinkage is still relevant for redundant representations. *IEEE Trans Info Theory*, 52:5559–5569, 2006.

[40] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736 –3745, dec. 2006.

[41] M. Elad, M. A. T. Figueiredo, and Y. M. Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[42] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky. A wide-angle view at iterated shrinkage algorithms. In *in SPIE (Wavelet XII*, pages 26–29, 2007.

[43] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*, 55(11):5302–5316, 2009.

[44] M. S. Fee, P. P. Mitra, and D. Kleinfeld. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *Journal of Neuroscience Methods*, 69(2):175 – 188, 1996.

[45] C. Fevotte, B. Torresani, L. Daudet, and S. Godsill. Sparse linear regression with structured priors and application to denoising of musical audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):174 –185, jan. 2008.

[46] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans Image Processing*, 12:906–916, 2003.

[47] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 16(12):2980–2991, 2007.

[48] P. Földiák. Learning invariance from transformation sequences. *Neural Comput.*, 3(2):194–200, June 1991.

[49] F. Franke, M. Natora, C. Boucsein, M. Munk, and K. Obermayer. An online spike detection and spike classification algorithm capable of in-

stantaneous resolution of overlapping spikes. *J. Comput. Neurosci.*, pages 127 – 148, 2009. in press.

[50] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans Computers*, C-23(9):881–890, 1974.

[51] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. Darpa timit acoustic phonetic continuous speech corpus cdrom, 1993.

[52] G. L. Gerstein and W. A. Clark. Simultaneous studies of firing patterns in several neurons. *Science*, 143(3612):1325–1327, 1964.

[53] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Trans. Signal Processing*, pages 600–616, 1997.

[54] M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, oct 2010.

[55] S. Greenberg, W. A. Ainsworth, A. N. Popper, R. R. Fay, N. Mogran, H. Bourlard, and H. Hermansky. Automatic speech recognition: An auditory perspective. In *Speech Processing in the Auditory System*, volume 18 of *Springer Handbook of Auditory Research*, pages 309–338. Springer New York, 2004.

[56] D. B. Grimes and R. P. N. Rao. Bilinear sparse coding for invariant vision. *Neural Comput.*, 17(1):47–73, Jan. 2005.

[57] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification. In *UAI*, 2007.

[58] P. Hamel and D. Eck. Learning features from music audio with deep belief networks. In *ISMIR*, pages 339–344, 2010.

[59] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, K. D, D. A. Henze, and J. Csicsvari. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J Neurophysiol*, 84:401–414, 2000.

[60] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[61] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *Proceedings of International Symposium on Music Information Retrieval (ISMIR'11)*, 2011.

[62] K. K. Herrity, A. C. Gilbert, and J. A. Tropp. Sparse approximation via iterative thresholding. In *Proc. IEEE Int Acoustics, Speech and Signal Processing Conf. ICASSP 2006*, volume 3, 2006.

[63] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[64] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[65] A. Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, Jul 2000.

[66] Y. Karklin and M. S. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, January 2009.

[67] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierachies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS 2010)*, 2010.

[68] M. Kivanc Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *Signal Processing Letters, IEEE*, 6(12):300 –303, dec. 1999.

[69] D. J. Klein, P. König, and K. P. Körding. Sparse spectrotemporal coding of sounds. *EURASIP J. Appl. Signal Process.*, 2003:659–667, Jan. 2003.

[70] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *Signal Processing Magazine, IEEE*, 13(4):67 –94, jul 1996.

[71] K. Y. Kwon and K. Oweiss. Wavelet footprints for detection and sorting of extracellular neural action potentials. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 609 –612, may 2011.

[72] P. Ladefoged and K. Johnson. *A Course in Phonetics*. Cengage Learning, 2010.

[73] Y. Lecun and Y. Bengio. *Convolutional Networks for Images, Speech and Time Series*, pages 255–258. The MIT Press, 1995.

[74] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989.

[75] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

[76] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.

[77] H. Lee, Y. Largman, P. Pham, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems 22*, pages 1096–1104, 2009.

[78] M. Lewicki. Bayesian modeling and classification of neural signals. *Neural Computation*, 6:1005–1030, 1994.

[79] M. S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network*, 9(4):R53–R78, Nov 1998.

[80] M. S. Lewicki and B. A. Olshausen. Inferring sparse, overcomplete image codes using an efficiient coding framework. In *NIPS*, 1997.

[81] M. S. Lewicki and B. A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A*, 16:1587–1601, 1999.

[82] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Comput.*, 12(2):337–365, Feb. 2000.

[83] S. Lyu and E. P. Simoncelli. Nonlinear extraction of 'independent components' of natural images using radial Gaussianization. *Neural Computation*, 21(6):1485–1519, Jun 2009.

[84] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, page 87, 2009.

[85] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11:674–693, 1989.

[86] S. Mallat. *A Wavelet Tour of Signal Processing*. Wavelet Analysis and Its Applications Series. Academic Press, 1999.

[87] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008.

[88] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans Sig Proc*, 41(12):3397–3415, December 1993.

[89] D. Mastronarde. Correlated firing of cat retinal ganglion cells. *Journal of Neurophysiology*, 49(2):303–324, 1989.

[90] P. McCullagh and J. A. Nelder. *Generalized linear models (Second edition)*. London: Chapman & Hall, 1989.

[91] M. Meister, J. Pine, and D. A. Baylor. Multi-neuronal signals from the retina: acquisition and analysis. *Journal of Neuroscience Methods*, 51(1):95 – 106, 1994.

[92] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[93] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Computation*, 22(6):1473–1492, June 2010.

[94] X. Miao and R. P. N. Rao. Learning the lie groups of visual invariance. *Neural Comput.*, 19(10):2665–2693, Oct. 2007.

[95] A. Mohamed, G. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):14 –22, jan. 2012.

[96] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227, 1995.

[97] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):310 –316, april 2010.

[98] S. R. Ness, T. Walters, and R. F. Lyon. *Auditory Sparse Coding*. 2011.

[99] D. North. An analysis of the factors which determine signal/noise discrimination in pulsed carrier systems. *Proceedings of the IEEE*, 51, July 1963.

[100] I. Obeid and P. Wolf. Evaluation of spike-detection algorithms fora brain-machine interface application. *Biomedical Engineering, IEEE Transactions on*, 51(6):905 –911, june 2004.

[101] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, Jun 1996.

[102] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993.

[103] R. D. Patterson. Auditory images: How complex sounds are represented in the auditory system. *Journal of the Acoustical Society of Japan*, 21(4):183–190, 2000.

[104] A. Pazienti and S. Grn. Robustness of the significance of spike synchrony with respect to sorting errors. *Journal of Computational Neuroscience*, 21:329–342, 2006. 10.1007/s10827-006-8899-7.

[105] G. Peyre. Manifold models for signals and images. *Computer vision and image understanding*, 113(2):249–260, 2009.

[106] J. W. Pillow, J. Shlens, E. Chichilnisky, and E. Simoncelli. Cross-correlation artifacts and undetected synchronous spikes in multi-neuron recordings: a model-based spike sorting algorithm. *Journal of Neuroscience Methods*, 2012.

[107] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signaling in a complete neuronal population. *Nature*, 454(7206):995–999, Aug 2008.

[108] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: from coding to source separation. *Proceedings of the IEEE.*, 98(6):995–1005, June 2010.

[109] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans Image Processing*, 12(11):1338–1351, November 2003. Recipient, IEEE Signal Processing Society Best Paper Award, 2008.

[110] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach. *J Neurophysiol*, 91(6):2910–2928, 2004.

[111] J. S. Prentice, J. Homann, K. D. Simmons, G. Tkaik, V. Balasubramanian, and P. C. Nelson. Fast, scalable, bayesian spike identification for multi-electrode arrays. *PLoS ONE*, 6(7):e19884, 07 2011.

[112] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.*, 16:1661–1687, August 2004.

[113] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[114] R. P. N. Rao and D. L. Ruderman. Learning lie groups for invariant visual perception. In *In NIPS*, 1999.

[115] S. P. Rebrik, B. D. Wright, A. A. Emondi, and K. D. Miller. Cross-channel correlations in tetrode recordings: implications for spike-sorting. *Neurocomputing*, 2627(0):1033 – 1038, 1999.

[116] R. W. Rodieck. Maintained activity of cat retinal ganglion cells. *Journal of Neurophysiology*, 30(5):1043–71, 1967.

[117] U. Rutishauser, E. M. Schuman, and A. N. Mamelak. Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. *Journal of Neuroscience Methods*, 154(12):204 – 224, 2006.

[118] M. Sahani. *Latent variable models for neural data analysis*. PhD thesis, California Institute of Technology, Pasadena, California, 1999.

[119] M. Sahani, J. S. Pezaris, and R. A. Andersen. On the separation of signals from neighboring cells in tetrode recordings. In *NIPS*, 1997.

[120] P. Sallee and B. A. Olshausen. Learning sparse multiscale image representations. In *NIPS*, pages 1327–1334, 2002.

[121] M. J. Schnitzer and M. Meister. Multineuronal firing patterns in the signal from eye to brain. *Neuron*, 37:499–511, 2003.

[122] I. Schoenberg. On trigonometric spline interpolation. *Journal of Math Mech.*, 13:795–825, 1964.

[123] O. Schwartz and E. P. Simoncelli. Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4(8):819–825, August 2001.

[124] O. Schwartz and E. P. Simoncelli. Natural sound statistics and divisive normalization in the auditory system. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Adv. Neural Information Processing Systems (NIPS*00)*, volume 13, pages 166–172, Cambridge, MA, May 2001. MIT Press.

[125] R. Segev, J. Goodhouse, J. Puchalla, and M. J. Berry. Recording spikes from a large fraction of the ganglion cells in a retinal patch. *Nature Neuroscience*, 7(10):1154–1161, Oct. 2004.

[126] L. Sendur and I. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *Signal Processing, IEEE Transactions on*, 50(11):2744 – 2756, nov 2002.

[127] J. Shlens, G. D. Field, J. L. Gauthier, M. Greschner, A. Sher, A. M. Litke, and E. J. Chichilnisky. The structure of large-scale synchronized firing in primate retina. *Journal of Neuroscience*, 29:5022–5031, April 2009.

[128] J. Shlens, F. Rieke, and E. Chichilnisky. Synchronized firing in the retina. *Current Opinion in Neurobiology*, 18(4):396 – 402, 2008. ¡ce:title¿Sensory systems¡/ce:title¿.

[129] S. Shoham, M. R. Fellows, and R. A. Normann. Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of Neuroscience Methods*, 127(2):111–122, 2003.

[130] P. Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition – tangent distance and tangent propagation. *International Journal of Imaging Systems and Technology*, 11(3), 2001.

[131] E. P. Simoncelli and E. H. Adelson. Noise removal via Bayesian wavelet coring. In *Proc 3rd IEEE Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne, Sep 16-19 1996. IEEE Sig Proc Society.

[132] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc 2nd IEEE Int'l Conf on Image Proc*, volume III, pages 444–447, Washington, DC, Oct 23-26 1995. IEEE Sig Proc Society.

[133] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

[134] M. Slaney and R. F. Lyon. On the importance of time - a temporal representation of sound, 1993.

[135] E. Smith and M. S. Lewicki. Efficient coding of time-relative structure using spikes. *Neural Computation*, 17(1):19–45, Jan 2005.

[136] E. C. Smith and M. Lewicki. Efficient auditory coding. *Nature*, 439(7079):800–805, 2006.

[137] J.-L. Starck, E. J. Cands, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, 2002.

[138] S. Takahashi, Y. Anzai, and Y. Sakurai. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. *Journal of Neurophysiology*, 89(4):2245–2258, 2003.

[139] S. Takahashi and Y. Sakurai. Real-time and automatic sorting of multi-neuronal activity for sub-millisecond interactions in vivo. *Neuroscience*, 134(1):301–315, 2005.

[140] G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*, page 2007. MIT Press, 2006.

[141] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Comput.*, 12(6):1247–1283, June 2000.

[142] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[143] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):2231–2242, 2004.

[144] M. Unser. Sampling-50 years after shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.

[145] J. Uriguen, Y. C. Eldar, P. L. Dragotti, and B.-H. Z. *Sampling at the Rate of Innovation: Theory and Applications*. Cambridge University Press, 2011.

[146] C. Vargas-Irwin and J. P. Donoghue. Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *Journal of Neuroscience Methods*, 164(1):1–18, 2007.

[147] M. Vetterli, P. Marziliano, and T. Blu. Sampling signals with finite rate of innovation. *IEEE Trans Sig Proc*, 50(6):1417 –1428, June 2002.

[148] M. Wehr, J. S. Pezaris, and M. Sahani. Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms. *Neurocomputing*, 26-27:1061–1068, 1999.

[149] J. Wild, Z. Prekopcsak, T. Sieger, D. Novak, and R. Jech. Performance comparison of extracellular spike sorting algorithms for single-channel recordings. *Journal of Neuroscience Methods*, 203(2):369 – 376, 2012.

[150] L. Wiskott. How does our visual system achieve shift and size invariance? Cognitive Sciences EPrint Archive (CogPrints) 3321, dec 2003.

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

[152] T. won Lee, M. S. Lewicki, M. Girolami, T. J. Sejnowski, and S. Member. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Sig. Proc. Lett*, pages 87–90, 1999.

[153] F. Wood and M. J. Black. A non-parametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173:1–12, 2008.

[154] F. Wood, M. J. Black, C. Vargas-irwin, M. Fellows, and J. P. Donoghue. On the variability of manual spike sorting. *IEEE Transactions on Biomedical Engineering*, 51:912–918, 2004.

[155] X. Yang, K. Wang, and S. Shamma. Auditory representations of acoustic signals. *Information Theory, IEEE Transactions on*, 38(2):824 –839, mar 1992.

[156] P.-M. Zhang, J.-Y. Wu, Y. Zhou, P.-J. Liang, and J.-Q. Yuan. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *Journal of Neuroscience Methods*, 135(12):55 – 65, 2004.