

Information-Theoretic Analysis of the Neural Code

Don H. Johnson^{*†}, Charlotte M. Gruner^{*}, Keith Baggerly[†], Chandran Seshagiri^{*}

Computer and Information Technology Institute

Department of Electrical and Computer Engineering^{}*

Department of Statistics[†]

Rice University, MS 366, Houston, Texas 77005–1892

dhj@rice.edu, cmkruger@rice.edu, kabagg@rice.edu, cshag@rice.edu

February 6, 1998

Abstract

We describe a family of new techniques for analyzing single- and multi-unit (ensemble) discharge patterns. These techniques are based on information theoretic distance measures and on empirical theories derived from work in universal signal processing. They are capable of determining transneuron statistical dependencies even when time-varying responses occur. Regardless of the nature of the neural code, these measures quantify that portion of the response contributing most to information coding and the fidelity of that encoding. Examples of how to apply these techniques are drawn from the single and multiple unit processing of sound amplitude and sound location.

1 Introduction

For about half a century, the information-bearing aspect of individual neuron’s discharge patterns has been thought to be the *times* at which discharges occur; any action potential waveform variations have second-order effects. In sensory systems and others, this presumption has been elaborated to the notion of a stimulus-response relationship: Discharge timing should somehow vary as the stimulus changes. Neural coding has been classified into two broadly defined types, rate codes —the average rate of spike discharge —or timing codes —the timing pattern of discharges. Recently, researchers have found single-unit response characterization inadequate to explain coding. For example, theoretical considerations indicate that single-neuron discharge patterns in the mammalian auditory pathway are too random to effectively represent sound [23]. Coordinated responses of neurons within sensory and motor nuclei have been found, with discharge timing relations among neural outputs having significance [1, 3, 9, 28]. Consequently, more recent work has focused on population activity, using the fundamental assumption that coordinated sequences of action potential occurrence times produced by groups of neurons collectively represent the stimulus-response relationship. Thus, today the “neural code” is taken to mean how groups of neurons, responding individually and collectively, represent sensory information with their discharge patterns [5]. Knowing the code would unlock the secrets of how neurons, working in concert, process and represent information.

Data analysis techniques for single-neuron discharges —the PST histogram, the interval histogram, and several joint interval statistical measures —were inspired by the mathematical model for single neuron discharges, the point process [23]. These measures were derived from the simplest point process model, the Poisson. Because of refractory effects, the Poisson model cannot apply in detail, but these measures remain the standard. Even if properly applied, these measures do not sufficiently quantify the response to reveal what stimulus aspects are being represented by neural discharges, when these representations occur, what the representations are, and what is the quality of these representations.

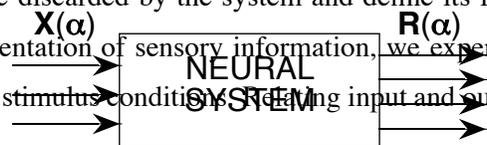
Using these techniques to measure population activity means that coordinated population coding is not directly probed. From the viewpoint of point process theory, population codes are equivalent to the intensity of an accurate vector-channel, point-process model [26] for the data. The intensity of the point process would summarize the (probably complex) dependence of discharge probability on discharges occurring in the same neuron (temporal dependence), in other neurons at the same time (spatial dependence), and in other neurons at different times (spatiotemporal dependence). Unfortunately, traditional optimal estimation techniques for point processes depend heavily on the intensity’s intrinsic structure (how one event depends on the timing of others) [5], which is part of the neural code we seek. Furthermore, stimulus changes induce time-varying responses, which confound many techniques for quantifying the population codes: Mutual information calculations [13, 29], cross- and autocorrelation techniques [2], and artificial neural networks [25] apply to *stationary* single-neuron or neural pair response patterns, and don’t generalize easily

Figure 1: A neural system has as inputs the vector quantity \mathbf{X} that depend on a collection of stimulus parameters denoted by the vector α . The output \mathbf{Y} thus also depends on the stimulus parameters. Both input and output implicitly depend on time. Note that the precise nature of the input is deliberately unclear. It can represent the stimulus itself or a population’s collective input.

to nonstationary neural ensembles. One recently published technique, based on defining an *ad hoc* distance between single-neuron discharge patterns [32], can deal with time-varying responses, but can only with difficulty be extended to ensembles. Furthermore, this and other *ad hoc* distance measures are difficult to interpret.

Beyond characterizing the response, we still have difficulty answering the question how, how well, and when is coding occurring even when the input-output relation for the neural system is known. For example, we developed a point process model for the tone-burst responses of single units located in the lateral superior olive (LSO) [34, 35] and a computational biophysical model [36]. Both models are so exact that responses they generate are difficult to discern from recordings. While these models provide a notion of the response’s structure, they do not help us determine the typical LSO unit’s information processing role and what processing function the variety of response types may engender. What has been left out is quantifying both the response’s significance and its effectiveness in representing sensory information. When a single primary auditory neuron responds to a suddenly applied stimulus (such as a tone burst), does the initial transient response or the later sustained response more effectively convey loudness? When inputs to a neural population have been measured and well characterized, how do we judge how well the population extracts sensory information? In fact, how would we know if a population did extract information or simply served as a relay? In short, having an accurate statistical characterization of a response, be it from individual neurons or an ensemble, does not mean that we have clarified the sensory processing role. We return to our central theme. We must be able to quantify the *neural code*: What aspect of a neural ensemble’s collective output represents information and what is the fidelity of this representation?

Consider the simple system shown in Figure 1. Conceptually, this system accepts inputs \mathbf{X} that represent a stimulus or a neural population conveying information (parameterized by α) and produces outputs \mathbf{Y} that collectively code some or all of the stimulus. The boldfaced symbols represent vectors, and are intended to convey the notion that our system—a neural ensemble—has multiple inputs and multiple outputs. Presumably, input stimulus features preserved in the output are those extracted by the system; those de-emphasized in the output are discarded by the system and define its feature extraction properties. To probe the system and its representation of sensory information, we experimentally measure the system’s output and its inputs as we vary stimulus conditions.



to measuring the intensity, which does not help quantify the effectiveness of neural coding. Instead, what we look for is how the inputs and the outputs change as the stimulus undergoes a controlled change. No change means no coding of the perturbed aspect of the stimulus; the bigger the change, the more the system accentuates that sensory aspect. To quantify change, we need a measure that quantifies its degree. In short, what we seek is a *distance measure*: Given two sets of stimulus conditions α_1, α_2 , we need to measure how different the corresponding responses $R(\alpha_1), R(\alpha_2)$ are — how far apart they are — with some distance metric $d(R(\alpha_1), R(\alpha_2))$. Assuming that population codes are subtle, this metric needs to apply to ensemble responses, to nonstationary as well as stationary response changes, to changes in transneural correlations, and to changes in temporal correlation structure.

While the merits of one measure versus another can be debated, we describe here a collection of information theoretic distances that have a clear, intuitive mathematical foundation. The underlying theory is not rooted in the classic results of Shannon, but in modern classification theory. In this theory, we try to assign a response to one of a set of pre-assigned response categories. The ease of classification depends on how different the categories are; it is through this aspect of the classification problem that distance measures arise. We use this classification theoretic approach because recent results from universal signal processing¹ provide distance measures and classification techniques that assume little about the data yet yield (in a certain sense) optimal classification results. In addition to the distance measure having a strong mathematical foundation, direct empirical results have been derived. For example, we can determine how complex a data analysis we can perform given a certain amount of data, and we have empirical methods for measuring distance that take into account the nature of the data.

Because our technique rests on modern classification theory and on universal data processing, areas that may not be familiar to most neuroscientists, we digress to describe them in sufficient detail to understand our approach. The relative newness of our technique means that we have not applied it extensively, and though mathematically well-founded, we have not completely developed response processing methods. Our example applications represent some response processing methodologies; we expect more will follow.

2 Modern classification theory

Classification theory concerns how observations can be optimally classified into predefined categories. Stating the problem formally, a set of observations $\mathbf{R} = \{R_1, \dots, R_L\}$ is to be classified as belonging to one of J categories. Two ways of defining categories create separate classification problems: (1) each category is defined according to a known probabilistic description (which may have unknown parameters or other controlled uncertainties) and (2) each category is defined according to training data. The first is known as the *classic* classification problem, and the latter the *empirical* classification problem. The classic problem was solved in the first half of this century, and recent results from information theory provide a solution to

¹The theory of how to process information universally without much regard to the underlying distribution of the data.

the empirical problem.

Classic classification. Given the probabilistic descriptions of the categories, the optimal rule for classifying observations, the likelihood ratio test, is well known [18]. Interestingly, what is very difficult to calculate is how well this optimal rule works. Developing approximations for calculating performance leads to important notions that directly apply to the neural response analysis problem.

The most frequently studied variant of this problem is the binary classification problem: which of two categories C_1 , C_2 best match the observations. Performance is using expressed in terms of error probabilities. Using terminology created by radar engineers, wherein C_1 means no target is present and C_2 one is, one error probability is the *false-alarm probability* $P_F = \Pr[\text{say } C_2 \mid C_1]$ (the probability that the classification rule announces C_2 when the data actually were produced according to C_1) and the second is the *miss probability* $P_M = \Pr[\text{say } C_1 \mid C_2]$. The *average error probability* P_e is the average of these individual error probabilities: $P_e = \Pr[C_1]P_F + \Pr[C_2]P_M$, where $\Pr[C_1]$, $\Pr[C_2]$ are the *a priori* probabilities that data conform to the categories. Note that in the two-category problem $\Pr[C_1] = 1 - \Pr[C_2]$. Classifier performance is usually judged by the average error probability. We can also formulate classifiers, so-called Neyman-Pearson classifiers, that operate under constraints on one of the component error probabilities and optimize the other. They are judged by the error probability not constrained. For both kinds of classification rules, comparing the likelihood ratio to a threshold is the optimal classification rule [27]. The choice of rule defines the threshold. In Neyman-Pearson classifiers, the threshold depends on how much data are available; in optimal P_e classifiers, the threshold is fixed.

One would expect that optimal P_e and Neyman-Pearson classifiers would produce different results, with each being optimal with respect to its design criterion. No general formulae for these error probabilities are known for these or any other optimal classifiers. In some special cases, like the classic Gaussian problem, we can analytically determine how well optimal classifiers work, but a formula that also applies to non-Gaussian data has not been found. What has been found are asymptotic error probabilities. We can answer the question ‘‘How does performance change as the amount of data becomes large?’’ When the observations \mathbf{R} are statistically independent and identically distributed under both categories, $p_{C_j}(\mathbf{R}) = \prod_l p_{C_j}(R_l)$, results known as Stein’s Lemma and Chernoff’s Bound [7: §12.8,12.9] state that error probabilities decay exponentially in the amount of data available. The first two describe the performance of Neyman-Pearson classifiers, the third optimal P_e classifiers.

$$\lim_{L \rightarrow \infty} \frac{\log P_F}{L} = -\mathcal{D}(p_{C_2}(R) \parallel p_{C_1}(R)) \quad \text{for fixed } P_M \quad (1a)$$

$$\lim_{L \rightarrow \infty} \frac{\log P_M}{L} = -\mathcal{D}(p_{C_1}(R) \parallel p_{C_2}(R)) \quad \text{for fixed } P_F \quad (1b)$$

$$\lim_{L \rightarrow \infty} \frac{\log P_e}{L} = -\mathcal{C}(p_{C_1}(R), p_{C_2}(R)), \quad (1c)$$

where $\mathcal{D}(p \parallel q)$ is known as the *Kullback-Leibler distance* between the probability densities p , q or between

two probability mass functions P, Q

$$\mathcal{D}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \quad \text{or} \quad \mathcal{D}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \quad (2)$$

and $\mathcal{C}(p, q)$ is the *Chernoff distance*.

$$\mathcal{C}(p, q) = \mathcal{D}(p^*||p) = \mathcal{D}(p^*||q) \quad \text{or} \quad \mathcal{C}(P, Q) = \mathcal{D}(P^*||P) = \mathcal{D}(P^*||Q) \quad (3)$$

The distribution p^* is equidistant from the two probability distributions p, q , and the Chernoff distance is this “halfway” distance. Finding these distributions so that the Chernoff distance can be calculated is equivalent to solving an optimization problem.

$$\mathcal{C}(p, q) = - \max_{0 \leq u \leq 1} \int [p(x)]^{1-u} [q(x)]^u dx \quad \text{or} \quad \mathcal{C}(P, Q) = - \max_{0 \leq u \leq 1} \sum_x [P(x)]^{1-u} [Q(x)]^u \quad (4)$$

Note that these definitions apply to both univariate and multivariate distributions. When the observations are not statistically independent, all these results apply to the multivariate distribution of the observations [21]: for example,

$$\lim_{L \rightarrow \infty} \frac{\log P_F}{L} = - \frac{\mathcal{D}(p_{C_2}(\mathbf{R})||p_{C_1}(\mathbf{R}))}{L} \quad \text{for fixed } P_M .$$

In these definitions, we use the base-two logarithm, which means that distance has units of bits.

Stein’s Lemma and Chernoff’s Bound (1) are not stated directly in term of error probabilities because of subtle technical details. Focusing on the false-alarm probability, Stein’s Lemma for the case of independent observations can be stated more directly as

$$P_F \rightarrow f(L)2^{-LD(p_{C_1}||p_{C_2})} \quad \text{for fixed } P_M ,$$

with $\lim_{L \rightarrow \infty} [\log f(L)]/L = 0$. The term $f(\cdot)$ changes more slowly in comparison to the exponential, and it depends on the problem at hand. What this formula means is that if we plot any of the error probabilities logarithmically against L linearly, we will always obtain a straight line for large values of L (see figure 2). Stein’s Lemma says that the false-alarm probability’s slope equals the negative of the Kullback-Leibler distance between the probability distributions defining our classification problem. Because of the presence of the problem-dependent quantity $f(\cdot)$, we cannot determine in general the vertical origin for the error probability and how large L must be for straight-line behavior to take over. Consequently, we cannot use asymptotic formulas to compute error probabilities, but we do know that they ultimately decay exponentially for *any* classification problem solved with the optimal classifier, and we know the rate of this decay. We can also say that if further observations increase any of these distances by one unit (a bit), the corresponding error probability decreases by a factor of two. Furthermore, having exponentially

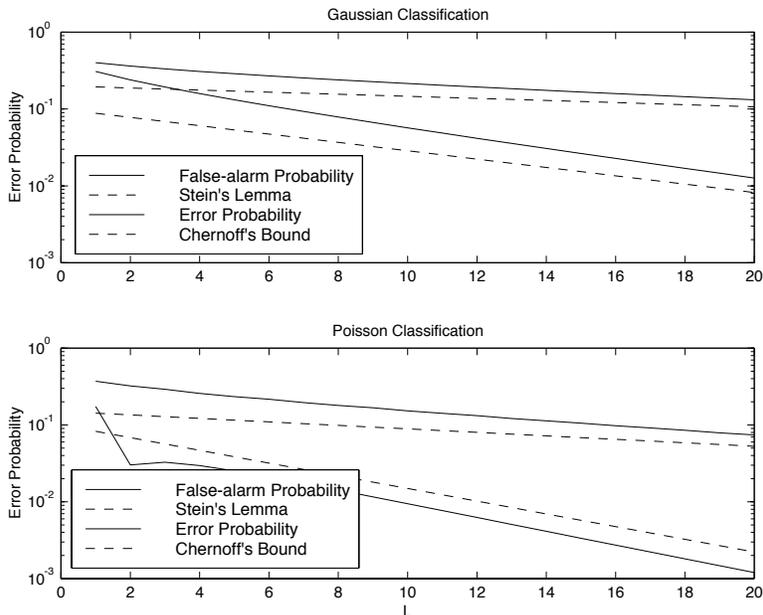


Figure 2: Using the Gaussian and Poisson classification problems as examples, we plot the false-alarm probability and average probability of error for each as a function of the amount of statistically independent data used by the classifier. The general shape of these curves—a rapidly decaying initial component followed by linear behavior—repeats in Gaussian and many non-Gaussian examples. The average error probability is the more slowly decaying solid line; the false-alarm probability is the more rapidly decaying, smaller probability solid line. The dashed lines depict the behavior of the error probability as predicted by asymptotic theory. In each case, these lines have been shifted vertically for ease of comparison.

decreasing error probabilities defines a set of “good” classifiers. Optimal classifiers produce error probabilities that decay exponentially with the quantity multiplying L equal to the Kullback-Leibler distance or the Chernoff distance. Suboptimal but “good” ones will have a smaller slope, with poor ones not yielding exponentially decaying error probabilities. Modern classification theory focuses on the slope of the error probability when plotted on semi-logarithmic coordinates. This slope, known as the *exponential rate*, cannot be steeper than the Chernoff distance for the classifier that optimizes average error probability and Kullback-Leibler distance for the Neyman-Pearson classifier. Thus, these distances define *any* classification problem’s difficulty. The greater the distance, the more quickly error probabilities decrease (the exponential rate is larger) and the “easier” the classification problem. Whether we use an optimal classifier or not, the Chernoff and Kullback-Leibler distances quantify the ultimate performance any classifier can achieve, and therefore measure intrinsic problem difficulty.

Note that the Kullback-Leibler distance (equation 2) is asymmetric with respect to the two distributions defining the classification problem. The false-alarm probability achieved under a fixed miss-probability constraint and the miss probability achieved under a fixed false-alarm-probability constraint not only have different values, they may have different exponential rates. When we have no particular choice for a reference distribution, choosing which error probability should be the focus is not only arbitrary, but also can

be misleading when analyzing classification difficulty. The Chernoff distance (equation 3) is symmetric with respect to the two probability distributions, and should be used to assess the classification problem. The optimization process (4) implicit in Chernoff distance's definition is easily determined analytically or computationally, but more calculations are needed for it than required to compute the Kullback-Leibler distance. Theoretical considerations detailed subsequently provide a streamlined, but approximate, calculation scheme that works well in applications of interest here.

In the Gaussian case (categories defined to have different means but the same variance), the Kullback-Leibler distance equals $d'^2/(2 \ln 2)$ bits and the Chernoff distance $d'^2/(8 \ln 2)$ bits, with $d' = |m_1 - m_2|/\sigma$. Thus, the Kullback-Leibler and Chernoff distances differ by a factor of four, and this difference typifies how much more slowly the probability of error decays in comparison to the false-alarm probability when the miss probability is held constant. The quantity d' is frequently used in psychophysics to assess how easily stimuli can be distinguished. When applied to non-Gaussian problems, both distance measures represent the generalization of d' to all binary classification problems: The larger these distances, the easier the classification problem. They measure how different two probability distributions are, and they, particularly the Kullback-Leibler distance, have several important properties.²

1. $\mathcal{D}(p||q) \geq 0$ and $\mathcal{D}(p||p) = 0$; $\mathcal{C}(p, q) \geq 0$ and $\mathcal{C}(p, p) = 0$.

The Kullback-Leibler and Chernoff distances are always non-negative, with zero distance occurring only when the probability distributions are the same.

2. $\mathcal{D}(p||q) = \infty$ whenever, for some x domain, $q(x) = 0$ and $p(x) \neq 0$.
3. When the underlying stochastic quantities are random vectors having statistically independent components with respect to both p and q , the Kullback-Leibler distance equals the sum of the component distances. Stated mathematically, if $p(\mathbf{x}) = \prod_i p(x_i)$ and $q(\mathbf{x}) = \prod_i q(x_i)$,

$$\mathcal{D}(p(\mathbf{x})||q(\mathbf{x})) = \sum_i \mathcal{D}(p(x_i)||q(x_i)) . \quad (5)$$

Furthermore, if p, q describe Markov data, the Kullback-Leibler distance has a similar summation property. Taking the first-order Markov case as an example, wherein $p(\mathbf{x}) = p(x_1) \prod_i p(x_{i+1}|x_i)$ and $q(\cdot)$ has a similar structure,

$$\mathcal{D}(p(\mathbf{x})||q(\mathbf{x})) = \mathcal{D}(p(x_1)||q(x_1)) + \sum_i \mathcal{D}(p(x_{i+1}|x_i)||q(x_{i+1}|x_i)) . \quad (6)$$

where

$$\mathcal{D}(p(x_{i+1}|x_i)||q(x_{i+1}|x_i)) = \int p(x_i, x_{i+1}) \log \frac{p(x_{i+1}|x_i)}{q(x_{i+1}|x_i)} dx_i dx_{i+1} \quad (7)$$

²The properties here are stated in terms of probability densities. They apply to probability mass functions as well.

The Chernoff distance obeys a similar property: When the random vectors have statistically independent components, $\max_u \log \sum_{\mathbf{x}} [p(\mathbf{x})]^{1-u} [q(\mathbf{x})]^u = \max_u \sum_i \log \sum_{x_i} [p(x_i)]^{1-u} [q(x_i)]^u$. However, the optimization must be calculated with respect to the *entire* sum, not individually. Thus, the Chernoff distance does *not* equal the sum of component distances.

4. $\mathcal{D}(p||q) \neq \mathcal{D}(q||p); \mathcal{C}(p, q) = \mathcal{C}(q, p)$.

The Kullback-Leibler distance is usually not a symmetric quantity. In some special cases, it can be symmetric (like the just described Gaussian example), but symmetry cannot, and should not, be expected. The underlying reason is that the false-alarm and miss probabilities need not have the same exponential rates. The Chernoff distance formula (1c) always yields a symmetric quantity.

5. $\mathcal{D}(p(x_1, x_2)||p(x_1)p(x_2)) = I(x_1; x_2)$.

The Kullback-Leibler distance between a joint probability density and the product of the marginal distributions equals what is known in information theory as the *mutual information* between the random variables. From the properties of the Kullback-Leibler distance, we see that the mutual information equals zero only when the random variables are statistically independent. Maximal distance and maximal mutual information occurs when the random variables equal each other, in which case the maximum equals the *entropy* of the random variable.

$$H(x) = - \int p(x) \log p(x) dx \quad \text{or} \quad H(x) = - \sum_x P(x) \log P(x)$$

The word “distance” should appear in quotes because $\mathcal{D}(\cdot||\cdot)$ and $\mathcal{C}(\cdot, \cdot)$ violate some of the fundamental properties a distance metric must have. A distance *must* be symmetric in its arguments; the Kullback-Leibler distance fails to meet this requirement. By the asymptotic results of (1), we say that $\mathcal{D}(p||q)$ is the distance from q to p , $\mathcal{D}(q||p)$ the distance from p to q . It is because false-alarm and miss probabilities may not have the same asymptotics that these two distances can differ. The Chernoff “distance” is symmetric, but does not obey the triangle inequality. Be that as it may, geometric theories of the classification problem show that *no* distance metric exists for it, and that the Kullback-Leibler distance is the distance-like quantity that should be used to assess how different two categories are [8]. The Gaussian example also indicates that both the Kullback-Leibler and Chernoff distances have the form of a *squared-distance*: When we have several statistically independent components, these distances are proportional to $\sum_i (m_1^{(i)} - m_2^{(i)})^2$, which corresponds to the square of the Euclidean distance. Thus, we have a second reason to put distance in quotes.

Both distances can be related to the ease of estimating parameters that often define the classification problem. Consider the situation where two categories differ slightly according to the values of a parameter α : symbolically, $p_{C_1} = p(\alpha)$ and $p_{C_2} = p(\alpha + \delta\alpha)$. Intuitively, if we can easily distinguish between two

such categories (small error probabilities), we should also be able to estimate the parameter accurately (the estimation error is smaller). For sufficiently small values of the difference $\delta\alpha$, the Kullback-Leibler and Chernoff distances are proportional to the reciprocal of the smallest mean-squared estimation error that can be achieved. The mathematical results are

$$\mathcal{D}(p(\alpha + \delta\alpha)||p(\alpha)) \approx \frac{1}{2}F(\alpha)(\delta\alpha)^2 \quad \mathcal{C}(p(\alpha + \delta\alpha), p(\alpha)) \approx \frac{1}{8}F(\alpha)(\delta\alpha)^2 \quad (8)$$

Here, $F(\alpha)$ denotes the Fisher information.

$$F(\alpha) = \mathcal{E} \left[\left(\frac{\partial \log p(\alpha)}{\partial \alpha} \right)^2 \right],$$

with $\mathcal{E}[\cdot]$ denoting expected value. The significance of these formulas rests in the *Cramér-Rao bound*, which states that the mean-squared error for *any* unbiased estimator $\hat{\alpha}$ of α cannot be smaller than $1/F(\alpha)$.

$$\mathcal{E} [(\hat{\alpha} - \alpha)^2] \geq \frac{1}{F(\alpha)} \quad (9)$$

When two or more parameters change, Fisher information becomes a matrix, and the distance formulas become what are known as quadratic forms.

$$\mathcal{D}(p(\boldsymbol{\alpha} + \delta\boldsymbol{\alpha})||p(\boldsymbol{\alpha})) \approx \frac{1}{2}\delta\boldsymbol{\alpha}'\mathbf{F}(\boldsymbol{\alpha})\delta\boldsymbol{\alpha} \quad \mathcal{C}(p(\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}), p(\boldsymbol{\alpha})) \approx \frac{1}{8}\delta\boldsymbol{\alpha}'\mathbf{F}(\boldsymbol{\alpha})\delta\boldsymbol{\alpha} \quad (10)$$

with $\mathbf{F}(\boldsymbol{\alpha}) = \mathcal{E} [(\nabla_{\boldsymbol{\alpha}} \log p(\boldsymbol{\alpha})) (\nabla_{\boldsymbol{\alpha}} \log p(\boldsymbol{\alpha}))']$. Here, $(\cdot)'$ means transpose and $\nabla_{\boldsymbol{\alpha}} \log p(\boldsymbol{\alpha})$ means the gradient of the log probability density function: $\nabla_{\boldsymbol{\alpha}} \log p(\boldsymbol{\alpha}) = \text{col}\{\frac{\partial}{\partial \alpha_1} \log p(\boldsymbol{\alpha}), \dots, \frac{\partial}{\partial \alpha_N} \log p(\boldsymbol{\alpha})\}$. The Cramér-Rao bound still holds, but in a more complicated form.

$$\mathcal{E}[(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})'] \geq \mathbf{F}^{-1}(\boldsymbol{\alpha})$$

What this result means is that the mean squared estimation error for any one parameter must be greater than the corresponding diagonal entry in the *inverse* of the Fisher information matrix: $\mathcal{E}[(\hat{\alpha}_i - \alpha_i)^2] \geq \mathbf{F}_{ii}^{-1}(\boldsymbol{\alpha})$. Thus for any given stimulus parameter perturbation $\delta\alpha$, the larger the Kullback-Leibler and Chernoff distances become (the further apart the distributions become), the larger the Fisher information (equation (8)), and the smallest possible mean-squared error in estimating the parameter becomes proportionally smaller. In short, larger distances mean smaller estimation errors. This relationship not only reinforces the notion that our distance measures do indeed measure how distinct two classification categories are, but also allows us to determine how well information can be gleaned from data.

Empirical classification. Instead of having a probabilistic description of the categories as in the classic classification problem, we have instead data. The scenario is that data are collected in each of J situations, and these datasets define the categories. Given a new observations, we are to classify them according to the data-derived categories. This problem formulation is similar to that of the classic artificial neural network

problem. Mimicking its jargon, we call the data defining the categories *training data*, and denote them by \mathbf{T}_j , $j = 1, \dots, J$. For simplicity, we assume that the amount of data in each training set is L_T . What we seek is the best possible empirical classifier: Which classifier produces the smallest possible classification error probabilities given *only* the availability of training data?

Gutman [17] found a classifier that is not only optimal in a certain sense, but will also, given enough training and observational data, produce error probabilities having the *same* exponential rate as the likelihood ratio classifier that clairvoyantly knows the underlying statistical model for the training data. As opposed to neural network classifiers that have an undefined structure (how many layers and nodes to solve a given problem) and that must somehow be trained, Gutman’s classifier is straightforward and computationally simple, and the training “algorithm” is efficient and encapsulates training data compactly. To apply his theory, we must assume that each observation can only assume a finite set of values: $R_l \in \{r_1, \dots, r_K\}$.³ Using the notation that $\mathbf{R} = \{R_1, \dots, R_{L_R}\}$ and $\mathbf{T}_j = \{T_1^{(j)}, \dots, T_{L_T}^{(j)}\}$, Gutman computes the test statistic G_j for each category,

$$G_j = \frac{L_T}{L_R} \mathcal{D}(\hat{P}_{\mathbf{T}_j} \| \bar{P}_{\mathbf{T}_j, \mathbf{R}}) + \mathcal{D}(\hat{P}_{\mathbf{R}} \| \bar{P}_{\mathbf{T}_j, \mathbf{R}}) \quad (11)$$

$$\bar{P}_{\mathbf{T}_j, \mathbf{R}} = \frac{L_T \hat{P}_{\mathbf{T}_j} + L_R \hat{P}_{\mathbf{R}}}{L_T + L_R} \quad (12)$$

where \hat{P} is the histogram estimate of the probability mass function

$$\hat{P}_{\mathbf{R}}(r_k) = \frac{(\# \text{ times } r_k \text{ occurs in } \mathbf{R})}{L_R} \quad \hat{P}_{\mathbf{T}_j}(r_k) = \frac{(\# \text{ times } r_k \text{ occurs in } \mathbf{T}_j)}{L_T}$$

and $\bar{P}_{\mathbf{T}_j, \mathbf{R}}$ is a weighted linear combination of the histograms computed from the data and the j^{th} training set. Thus, training data are *entirely* represented by a histogram. In information theory, such histograms are known as *types* [7: Chap. 12], and they have interesting theoretical properties we exploit here.

Gutman’s classification rule seeks not only to find the best-matching category, but also to determine if the data and training sets do indeed warrant a decision. A constant γ is chosen; if one statistic is less than this constant *and* the others exceed it, categorization is justified and we choose the category corresponding to the smallest statistic.

$$\text{Choose } i \text{ if } G_i < \gamma \text{ and } G_j > \gamma, i \neq j$$

Otherwise, the data do not warrant classification, and no choice is made. To understand the rule, we note that the quantities in (11) are Kullback-Leibler distances. Rather than computing the distance between the data and each training set, a new quantity, the weighted average of the types $\bar{P}_{\mathbf{T}_j, \mathbf{R}}$ is first derived, and the distance of each set from the weighted-average histogram is computed. If the training data for a category

³We will show later that this restriction poses no fundamental problem to us. Because we focus on finitely valued random variables, we now employ only probability mass functions in our formulas.

and the data are similar, the concatenated histogram will resemble both, and the two distances will be small. If they differ, both distances should be large. The constant γ thus indirectly defines how “close” we demand the data be from a training set *and* how different they be from all other training sets. The larger the value of γ , the more likely we are to properly classify the data. Gutman showed that the probability of making a correct decision with this rule decreased exponentially in the size of the dataset L_R with an exponential rate equal to γ . However, increasing γ arbitrarily will also increase the probability of making no decision. He showed that as long as γ is less than a certain threshold value γ_0 , the probability of making no classification *also* decreased exponentially. He further showed that no other empirical classifier can have an exponential rate of no-classification decisions larger than his. Thus, the overall performance of the Gutman classifier, as judged by exponential rates, is optimal.

The technical details here are interesting and important. Gutman’s results apply when the size of the training sets and the data grow large. In particular, we must have $\lim_{L_R \rightarrow \infty} L_T/L_R = C > 0$ for his results to hold. The interpretation of this constraint is that the training set and datasets must be sufficiently large for “good” classification performance to result. How large “large” is will depend on the problem and theoretical results from empirical classification theory hint that no general rule of thumb can be found [10]. What we have found is that moderate data and training set sizes suffice to yield exponentially decreasing error probabilities (roughly 100–1000) [19, 20]. Furthermore, Gutman showed that if $C = \infty$ —the training set size increases more rapidly than the dataset size —that the exponential rate of the empirical rule *equaled* that of the optimal clairvoyant classifier.⁴

The Gutman statistic has several important properties.

1. G is a symmetric function of the histograms derived from the training and observation data when these datasets are of equal size.
2. $0 \leq G \leq \log(1 + L_T/L_R) + (L_T/L_R) \log(1 + L_R/L_T)$.
The Gutman statistic is, like the Kullback-Leibler and Chernoff distances, non-negative, equaling zero only when the two histograms are identical. However, the distance is bounded, with the maximal value achieved when the two types do not overlap. An example of subsequent importance occurs when the datasets have equal size ($L_R = L_T$); in this special case, $0 \leq G \leq 2$.
3. G can be applied to true probability mass functions as well as to types. When $L_R = L_T$, G becomes a symmetric function of the two distributions.
4. The Gutman distance between the joint distributions of independent random variables does *not* equal

⁴Our work has found a small error in Gutman’s proof: The ratio L_T/L_R must not increase too rapidly ($\lim_{L_R \rightarrow \infty} \log(L_R + L_T)/L_R < \infty$). Essentially, increasing training set sizes too fast amounts to overtraining, and suboptimal performance will result.

the sum of the component Gutman distances. The mathematical reason is that the weighted linear combination of products, as found in (12), does not equal a product.

5. Applying Gutman’s statistic to two probability mass functions that are parametric perturbations of each other, calculations show that $G \approx \frac{1}{4}F(\alpha)(\delta\alpha)^2$.

Summary. What have we learned from this sojourn into classification theory? Suppose we set up a hypothetical classification problem: one category is that the measured response corresponds to some known probabilistic model, and the second category is that the response came from some other model. Classification theory suggests that the ability to infer a stimulus-induced change in a nominal response is related to the probability of judging incorrectly from the two responses that no change occurred. Error probabilities in both standard and empirical situations are very problem-dependent, but as the number of observations becomes large, error probabilities of optimal classifiers will *always* decay exponentially (equation 1). Because of this consistent behavior, our ability and that of optimal systems to assess change is directly related to the constant of this exponential decay. In the two-class classification problem when the classes differ by a small perturbation, the exponential decay constant is proportional to the reciprocal of the smallest possible mean-square estimation error (equation 8).

Returning to figure 1, quantifying stimulus-induced change in a response is equivalent to determining what information processing the system is performing and how well the processed information is encoded. In information theory, one of the fundamental signal processing results is the Data Processing Theorem [7: §2.8]: The quality of information representation in a system’s output cannot exceed that contained in its inputs. More precisely stated, the Kullback-Leibler distance between outputs obtained for two sets of stimulus parameters cannot exceed that of the system’s inputs: $\mathcal{D}(p_{\mathbf{Y}}(\alpha_1)||p_{\mathbf{Y}}(\alpha_2)) \leq \mathcal{D}(p_{\mathbf{X}}(\alpha_1)||p_{\mathbf{X}}(\alpha_2))$. Thus, classification and estimation problems do not become inherently easier by performing data processing, be the processing performed by neural systems or by the researcher in analyzing data. This result may seem at first confusing, but it merely means that information cannot be created by systems by applying any signal processing strategy.⁵ Feature extraction occurs when some stimulus components have *little* representation in the system’s outputs in comparison to others. In terms of our formalism, some stimulus perturbations, those extracted by the system, yield little change in distance while others, those removed, yield larger distance changes. None of these changes can exceed that contained in the inputs.

If we knew the neural code, we would have a point process model that related discharge probability to the stimulus. In terms of classification theory, this situation corresponds to classic classification, wherein we know the underlying probability distributions. To assess response difference, we would compute the Chernoff distance between the underlying probability distributions governing two response patterns or use

⁵You might think that optimal systems could be sought by maximizing the output distance. The hurdle this approach faces is that the “do nothing” system achieves the upper bound, and this solution must somehow be eliminated.

the Kullback-Leibler distance when one response can serve as the nominal response. Because we seek the neural code, the required probability distributions are precisely what we don't know. To assess neural coding, the approach we take here is to estimate the Chernoff or Kullback-Leibler distances from the data. The estimation procedure is beset by statistical issues, bias and numerical issues among them; these are addressed in a subsequent section. We can also use empirical classification theory to help analyze responses. The example we explore here (section 7.3) is to measure when a response changes because of an additional input.

3 Digital representation of neural responses

To develop a measure of the population's response, we first convert the population's discharge pattern into a convenient representation for computational analysis (figure 3). Here, a neural population's response during the b^{th} bin is summarized by a single number R_b that equals a binary coding for which neurons, if any, discharged during the bin. This procedure generalizes the approach taken for the single neuron case, wherein the occurrence of a spike in a bin was represented by a zero or a one. Note that this digitization process for a neural ensemble is reversible (up to the temporal precision of the binwidth): The sequence $\{R_b\}$ completely characterizes the population response, and the entire discharge pattern can be recreated from it. *In developing techniques to analyze neural coding, we need only consider the statistical structure of this sequence.* When we employ a periodic stimulus, the response presumably also repeats. When it does, the probability law governing the response varies with post-stimulus time, and then repeats in synchrony with the stimulus. In absolute terms, such responses are non-stationary. To capture the fact that the probability law repeats periodically, thereby justifying estimation of it by judicious averaging, such stochastic sequences are said to be *cyclostationary* [14]: At each post-stimulus time, the response has the same probability distribution from trial to trial, and this probability distribution varies with post-stimulus time. We represent such datasets by the collection of response vectors $\mathbf{R} = \{\mathbf{R}_1, \dots, \mathbf{R}_M\}$, where M denotes the number of stimulus repetitions and each component response \mathbf{R}_m equals $\{R_{m,1}, \dots, R_{m,b}, \dots, R_{m,B}\}$.

One such tool that has been used to analyze single-unit responses is the PST histogram. The number of discharges occurring in the b^{th} bin for all stimulus repetitions is used to estimate discharge rate in that bin [23]. Note that this number divided by the number of repetitions M estimates discharge probability in that bins, thus equaling the type of the response evaluated at the value $r = 1$.

$$\hat{P}_{R_b}(R_b = 1) = \frac{(\# \text{ times } R_b = 1)}{M}$$

For this single-neuron case, the only other remaining value of the type — $\hat{P}_{R_b}(R_b = 0)$ — is found by subtracting from one. *Thus, calculating a PST histogram corresponds to the calculation of a type.* The value of L , the number of observations used to compute the type, equals the number of stimulus repetitions M .

When we have a neural population, in which case R_b takes on values in $(0, \dots, 2^N - 1)$, we could

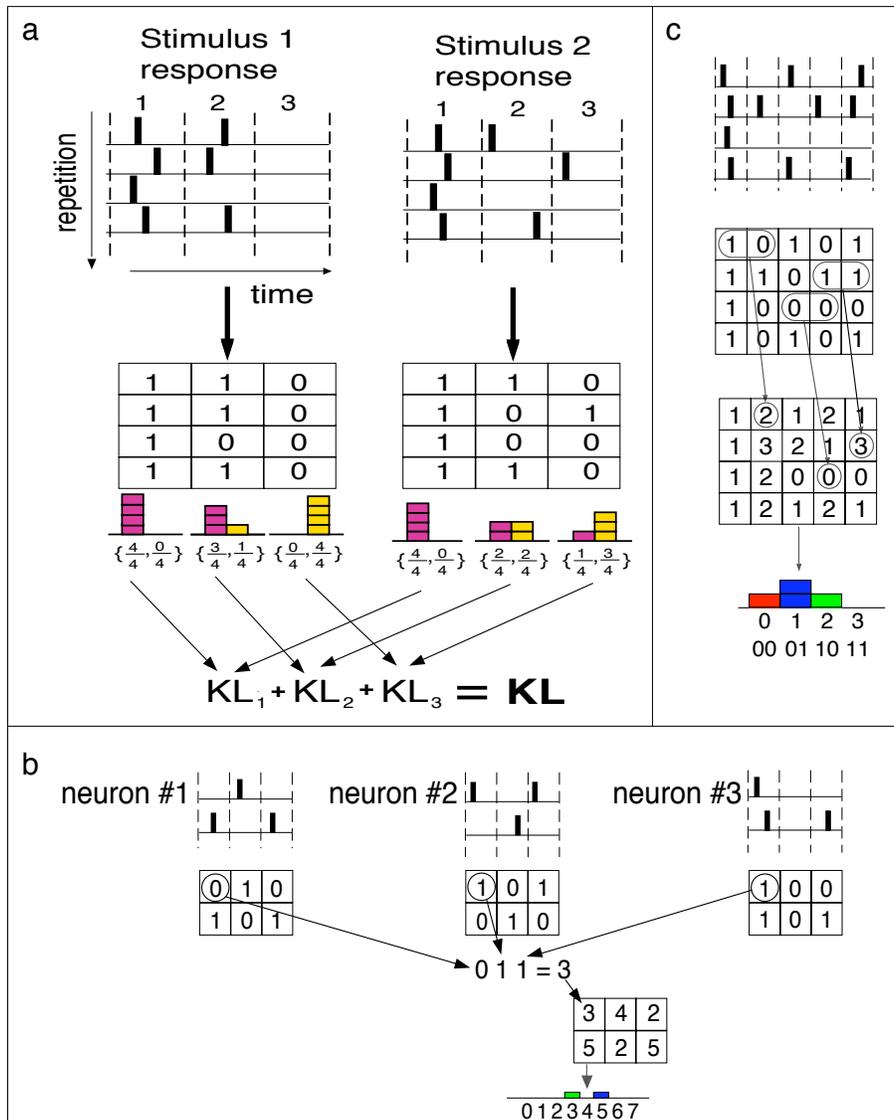


Figure 3: Panel (a) portrays how we estimate the Kullback-Leibler distance between single neuron responses to different stimuli. The response to each stimulus repetition is time aligned as in PST histogram computation, and a table is formed from the spike occurrence times (denoted by an “ \times ”) quantized to the binwidth Δ . For each bin, a “1” indicates that a spike occurred in a bin and a “0” indicates that a spike did not occur. We accumulate the type for each bin, forming a histogram of spike occurrences and nonoccurrences separately for each bin from the M stimulus presentations (four are shown in the figure). A similar set of types are computed from the responses to a second stimulus. When we assume the Markov order D to be zero, we compute the Kullback-Leibler distance between corresponding bins and sum the results. Panel (b) generalizes the computations of panel (a) to the multi-neuron case. In the depicted ensemble of three neurons, the spike pattern at any bin could be one of eight ($2^3 = 8$) possible patterns. Each possible pattern is represented by an integer between 0 and 7 in the table to the lower right. Types are formed from these quantities and distances are again computed separately between corresponding bins and summed when $D = 0$. Panel (c) illustrates how first-order distance analysis is computed. For each neuron’s (or ensemble’s) responses, the response pattern for two bins at a time is represented by an integer between 0 and 3 in the bottom table. Note that the first bin is special as no bin precedes it. This edge effect corresponds to the first term in equation (6). We compute the zeroth order distance for it and the first-order distances for the others, then sum the result to form the total distance.

perform a similar procedure to yield the multiunit PST histogram (figure 3b). The main difference is that we need to segregate the various values of R_b and estimate the probability of each value occurring in a bin. As with the single neuron situation, only nonzero values of R_b need appear in the histogram. By accumulating this multineuron PST histogram in this way, we obtain the distribution of neural discharge occurrence across the entire population within each bin. The multiunit PST histogram estimates $\Pr[R_b = k]$, $k = 0, \dots, 2^N - 1$. In the three-neuron example shown in figure 3, the value $R_b = 5$ means the first and third neurons discharged and the second did not within a given bin.

Just as in the usual PST histogram, this multiunit histogram does not faithfully represent temporal dependence that may be present in the ensemble response. The multiunit PST histogram essentially assumes responses occur independently from bin to bin —what amounts to a Poisson assumption —because no record is kept of what preceded a particular population discharge pattern in each bin. This assumption is more serious here than in the single-unit case: While departures from Poisson behavior may not be significant in the single-unit case, a discharge in one neuron may well affect another’s discharge occurring several bins later. We want our analysis techniques to be sensitive to this possibility, and go beyond the PST histogram in providing insight into the neural code. As shown in figure 3c, temporal dependence is easily incorporated into type-based distance calculations.

4 Calculating distance between responses

Let $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ represent the responses of a neural population to two stimulus conditions. What we want to measure is the distance between the *joint* probability distributions corresponding to these responses. Using the Kullback-Leibler as an example, we would want to find $\mathcal{D}(p(\mathbf{R}^{(2)})||p(\mathbf{R}^{(1)}))$. We emphasize that for this calculation to have meaning, we need the *joint* probability of the sequence of discharges occurring in each response. This computation is illustrated in figure 3. This joint probability is exceedingly difficult to calculate because we simply don’t have enough data. If we have N neurons whose cyclostationary responses are measured across B bins, the number of possible response values is 2^{NB} , and to estimate a probability for each of these values, the number of stimulus repetitions would need to exceed this typically large number. To manage this statistical complexity, we must assume that the response in a given bin depends (in the statistical and practical sense) *only* on the responses that occur in the immediately preceding D bins. This validity of this assumption can be verified in several ways. One is first principles based on neural biophysics. The second is statistical, which we describe in a subsequent section. Once this *analysis dependence order* is chosen, the distance calculation generalizes equation (6).

$$\begin{aligned} \mathcal{D}(p(\mathbf{R}^{(2)})||p(\mathbf{R}^{(1)})) &\cong \mathcal{D}(p(R_1^{(2)}, \dots, R_D^{(2)})||p(R_1^{(1)}, \dots, R_D^{(1)})) \\ &+ \sum_{i=D+1}^B \mathcal{D}(p(R_i^{(2)}|R_{i-1}^{(2)}, \dots, R_{i-D}^{(2)})||p(R_i^{(1)}|R_{i-1}^{(1)}, \dots, R_{i-D}^{(1)})) \end{aligned} \quad (13)$$

The “ \cong ” relation means that this relation is only true according to assumption, and the data’s actual dependence structure may differ. If D equals or exceeds the memory present in the responses, this equation holds: Picking D too large does not hurt. The problem arises when D is chosen too small; in this case, the two sides of equation (13) are not equal. Furthermore, mathematical analysis suggests that Kullback-Leibler distances calculated using a smaller-than-required dependence order can be smaller or larger than the actual value.

To estimate these quantities, we calculate *joint types*, which capture both temporal and spatial dependencies (figure 3). For each bin, a joint type estimates the joint probability that a given ensemble response pattern occurs in it and the preceding D bins. Extending our example, rather than just counting the number of times R_b assumes various values, we need to know the joint distribution of (R_{b-1}, R_b) to assess first-order dependence. Because the PST histogram is equivalent to zeroth-order analysis, employing joint types in measuring response differences can reveal response changes *not* revealed by the PST histogram, be it a single- or multi-unit histogram. Temporal dependence in discharge probabilities can arise in a variety of ways: among them are dependence on discharge history [22, 34, 35], non-exponential interval distributions⁶, and syn-fire response patterns in ensembles [1, 28].

Recent results in information theory [33] proscribe how much data are needed to measure a given degree of dependence:

$$D \leq \frac{\log(L+1)}{\log(2^N+1)}, \quad (14)$$

where L is the amount of averaging used in the type calculation and N is the number of neurons. For nonstationary responses L equals the number of stimulus repetitions M while for stationary responses it equals the number of bins in the measured response. This result makes the point that the amount of data needed grows exponentially in the dependence order and in the number of neurons in the ensemble: $L > 2^{D \cdot N}$. Computational experiments indicate that this bound is not particularly tight, and we do not analyze data to as high an order as the bound permits.

Similar considerations apply to the Chernoff distance. Prior to optimization, we form joint types and accumulate the components of the defining expression of equation (1c). To compute how Chernoff distance increases with time, we must recalculate the optimization with respect to u as each bin’s response is incorporated. This necessity creates a much heavier computational load than encountered in Kullback-Leibler distance calculations. We have derived relations between the Chernoff and Kullback-Leibler distance mea-

⁶This situation is particularly subtle. Even when the response can be well-modeled as a renewal process (interspike intervals are statistically independent from each other), the probability of a discharge in a bin depends on how long ago the previous discharge occurred.

tures, and have found a very good approximation to the Chernoff distance that does not require optimization.

$$\mathcal{R}(p_1, p_2) \geq \mathcal{C}(p_1, p_2) \geq \frac{1}{2}\mathcal{R}(p_1, p_2), \quad (15)$$

where $\mathcal{R}(p_1, p_2)$ denotes the so-called *resistor-average* of the two Kullback-Leibler distances.

$$\mathcal{R}(p_1, p_2) = \frac{\mathcal{D}(p_1||p_2)\mathcal{D}(p_2||p_1)}{\mathcal{D}(p_1||p_2) + \mathcal{D}(p_2||p_1)} \quad (16)$$

Note that it too is symmetric in the two underlying probability distributions, and that it requires about twice the computations as one Kullback-Leibler computation. Examples indicate that the half the resistor-average approximates the Chernoff distance to within about 10% in many situations. In empirical situations, distance estimation error usually exceeds the approximation discrepancy. We therefore compute the resistor-average distance to determine how different two responses are, saving exact computation of the Chernoff distance for those response pairs deemed particularly interesting.

5 Statistical properties

5.1 Estimation of distance measures

The most direct approach to estimating distance measures is to use types in their definitions. The Gutman distance was derived from empirical considerations, and alternative estimates need not be considered. However, estimating the most important distance measure, Kullback-Leibler distance, does present difficulties. When the type for the reference distribution has a zero-valued probability estimate for some letter at which the other type is nonzero, we obtain an infinite answer, which may not be accurate (the true reference distribution has a nonzero probability for the offending letter). To alleviate this problem, the so-called *K-T estimate* [24] is employed. Each type is modified by adding one half to the histogram estimate *before* it is normalized to yield a type. Thus, for the k^{th} letter, the K-T estimate is

$$\hat{P}_{\mathbf{R}}^{\text{KT}}(r_k) = \frac{(\# \text{times } r_k \text{ occurs in } \mathbf{R}) + \frac{1}{2}}{L_R + \frac{K}{2}}$$

Now, no letter will be assigned a zero estimate of its probability of occurrence *and* the estimate remains asymptotically unbiased with increasing number of observations. When applied to joint types, we add 1/2 to each bin and normalize according to the total number of letters in the joint type. For second-order dependence analysis, we need the joint type defined over three successive bins, and we apply the K-T procedure according to

$$\hat{P}_{\mathbf{R}}^{\text{KT}}(r_{k_1}, r_{k_2}, r_{k_3}) = \frac{(\# \text{ times sequence } r_{k_1}, r_{k_2}, r_{k_3} \text{ occurs in } \mathbf{R}) + \frac{1}{2}}{L_R + \frac{K^3}{2}}$$

This estimation procedure is not arbitrary: It is based on theoretical considerations of what *a priori* distribution for the probabilities estimated by a type sways the estimate the least.

5.2 Bootstrap removal of bias

All distance measures presented here have the property that they can only attain non-negative values, and, in addition, the Gutman distance is bounded. Any quantity having these properties cannot be estimated without bias. For example, if the true distributions are identical, distance measures are zero, but types calculated from two datasets drawn from the same distribution are unlikely to themselves be equal, which means the measured distances will be positive. Thus the estimates are biased. While the estimates are asymptotically unbiased, in our experience the bias is significant even for large datasets, and can lead to analysis difficulties. Analytic expressions for the bias of a related quantity —entropy— are known [6], and they indicate that bias expressions will depend on the underlying distribution in complicated ways.

Fortunately, recent work in statistics provides a way of estimating the bias and removing it from *any* estimator without requiring additional data. The essence of this procedure, known as the *bootstrap*, is to employ computation as a substitute for a larger dataset. The bootstrap procedure is one of several *resampling* techniques that attempt to provide auxiliary information —variance, bias, confidence intervals— about a statistical estimate. Another method in this family is the so-called jackknife method, and it has been used for removal of bias in entropy calculations [12]. The book by Efron and Tibshirani [11] provides excellent descriptions of the bootstrap procedure and its theoretical properties.

In a general setting, let $\mathbf{R} = \{R_1, \dots, R_L\}$ denote a dataset from which we estimate the quantity $\theta(\mathbf{R})$. Our quantities of interest here are the Kullback-Leibler, Chernoff, resistor-average, and Gutman distance measures. We create a sequence of bootstrap datasets $\mathbf{R}_l^* = \{R_{1,l}^*, \dots, R_{L,l}^*\}$, $l = 1, \dots, L_B$. Each bootstrap dataset has the same number of elements as the original, and is created by selecting elements from the original randomly and with replacement. Thus, elements in the original dataset may or may not appear in a given bootstrap dataset, and each can appear more than once.⁷ For example, suppose we had a dataset having four data elements $\{R_1, R_2, R_3, R_4\}$; a possible bootstrap dataset might be $\mathbf{R}^* = \{R_2, R_3, R_1, R_1\}$. The parameter estimated from the l^{th} bootstrap dataset is denoted by $\theta_l^* = \theta(\mathbf{R}_l^*)$. From the L_B bootstrap datasets, we estimate the quantity of interest, obtaining the sequence of estimates $\{\theta_1^*, \dots, \theta_{L_B}^*\}$. The suggested number of bootstrap datasets and estimates is several hundred [11].

The bootstrap estimates cannot be used improve the precision of the original estimate, but they can provide estimates of $\theta(\mathbf{R})$'s auxiliary statistics, such as variance, bias, and confidence intervals. The *bootstrap estimate of bias* is found by averaging the bootstrap estimates, and subtracting from this average the original estimate: $\text{bias} = \frac{1}{L_B} \sum_l \theta_l^* - \theta(\mathbf{R})$. The bootstrap-debiased estimate is, therefore, $2\theta(\mathbf{R}) - \frac{1}{L_B} \sum_l \theta_l^*$. Calculation of bootstrap-debiased distances can result in negative distances when the actual distance is small.

Confidence intervals of level β can also be estimated from the bootstrap estimates by sorting them,

⁷Programming the creation of a bootstrap dataset is easy. If $\text{rand}(i1, i2, L)$ is a function generating L integers ranging between $i1$ and $i2$ inclusively, $\mathbf{R}^* = \mathbf{R}(\text{rand}(i1, i2, L))$.

and determining which values correspond to the $\beta/2$ and $1 - \beta/2$ quantiles. Let $\{\theta_{(1)}^*, \dots, \theta_{(L_B)}^*\}$ denote the sorted (from smallest to largest) estimates. A raw confidence interval estimate corresponds to $[\theta_{(\lfloor L_B - \beta L_B / 2 \rfloor)}^*, \theta_{(\lceil \beta L_B / 2 \rceil)}^*]$. Thus, for the 90% confidence interval, $\beta = 0.9$, and the raw confidence interval corresponds to the 5th and 95th percentiles. Because we want confidence intervals on the bootstrap-debiased estimate rather than the original, we reverse the interval and center it around the debiased estimate: $[2\theta(\mathbf{R}) - \theta_{(\lceil \beta L_B / 2 \rceil)}^*, 2\theta(\mathbf{R}) - \theta_{(\lfloor L_B - \beta L_B / 2 \rfloor)}^*]$.

5.3 Dependence on binwidth

Ideally, the calculation of distance measures between two responses would not depend on the binwidth Δ used in the digitization process. However, discharge probability at any specific time varies as binwidth varies. Since distances measure how different two probability distributions are, we expect that distance calculations do depend on binwidth. To analyze this situation, let's assume a single neuron population, with the probability of an event equaling some rate times the binwidth: $\Pr[R_b = 1] = \lambda\Delta$ and $\Pr[R_b = 0] = 1 - \lambda\Delta$. The Kullback-Leibler distance between two such random variables (having rates λ_1 and λ_2) is given by

$$\mathcal{D}(\lambda_2 || \lambda_1) = \lambda_2 \Delta \log \frac{\lambda_2 \Delta}{\lambda_1 \Delta} + (1 - \lambda_2 \Delta) \log \frac{1 - \lambda_2 \Delta}{1 - \lambda_1 \Delta}.$$

The first term is clearly proportional to binwidth; if we assume that the discharge probability is small ($\lambda\Delta \ll 1$), then the total expression is proportional to the binwidth.

$$\mathcal{D}(\lambda_2 || \lambda_1) \approx \left(\lambda_2 \log \frac{\lambda_2}{\lambda_1} + \lambda_1 - \lambda_2 \right) \Delta$$

All the other distances are also proportional to binwidth when $\lambda\Delta \ll 1$.

When we accumulate the distance across bins that span a given time interval having duration T , as suggested in property 3 and equation (13), the number of bins equals T/Δ . If the discharge rates are such that discharge probabilities are small, the accumulation *over a given time interval* cancels the binwidth dependence, which leaves the accumulated distance independent of the binwidth. In this way, the accumulated distance will not depend on how we digitize the population response. For this reason, we prefer plotting accumulated distance (as expressed in equations (5) and (13) in the independent and Markov cases respectively) across the response. To judge how much each response over some time interval contributes to the total distance, we need only subtract the final and beginning values of the distance accumulated over the interval in question.

More subtle dependencies of distance calculations on binwidth can also occur. To take into account any temporal dependencies, we want to calculate distances spanning as much time as possible. Because the statistical bound on analysis dependence (equation 14) is expressed in bins, not time, it behooves us to use large binwidths. Counteracting this preference are the desires to obtain as much temporal detail as

possible and to avoid obtaining multiple spikes within a bin. Thus, choosing binwidth is a somewhat more complicated issue than in PST histogram calculations.

6 Basic analysis procedure

The goal of information-theoretic distance analysis is to compute the distance between responses. We explore several ideas on how these distance calculations can be used to measure and assess the neural code in a subsequent section. In all of these, the basic procedure is as follows.

1. Given sets of individual or simultaneous recordings, the analysis of the population’s response begins with the digitization process described in section 3. The important consequence of this procedure is that single and multi-unit recordings have a common data representation.
2. Compute the joint type of user-specified order D , employing the K-T modification if the Kullback-Leibler distance is needed.

$$\hat{P}_{R_b, \dots, R_{b-D}}(r_0, \dots, r_D) = \frac{(\#\text{times } R_b = r_0, \dots, R_{b-D} = r_D) + \frac{1}{2}}{L + 2^{D \cdot N - 1}}$$

3. Compute the Kullback-Leibler, Chernoff, or Gutman distance using the Markov decomposition expressed in equation (13). The conditional distribution needed in this computation is found from the joint type by a formula that mimics the definition of conditional probabilities.

$$\hat{P}_{R_b | R_{b-1}, \dots, R_{b-D}}(r_0 | r_1, \dots, r_D) = \frac{\hat{P}_{R_b, \dots, R_{b-D}}(r_0, r_1, \dots, r_D)}{\sum_r \hat{P}_{R_b, R_{b-1}, \dots, R_{b-D}}(r, r_1, \dots, r_D)}$$

4. Use the bootstrap debiasing and confidence interval procedure on the distance thus calculated. When analyzing cyclostationary responses, we consider the responses to individual stimulus presentations as the fundamental “data quantum.” Our bootstrap samples are drawn from this collection of M datasets.
5. Our examples plot the cumulative debiased distance as each term is accumulated (using an expression similar to that of equation (13)).

Figure 4 illustrates a simple application of this procedure for simulated (Poisson) single-neuron discharges. The two ways of computing the Kullback-Leibler distance from the simulated responses differ substantially. We find that this difference is statistically significant, and occurs frequently in simulations and in actual recordings. Consequently, to compare two responses, we use the Chernoff distance or its resistor average approximation. The resistor average depicted in the top right panel consists of a series of straight lines, which correspond to time segments of constant rate differences between the responses. The greater slopes correspond to greater rate differences. Note that when the rates are equal, the distances do

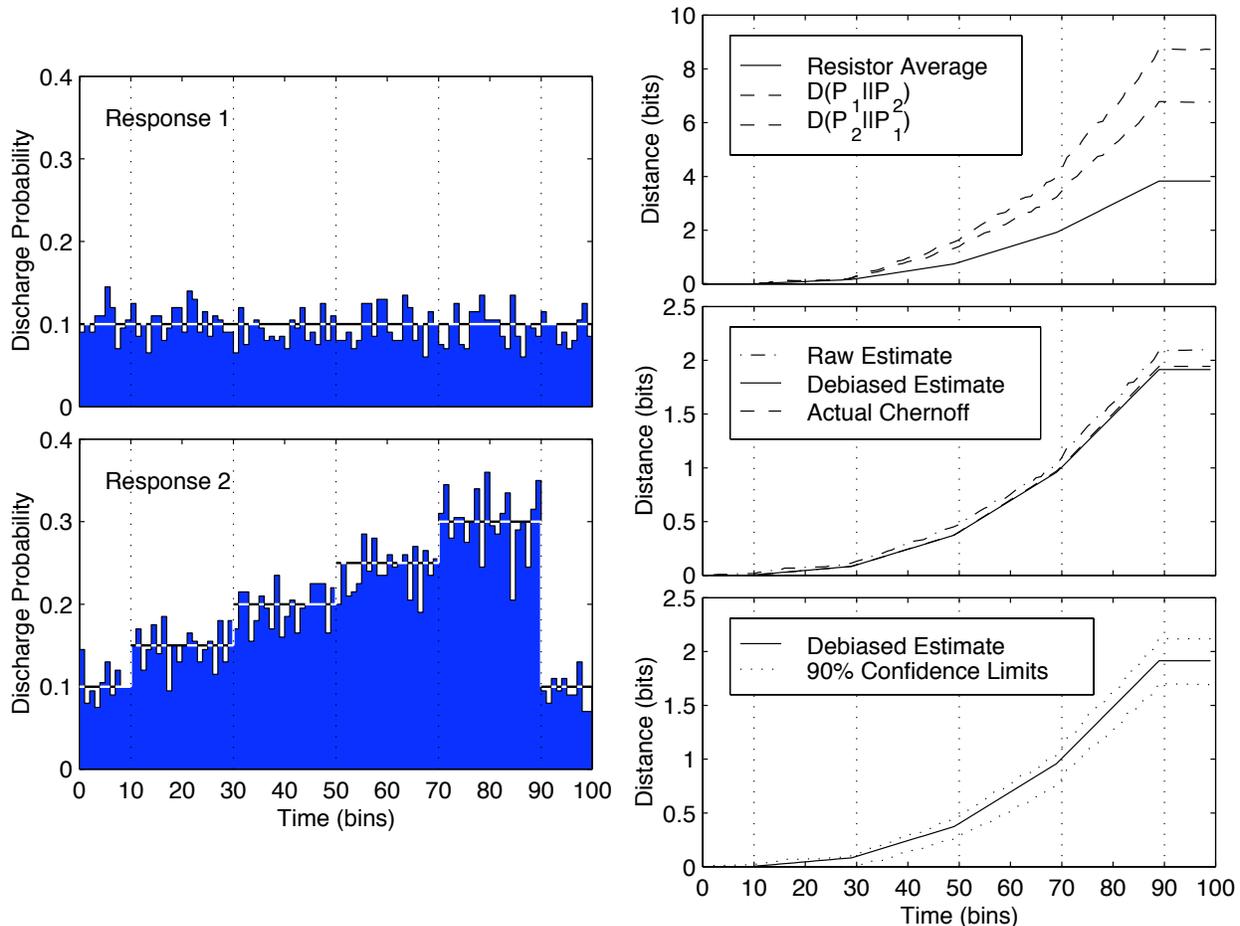


Figure 4: Single-neuron responses were simulated based on a Poisson discharge model. The first response had a constant rate, and the second response was a staircase; these constitute an example chosen to illustrate type-based analysis. These two responses equaled each other during the initial and final ten bins. The discharge probabilities controlled the occurrence of $M = 200$ simulated responses. The resulting PST histograms are shown, with the actual discharge probability shown by dashed lines and the dotted vertical lines indicating when rate changes occurred. The right column displays the various information-theoretic distance measures calculated from these responses. The top panel shows the accumulated Kullback-Leibler distances estimated with the K-T modification using each response as the reference (dashed lines), along with the resistor-average of these two shown (solid line). All of these were debiased using the bootstrap. In the middle panel, the resistor-average (scaled by two) before (dot-dashed) and after (solid) applying the bootstrap is compared with the theoretical Chernoff distance (dashed). The bottom panel again shows the debiased resistor-average (again scaled by two) along with the 90% confidence limits (dotted) estimated via the bootstrap. In all cases, two hundred bootstrap samples were used.

not change, indicating no response difference. The middle plot shows that the bias in the initial estimate of the resistor average is quite large. We have found the bootstrap bias compensation procedure described in section 5.2 to be necessary for obtaining accurate distance estimates. To employ bootstrap in the cyclostationary case, we consider our dataset to consist of the responses to individual stimulus presentations for a given parameter setting: $\mathbf{R}(\alpha) = \{\mathbf{R}_1(\alpha), \dots, \mathbf{R}_M(\alpha)\}$, and our bootstrap datasets contain M responses selected randomly from this original. We independently perform the bootstrap on each response resulting from each stimulus condition, compute types from each bootstrap sample, and calculate the distance be-

tween these samples. As illustrated in figure 4, the bootstrap substantially removes the inherent positive bias. Also shown in this panel is that half the resistor-average distance quite closely approximates the actual Chernoff distance between the responses. Examining the bottom right panel of figure 4 shows that the actual Chernoff distance lies well within the 90% confidence interval. Hence, we use the computationally simpler resistor-average distance measure. The confidence interval widens as we progress across the response. This effect occurs naturally because we are adding more and more statistical quantities calculated for each bin as we accumulate the total distance. These intervals would be substantially smaller if we accumulated distance only over portions of the response.

To interpret this distance calculation, we refer to modern classification theory reviewed in section 2. Because Chernoff distance is related through equation (1c) to the classification error rate, it reveals how easily the two responses can be distinguished: The bigger the distance, the smaller the probability of an error in distinguishing the two. Note that this error probability is known only up to a constant: We cannot compute it precisely. Asymptotic error probability changes with time roughly according to $2^{-d(t)}$, where $d(t)$ is the accumulated distance, be it Kullback-Leibler or Chernoff, and t is post-stimulus time. Thus, each unit (one bit) increase in distance corresponds to a factor of two smaller error probability. The accumulation of distance with time is not an arbitrary choice. This procedure corresponds to the Kullback-Leibler distance's property 3, which states that the distance between the joint probability distributions characterizing a response over a given number of bins equals the sum of the component distances.

As the two responses are identical over the first ten bins, no distance is accumulated. As the rates differ more in each twenty-bin section, we see that the distance accumulated in each section increases. In this example, the accumulated distance increases from the beginning to the end of each section are 0.1, 0.3, 0.55, and 0.95 bits. These quantities were calculated by subtracting the accumulated distance at the beginning from its value at the end. Finally, the responses have identical rates during the last ten bins, and we see distance does not increase further. When we analyze responses, we concentrate on those portions of the response that contribute most to accumulated distance since they provide the most effective coding (in terms of classification errors). In our simple example, the response during bins 70–90 contributes most because the rate difference is greater there. As we consider more complicated examples of coding, it becomes increasingly important that we can use type-based analysis to determine important sections of the response *without* assuming the nature of the code.

Figure 5 portrays how choice of analysis order can affect distance calculations. Recall that exploring nonzero analysis orders amounts to seeking response differences *not* conveyed by the PST histogram. During the first few milliseconds, no significant response differences are evident. After about 5 ms, significant differences occur, with the various choices of analysis orders yielding about the same result. These distances then depart at about 7 ms, with the $D = 4$ curve being significantly larger. This result indicates significant

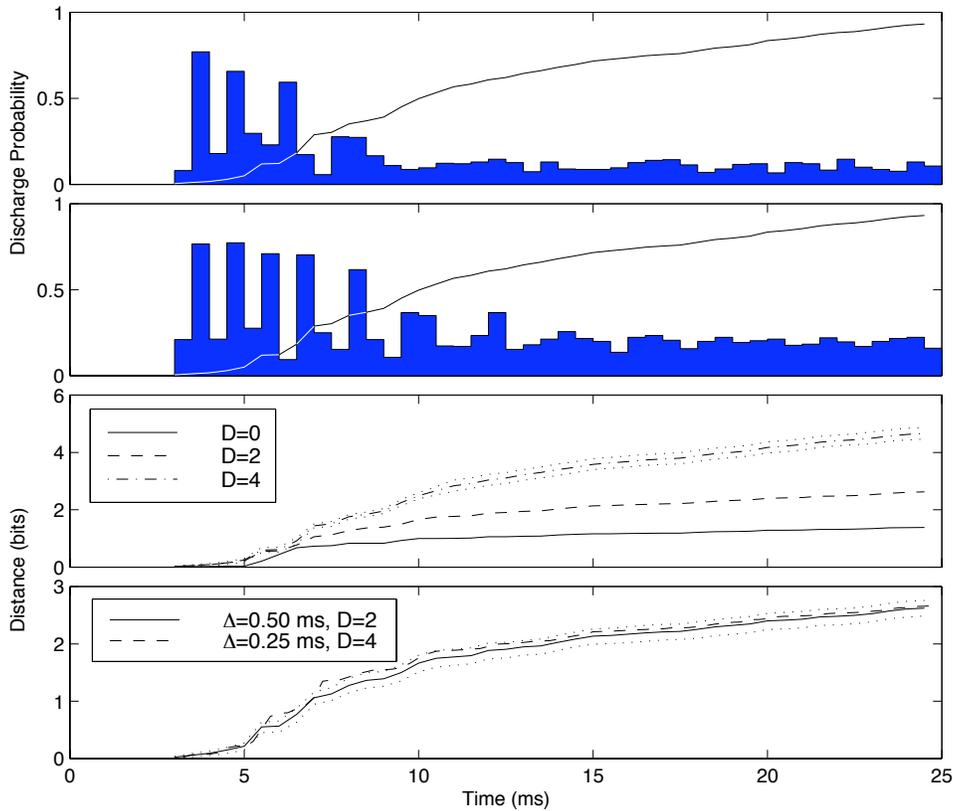


Figure 5: The upper panels show the PST histograms of a simulated lateral superior olive neuron’s response to two choices of stimulus level (binwidth equals 0.5 ms). The simulations modeled the neuron’s biophysics [36]. The bottom panels show the resistor-average distance between these two responses; the computations were performed under several conditions. The first of these shows the resistor-average distance (divided by two) between these responses computed for $D = 0, 2, 4$ bins (corresponding to 0, 1, and 2 ms of temporal dependence, respectively). The dotted lines straddling the $D = 4$ curve portray the 90% confidence interval. The curve superimposed upon the PST histograms is the $D = 4$ curve. Finally, the bottom plot displays the resistor-average distance (divided by two) between the responses for two choices of binwidth, but with the dependence parameter D chosen so that the assumed temporal dependence for each spans the same time interval. The 90% confidence interval for the $\Delta = 0.5$ ms is displayed with dotted lines.

temporal dependence in the responses as it differs greatly from the $D = 0$ curve, which always corresponds to assuming the data are statistically independent from bin to bin. The value of dependence parameter D is one of the few assumptions our information-theoretic approach must take. Ideally, all values that can be computed based on the amount of available data (equation 14) should be explored. As D increases, the distance calculations will eventually not change, and the best value for the dependence parameter is the smallest of these. In the example portrayed in figure 5, the resistor-average distance kept increasing, leaving us no choice but to use the largest possible value. Using the $D = 4$ result, the distance between the responses increases most sharply during the second portion of the transient response. Note that during the latter portion of the response the distance measures increase roughly linearly. This effect usually indicates a difference in sustained rates, which can be discerned from the PST histograms. Furthermore, about half the total distance accumulated over 25 ms (4.65 bits) is garnered in the first 10 ms. We conclude that the initial

transient of the response allows equal discriminability in the first 10 ms (actually 7 ms as there is about a 3 ms latency) as does the response obtained during the last 13 ms. Thus, the initial portion of the response conveys as much about the stimulus as does the latter portion in less time.

This result also illustrates our general finding that the distance measures smooth response variations found in PST histograms. Although the displayed responses came from simulations, actual recordings also demonstrate rapid rate oscillations. The ability of the distance measures to assess response differences without regard to whether response rates are time varying or not is one of our analysis technique's most powerful features.

Binwidth effects are also demonstrated in figure 5. From the example shown there, we conclude that the larger binwidth of 0.5 ms would suffice as joint types computed over the same time span but with different binwidths yield nearly the same results. The time epochs over which the distance calculations disagree most occurs during the high-probability-of-discharge segments of both responses, a result consistent with the analysis of section 5.3.

7 Applications

7.1 Assessing neural codes

The simplest application of distance analysis is assessing which part of the response changes significantly as with stimulus changes. Perhaps the most powerful aspect of type-based analysis is that it makes no *a priori* assumption about the nature of neural encoding. Calculating response distance quantifies how well the code expresses stimulus changes regardless of its form, whether it be a timing code, a rate code, or some combination of these. "Significant change" has two meanings here. The first is whether the distance measure is significantly different from zero during some portion of the response. Inferring this statistical significance is the role for confidence intervals, which we compute using the bootstrap. The second type of significance is which portion of the response contributes most to accumulated distance. We judge this by computing how much distance changes over a given time interval. One consequence of making this kind of calculation is that we can *directly* evaluate one response component's importance relative to another's. For well-defined portions of the response, like the initial transient and later sustained response that typifies auditory neuron responses to tone bursts, we can directly compare how different portions are. Furthermore, the cumulative distance reveals how long it takes to yield a certain level of discrimination. We can then begin to answer questions such as how long it takes to determine from the population's response a just noticeable stimulus change.

An example of this analysis for the single neuron case is displayed in figure 4. Figure 6 illustrates the applying this approach to a simple population of three neurons. Both a stimulus-induced rate response and a transneuronal correlation can be detected, and the relative contribution of each response component to sensory

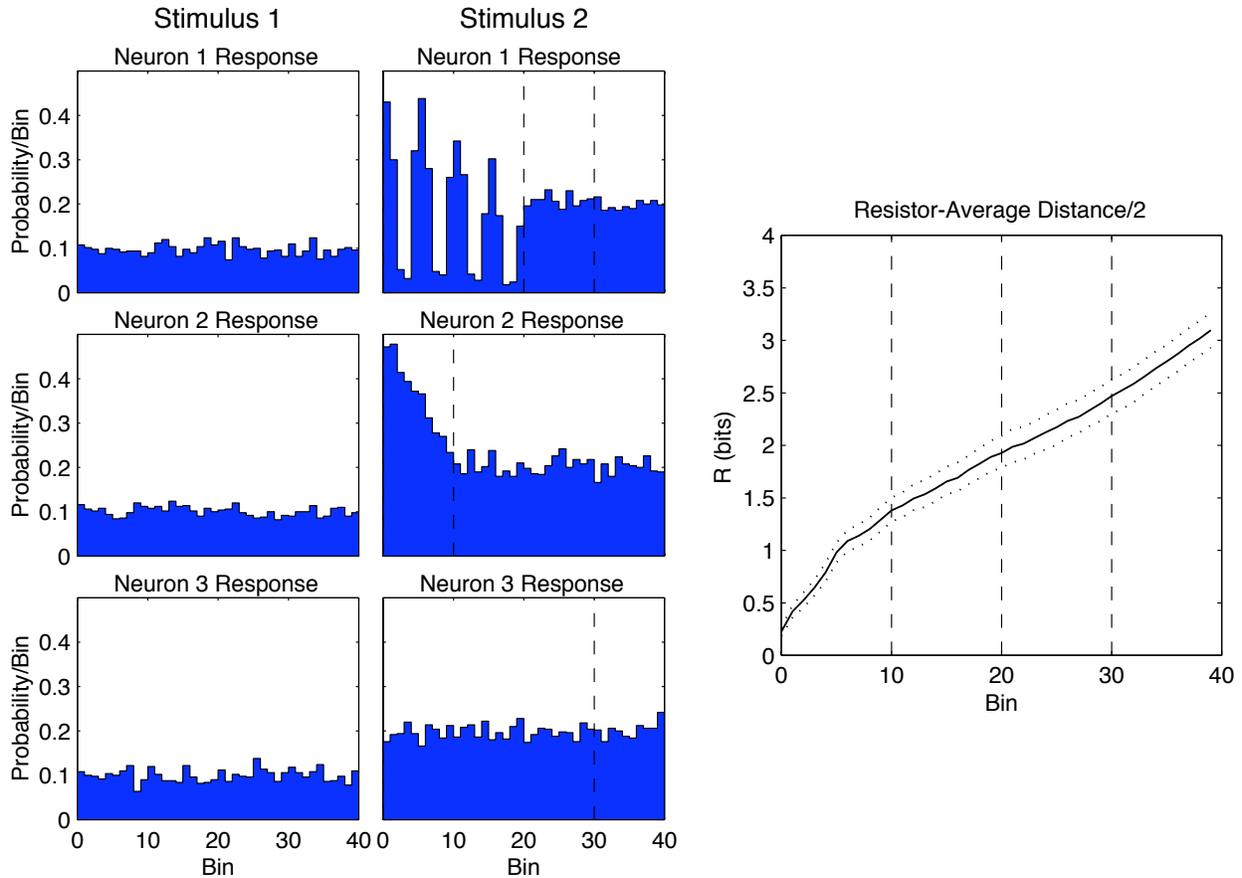


Figure 6: We simulated a three-neuron ensemble responding to two stimulus conditions. The left portion of the display shows PST histograms of each neuron. As far as can be discerned from these histograms, the first stimulus yielded a constant-rate response in each simulated neuron. The second stimulus produced different responses in each neuron: The first had an oscillatory response lasting 20 bins, the second a transient rate increase for 10 bins, and the third a rate change. The dashed vertical lines in the PST histograms indicate the boundaries of these various response portions. During the first stimulus, and until the last ten bins of the second stimulus, the neurons produced discharges statistically independent of the others. In the last ten bins, the first and third neurons' discharges became correlated (coefficient = 0.6). Throughout all responses, the responses were produced by a first-order Markov model having a correlation coefficient of -0.1 . The right panel shows the result of computing the resistor-average distance between the two responses. The solid line shows half the resistor-average distance, with its 90% confidence interval shown with a dotted line. Dashed vertical lines correspond to stimulus 2 response components.

discrimination quantified. Clearly, the initial portion of the response produced the greatest distance change. During the next ten bins, when the latter portion of neuron #0's oscillatory response and the rate responses of the other two are present, about 0.5 bits of distance were gained. This increase means that the probability of not being able to discriminate between the two stimulus conditions decreased by a factor of about $2^{0.5} = 1.4$. A much larger change (1.4 bits) occurred during the first ten bins. Consequently, the first portion of the response contributes much more to stimulus discrimination than the second. The third portion of the response contains only constant discharge rates. The distance accumulated during this time (bins 20–29) roughly equals the distance accumulated during the previous ten bins, when neuron #1's response contained an oscillatory component. This equality of accumulated distance means that the oscillatory response and

the constant-rate response are equally effective in representing the stimulus difference. Interestingly, the introduction of spatial correlation increased only slightly the accumulated distance beyond what the rate response by itself would have.

7.2 Uncovering neural codes

The calculation of distances between responses quantifies neural coding without revealing what the code is. Distance calculations can offer some insights as well into what aspects of the response contribute to the code. For example, we can determine the presence of correlation in an ensemble’s response, be it stimulus- or connectivity-induced. In the former case, spike trains can be correlated merely because neurons are responding to the same stimulus. In the latter, the neurons receive common inputs or are interconnected. We compute the type of the measured ensemble response and derive from it the type that would have been produced by the ensemble if it had statistically independent members (spatial dependence) and/or had no temporal dependence. Referring to figure 3 for an example, the probability of each neuron discharging in each bin can be calculated from the joint probability of various response patterns occurring in a bin (e.g., $\Pr[\text{discharge in neuron \#1}] = \Pr[R_n = 4] + \Pr[R_n = 5] + \Pr[R_n = 6] + \Pr[R_n = 7]$). From these component probabilities, we estimate the probability of all possible ensemble response patterns by multiplying according to the ensemble response the probabilities of each neuron discharging or not ($\Pr[R_n = 3] = \Pr[\text{nodischarge in neuron \#1}] \cdot \Pr[\text{discharge in neuron \#2}] \cdot \Pr[\text{discharge in neuron \#3}]$). By calculating the distance between these two types, we can infer when correlated responses are present; figure 7 illustrates an example.

In this analysis, we can also use the Kullback-Leibler distance directly. It equals the mutual information between the component discharge patterns of the population (property 5). Zero mutual information corresponds to statistically independent responses. We note that from an information transfer viewpoint, statistically independent responses do not always correspond to the best situation [15]. As this measure increases, the discharge patterns must be more interdependent, with achieving the maximum value meaning that the discharge patterns are identical.

7.3 Measuring response latency

Frequently, we have the situation wherein two responses are the same up to some time, then differ in some way. In our case, a suddenly applied stimulus evoked the responses, which meant that the both were cyclostationary with unknown time-varying statistics during the entire response course. While we could have computed the cumulative distance between the types underlying the histogram and determined when this distance exceeded a threshold, we opted for a technique that is more attuned to a persistent response difference, not a momentary one. To look for a persistent difference—a *change*—we designed a two-stage based on empirical classification and distributed detection ideas. This problem can be approached by

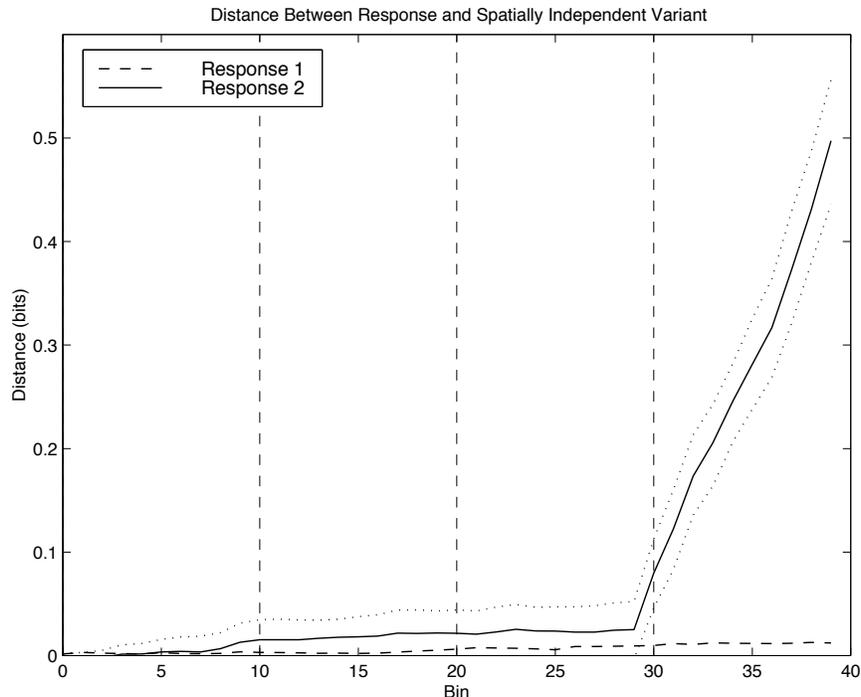


Figure 7: The resistor average (divided by two) between the type computed from response and the type computed derived from it that forces a spatially independent ensemble response structure is shown for the two stimulus conditions used in figure 6. The dashed line shows the result for the first response (histograms shown in the left column of figure 6), the solid line for the second (center column). As was simulated, the responses to stimulus 1 demonstrated no transneural correlation. The second stimulus did induce a correlation in the latter portion of the response, and the distance clearly indicates the presence of such correlation. The 90% confidence interval for the second response is indicated by the dotted lines. Note that the confidence interval’s lower edge was less than zero for the first 30 bins.

the classic change-detection solution [4]. Here, the traditional approach is to consider this problem as a multihypothesis hypothesis testing problem, with each hypothesis indicating the index at which the change occurs. However, this approach does not work well in our case because the probability distributions before the change are usually assumed known. Because of the time-varying nature of the response to each stimulus repetition, the required probability laws are not known *a priori*.

A new approach based on *distributed detection* demonstrates better results (figure 8). For each bin, a front-end detector decides whether or not $R_b^{(1)}$ and $R_b^{(2)}$ are commonly distributed, and indicates that decision to a fusion center. The b^{th} detector’s output d_b is “0” if they are determined to commonly distributed, “1” if they are not. The fusion center must estimate from these individual decisions where the change C occurred. Note that the fusion center needs only to solve a simple change-point problem, since the bank of detectors has removed the nonstationarities *if* the detector performance probabilities can be made uniform. Because we do not know the probabilities of correct and incorrect decisions, the more difficult task is designing the front-end detectors.

We use a type-based empirical detector for the front-end processors. Here, the types of each pair $R_b^{(1)}$ and $R_b^{(2)}$ are compared to judge whether or not their members are commonly distributed using Gutman’s

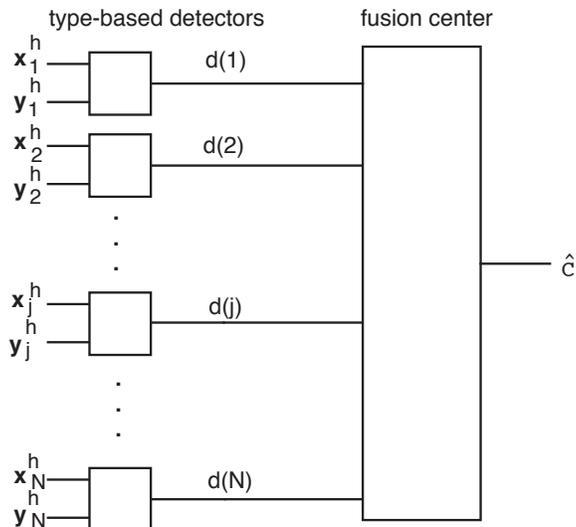


Figure 8: The algorithm consists of two stages. First, type-based detectors take the observations from each response, and decide at each bin whether or not the observations are emitted from the same or different sources. These decisions are fed a fusion center, which then estimates \hat{C} , the change point.

classifier. If this decision rule is used, the false-alarm probability P_F is *guaranteed* to satisfy a performance standard that does *not* depend on the underlying distributions [17]: $\lim_{L \rightarrow \infty} \frac{1}{L} \log P_F \leq -\gamma$. This standard is controlled parametrically by our choice for the constant γ .

Calling \mathbf{d} the vector of individual bin decisions, our fusion center takes the collected decisions and calculates the estimate \hat{C} . We have considered two possible fusion centers. In the threshold fusion center, we choose the first non-zero location in the vector \mathbf{d} as the change estimate. This simple method does not work very well because isolated false alarms often cause bad estimates. Such errors could be reduced by controlling γ ; as with any detector strategy, imposing too small a choice for P_F will lead to small detection probabilities, which means each detector becomes too conservative, never announcing a difference. A better method is to choose \hat{C} based on a least-squares fit of a unit step to the vector \mathbf{d} . An example of results using this algorithm is shown in figure 9.

7.4 Uncovering feature extraction

As part of developing these new techniques, we reexamined how the signal processing function of any system should be assessed. Consider a nonlinear, adaptive system —a neural ensemble —that accepts inputs and produces outputs (as shown in figure 1), and about which we have only general insight into the system’s function (for example, it processes acoustic information). Assume that the inputs depend on a collection of stimulus parameters represented by the vector α . Curiously, knowing the system’s input-output relation may not be helpful in understanding its signal processing function: Nonlinear systems are just too complicated. Our approach examines response sensitivity to stimulus changes and derives from it the ability of an optimal signal processing system to estimate the stimulus parameters. The key idea underlying this approach is the

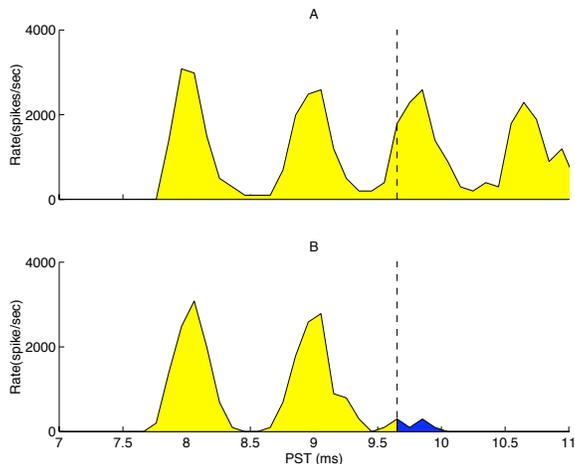


Figure 9: The top panel shows the PST histogram of an LSO unit’s response to a tone burst present to one ear. The typical oscillatory response (called chopping) occurring during the first few milliseconds is shown. The bottom panel shows the response of the same unit when the other ear was also stimulated. The algorithmically estimated change point indicated at 9.7 ms, as indicated by the dashed line.

perturbational result of equation (8), which relates distance measure changes to the Fisher information matrix.

In our approach, we measure responses recorded in response to a reference stimulus parameterized by α_0 and a family of responses parameterized by $\alpha_0 + \delta\alpha$, with $\delta\alpha$ a perturbation [16]. We compute types from ensemble responses to both stimuli and quantify the “distance” between them. We use the Kullback-Leibler distance in this application since it yields larger values for a given perturbation (equation 8) and we have a natural choice for a reference response. Figure 10 shows the surfaces generated by perturbing two stimulus parameters —sound amplitude and azimuthal location of the sound —about a reference stimulus. Interestingly, our responses, obtained from accurate biophysical simulations of binaurally sensitive lateral superior olive (LSO) neurons [36], indicate that during different portions of the response, the two stimulus features are coded with differing fidelity. We measure fidelity as the ability (standard deviation of error) of an optimal system to estimate the stimulus parameters from the response. Early on, the transient response encodes both stimulus features well. Twenty milliseconds later, the fidelity of angle encoding remains about the same, although the form of the response has changed from a transient to a gradual rate change. During this period, the amplitude encoding has greatly worsened, with the standard deviation increasing by over a factor of five. During the constant-rate portion of the response starting 20 ms later, the amplitude estimate has worsened more with the angle estimate’s quality remaining about the same.

What these results indicate is that this LSO neuron is processing its inputs (which greatly resemble the primary neural outputs of the two ears) in such a way that stimulus amplitude and angle are encoded in its response [16]. In short, the neuron’s discharge pattern multiplexes stimulus information. The fidelity of this representation changes rapidly with time after stimulus onset, with the azimuth being the primary stimulus

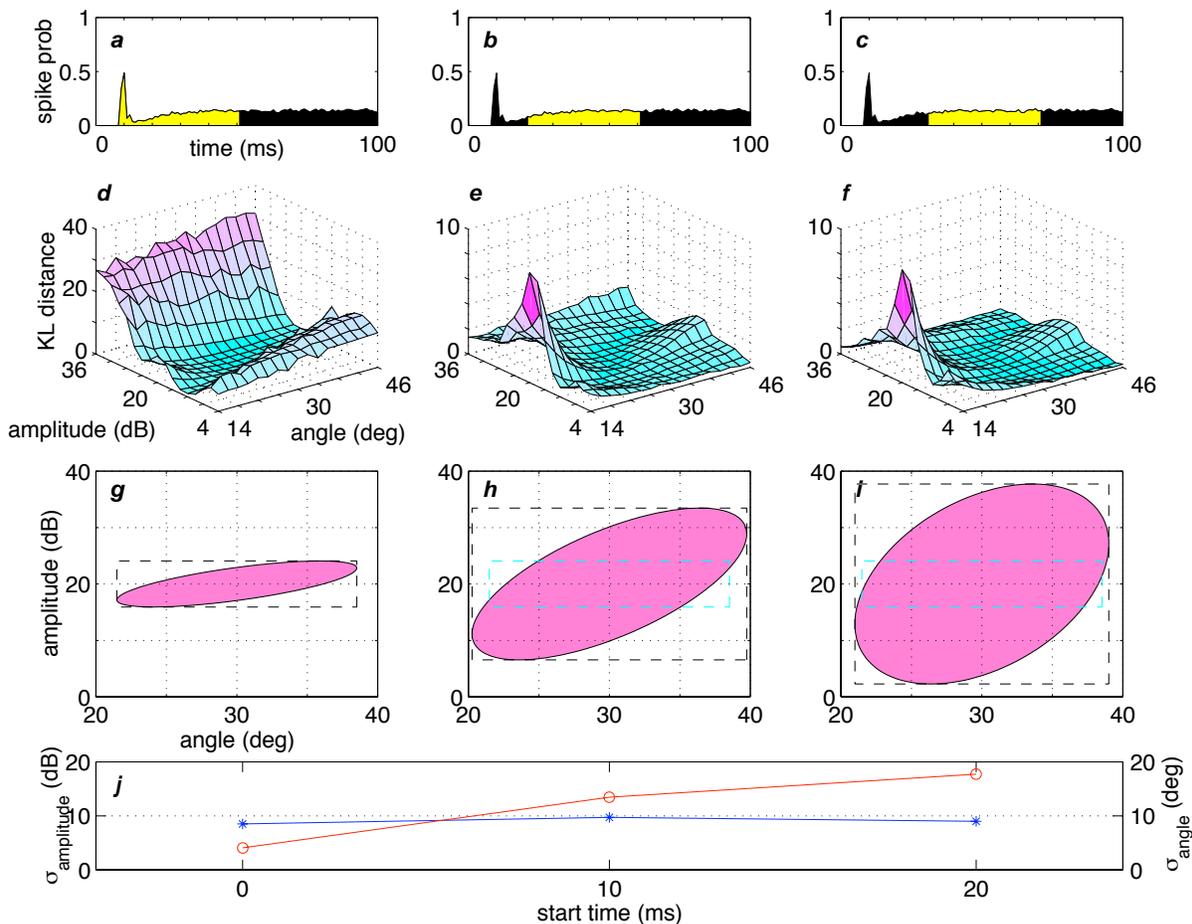


Figure 10: We simulated [36] the response of a single lateral superior olive neuron [30, 31] to high-frequency tone bursts presented at various amplitudes and azimuthal locations. The top row (panels a-c) shows the PST histogram of the simulated response at the reference condition (20 dB, 30°). The light areas in each indicate the 40 ms portion of the response subjected to type-based analysis. The next row (panels d-f) shows three-dimensional surfaces of the corresponding values of Kullback-Leibler distance between the reference response and the responses resulting from varying stimulus amplitude and angle. From these surfaces, we fit a two-dimensional third-order polynomial, and used its parabolic terms to estimate the elements of the Fisher information matrix according to equation (8). The inverse of this matrix provides lower bounds on estimates of angle and amplitude derived from the analyzed portion of the response. Panels g-i show sensitivity ellipses that trace one standard deviation that an optimal system would yield if it estimated amplitude and angle simultaneously. The horizontal and vertical extents of these ellipses correspond to the standard deviations of angle and amplitude estimates, respectively, and these determined the rectangles shown in each panel. Panel g’s rectangle is repeated in the other panels for comparison. The bottom panel shows how these standard deviations changed during the response. The circles indicate the standard deviation of the amplitude estimate (left vertical scale) and the asterisks the standard deviation of the angle estimate (right scale). Taken from [16].

attribute encoded in the response. Thus, the information coding provided by this neuron’s discharges is multidimensional and time-varying. The initial portion of the response could be used along with other neural responses in the auditory pathway to estimate stimulus amplitude, but later portions are less useful. We would conclude the primary, but not only, role for the lateral superior olive, is sound localization.

Because these results were obtained from simulations, we have access to the inputs as well, and these can be subjected to the same analysis to understand not only how well our two stimulus attributes are represented

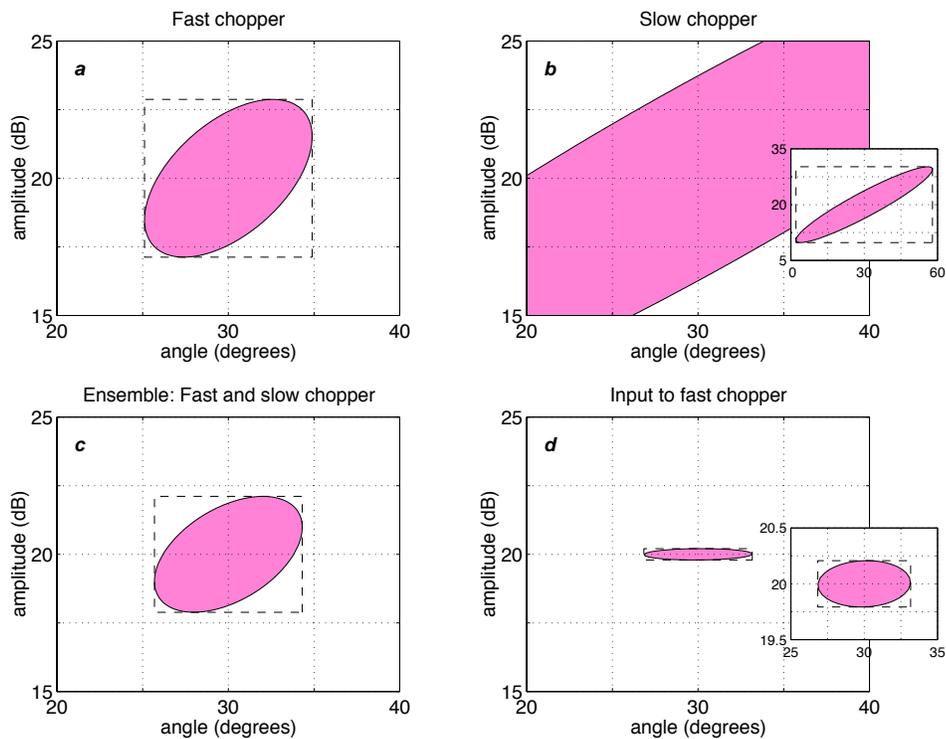


Figure 11: Sensitivity ellipses were computed for the entire response, which lasted 100 ms. Panel a corresponds to the fast-chopper simulation for which results are shown in figure 10. Panel b shows the results of a slow chopper, which clearly encodes both amplitude and angle information much less well than the fast chopper. The inset shows the entire sensitivity ellipse. Some of the fast chopper’s inputs served as all the slow chopper’s inputs. Panel c shows the sensitivity ellipse for the two-neuron ensemble comprised of these two simulated neurons. The sensitivity ellipse computed from the total input to these neurons is shown in panel d, with a detailed view shown as an inset. The amplitude extent of the input sensitivity ellipse is much less than that of the ensemble’s outputs, an indication that the neurons removed much amplitude information found in their inputs. In contrast, the angle extents are comparable, meaning that the ensemble is preserving the angle information contained in its inputs. Taken from [16].

by the neuron’s inputs, but also how well the LSO is processing these inputs to produce azimuthal estimates. As shown in figure 11, the inputs provide much better estimates of both stimulus parameters than does a single LSO neuron [16]. Because of the Data Processing Theorem in information theory [7: §2.8], no system can yield an output containing more fidelity about a parameter than contained in its inputs. Thus, the fact that amplitude and angle are better represented in the neuron’s inputs is not surprising; the dramatic decrease in output fidelity for both parameters does surprise.

Neurons in the LSO are not homogeneous, with at least three response classes defined [22]. We must therefore consider the population as well. Typed-based analysis can be applied to the population of neurons to judge how well the population codes stimulus information. We simulated a two-neuron ensemble consisting of two LSO unit types (so-called fast and slow choppers), with the slow chopper’s input also projecting to the fast chopper. The fast chopper had additional inputs. The results, shown in figure 11, indicate that this simple ensemble does produce a combined output having a fidelity of angle coding comparable to that

found in the ensemble’s input. The fidelity of amplitude coding decreases much more (by a factor of four) from input to output. Thus, the LSO as a population provides the more central portions of the auditory pathway with as much azimuthal location information as is possible, and removes to some degree other stimulus features. Our information theoretic analysis thus quantifies both feature extraction and the quality of information representation.

8 Conclusions and discussion

We have presented several information theoretic distance measures that can be used to quantify neural coding, and described techniques that exploit them. These distances depend on the probabilistic descriptions of the neural discharges, about which we want to assume as little as possible. The theory of types suggests that empirical estimates of these distributions can be used to accurately compute these distance measures, with the sole modeling assumption being the Markov dependence parameter. Given sufficient data, this parameter can also be determined solely by the data. The examples we have presented here, particularly the feature extraction one, demonstrate that neural information coding can be quite complex, being both time-varying and expressed by both discharge timing and rate. Thus, any technique for assessing information coding fidelity must make as few assumptions as possible; the type-based analysis described here fulfills that criterion.

A second powerful aspect of our approach is its ability to cope with ensemble responses. As shown in figure 3, the analysis technique can conceptually be applied to any sized population. The information bound (equation 14) suggests that the amount of data required for a given level of analysis grows *exponentially* in the number of neurons. In practical terms, our technique can be used only for small populations. However, since the bound is procedure independent, any technique that makes no assumptions about the structure of the data is subject to the same limitations.

For judging coding quality, we prefer the Chernoff distance. Because of its computational complexity, we use the resistor-average distance to approximate it. The Kullback-Leibler distance, despite its theoretical importance, is difficult to use empirically because it is asymmetric with respect to the two component responses. We used it in the stimulus perturbation analysis because a natural reference response emerges, and it is the simplest computationally to estimate. The Gutman statistic must be used when we apply classification techniques to neural responses. Gutman’s empirical classification theory [17] suggest that no other *empirical* classification algorithm (no access to the true distributions) can produce better classification results (in the sense of exponential rates of classification error probabilities). This result and the properties of Gutman’s classifier can be used in several ways. First of all, its properties place limits on the performance levels competing techniques can provide. More importantly, Gutman’s classification algorithm is quite simple, and the way in which training data are used is explicit and simply and concisely represented by types. We use it to precisely measure when a response changes from a reference. We exploited a detailed

property of Gutman's classifier: Its performance levels are controlled by us and don't depend greatly on the underlying probability distributions.

The procedures we have described here can assess neural coding, but they do not directly reveal what the code is. We showed one approach to assessing transneural correlation in figure 7. In general, coding mechanisms can be inferred from the component types; precisely how we have not yet determined. Be that as it may, the information-theoretic procedures developed here offer flexible but computationally intense analysis techniques that can meaningfully quantify the nature of neural coding.

9 References

1. M. Abeles. *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press, New York, 1991.
2. M. Abeles and M. H. Goldstein, Jr. Multispikes train analysis. *Proc. IEEE*, 65:762–773, 1977.
3. J.-M. Alonso, W. M. Usrey, and R. C. Reid. Precisely correlated firing in cells of the lateral geniculate nucleus. *Nature*, 383:815–818, 1996.
4. M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes*. Prentice Hall, 1993.
5. W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, and D. Warland. Reading a neural code. *Science*, 252:1852–1856, 1991.
6. A. G. Carlton. On the bias of information estimates. *Psychological Bulletin*, 71:108–109, 1969.
7. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
8. A.G. Dabak. *A Geometry for Detection Theory*. PhD thesis, Rice University, Houston, TX, 1992.
9. R.C. deCharms and M.M. Merzenich. Primary cortical representation of sounds by the coordination of action-potential timing. *Nature*, 381:610–613, 1996.
10. L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer-Verlag, New York, 1996.
11. B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.
12. R. M. Fagan. Information measures: Statistical confidence limits and inference. *J. Th. Biol.*, 73:61–79, 1978.
13. F. Gabbiani and C. Koch. Coding of time-varying signals in spike trains of integrate-and-fire neurons with random threshold. *Neural Computation*, 8:44–66, 1996.
14. W. A. Gardner. An introduction to cyclostationary signals. In *Cyclostationarity in Communications and Signal Processing*, chapter 1. IEEE Press, New York, 1994.
15. C. M. Gruner and D. H. Johnson. Correlation and neural information coding efficiency. In *Computational Neuroscience*, Santa Barbara, CA, 1998.
16. C. M. Gruner and D. H. Johnson. Time-varying, multiplexed feature coding in the lateral superior olive. *Nature*, 1998. Submitted.

17. M. Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Trans. Info. Th.*, 35:401–408, 1989.
18. R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Prentice Hall, Englewood Cliffs, NJ, fifth edition, 1995.
19. D. H. Johnson et al. Empirical classification based on types. *Signal Processing Letters*, 1997. Submitted.
20. D. H. Johnson, Y. K. Lee, O. E. Kelly, and J. L. Pistole. Type-based detection for unknown channels. In *ICASSP Proc.*, Atlanta, GA, 1996.
21. D. H. Johnson and G. C. Orsak. Performance of optimal non-Gaussian detectors. *IEEE Trans. Comm.*, 41:1319–1328, 1993.
22. D. H. Johnson, C. Tsuchitani, D. A. Linebarger, and M. Johnson. The application of a point process model to the single unit responses of the cat lateral superior olive to ipsilaterally presented tones. *Hearing Res.*, 21:135–159, 1986.
23. D.H. Johnson. Point process models of single-neuron discharges. *J. Computational Neuroscience*, 3:275–299, 1996.
24. R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Trans. Info. Th.*, IT-27:199–207, 1981.
25. J. C. Middlebrooks, A. E. Clock, L. Xu, and D. M. Green. A panoramic code for sound location by cortical neurons. *Science*, 264:842–844, 6 May 1994.
26. M. I. Miller and D. L. Snyder. *Random Point Processes in Space and Time*. Springer-Verlag, New York, second edition, 1991.
27. H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, New York, 1988.
28. A. Riehle, S. Grun, M. Diesmann, and A. Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278:1950–1953, 1997.
29. F. Rieke, D.A. Bodnar, and W. Bialek. Naturalistic stimuli increase the rate and efficiency of information transmission by primary auditory efferents. *Proc. R. Soc. Lond. B*, 262:259–265, 1995.
30. C. Tsuchitani. The inhibition of cat lateral superior olivary unit excitatory responses to binaural tone bursts: I. The transient chopper discharges. *J. Neurophysiol.*, 59:164–183, 1988.
31. C. Tsuchitani. The inhibition of cat lateral superior olivary unit excitatory responses to binaural tone bursts: II. The sustained discharges. *J. Neurophysiol.*, 59:184–211, 1988.
32. J. D. Victor and K. P. Purpura. Metric-space analysis of spike trains: theory, algorithms and applications. *Network: Comput. Neural Sys.*, 8:127–164, 1997.
33. M. J. Wienberger, J. J. Rissanen, and M. Feder. A universal finite memory source. *IEEE Trans. Info. Th.*, 41:643–652, 1995.

34. M. Zacksenhouse, D. H. Johnson, and C. Tsuchitani. Excitatory/inhibitory interaction in the LSO revealed by point process modeling. *Hearing Res.*, 62:105–123, 1992.
35. M. Zacksenhouse, D. H. Johnson, and C. Tsuchitani. Excitation effects on LSO unit sustained responses: Point process characterization. *Hearing Res.*, 68:202–216, 1993.
36. M. Zacksenhouse, D. H. Johnson, J. Williams, and C. Tsuchitani. Single-neuron modeling of LSO unit responses. *J. Neurophysiol.*, 1998. To appear.