

Article

Function Identification in Neuron Populations via Information Bottleneck

S. Kartik Buddha ¹, Kelvin So ², Jose M. Carmena ^{2,3} and Michael C. Gastpar ^{1,2,*}

¹ School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne, BC 106, Station 14, CH-1015 Lausanne, Switzerland; E-Mail: buddha@swisslitho.com (S.K.B.)

² Department of EECS, University of California, Berkeley, CA 94720-1770, USA; E-Mails: sokelvin@eecs.berkeley.edu (K.S.); carmena@eecs.berkeley.edu (J.M.C.)

³ Helen Wills Neuroscience Institute and the UCB/UCSF Joint Graduate Group in Bioengineering, University of California, Berkeley, CA 94720, USA

* Author to whom correspondence should be addressed; E-Mail: michael.gastpar@epfl.ch; Tel.: +41-21-693-7523; Fax: +41-21-693-6770

Received: 26 February 2013; in revised form: 27 March 2013 / Accepted: 22 April 2013 /

Published: 6 May 2013

Abstract: It is plausible to hypothesize that the spiking responses of certain neurons represent *functions* of the spiking signals of other neurons. A natural ensuing question concerns how to use experimental data to infer what kind of a function is being computed. Model-based approaches typically require assumptions on how information is represented. By contrast, information measures are sensitive only to relative behavior: information is unchanged by applying arbitrary invertible transformations to the involved random variables. This paper develops an approach based on the information bottleneck method that attempts to find such functional relationships in a neuron population. Specifically, the information bottleneck method is used to provide appropriate compact representations which can then be parsed to infer functional relationships. In the present paper, the parsing step is specialized to the case of remapped-linear functions. The approach is validated on artificial data and then applied to recordings from the motor cortex of a macaque monkey performing an arm-reaching task. Functional relationships are identified and shown to exhibit some degree of persistence across multiple trials of the same experiment.

Keywords: information theory; information bottleneck method; neuroscience

1. Introduction

Information measures have been used frequently in neuroscience research mainly for answering questions about neural coding. Neural coding is a fundamental aspect of neuroscience concerned with the representation of sensory, motor, and other information in the brain by networks of neurons. It characterizes the relationship between external sensory stimuli and the corresponding neural activity in the form of time-dependent sequences of discrete action potentials known as *spike trains* [1]. Information theory addresses issues similar to the ones posed in neural coding, such as: How is information encoded and decoded? What does a response (output) tell us about a stimulus (input)? It is therefore used as a general framework in neural coding for measuring how the neural responses vary with different stimuli (see e.g., [2,3]). In classical neuroscience experiments, the responses of a single neuron to several stimuli are recorded and information-theoretic tools are used to quantify neural code reliability by measuring how much information about the stimuli is contained in neural responses.

New measurement techniques such as implanted tungsten micro-wire arrays lead to larger datasets. They simultaneously measure the neural activity of multiple neurons. Consequently, on datasets of this nature, additional questions pertaining to the network behavior of the neurons can be asked. Statistical methods based on information measures such as mutual information and directed information have been used to estimate fundamental properties from the data. For example, considerable research has concerned the redundancy present in neural populations, see e.g., [4–8] and for the experimental data setup considered in this paper, a redundancy study was presented in [9]. Another recent example is the use of directed information to infer causal relationship between measured neurons (without any physiological side information concerning, for example, monosynaptic connections), see [10]. For the experimental data setup considered in this paper, a directed information study was presented in [11].

In this paper, we explore a novel application of information measures in neuroscience. Consider the spiking signals of three neurons: we seek to explore whether one of those neurons might represent a *function* of the other two neurons. Note that we are not assuming any knowledge or facts about the synaptic connectivity of these neurons, such as direct monosynaptic connections. More abstractly and generally, given samples from three processes, can we find evidence that one of these processes represents a function of the other two? This question should not generally be expected to have a clear answer since for repeated applications of the same input configuration, we would typically not expect to observe the same output. Often, the question has to be posed in a relaxed setting: Does one of the processes *approximately* represent a function of the others? If so, what function would this be? In the present paper, we refer to this problem as *function identification*.

Many methods can be employed to find interesting solutions to the function identification problem. For starters, we might restrict attention to a discrete and finite dictionary of functions and ask whether one of these functions provides a good match. It is clear in this case, as long as we are not worried about computational complexity, that we can simply try all functions in the dictionary. Then, using some measure of distance, we can select the best match, and assess how close of a fit this represents. One approach to measure distance could be to impose a parametric probabilistic model, where the parameters would model the best function match as well as the characteristics of the disturbance (the “noise”). For example, for neural responses, one could impose a so-called generalized linear model for the conditional

distribution of the firing patterns of one neuron, given two (or more) other neurons. The parameters of the model could then be selected via the usual maximum-likelihood approach, as done for example in [11]. Many other probabilistic models may be of interest. A related interesting approach to this problem has been developed in [12].

A common property of all model-based approaches is that they require assumptions to be made concerning how the observed signals represent information. To illustrate this issue, suppose that both input processes as well as the output process take values in a discrete set, which we will refer to as their *representation alphabet*. Let the ground truth be that the output process is simply a weighted sum of the two input processes, subject to additive noise. Then, given samples of all three processes, it is easy to find the weights of the sum by, for example, applying linear regression. Now, however, suppose that the output process is subject to an arbitrary permutation of the representation alphabet. In this revised setting, not knowing what permutation was applied, it is impossible for linear regression to identify the weights of the sum. The deeper reason is that linear regression (and all related model-based approaches) must assume a concrete meaning for the letters of the representation alphabet.

By contrast, if we look at this problem through the lens of information measures, we obtain a different insight: the information between the input processes and the output process is unchanged by the permutation operation, or in fact, by arbitrary invertible remappings. It is precisely this feature that we aim to exploit in the present paper. Specifically, we tackle this problem through the lens of the *information bottleneck* (IB) method, developed by Tishby *et al.* in [13]. We note that the IB method has been used previously to study, understand, and interpret neural behavior, see e.g., [14,15]. At a high level, the IB method produces a compact representation of the input processes with the property that as much information as possible is retained about the output process. This compact representation depends only on the relative (probabilistic) structure, rather than on the absolute representation alphabets. In this sense, the IB method provides a non-parametric approach to the function identification problem.

To be more specific, the proposed method for function identification proceeds as follows: In the first step, the data (e.g., the spiking sequences of the three neurons) is digested into probability distributions. In this paper, we achieve this simply via histograms, but one could also use a parametric model for the joint distribution and fit the model to the data. In the second step, the information bottleneck method is used to produce a “compact” representation of two of the processes with respect to the third. In the third step, this compact representation is parsed to identify functional behavior, if any. In the fourth and final step, the closeness of the proposed fit is numerically assessed in order to establish whether there is sufficient evidence to claim the identified function.

For the scope of the present paper, we consider simplified versions of the third step, the parsing of the compact representation that was found by the IB method. Specifically, throughout the paper, we consider what we refer to as *remapped-linear* functions. By this, we mean that we allow each of the input processes to be arbitrarily remapped, but assume that the output process is a weighted sum of these (possibly remapped) input processes, subject to a final remapping. A main contribution of the present paper is the development of algorithms that take the compact representation provided by the IB method and output good fits for the weights appearing in the sum. Our algorithms are of low complexity and scale well to larger populations.

When we apply our method to spiking data, we preprocess the spiking responses by partitioning the time axis into bins of appropriate width, and merely retain the number of spikes for each bin. Hence, the data in this special case takes values in the non-negative integers. Therefore, in this application, we further restrict the weights appearing in the sum to be non-negative integers, too.

2. The Function Identification Problem

The function identification problem can be made precise in a number of ways. For the purpose of the present paper, we phrase it in a probabilistic manner. We consider a multivariate distribution of $(n + 1)$ random variables, denoted by $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ and Y . We denote the distribution by $p(\mathbf{x}, y)$, and we think of X_1, X_2, \dots, X_n as the inputs, and of Y as the output. Then, the function identification problem consists in finding a function $f(X_1, X_2, \dots, X_n)$ such that

$$Y \approx f(\mathbf{X}) \quad (1)$$

The key question, of course, concerns the approximation in this expression: How closely do Y and $f(\mathbf{X})$ agree with each other? Many approaches can be envisioned. For example, one could aim to find the function $f(\mathbf{X})$ that minimizes the mean-squared error $\mathbb{E}[(Y - f(\mathbf{X}))^2]$. This is a classical problem from estimation theory whose solution can be expressed as $f(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. It is not generally possible to evaluate this expectation in a useful fashion, and it is also not clear whether the mean-squared error is a meaningful criterion for the function identification problem. In the present paper, we will use an information criterion to determine the goodness of the functional fit. That is, we aim to select the function $f(\mathbf{X})$ in such a way as to maximize $I(f(\mathbf{X}); Y)$.

3. Approach via Information Measures

The problem of finding functional relationships encompassed by a set of random variables can be tackled in different ways. In this paper, we attempt to solve the problem stated in Section 2 with the help of information measures such as mutual information, see e.g., [16]. The mutual information $I(f(\mathbf{X}); Y)$ between $f(\mathbf{X})$ and Y can be perceived as a quantitative measure for evaluating the degree of closeness between $f(\mathbf{X})$ and Y . Therefore, $I(f(\mathbf{X}); Y)$ can be set as an objective function to be maximized over functions f .

By itself, this criterion is not useful—the trivial solution $f(\mathbf{X}) = \mathbf{X}$ (*i.e.*, the identity function) maximizes the mutual information without revealing any functional structure. The key to making this approach meaningful is to constrain the function to be as compact as possible. For the scope of the present paper, we consider compactness also via the lens of information measures. A first, intuitively pleasing measure of compactness is to simply impose an upper bound constraint on the cardinality of the function $f(\mathbf{X})$, *i.e.*, the number of different output values the function has. If we denote this cardinality by $|f(\mathbf{X})|$, we can express the function identification problem as

$$\max_{\substack{f: \\ |f(\mathbf{X})| \leq \Gamma}} I(f(\mathbf{X}); Y) \quad (2)$$

This problem does not appear to have a simple (algorithmic) solution. The key step enabling the method proposed in this paper is to (temporarily) *generalize* the notion of a function. To this end, let us identify the function value with a new random variable Z , defined as follows:

$$Z = f(\mathbf{X}) \tag{3}$$

At this point, it is natural to study the conditional probability distribution $p(z|\mathbf{x})$. As long as we have $Z = f(\mathbf{X})$, this conditional probability distribution is degenerate— $p(z|\mathbf{x})$ is equal to 1 whenever $z = f(\mathbf{x})$, and zero otherwise. Therefore, a tempting relaxation of the original problem is to optimize over *general* conditional distributions $p(z|\mathbf{x})$.

The next question is how to phrase the cardinality constraint in Equation (2) in this new setting. Simply constraining the cardinality is not necessarily meaningful: it is acceptable for Z to assume many different values, as long as most of them occur with small probability. Hence, an intuitively pleasing option might be to constrain the entropy $H(Z)$, but this does not appear to lead to a tractable solution. Another option is to constrain the mutual information term

$$I(Z; \mathbf{X}) = H(Z) - H(Z|\mathbf{X}) \tag{4}$$

which can also be interpreted as capturing the compactness of the mapping (see Remark 1 below) and is sometimes referred to as the *compression-information*. Thus, we arrive at the following formulation:

$$\begin{aligned} \max_{\substack{p(z|\mathbf{x}): \\ I(Z; \mathbf{X}) \leq \Gamma}} I(Z; Y) \end{aligned} \tag{5}$$

This formulation is precisely the problem known as the IB method. There exist several algorithms to solve for the maximizing conditional distribution $p(z|\mathbf{x})$, see e.g., [17].

Remark 1 (Intuitive Interpretation of $I(Z; \mathbf{X})$ as Compactness). Using the Asymptotic Equipartition Property (AEP) [16], the probability $p(\mathbf{x})$ assigned to an observed input will be close to $2^{-H(\mathbf{X})}$ and the total number of (typical) inputs is $\approx 2^{H(\mathbf{X})}$. In that sense, $2^{H(\mathbf{X})}$ can be seen as the *volume* of \mathbf{X} . Also, for each (typical) value z of Z , there are $2^{H(\mathbf{X}|Z)}$ possible \mathbf{x} input values which map to z , all of which are equally likely. To ensure that no two input vectors map to the same z , the set of possible inputs \mathbf{x} has to be divided into subsets of size $2^{H(\mathbf{X}|Z)}$, where each subset corresponds to a different value of Z . Thus, the average cardinality of the mapping (partition) of \mathbf{X} is given by the ratio of the volume of \mathbf{X} to that of the mean partition:

$$\frac{2^{H(\mathbf{X})}}{2^{H(\mathbf{X}|Z)}} = 2^{I(Z; \mathbf{X})} \tag{6}$$

By this reasoning, the quantity $I(Z; \mathbf{X})$ can be intuitively seen as a measure of compactness of Z . Lower values of $I(Z; \mathbf{X})$ correspond to a more compact Z and higher values for $I(Z; \mathbf{X})$ correspond to higher cardinalities of the functional mapping between \mathbf{X} and Z .

Remark 2 (Limitation of Information Measures for Function Identification). Define Z' as follows:

$$Z' = \mathcal{G}(\gamma Z) \tag{7}$$

where \mathcal{G} is a uniquely invertible one-to-one mapping and $\gamma \in \mathbb{R}$ is a constant.

For any Z' defined in such a way, $I(Z; Y) = I(Z'; Y)$. This result is trivial for the case where all the involved variables are discrete as there would be a one-to-one mapping between \mathcal{Z} (the support set or *alphabet* of Z) and \mathcal{Z}' (the alphabet of Z'), making $p(z, y) = p(z', y)$ and thus $I(Z; Y) = I(Z'; Y)$. This result also holds true for the continuous case due to the following argument:

If Z' is a *homeomorphism* (smooth and uniquely invertible map) of Z and $J_Z = \|\partial Z / \partial Z'\|$ is the Jacobian determinant of the transformation, then

$$p(z') = J_Z(z')p(z) \text{ and } p(z', y) = J_Z(z')p(z, y) \quad (8)$$

which gives

$$\begin{aligned} I(Z'; Y) &= \int \int dz' dy p(z', y) \log \frac{p(z', y)}{p(z')p(y)} \\ &= \int \int dz dy p(z, y) \log \frac{p(z, y)}{p(z)p(y)} \\ &= I(Z; Y) \end{aligned} \quad (9)$$

This result implies that tackling the function estimation problem using an approach involving mutual information cannot uniquely determine the function we are after. The solution we can expect to obtain is a class of equivalent functions that can be transformed from one to another through uniquely invertible maps.

4. Algorithm Using the Information Bottleneck Method

4.1. The IB Method

The basic idea of the Information Bottleneck (IB) method, originally introduced by Tishby *et al.* [13] is as follows: assuming that the joint probability distribution $p(x, y)$ of two random variables X and Y is known, we are interested in finding a compressed representation (or quantized codebook) for X , say Z , which is as informative as possible about the random variable Y . In this paper, we use an extension of the IB method for n input variables: $\mathbf{X} = (X_1, \dots, X_n)^T$ instead of one input variable X .

This compressed representation Z of \mathbf{X} is characterized through a conditional probability distribution $p(z|\mathbf{x})$ that gives a mapping between the values of X and Z . Each value of \mathbf{X} is associated with all the values taken by the random variable Z , according to this conditional probability. Intuitively, this approach can be viewed as squeezing the information that the multivariate random variable \mathbf{X} provides about the random variable Y through a *bottleneck* formed by a limited set of codewords Z . The IB method offers a fundamental trade-off between the complexity of a model and its precision which are respectively reflected by the extent of compression of \mathbf{X} and the amount of information the compressed variable Z preserves about Y .

As mentioned in Equation (5), the IB method can be formulated as an optimization problem where we maximize the relevant information $I(Z; Y)$ while constraining the compression-information $I(Z; \mathbf{X})$

below some maximal value. Equivalently, the same problem can be formulated as a minimization problem where we minimize $I(Z; \mathbf{X})$ while preserving $I(Z; Y)$ above some minimal level as follows:

$$\min_{\substack{p(z|\mathbf{x}): \\ I(Z;Y) \geq \Gamma}} I(Z; \mathbf{X}) \tag{10}$$

where Γ is a parameter which lower bounds the relevant information $I(Z; Y)$ while minimizing the compression-information $I(Z; \mathbf{X})$.

The IB objective function $I(Z; \mathbf{X})$ is a concave function of $p(\mathbf{x})$ for fixed $p(z|\mathbf{x})$, and a convex function of $p(z|\mathbf{x})$ for a fixed $p(\mathbf{x})$. Therefore, this is a constrained minimization problem of a convex function over the convex set of all $p(z|\mathbf{x})$ which satisfy the lower bound constraint on the relevant information $I(Z; Y)$. This is a variational problem that can be solved by introducing Lagrange multipliers, β for the relevant information constraint and $\lambda(\mathbf{x})$ for the normalization of the conditional distributions $p(z|\mathbf{x})$ at each \mathbf{x} . Proceeding along similar lines as in [13] we arrive at the following set of self-consistent equations in order to solve for the mapping $p(z|\mathbf{x})$:

$$p(z|\mathbf{x}) = \frac{p(z)}{Z(\mathbf{x}, \beta)} e^{-\beta D_{KL}[p(y|\mathbf{x})||p(y|z)]}, \quad \forall \mathbf{x}, \forall z \tag{11}$$

$$p(z) = \sum_{\mathbf{x}} p(\mathbf{x})p(z|\mathbf{x}) \tag{12}$$

$$p(y|z) = \frac{1}{p(z)} \sum_{\mathbf{x}} p(\mathbf{x}, y, z) = \frac{1}{p(z)} \sum_{\mathbf{x}} p(\mathbf{x}, y)p(z|\mathbf{x}) = \sum_{\mathbf{x}} p(y|\mathbf{x})p(\mathbf{x}|z) \tag{13}$$

This is a formal solution since $p(z)$ and $p(y|z)$ on the right hand side of Equation (11) are implicitly determined using $p(z|\mathbf{x})$ (Equations (12) and (13)). The final solution in Equation (11) along with these two equations, self-consistently determine the optimal solution. An iterative algorithm can then be used for obtaining $p(z|\mathbf{x})$ using these three equations with the joint distribution $p(\mathbf{x}, y)$, the cardinality of Z and the trade-off parameter β as inputs for the algorithm. A convenient application of this iterative algorithm involves achieving the trade-off between precision and complexity by restricting the cardinality of Z and choosing high values for β . The next subsection discusses few heuristics for estimating the functional relationship between \mathbf{X} and Y once we compute this mapping $p(z|\mathbf{x})$.

4.2. Parsing $p(z|\mathbf{x})$ for Function Identification

Given an input random vector \mathbf{X} and an output variable Y , Section 4.1 outlines a procedure using the information bottleneck method for finding a variable Z that is a compact representation of \mathbf{X} and retains as much information as possible about Y . This variable Z is represented using the probabilities $p(z|\mathbf{x})$, $p(z)$ and $p(y|z)$ which are the outputs of the iterative IB algorithm. The next step is to parse the conditional probability distribution $p(z|\mathbf{x})$ in such a way as to infer a function $f(\mathbf{X})$ that explains the relationship between \mathbf{X} and Y in the best possible way.

In the ideal case, the conditional distribution $p(z|\mathbf{x})$ found by the IB algorithm represents exactly a function, meaning that $p(z|\mathbf{x})$ only assumes the values 0 and 1. In this case, $p(z|\mathbf{x})$ characterizes the function directly and no further work is necessary. Now, if we deviate slightly from this ideal scenario,

there is the case where $p(z|\mathbf{x})$ only assumes values that are either very close to one or very close to zero. In that case, a natural way to extract the functional relationship is simply to suppose that the function value $f(\mathbf{x})$ for the particular input configuration \mathbf{x} is given by

$$z^*(\mathbf{x}) = \max_{z \in \{z_1 \dots z_M\}} p(z|\mathbf{x}) \quad (14)$$

This leads to a lookup table representation of the function.

More generally, however, the conditional distribution $p(z|\mathbf{x})$ found by the IB algorithm has arbitrary values. Even in these cases, there may still exist a function $f(\mathbf{x})$ that reasonably and meaningfully matches the observed data. However, to extract this function from $p(z|\mathbf{x})$ will now require an additional effort. Typically, this will involve making some assumptions about the *structure* of the function. For the scope of the present paper, we restrict attention to a class of functions we will refer to as *remapped-linear functions*, which we discuss in detail in the following subsections.

4.2.1. The Case of Remapped-Linear Functions

In the present paper, we restrict attention to remapped-linear functions, by which we mean that the function $f(\mathbf{X})$ appearing in Equation (1) takes the form

$$\mathcal{F} \left(\sum_{i=1}^n \alpha_i \phi_i(X_i) \right) \quad (15)$$

where $\mathcal{F}(\cdot)$ is an unknown one-to-one function, and $\phi_i(\cdot)$ are arbitrary functions. The most challenging part of this formula is to determine the coefficients α_i . We note that once these coefficients are determined, it is a simple exercise to extract the mapping $\mathcal{F}(\cdot)$.

The problem of finding the function $f(\mathbf{X})$ now amounts to evaluating these coefficients $\alpha_i \in \mathbb{R}$, for $i \in [1 \dots n]$. We assume these coefficients to be real-valued. Following the discussion in Remark 2, since we proceed via information measures, the coefficients α_i cannot be uniquely determined and can only be estimated up to a scale factor γ . As a result, there will be ambiguity in the scale γ of these estimates. It should be noted that this is not a limitation caused by using the IB method, but is an inherent limitation of using only information measures for solving this problem. One can only expect to estimate the ratios $\frac{\alpha_1}{\alpha_k}, \frac{\alpha_2}{\alpha_k}, \dots, \frac{\alpha_n}{\alpha_k}$, for all $k \in \{1, \dots, n\}, \alpha_k \neq 0$.

We now propose the following three heuristic methods for estimating the coefficients α_i using $z^*(\mathbf{x})$ as identified according to Equation (14).

Method 1

This method does not make any assumptions on the support of Z for estimating the coefficients and uses only the *labels* $z^*(\mathbf{x})$ for which each input value \mathbf{x} has a mapping. Furthermore, using this method, we try to directly estimate the coefficients scaled with respect to one of the coefficients.

Consider a pair of inputs \mathbf{x} and \mathbf{x}' which map to the same z label, that is $z^*(\mathbf{x}) = z^*(\mathbf{x}')$. We can dispose of the z label by taking the difference of the two resulting equations as indicated below:

$$\sum_{i=1}^n \alpha_i \phi_i(x_i) = \sum_{i=1}^n \alpha_i \phi_i(x'_i) \tag{16}$$

$$\implies \sum_{i=1}^n \frac{\alpha_i}{\alpha_k} \phi_i(x_i) - \sum_{i=1}^n \frac{\alpha_i}{\alpha_k} \phi_i(x'_i) = 0 \tag{17}$$

We obtain a system of linear equations if we proceed in a similar way for all pairs of inputs which lead to the same mapping to the compressed variable. We can then solve for $\frac{\alpha_1}{\alpha_k}, \frac{\alpha_2}{\alpha_k}, \dots, \frac{\alpha_n}{\alpha_k}$ from the resulting system of equations given below:

$$\sum_{i=1}^n \frac{\alpha_i}{\alpha_k} \phi_i(x_i) - \sum_{i=1}^n \frac{\alpha_i}{\alpha_k} \phi_i(x'_i) = 0, \quad \forall \mathbf{x} \text{ and } \mathbf{x}' \text{ such that } z^*(\mathbf{x}) = z^*(\mathbf{x}') \tag{18}$$

Method 2

Method 1 described above could be computationally expensive even though solving a system of linear equations can be performed in polynomial time. This is because we look at all pairs of inputs which map to the same z . Furthermore, if there is only one input mapped to all of the compressed variable values, this method for identifying the coefficients cannot be applied.

An alternative way to estimate these α_i s could be to adopt some heuristics for assigning some values for z . In this method, we assign the support \mathcal{Z} of the variable Z using the support \mathcal{Y} of the output variable Y . In order to do so, we make use of the probability $p(y|z)$ which is also an output of the IB algorithm along with $p(z|\mathbf{x})$ and $p(z)$. We associate the value of the cluster centroid $\mathbb{E}[Y|z^*(\mathbf{x})]$ to each $z^*(\mathbf{x})$. Accordingly, we now solve the resulting overdetermined system of linear equations given below in a least squares sense for $(\alpha_1, \dots, \alpha_n)$.

$$\sum_{i=1}^n \alpha_i \phi_i(x_i) = \mathbb{E}[Y|z^*(\mathbf{x})], \quad \forall \mathbf{x} \tag{19}$$

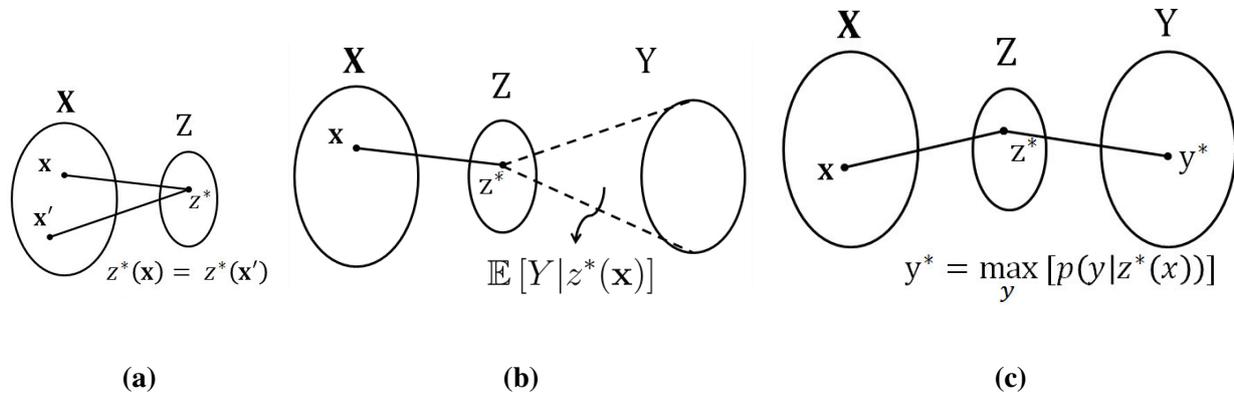
Method 3

Method 3 proceeds along similar lines to Method 2 by assigning values to z using the values of y . Instead of setting the expected value of y for each $z^*(\mathbf{x})$ like in Method 2, we now associate $z^*(\mathbf{x})$ with the value of Y which has the maximum value for $p(y|z^*(\mathbf{x}))$. Accordingly, we can solve the below set of linear equations in a least square sense for $(\alpha_1, \dots, \alpha_n)$.

$$\sum_{i=1}^n \alpha_i \phi_i(x_i) = \max_y \{p(y|z^*(\mathbf{x}))\}, \quad \forall \mathbf{x} \tag{20}$$

Although methods 2 and 3 solve explicitly for $(\alpha_1, \dots, \alpha_n)$, it is only the ratios $\left(\frac{\alpha_1}{\alpha_k}, \frac{\alpha_2}{\alpha_k}, \dots, \frac{\alpha_n}{\alpha_k}\right)$ which have to be considered as those are the best one could expect to be able to retrieve in this setup. These two methods only give an appropriate scaling for the possible values of Z . Figure 1 gives a pictorial representation explaining these three methods.

Figure 1. Methods 1, 2 and 3 for estimating the coefficients α_i in $Z = \sum_{i=1}^n \alpha_i \phi_i(X_i)$ from the output $p(z|\mathbf{x})$ of the IB algorithm. Method 1 looks at all $\{\mathbf{x}, \mathbf{x}'\}$ such that $z^*(\mathbf{x}) = z^*(\mathbf{x}')$, Method 2 sets $z^*(\mathbf{x}) = \mathbb{E}[Y|z^*(\mathbf{x})]$ and Method 3 sets $z^*(\mathbf{x}) = \max_y \{p(y|z^*(\mathbf{x}))\}$. (a) Method 1; (b) Method 2; (c) Method 3.



The above three methods are constructed in such a way that they always output *some* coefficients to explain Y as a function of the inputs, irrespective of whether a functional relationship really exists between the input and observed random variables. Therefore, an additional final check needs to be performed to ensure that the coefficients obtained from these three methods actually correspond to a compact function. The next section describes how this final test can be performed.

4.3. Sufficient Evidence

Given any joint distribution between the input and output random variables $p(\mathbf{x}, y)$, our algorithms will *always* output some collection of coefficients $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. But in some cases, these coefficients may be spurious. Therefore, we now develop a criterion to decide whether there is sufficient evidence that the claimed coefficient indeed represent true behavior. In keeping with the information measures used throughout the present paper, it is natural to introduce the new random variable

$$\tilde{Y} = \sum_{i=1}^n \alpha_i \phi_i(X_i) \tag{21}$$

This random variable represents the hypothesized function (without the remapping $\mathcal{F}(\cdot)$, which has no bearing on the involved information measures) and its joint distribution with Y can be easily computed according to

$$p(\tilde{y}, y) = \sum_{\mathbf{x}} p(\mathbf{x}, y, \tilde{y}) \quad \forall \tilde{y} : \tilde{y} = \alpha_1 \phi_1(x_1) + \dots + \alpha_n \phi_n(x_n) \tag{22}$$

We will say that there is sufficient evidence if the random variable \tilde{Y} captures a significant portion of the entire mutual information between \mathbf{X} and Y , that is, if the quantity

$$\frac{I(\tilde{Y}; Y)}{I(\mathbf{X}; Y)} \tag{23}$$

is large. Note that by the data processing inequality, this quantity cannot exceed one, and it is trivially non-negative.

Now, it is obvious that if the claimed function is very compact, then we cannot expect it to capture a significant portion of the entire mutual information between \mathbf{X} and Y , and thus, the quantity $I(\tilde{Y}; Y)/I(\mathbf{X}; Y)$ would be small. Hence, just considering $I(\tilde{Y}; Y)/I(\mathbf{X}; Y)$ by itself would not lead to a useful criterion. Rather, we need to compare this quantity to the compactness of the claimed function: If the function is very compact, then even if $I(\tilde{Y}; Y)/I(\mathbf{X}; Y)$ is small, there might be sufficient evidence. Thus, we want to also normalize by the compression-information term $I(\tilde{Y}; \mathbf{X})$. Specifically, we will consider the ratio $I(\tilde{Y}; \mathbf{X})/H(\mathbf{X})$ and introduce the following score function:

$$\Theta = \frac{I(\tilde{Y}; Y)/I(\mathbf{X}; Y)}{I(\tilde{Y}; \mathbf{X})/H(\mathbf{X})} \tag{24}$$

This score function has the desired behavior: if a function is very compact, $I(\tilde{Y}; Y)/I(\mathbf{X}; Y)$ might be small, but $I(\tilde{Y}; \mathbf{X})/H(\mathbf{X})$ would also be small, making the score Θ large. Thus, for the purported function $f(\cdot)$, the larger the score Θ , the more significant the evidence that the considered $p(\mathbf{x}, y)$ represents the claimed functional behavior. Therefore, we will accept the claimed coefficients whenever Θ is greater than some threshold θ (i.e., if $\Theta > \theta$). A typical value for this threshold could be 1, as this indicates that the coefficients represent a compact random variable which has more normalized information about Y than the normalized information about X .

4.4. Normal Variables and Linear Functions

To gain some insight into the information measures introduced so far (and hence, into the workings of the proposed algorithm), we consider a very simple special case in this section. Namely, we suppose that the joint distribution $p(\mathbf{x}, y)$ is a multivariate normal (Gaussian) distribution where all the input random variables X_i are independent of each other, with mean zero and variance P . Without loss of generality we can write

$$Y = \sum_{i=1}^n \alpha_i X_i + W \tag{25}$$

where W is also a normal random variable of mean zero and variance N .

Now, let us define the random variable V in the following form:

$$V = \sum_{i=1}^n \hat{\alpha}_i X_i \tag{26}$$

and point out that this random variable represents the desired function (in the general context of Section 4.3, we called this random variable \tilde{Y} , but to avoid confusion, we use a different symbol here).

To understand how the method discussed in this paper proceeds, the crucial quantity is the amount of information that the function captures, as discussed about in Equation (23). Therefore, we introduce the quantity $\rho(V)$ as follows:

$$\rho(V) = \frac{I(V; Y)}{I(\mathbf{X}; Y)} \tag{27}$$

Due to the data processing inequality, this quantity $\rho(V)$ is upper bounded by 1 which is attained when $V = \mathbf{X}$. If we denote the vectors $[\alpha_1 \dots \alpha_2]^T$ and $[\hat{\alpha}_1 \dots \hat{\alpha}_2]^T$ by $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\alpha}}$ respectively, then we have (a derivation is given in the Appendix):

$$I(\mathbf{X}; Y) = \frac{1}{2} \ln \left[1 + \|\boldsymbol{\alpha}\|^2 \frac{P}{N} \right] \tag{28}$$

$$I(V; Y) = \frac{1}{2} \ln \left[\frac{1 + \|\boldsymbol{\alpha}\|^2 \frac{P}{N}}{1 + \frac{\|\boldsymbol{\alpha}\|^2 \|\hat{\boldsymbol{\alpha}}\|^2 - \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle^2 P}{\|\hat{\boldsymbol{\alpha}}\|^2 N}} \right] \tag{29}$$

From the above two equations we get,

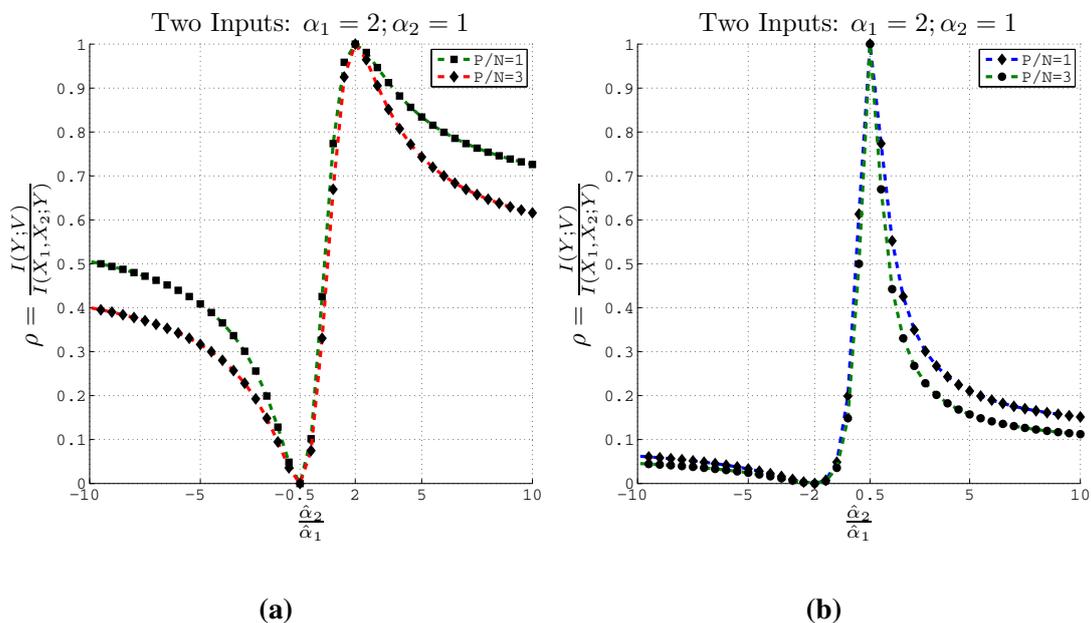
$$\rho(V) = 1 - \frac{\ln \left[1 + \frac{\|\boldsymbol{\alpha}\|^2 \|\hat{\boldsymbol{\alpha}}\|^2 - \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle^2 P}{\|\hat{\boldsymbol{\alpha}}\|^2 N} \right]}{\ln \left[1 + \|\boldsymbol{\alpha}\|^2 \frac{P}{N} \right]} \tag{30}$$

This expression attains its maximum value of 1, when

$$\begin{aligned} \|\boldsymbol{\alpha}\|^2 \|\hat{\boldsymbol{\alpha}}\|^2 - \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle^2 &= 0 \\ \text{i.e., } \hat{\boldsymbol{\alpha}} &= \gamma \boldsymbol{\alpha}, \text{ for some } \gamma \end{aligned} \tag{31}$$

From this equation we see that $I(V; Y)$ becomes equal to $I(\mathbf{X}; Y)$ for all estimates $\hat{\boldsymbol{\alpha}}$ that are multiples of the original coefficients $\boldsymbol{\alpha}$. This result is consistent with the discussion in Remark 2 where we argued that the coefficients cannot be uniquely determined using information measures.

Figure 2. $\rho\left(\frac{\hat{\alpha}_1}{\hat{\alpha}_2}\right)$ and $\rho\left(\frac{\hat{\alpha}_2}{\hat{\alpha}_1}\right)$ for 2 Gaussian inputs at different SNR levels. We see sharp peaks where the coefficients of V are equal to the actual coefficients of Y up to a scale factor. **(a)** $\rho\left(\frac{\hat{\alpha}_1}{\hat{\alpha}_2}\right)$; **(b)** $\rho\left(\frac{\hat{\alpha}_2}{\hat{\alpha}_1}\right)$.



For the two input Gaussian case ($n = 2$), Figure 2 depicts $\rho(V)$ computed according to Equation (30) as a function of the ratios of the estimated coefficients $\hat{\alpha}_1/\hat{\alpha}_2$ and $\hat{\alpha}_2/\hat{\alpha}_1$. From these plots we see that if the computed $\hat{\alpha}$ are such that, $I(V; Y)$ is close to $I(X_1, \dots, X_n; Y)$, then these estimated coefficients are also close to the original coefficients α up to a scale factor, due to the sharp peaks in the plots at these points.

Therefore, by using the information bottleneck if we are able to find a compact V such that $I(V; Y)$ is as close as possible to $I(\mathbf{X}; Y)$, then the computed coefficients from this V reflect the original functional relationship between \mathbf{X} and Y up to a scale factor.

5. Results on Artificial Data: Remapped-Linear Functions

In this section, we apply the proposed algorithms to artificial data. In order to set the stage for the application to experimental data, presented in Section 6 below, we consider an example where the data is integer-valued, and where we look for functions that are integer linear combinations. Specifically, we consider the following joint probability mass function $p(\mathbf{x}, y)$: We let the inputs X_i be independent of each other and uniformly distributed with support $\mathcal{X}_i = \{-M, \dots, 0, \dots, M\}$, where $M \in \mathbb{Z}$. Moreover, we let the output Y be

$$Y = \pi \left(\sum_{i=1}^n \alpha_i X_i + W \right) \quad (32)$$

where W is an additive independent noise following the same distribution as the X_i s. Here $\pi(\cdot)$ is an arbitrary permutation function on the support set (which we refer to as *alphabet* in this paper, matching the standard terminology in the information-theoretic literature). We chose this complex model to test our method because other methods such as linear regression will also be able to identify the coefficients when it is a simple linear model without any outer unknown function.

In this modified linear function setting of randomly generated data, we investigate whether the algorithm outlined in Section 4 can recover these coefficients α_i . While doing so, we set the cardinality $|\mathcal{Z}|$ of the compressed variable required for the iterative IB algorithm, to be much smaller than the true cardinality $|\mathcal{Y}|$ of Y so as to retrieve a compact function.

For example, consider the case when we have two inputs ($n = 2$) with $M = 5$ and the actual coefficients $\alpha_1 = 1$ and $\alpha_2 = 2$. Then the support of X_1 and X_2 becomes $\{-5, \dots, 0, \dots, 5\}$. In this scenario the true cardinality $|\mathcal{Y}| = 31$. We then run our algorithm for estimating the coefficients by setting $|\mathcal{Z}| = 5$. As Methods 2 and 3 are more computationally efficient than Method 1, we focus on these two methods in the rest of the report. Figure 3 plots the estimated normalized coefficients $\hat{\alpha}_1$ and $\hat{\alpha}_2$ for two inputs using both Method 2 and Method 3 at different values of the trade-off parameter β . Similar plots are also depicted in Figure 4 for three inputs with the same support and the actual coefficients set as $\alpha_1 = 1$, $\alpha_2 = 5$ and $\alpha_3 = -2$. In this case, the true cardinality $|\mathcal{Y}| = 81$ and the cardinality set in the IB algorithm $|\mathcal{Z}| = 10$.

Figure 3. Estimating coefficients $\hat{\alpha}_1$ and $\hat{\alpha}_2$ using Methods 2 and 3 on artificial data with 2 inputs of support $\{-5, \dots, 5\}$ at different values of the β parameter used in the IB algorithm. Here $[\alpha_1 = 1; \alpha_2 = 2]$, $|\mathcal{Y}| = 31$ and $|\mathcal{Z}| = 5$. (a) Using Method 2; (b) Using Method 3.

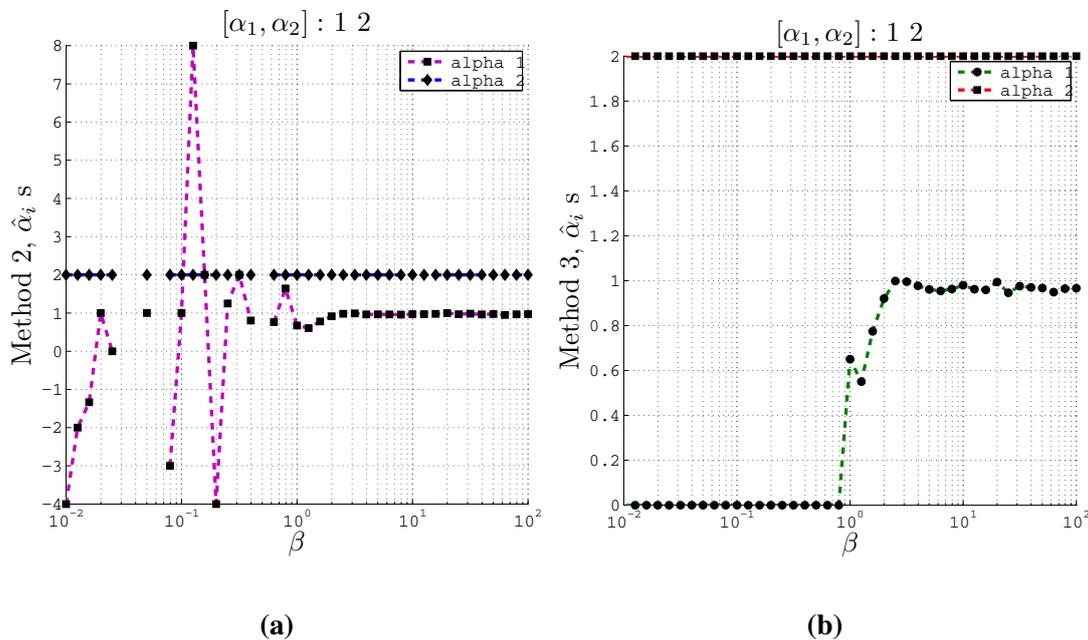
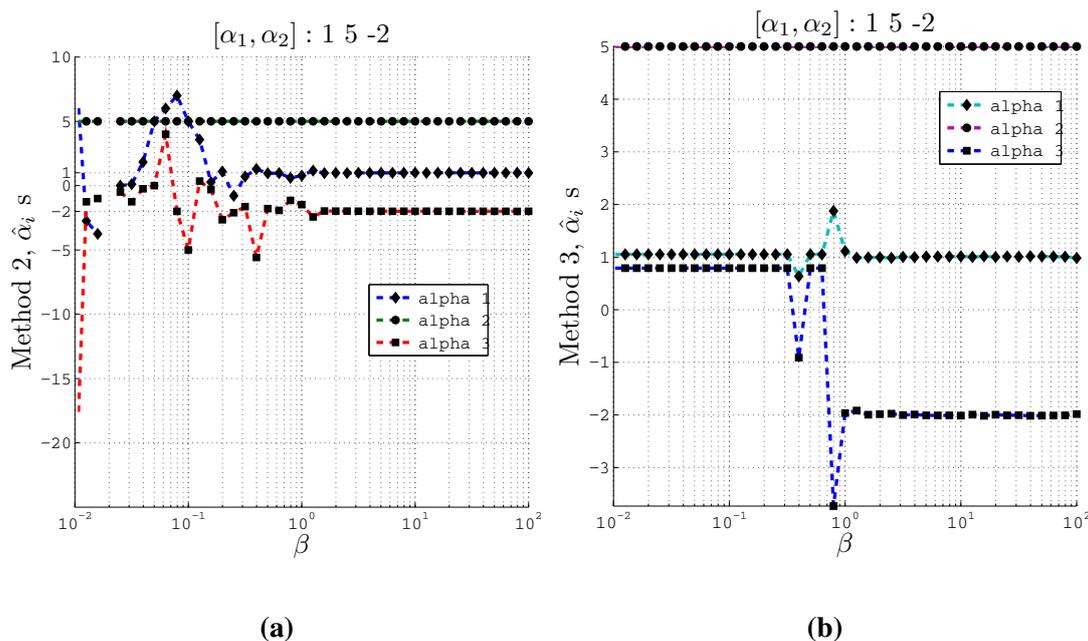


Figure 4. Estimating coefficients $\hat{\alpha}_1, \hat{\alpha}_2$ and $\hat{\alpha}_3$ using Methods 2 and 3 on artificial data with 3 inputs of support $\{-5, \dots, 5\}$ at different values of the β parameter used in the IB algorithm. Here $[\alpha_1 = 1; \alpha_2 = 5; \alpha_3 = -2]$, $|\mathcal{Y}| = 81$ and $|\mathcal{Z}| = 10$. (a) Using Method 2; (b) Using Method 3.



From these plots (Figures 3 and 4) we observe that at relatively small β values, the estimated $\hat{\alpha}_i$ s converge to the actual coefficients α_i s even when the cardinality is set such that $|\mathcal{Z}| \ll |\mathcal{Y}|$. Moreover,

Method 2 and Method 3 converge to the actual coefficients in different ways. Method 2 fluctuates greatly at very small β values before it converges to the actual coefficient values. On the other hand, Method 3 is stable at small β values and at a particular β value, it converges to the actual coefficients.

Furthermore, the Θ values computed at different values of β correspond to the recovery of the original coefficients. At the values of β where the estimated coefficients become close to the original coefficients, we obtain high values for Θ . We observe a gradual increase of Θ value for increasing values of β . For example, in the 2 input case, using method 2, at $\beta = 10$, the computed Θ value is 1.347 which is high enough to accept the estimated coefficients from our algorithm. Similar high values for Θ are obtained for the 3 input case as well.

5.1. Comparison with Linear Regression and Related Model-Based Approaches

When parsing the compact representation Z to an actual function exhibiting simple mathematical structure, we restricted attending to linear functions in the present paper. It is therefore tempting to compare to methods that start out with a linear model from scratch (rather than bringing it in at the end, as we are doing in the proposed method). For example, consider linear regression: Given data from the considered three neurons, x_1, x_2, y , we could simply run linear regression for y based on x_1 and x_2 , and this would provide us with the regression coefficients. This appears to be a much more direct and simpler approach to the function identification problem.

However, there is a significant downside to this approach: it requires one to separately impose how information is represented in absolute terms, with respect to the real numbers and mean-squared error. This information is crucially exploited by linear regression, though such a feature appears to violate the spirit of the function identification problem: functional behavior should be *relative*, not connected to absolute representations.

To illustrate this issue more concretely, the remapped-linear case is instructive. That is, suppose that the underlying ground truth is given by

$$Y = 2^{X_1+2X_2+W} \quad (33)$$

It should be immediately obvious that linear regression will fail when applied directly to this data. Indeed, using the probability distribution leading to Figure 3 and generating data from that distribution, linear regression returns the coefficients 912 and 1,045, which are very far from the true coefficients, 1 and 2 (not surprisingly).

6. Results on Experimental Data: Linear Function

6.1. Data Description

In this experiment, an adult male rhesus monkey (*Macaca mulatta*) performs a behavioral task for a duration of about 15 minutes (1,080,353 milliseconds, to be precise), while the resulting voltage traces are simultaneously measured in the primary motor cortex (M1) region of the brain using 64 Teflon-coated tungsten multielectrodes (35 μm diameter, 500 μm electrode spacing, in 8 by 8 configuration: CD Neural Engineering, Durham, NC, USA). The arrays were implanted bilaterally in the hand/arm

area of M1, positioned at a depth of 3 mm targeting five pyramidal neurons. Localization of target areas was performed using stereotactic coordinates of the rhesus brain. We then use a low-pass filtered version of these voltage traces to obtain the neural responses from 184 neurons.

The monkey was trained to perform a delayed center-out reaching task using his right arm. The task involved cursor movements from the center toward one of eight targets distributed evenly on a 14-cm diameter circle. Target radius was set at 0.75 cm. Each trial began with a brief hold period at the center target, followed by a GO cue (center changed color) to signal the reach toward the target. The monkey was then required to reach and hold briefly (0.2–0.5 s) at the target in order to receive a liquid reward. Reaching was performed using a Kinarm (BKIN Technologies, Kingston, ON, Canada) exoskeleton where the monkey's shoulder and elbow were constrained to move the device on a 2D plane. Over the course of the entire experiment, the reaching task is performed in different directions: 0° , 45° , 90° , 180° , etc. Moreover, the reaching task in a particular direction is repeated several times; for example, the 180° reaching task is repeated 36 times at different starting points in the entire duration of 15 minutes.

All procedures were conducted in compliance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and were approved by the University of California at Berkeley Institutional Animal Care and Use Committee.

6.2. Applying the Proposed Algorithm on Data

The functional identification algorithm outlined in Section 4 is applied on this dataset to infer some structure present in the data. Before doing that, we first need to decide how to estimate the required probability distributions $p(\mathbf{x}, y)$ from the data. Additionally, we also need to decide a way to deal with the temporal aspect of the neural spike trains from different neurons.

Let $S_i^t(\Delta)$ denote the spike train of neuron i starting from time t and lasting for Δ milliseconds, *i.e.*, we are looking at the neural response of neuron with id i from time t ms to $(t + \Delta)$ ms with a millisecond precision. $S_i^t(\Delta)$ can be seen as a vector of length Δ comprising of 0s and 1s where 0 represents no spike and 1 represents a spike. The number of spikes we have in this time window is denoted by $|S_i^t(\Delta)|$.

Then a random variable, denoted by $R_i^t(\Delta, b)$, is estimated from $S_i^t(\Delta)$ in the following way (b here, is a binning parameter): c the histogram of the realizations $r_i^t(\Delta, b)$ given by:

$$r_i^t(\Delta, b) = |S_i^{t'}(b)|, \forall t' \in \{t, \dots, \Delta - b\} \quad (34)$$

and normalize this histogram to get the probability distribution $p(R_i^t(\Delta, b) = r_i^t(\Delta, b))$. In other words, this procedure maintains a sliding window of length b ms starting from the beginning of the spike train $S_i^t(\Delta)$, counts the number of spikes in this window while stepping this window to the right until we reach the end of the spike train $S_i^t(\Delta)$ and normalizes this binned histogram to obtain the probability distribution of the random variable $R_i^t(\Delta, b)$ (Figure 5). Accordingly, the support of this random variable is the number of spikes observed in any contiguous segment of length b ms of the spike train $S_i^t(\Delta)$.

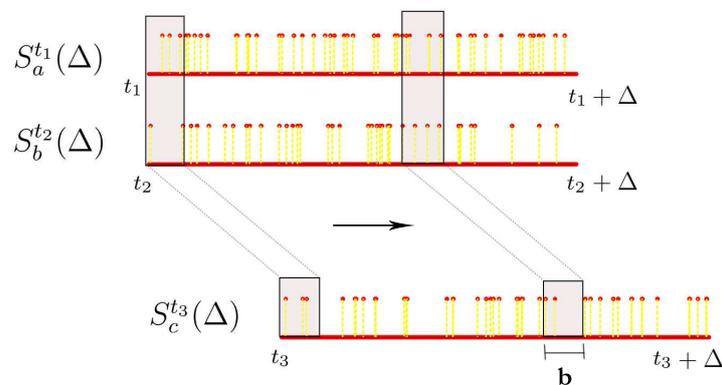
The above procedure can be extended for estimating the joint probability distribution from multiple spike trains. In this project, we restrict ourselves to the case where given two spike train segments $S_a^{t_1}(\Delta)$ and $S_b^{t_2}(\Delta)$, we want to know if there exists a linear functional relationship between these two spike train segments in order to explain a third spike train segment $S_c^{t_3}(\Delta)$. Therefore, we need to estimate the joint probability distribution of the three random variables $R_a^{t_1}(\Delta, b)$, $R_b^{t_2}(\Delta, b)$ and $R_c^{t_3}(\Delta, b)$ associated with

these three spike train segments. To do this, we ensure that the sliding window is appropriately aligned across all these three spike trains while obtaining the joint histogram. This procedure is illustrated in Figure 5. Once we have this joint histogram we can use the procedure outlined earlier in this chapter to estimate α_1 and α_2 such that the below functional relationship holds:

$$\alpha_1 R_a^{t_1}(\Delta, b) + \alpha_2 R_b^{t_2}(\Delta, b) = R_c^{t_3}(\Delta, b) \tag{35}$$

$R_a^{t_1}(\Delta, b)$ and $R_b^{t_2}(\Delta, b)$ are the input random variables (X_1, X_2 , as in the notation used in Section 3) and $R_c^{t_3}(\Delta, b)$ is the output random variable (Y). It should be noted that we should expect to be able to identify such a relationship only occasionally from the data, as neurons generally do not behave in a predictable and deterministic way. We need to perform an exhaustive search to find the *right* neurons (a, b, c), the time frames (t_1, t_2, t_3) when these neurons have interesting behaviors and also the suitable parameters Δ and b for which such relationships exist and can be identified by our method. Accordingly, in order to reduce the search space, we assume that the two inputs neuron spike trains are aligned and start at the same time, *i.e.*, $t_1 = t_2$. We then try different delays δ in the output neuron spike train, *i.e.*, $t_3 > t_1 = t_2$ and $\delta = t_3 - t_1$.

Figure 5. Estimating joint histograms from spike trains, where we consider overlapping bins using a sliding window.



6.3. Functions Identified in Data

Consider a particular trial of the 180° experiment which lasts from time 40,718 ms to time = 43,457 ms. The two input neuron spike trains are set to start at the advent of different events (like: center appears, hand enters center, go cue, hand enters target, *etc.*) and the output neuron spike train is set to start at different delay values with respect to the input neurons with a maximum delay of 800 ms. We set the parameters for obtaining the joint histograms as $\Delta = 200$ ms, $b = 10$ ms and the threshold θ is set to 1.25. Below are a few functions obtained at the event when *hand enters center* at time $t = 41,080$ of this experimental trial. For the sake of simplicity, we drop the superscript for the random variables corresponding to the input neurons with the understanding that both the input neuron spike trains start at $t = 41,080$. Also, in the superscript of the output neuron’s random variable, we indicate only the delay δ with respect to the input neurons. As it turned out, *all* the functions we

found were direct, unweighted sums. The functions are sorted in the descending order of the confidence parameter Θ .

1. $R_{139} + R_{114} = R_{98}^{250}$ with $\Theta = 1.456$.
2. $R_{28} + R_{114} = R_{98}^{370}$ with $\Theta = 1.321$.
3. $R_{139} + R_{98} = R_{28}^{310}$ with $\Theta = 1.278$.
4. $R_{139} + R_{28} = R_{114}^{750}$ with $\Theta = 1.273$.
5. $R_{114} + R_{28} = R_{63}^{450}$ with $\Theta = 1.267$.

We observe that most of the normalized linear coefficients estimated by our method on this dataset are equal to 1, with rare occurrences of 2, and no values greater than 2. An example of such a weighted function was found between neurons 63, 114, and 28 in the case where we set the input spike trains to start at the event when the *hand enters the target* (at time 42,705 ms) as given below:

$$R_{63}^{42705} + 2R_{114}^{42705} = R_{28}^{42705+570} \tag{36}$$

However, such weighted functions were not frequently identified. This can be explained by looking at the support of the different random variables estimated from the spike trains. These supports are compact and concentrated in a particular range for most of the spike trains. Therefore, we do not observe higher normalized coefficient estimates such as 3 or 4. Functions at different events for different experimental trials can be identified in a similar way.

In order to better validate the results obtained using our proposed algorithm, we need to verify whether the functional relationships listed above are replicated in different trials of the same experiment. The next paragraph goes through a particular case study where the functional relationships between a particular triplet of neurons are analyzed over different trails of the same experiment.

Consider the following three neurons: 28, 114 and 98, with neurons 28 and 114 as the input neurons and neuron 98 as the output neuron. We want to observe the behavior of these three neurons across all 36 trials of the 180° reaching tasks. Out of these 36 trials, only 26 are as successful and the rest are considered unsuccessful as the monkey’s hand leaves the center before the go cue is given. The functional relationships obtained from one of the 26 successful trials that lasts from $t = 40,718$ ms to $t = 43,457$ ms were listed above. Let us look at one particular function, namely, the second item on the previous list:

$$R_{28}^{41080} + R_{114}^{41080} = R_{98}^{41080+370} \tag{37}$$

The functional relationship in this equation implies that the *sum* of neuron 28 and neuron 114 at time 41,080 (which corresponds to the action of the hand entering center) is *equal* to neuron 98 after a delay of 370 ms. If we consider this as a *reference* trial, we are interested in knowing if a similar function exists between these neurons during other trials at this delay of 370 ms (which corresponds to when the hand enters center).

Accordingly, we apply this above function obtained between neurons (28,114,98) during this reference trial, at identical stages of the different trials of the 180° reaching task. There are 36 such trials in the same direction. We then plot $I(\tilde{Y}^k(370); Y^k(370))/I(X_1^k, X_2^k; Y^k(370))$ versus $(I(\tilde{Y}^k(370); X_1^k, X_2^k)/H(X_1^k, X_2^k))$, for all these scenarios (Figure 6). Here $X_1^k \equiv R_{28}^t(200, 10)$,

$X_2^k \equiv R_{114}^t(200, 10)$ are the 2 input neurons (28 and 114) starting at time t where the hand enters the center for trial k and $Y^k(370) \equiv R_{98}^{t+370}(200, 10)$ is the variable corresponding to the output neuron (98) in trial k after a delay of 370 ms w.r.t the input neurons. Here, $\tilde{Y}^k(370)$ is computed as described in Section 4.3 from the estimated coefficients in order to check for sufficient evidence for the function found between X_1^k, X_2^k and $Y^k(370)$.

Figure 6. We check if the function (Equation (37)) obtained between neurons 28, 114 and 98 in the reference trial ($t = 40,718$) is valid across all 36 trials of the 180° reaching task with a threshold $\theta = 1$.

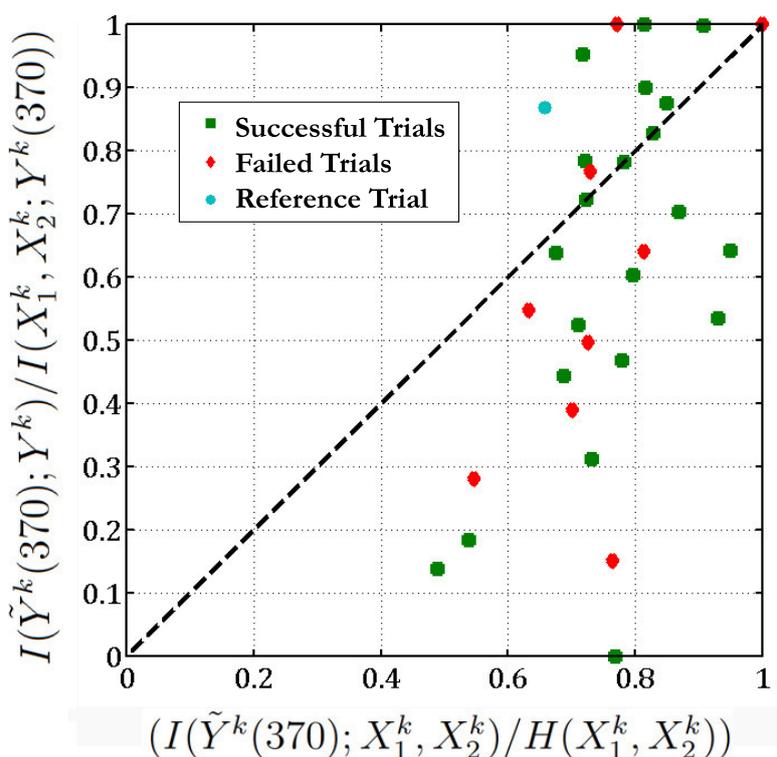


Figure 6 implies that in 13 out of the 36 different trials of the 180° reaching task, the *sum* of neurons 28 and 114 is equal to the response of neuron 98 as the points corresponding to these trials lie above the 45° line (here we set the threshold $\theta = 1$). In one-third of the experimental trials, the functional relationship given in Equation (37) holds. Moreover, if we exclude the unsuccessful trials, then 10 out of 26 of the successful trials follow the above functional relationship between neurons 28, 114 and 98. Most of the unsuccessful trials lie below the 45° line ($\Theta < 1$) which indicates that these neurons behave in a different manner during an unsuccessful trial. The below table lists the Θ values corresponding to all the trials (we exclude 5 trials which give Θ of the form 0/0 and 1/0):

Successful Trials		Unsuccessful Trials	
$\Theta \geq 1$	$\Theta < 1$	$\Theta \geq 1$	$\Theta < 1$
1.331	0.947	1.000	0.197
1.321	0.811	1.052	0.513
1.229	0.758	1.293	0.555
1.114	0.739		0.685
1.109	0.676		0.787
1.102	0.647		0.865
1.091	0.601		
1.000	0.574		
1.000	0.427		
1.000	0.341		
	0.283		
	0.000		

For successful trials, we can consider the cases when $\Theta \geq \theta$ as *true positives* (TP) and the cases when $\Theta < \theta$ as *false negatives* (FN). Similarly for the unsuccessful trials, we can consider the cases when $\Theta \geq \theta$ as *false positives* (FP) and the case when $\Theta < \theta$ as *true negatives* (TN). Then, at this value of the threshold $\theta = 1$ we can compute the following quantities:

- True Positive Rate (TPR) = $TP/(TP+FP) = 76.92\%$
- True Negative Rate (TNR) = $TN/((TN+FN) = 33.33\%$
- Sensitivity = $TP/(TP+FN) = 45.45\%$
- Specificity = $TN/(TN+FP) = 66.67\%$

We achieve high TPR but not such a high TNR as there are many false negatives. It should be noted that these values depend on the value of the threshold θ . These numbers indicate that the functions identified by our algorithm in a particular trial are somewhat consistent across different trials of the reaching task of the same type.

7. Conclusions

This paper explores a novel application of the Information Bottleneck (IB) method in the context of neuroscience. While most direct practical applications of the IB method are in the domain of supervised and unsupervised clustering, we use the IB method in an entirely different way for identifying compact linear functional relationships between different random variables. In this paper, we attempted to answer the following questions: When can we say that a functional relationship exists between random variables? How can we estimate these coefficients that explain linear dependencies between random variables? How reliable are these estimates? This approach is then tested on artificial data to investigate the performance of the proposed algorithm. We then applied our proposed algorithm on experimental data characterizing the neural activity of a large population of neurons recorded during a

macaque experiment involving behavioral tasks. Sifting through the data and considering a large number of neuron triples, we were able to identify several occurrences of neurons that appear to exhibit linear relationships towards other neurons. We selected one neuron triple and followed it through all 36 runs of that particular experiment, finding a certain degree of consistency of the functional relationship that was identified.

Appendix

Here, we provide a short derivation of Equations (28) and (29). For both, the main argument is that the entropy of the n -dimensional multivariate Normal distribution with covariance matrix Σ is given by $\frac{1}{2} \log_2(2\pi e)^n \det(\Sigma)$, see e.g., ([16][p.254]). Specifically, for Equation (28), we can derive

$$\begin{aligned}
 I(\mathbf{X}; Y) &= H(\mathbf{X}) + H(Y) - H(\mathbf{X}, Y) = \frac{1}{2} \ln \left[\frac{(2\pi e)^n \begin{vmatrix} P & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & P \end{vmatrix} (2\pi e)(\|\boldsymbol{\alpha}\|^2 P + N)}{(2\pi e)^{n+1} \begin{vmatrix} P & \dots & 0 & \alpha_1 P \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & P & \alpha_n P \\ \alpha_1 P & \dots & \alpha_n P & \|\boldsymbol{\alpha}\|^2 P + N \end{vmatrix}} \right] \\
 &= \frac{1}{2} \ln \left[\frac{P^n (\|\boldsymbol{\alpha}\|^2 P + N)}{P^n N} \right] = \frac{1}{2} \ln \left[1 + \|\boldsymbol{\alpha}\|^2 \frac{P}{N} \right] \tag{38}
 \end{aligned}$$

For Equation (29), by the same argument, we find

$$\begin{aligned}
 I(V; Y) &= H(V) + H(Y) - H(V, Y) = \frac{1}{2} \ln \left[\frac{(2\pi e)(\|\hat{\boldsymbol{\alpha}}\|^2 P)(2\pi e)(\|\boldsymbol{\alpha}\|^2 P + N)}{(2\pi e)^2 \begin{vmatrix} \|\hat{\boldsymbol{\alpha}}\|^2 P & \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle P \\ \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle P & \|\boldsymbol{\alpha}\|^2 P + N \end{vmatrix}} \right] \\
 &= \frac{1}{2} \ln \left[\frac{\|\hat{\boldsymbol{\alpha}}\|^2 (\|\boldsymbol{\alpha}\|^2 P + N)}{P(\|\boldsymbol{\alpha}\|^2 \|\hat{\boldsymbol{\alpha}}\|^2 - \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle^2) + N \|\hat{\boldsymbol{\alpha}}\|^2} \right] = \frac{1}{2} \ln \left[\frac{1 + \|\boldsymbol{\alpha}\|^2 \frac{P}{N}}{1 + \frac{\|\boldsymbol{\alpha}\|^2 \|\hat{\boldsymbol{\alpha}}\|^2 - \langle \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \rangle^2 P}{\|\hat{\boldsymbol{\alpha}}\|^2 N}} \right] \tag{39}
 \end{aligned}$$

Acknowledgements

The authors thank the anonymous reviewers whose comments have helped to significantly improve this manuscript. The research reported here was supported in part by the U.S. National Science Foundation under CDI Type-I Grant 0941343.

References

1. Rieke, F.; Warland, D.; Rob.; Bialek, W. *Spikes: Exploring the Neural Code*, 1st ed.; MIT Press: Cambridge, MA, USA, 1997.
2. Borst, A.; Theunissen, F.E. Information theory and neural coding. *Nat. Neurosci.* **1999**, *2*, 947–957.

3. Dayan, P.; Abbott, L.F. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, 1st ed.; The MIT Press: Cambridge, MA, USA, 2001.
4. Abbott, L.F.; Dayan, P. The effect of correlated variability on the accuracy of a population code. *Neural Comput.* **1999**, *11*, 91–101.
5. Schneidman, E.; Bialek, W.; Berry, M.J. Synergy, redundancy, and independence in population codes. *J. Neurosci.* **2003**, *23*, 11539–11553.
6. Narayanan, N.S.; Kimchi, E.Y.; Laubach, M. Redundancy and synergy of neuronal ensembles in motor cortex. *J. Neurosci.* **2005**, *25*, 4207–4216.
7. Latham, P.; Nirenberg, S. Synergy, redundancy, and independence in population codes, revisited. *J. Neurosci.* **2005**, *25*, 5195–5206.
8. Averbeck, B.B.; Lee, D. Effects of noise correlations on information encoding and decoding. *J. Neurophysiol.* **2006**, *95*, 3633–3644.
9. So, K.; Ganguly, K.; Jimenez, J.; Gastpar, M.C.; Carmena, J.M. Redundant information encoding in primary motor cortex during natural and prosthetic motor control. *J. Comput. Neurosci.* **2011**, *32*, 555–561.
10. Quinn, C.; Coleman, T.; Kiyavash, N.; Hatsopoulos, N. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *J. Comput. Neurosci.* **2010**, *30*, 17–44.
11. So, K.; Koralek, A.C.; Ganguly, K.; Gastpar, M.C.; Carmena, J.M. Assessing functional connectivity of neural ensembles using directed information. *J. Neural Eng.* **2012**, *9*, doi:10.1088/1741-2560/9/2/026004.
12. Schmidt, M.; Lipson, H. Distilling free-form natural laws from experimental data. *Science* **2009**, *324*, 81–85.
13. Tishby, N.; Pereira, F.C.; Bialek, W. The Information Bottleneck Method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, September 1999; IEEE Press: Piscataway, NJ, USA, 1999; pp. 368–377.
14. Klampfl, S.; Legenstein, R.; Maass, W. Spiking neurons can learn to solve information bottleneck problems and extract independent components. *Neural Comput.* **2009**, *21*, 911–959.
15. Buesing, L.; Maass, W. A spiking neuron as information bottleneck. *Neural Comput.* **2010**, *22*, 1961–1992.
16. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley-Interscience: Hoboken, NJ, USA, 2006.
17. Slonim, N. The Information Bottleneck: Theory and Applications. PhD thesis, The Hebrew University, Jerusalem, Israel, 2003.