Published in final edited form as:

J Neural Eng. 2012 October ; 9(5): 054001. doi:10.1088/1741-2560/9/5/054001.

Sparse decoding of multiple spike trains for brain-machine interfaces

Ariel Tankus^{1,2}, Itzhak Fried^{1,3,4}, and Shy Shoham²

Shy Shoham: sshoham@bm.technion.ac.il

¹Department of Neurosurgery, University of California, Los Angeles, CA 90095, USA

²Faculty of Biomedical Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel

³Functional Neurosurgery Unit, Tel-Aviv Medical Center, Tel-Aviv 64239, Israel

⁴Sackler School of Medicine, Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

Brain-machine interfaces (BMIs) rely on decoding neuronal activity from a large number of electrodes. The implantation procedures, however, do not guarantee that all recorded units encode task-relevant information: selection of task-relevant neurons is critical to performance but is typically performed heuristically. Here, we describe an algorithm for decoding/classification of volitional actions from multiple spike trains, which automatically selects the relevant neurons. The method is based on sparse decomposition of the high-dimensional neuronal feature space, projecting it onto a low-dimensional space of codes serving as unique class labels. The new method is tested against a range of existing methods using simulations and recordings of the activity of 1592 neurons in 23 neurosurgical patients who performed motor or speech tasks. The parameter estimation algorithm is orders of magnitude faster than existing methods, and achieves significantly higher accuracies for both simulations and human data, rendering sparse decomposition highly attractive for BMIs.

1. Introduction

The faithful decoding of neuronal activity is widely applied in neural coding research and is also a fundamental stage for brain-machine interfacing. Brain-machine interface (BMI) research has employed a wide variety of decoding methodologies (reviewed in (L. Paninski et al. 2007; B. M. Yu et al. 2008; Zacksenhouse & Nemets 2008; Quiñero Quiroga & Panzeri 2009; Koyama et al. 2010)), including finite impulse response linear decoders (Humphrey et al. 1970; Georgopoulos et al. 1986; Wessberg et al. 2000; Carmena et al. 2003; Hochberg et al. 2006; Velliste et al. 2008; Herzfeld & Beardsley 2010; O'Keefe et al. 2007), recursive (infinite impulse response) linear decoders (Wu et al. 2004; S.-P. Kim et al. 2008; Z. Li et al. 2009; Gupta & Ashe 2009; Wu et al. 2009), and non-linear methods including Bayesian decoders (Sanger 1996; Gerwinn et al. 2009; Shin et al. 2010), sequential Monte Carlo filters (Shoham 2001; Shoham et al. 2005; Brockwell et al. 2007; Wang et al. 2009a;

Wagenaar et al. 2011), and artificial neural networks (Carmena et al. 2003; O'Keefe et al. 2007).

When multiple spike trains are to be used for decoding, it is important to decide which of the recorded channels encode information relevant to the task at hand. Input from task-unrelated channels should be discarded or strongly down-weighted due to the noise it inserts into the decoding process (Fraser et al. 2009; Vargas-Irwin et al. 2010). The common heuristics for channel selection is neuron dropping (Carmena et al. 2003; Li et al. 2009b; Ganguly & Carmena 2009; Hélot et al. 2010). This procedure selects the task-relevant neurons based on a training set. It begins with computing the cross-validated decoding accuracy based on the group of all n neurons. At the first stage, each neuron is dropped, and accuracy is re-computed for the remaining $n-1$ neurons, yielding n values (one for each neuron dropping). The neuron whose dropping yielded the highest of the n accuracies is then dropped. The second stage now begins with the remaining $n-1$ neurons, dropping one neuron at a time, and so forth; the process is repeated n times. The stage leading to the highest accuracy of all stages determines the task-relevant neurons to participate in decoding of the test set. Variants of this method, such as adding neurons by the correlation of their activity to behavior (Vargas-Irwin et al. 2010), or stopping criteria are common in the literature (Sanchez et al. 2004). Other approaches include methods based on information theory (Wang et al. 2009b), sensitivity analysis (Singhal et al. 2010) or variational Bayesian least squares (Ting et al. 2009). We note that although neuron selection is usually a first step in decoding neural data (figure 1), some studies do not describe the method used. For example, (Hochberg et al. 2006) provide only a qualitative description of comparing the neural activity during “rest” periods to active performance of a motor task, whereas (Carmena et al. 2003) do not specify how they selected the subset of isolated units used for control, even though they sometimes dropped over half the neurons (monkey #1: 10-30 units out of 20-50 isolated units; Monkey #2: 60-120 out of 150-180 isolated units).

The implicit assumption of neuron selection methods is that usually only a small subset of the recorded neurons are task-related. The level of sparseness is expected to be especially high when electrode placement is based only on clinical criteria or when widely distributed arrays are used, suggesting the use of methods based on sparseness priors. Sparse representation is an emerging field of signal processing (Candes & Wakin 2008; Elad 2010; Eldar & Kutyniok 2012), and has already been employed to analyze neural data in the context of multichannel electroencephalography (EEG) (Wu et al. 2011), perceptual decoding from simulated visual system neurons (Shi et al. 2009) and decoding of high dimensional functional neuroimaging (Yamashita et al. 2008; Li et al. 2009a; Li et al. 2011; de Brecht & Yamagishi 2012). Here, we apply and test sparse decomposition as an alternative feature selection strategy when decoding neuronal activity in a BMI: the suggested sparse decoding algorithm jointly estimates a linear decoder with automatic selection of parsimonious features (section 2). In section 3, we compare the performance of sparse decoding to other BMI feature selection methods, using both simulated responses and real neuronal activity recorded from the human brain during hand movements and speech. The results demonstrate the superior performance of the new procedure in terms of both accuracy and runtime. Concluding remarks appear in Section 4.

2. Materials and Methods

2.1. Patients and Electrophysiology

Twenty three patients with pharmacologically resistant epilepsy undergoing invasive monitoring with intracranial depth electrodes to identify the seizure focus for potential surgical treatment (19-53 years old (mean: 34.4 years; SD=11.0), 10 right-handed females, 2 left-handed females, 8 right-handed males, 3 left-handed males) participated in a total of 41 recording sessions, each on a different day. Based exclusively on clinical criteria, each patient had 7-13 electrodes, each of which terminated with a set of nine 40- μ m platinum–iridium microwires. Their locations were verified by MRI or by computer tomography coregistered to preoperative MRI. Bandpass filtered signals (0.3–3kHz) from these microwires and the sound track were synchronously recorded at 30kHz using a 128-channel acquisition system (Neuroport, Blackrock, Salt Lake City, UT). Sorted units (WaveClus (Quian-Quiroga et al. 2004), SUMU (Tankus et al. 2009)) recorded in different sessions are treated as different in this study. All studies conformed to the guidelines of the Medical Institutional Review Board at the University of California Los Angeles.

2.2. Experimental Paradigms

Aiming for BMI systems that will reconstruct either hand movement direction or speech, we performed offline tasks that allow decoding these types of information.

2.2.1. Simulations—Neuronal activity was modeled as an inhomogeneous Poisson point process with stochastic intensity λ ($\lambda = 3$ spikes/s during the baseline period; SNR = $\lambda_{\text{response}} / \lambda_{\text{baseline}} = 2$ during response periods). The mean firing rate was computed at 100ms time bins, 10 of which preceded (“baseline”) and 10 followed (“response”) trial onset. Each simulated run simulated 100 neurons at 5 different task conditions with 20 trials per condition. A varying percentage of these neurons responded to the different task conditions (“responsive neurons”).

To simulate a more realistic case where a neuron may respond to more than one condition, we introduced overlaps between the groups of neurons responsive to one condition and the group of neurons responsive to the next condition. Initially, each task condition had 5 responsive neurons, which did not respond to any other condition. We then increased the level of overlap: the first and second task conditions had 1 neuron responsive to both; the second and third conditions had 1 neuron responsive to both, etc. (overlap level: 1). For overlap level of 2, the first and second task conditions had 2 neurons responsive to both, and so on.

2.2.2. The directed movement task—Fourteen patients performed the motor task in 27 sessions. They performed hand movements to control a computer cursor using a joystick (Attack3, Logitech, Newark, CA). They first had to position the cursor inside a ring-shaped target at the center of the screen. Following a short delay period (random between 2-3 seconds), the target would “jump” to one of four randomly-lit targets, equi-spaced on a virtual circle around the center. The patients would then move the cursor to reach the target (“Center-Out”) (Moran & Schwartz 1999). The target would return to the center, with the

patients following it, and the task will repeat. Patients performed between 32 and 102 trials per session (mean=79.5 trials; SD=18.0).

2.2.3. The speech task—Eleven patients performed the speech task in 14 sessions conducted at the patient's quiet bed-side. They first listened to isolated auditory cues (beeps) and to another individual uttering the vowel sounds ("auditory controls"). Then, following an oral instruction, patients uttered the instructed monophthongal vowels (a /a/, e /e/, i /i/, o /o/, u /u/; between slashes is the phonemic transcription) multiple times, each following a randomly-spaced (2-3sec.) beep. For simplicity, this paper employs the English rather than phonemic transcription as described above.

2.3. Data Analysis

2.3.1. Features

Simulation: For each simulated unit, 4 features were extracted: the average firing rate in two 500ms bins before and two 500ms bins after trial onset.

Directed movement: For each unit, the selected features consisted of the average firing rate in ten 100ms bins between -400ms and 600ms after movement onset.

Speech: For each unit, the selected features consisted of the average firing rate in 2 bins: baseline between -1000ms and cue onset time (0) and in the 200ms time window starting at speech onset.

Features with no variance on the training set were omitted.

2.3.2. Algorithms—All decoding results were 6-fold cross-validated using trials not used for training. The data was partitioned into 6 groups out of which 5 were used for training and the 6th, for testing (repeated 6 times). All algorithms were implemented in MATLAB (Mathworks, Natick, MA). All runtimes were measured on the same computer (Dell Latitude E6500, Dell, Round Rock, TX) and represent CPU time.

Sparse Decomposition

The suggested decoding strategy incorporates the stage of selecting relevant channels into the decoder itself. Instead of selecting relevant channels explicitly and projecting their spike trains onto a low dimensional space for classification (figure 1 (a)), the new methodology suggests to project *all* spike trains directly onto the low dimensional classification space using a sparse projection (figure 1(b)). We constrain a linear projection, so that the projected feature vector is sparse, resulting in nontrivial weights for features relevant to classification.

For sparse decomposition, one would like to minimize the number of non-zero elements of the linear decoder x , known as the L_0 "norm" of x , while maintaining a faithful decoding, $Ax \approx b$, of the output labels b using the activity vectors A . As an approximation, L_1 -norm regularized least squares solvers can be employed to minimize $\|x\|_{L_1}$ subject to $\|Ax-b\|_{L_2} \leq \sigma$ (Basis Pursuit Denoising problem (BPDN) (van den Berg & Friedlander 2008)). Here, A is a matrix whose rows are the feature inputs to the decoder (all spike counts of all units in the

relevant time bins) and b is a vector of unique “label” codes for each class. We utilized the original implementation of the Spectral Projected Gradient for L_1 minimization (SPGL1) (van den Berg & Friedlander 2007). BPDN fits the least-squares problem only approximately, and a single parameter (σ) determines a curve that traces the optimal trade-off between the least-squares fit and the one-norm of the solution. At the heart of the approach is the ability to efficiently solve a sequence of Lasso problems (Tibshirani 1996) of the form: $\text{minimize}_x \|Ax - b\|_{L_2}$ subject to $\|x\|_{L_1} \leq \tau$ using a spectral projected-gradient (SPG) algorithm (Birgin et al. 2000; Birgin et al. 2003). The dual solution of this Lasso problem yields vital information on how to update τ so that the next solution of a Lasso problem (with the updated τ) is much closer to the solution of the BPDN problem.

In linear decoding, the labels b serve as low dimensional targets for the projection of the high dimensional feature space. Due to intra-class dispersion, the projection usually results in clusters around the labels b . To allow classification, the labels should therefore be far away from one another in comparison with the intra-class variance. The structure imposed on the low dimensional space by the labels (for example, their order or cyclicity) constrains the projection. To reduce the dependency between labels, we have used a separate dimension for the label of each class (Lin et al. 2006). For example, in the 5-class speech decoding problem we have used 5-element binary vectors as labels for the individual vowels. We therefore had to employ a multiple measurement vectors version of the Basis Pursuit Denoising problem, in which we minimized $\|X\|_{L_{1,2}}$ subject to $\|AX - B\|_F \leq \sigma$,

where: $\|X\|_{L_{1,2}} = \sum x_{L,2}^{(i)}$, $x^{(i)}$ is the i -th row of X , and $\|A\|_F$ is the Frobenius norm:

$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$. The decoding process appears to be insensitive to the parameter σ which was set to $\sigma = 0.99$ manually, but could potentially be chosen automatically as in other regularization methods (Zacksenhouse et al. 2009). In the test phase, given a new set of measurement features (neuronal firing rates at certain time delays) a_{new} , the predicted target (movement direction, phoneme, etc.) will be: $a_{new}X_{reduced}$, where $X_{reduced}$ is the matrix X limited to the task-relevant rows (i.e., those below a small threshold; a_{new} refers to task-relevant features only). The target nearest to the prediction (Euclidean distance) is taken to be the predicted target. Figure 2 is an example of the resultant X matrix for speech decoding based on the aforementioned 5-dimensional labels. The vast majority of neuronal features (776/1011=77%) were given weights close to zero in the decoder matrix X , thus selecting only a sparse subset of neurons that will participate in decoding.

Neuron-dropping

The basic procedure to isolate task-relevant neurons by Neuron Dropping (“Neuron Drop”) was delineated in the Introduction. We have implemented two variants of the method, one which uses an optimal linear estimator for prediction (a common procedure), and another which employs a naïve Bayes classifier.

Add correlated neurons

The “add correlated neurons” (“AddCorr”) method (Vargas-Irwin et al. 2010) is a first order incremental feature selection method based on the correlation coefficient r between the

firing rate of each of the n neurons and behavioral data on part of the training set (“double cross-validation”). For each $k=1,\dots,n$, it constructs a decoder based on the group of k neurons with highest r values, resulting in n decoders. The best decoder (on a testing part of the training set) is then selected. This reduces the complexity from exponential to linear.

Feature selection based on mutual information

Extraction of the important neuron subsets by information theoretical analysis has been suggested for point process decoding for BMI (Wang et al. 2009b). The procedure based on mutual information is similar in nature to the aforementioned “add correlated neurons” procedure, but the best features are selected by Minimal-Redundancy-Maximal-Relevance criterion (“mRMR linear”) (Peng et al. 2005). Maximal-Relevance to the task is defined as the highest mutual information between the firing rate and behavior. However, the combinations of individually good features do not necessarily lead to good classification performance (“the m best features are not the best m features”). To select mutually exclusive features, we add a requirement for minimal dependency, minimizing the average mutual information between selected features.

3. Results

All error rates and runtimes in this section refer to the average for the 6-fold cross-validation, and are accompanied by error bars denoting $\pm$ standard error. The different algorithms are labeled as follows: *Sparse Decomp*: The suggested Sparse Decomposition method; *AddCorr Linear*: Add correlated neurons with linear classifier; *Neuron Drop NaiveBayes*: Neuron dropping with naïve Bayes classifier; *Neuron Drop Linear*: Neuron dropping with linear classifier; *AddCorr NaiveBayes*: Add correlated neurons with naïve Bayes classifier; *mRMR linear*: Feature selection based on mutual information with linear classifier.

Simulation Study

We simulated the responses of 100 neurons to 5 discrete conditions (analog to the 5 vowels), of which 8 different neurons responded to each condition. We compared 6 decoding algorithms: sparse decomposition, neuron dropping with optimal linear classifier, neuron dropping with naïve Bayes classifier, add correlated neurons with optimal linear classifier, add correlated neurons with naïve Bayes classifier, and mutual information-based neuron selection. Sparse decomposition obtained the lowest error rates (7%; figure 3(a)) and the lowest run time (3.3s), over 1 order of magnitude faster than the next method: add correlated neurons with linear classifier (figure 3(b)).

We next examined the dependency of classification accuracy on the level of sparsity of the task-relevant neurons among recorded neurons. Four levels of sparsity were checked: 3, 5, 8, and 10 task-relevant neurons per condition, comparing the two most accurate methods: sparse decomposition and add correlated neurons with linear classifier (figure 3(c)). At all sparsity levels, sparse decomposition obtained significantly lower error rates than the add correlated neurons method (paired-sample one-sided t-test; $p < 0.015$). The graphs also show that the lower the sparsity, the lower the error rate for both methods.

We also varied the amount of overlap of the groups of neurons responsive to each condition (figure 3(d)). Each condition had 5 responsive neurons. Initially, these were distinct groups of neurons (0 overlap). Then, 1 neuron responded to both the first and second conditions, 1 neuron responded to both second and third conditions, and so on (overlap level: 1). For all nontrivial overlap levels (0, 1, ..., 4), sparse decomposition gained significantly lower error rates than the linear add correlated neurons method (paired-sample, one-sided t-test; $p < 0.005$). Not surprisingly, the higher the level of overlap, the higher the error rate.

Directed Movement Decoding

We compared the six aforementioned methods in decoding real human neuronal activity to infer the movement direction in the center-out task (figure 4). We recorded the activity of 928 units during this task. Our samples includes classical cortical motor areas: supplementary motor area proper (83 units) and pre-supplementary motor area (42 units), as well as other areas: hippocampus (183), parahippocampal gyrus (152), cingulate cortex (108), medial pre-frontal cortex (99), entorhinal cortex (91), amygdala (82), temporal-occipital lobes (37), inferior Sylvian fissure (28), occipital lobe (13), parietal lobe (10). Again, sparse decomposition achieved the least error rate (5.6%) at the lowest runtime (0.79s), 3 orders of magnitude faster than linear add correlated neurons, the second fastest method. The average decoder sparsity (over 6 cross-validation runs), defined as the percentage of neuronal features for which all assigned weights were smaller than 0.05% of the maximal weight, was 88% (SE: 1.8%; range: 80-91%).

Speech Decoding

Finally, we compared the six algorithms on real data recorded while human subjects uttered the five vowels (Tankus et al. 2012) (figure 5). We recorded 664 units during this task, and analyzed 579 that were not responsive during any auditory control (see section 6) (rostral anterior cingulate and adjacent medial orbitofrontal cortex (rAC/MOF): 134 of 167; dorsal and subcallosal anterior cingulate cortex (d/sACC): 57/61; entorhinal cortex: 124/138; hippocampus: 103/114; amygdala: 92/106; parahippocampal gyrus: 64/66; superior temporal gyrus: 32/64). The anatomical sub-divisions of the ACC are according McCormick et al. (2006). Sparse decomposition yielded the lowest error rate (6.7%) at minimal runtime (0.50s), 2 orders of magnitude faster than the next fastest method (mutual information-based) and 5 orders of magnitude faster than the slowest (neuron dropping with naïve Bayes classifier). The average decoder sparsity (over 6 cross-validation runs) was 66% (SE: 9.2%; range: 35-85%).

4. Discussion

This study proposed a method for decoding multiple spike trains for BMIs, which automatically selects the task-relevant neurons based on sparse projection of the high dimensional response features onto the low-dimensional classification space. This development could be particularly advantageous for BMIs combining several modalities (e.g., hand movements, leg movement, speech, and audio), where the level of sparseness in recorded neurons may significantly increase. The proposed method outperformed existing feature-selecting decoders, in both simulations and real data recorded from the human brain

while participants uttered vowels or performed directed hand movements. Our results also show that the proposed method performs better regardless of the level of sparsity in the data (figure 3(c)).

The sparse decomposition algorithm also performed several orders of magnitude faster than existing methods, rendering it especially suitable for BMIs, where short training times of the decoder are necessary in order for the patient to operate the device. Although we have examined here only offline decoding, our results imply that following a short training session, a decoder can be constructed in about 1 second (see Results), and then applied online to linearly project neuronal activity vectors for real time decoding. As described above, linear decoders are widespread in the BMI literature and are commonly implemented for real-time online decoding in closed-loop BMIs – decoder training and feature selection are often the time-limiting steps. The short training times obtained here may also allow re-training the decoder when more data is collected in order to improve performance.

Our results also show that the algorithm used here performs better than other neuronal feature selectors regardless of the level of sparsity in the data (figure 3(c)), and that in practice it seems to handle well the very noisy statistics of real human neuronal data from two different modalities. There are several potential advantages in the SPGL1 algorithm that may explain this relatively robust performance. First, formulating decoding as a basis pursuit denoising problem that allows for an error in the approximate least square solution rather than an exact basis pursuit problem (as in (Li et al. 2009a; Li et al. 2011)) could make the method more robust to noise. The noise parameter τ is adaptively computed, rather than set a-priori as in penalized least-squares problem (Shi et al. 2009). Finally, the SPGL1 method uses only matrix-vector operations, and was shown to scale well to large problems.

Sparse coding has been proposed as a guiding principle in neural representations of sensory input, particularly in the visual system (Graham & Field 2006; Ohiorhenuan et al. 2010), but also in the auditory (Greene et al. 2009), olfactory (Jortner et al. 2007) and taste systems (Spector & Travers 2005). The suggested sparse decoding is, in a sense, a complementary process, being a rigorous formulation of the classification of the sparse neuronal code into classes of world representations (for sensory decoding) or behaviors (for output decoding).

Acknowledgments

We thank D. Pourshaban, E. Behnke, T. Fields, and Prof. P. Keating of UCLA and A. Alfassy of the Technion for assistance, Prof. M. Zacksenhouse and A. Shimron of the Technion for insightful comments on the manuscript, and the European Research Council (STG #211055), NINDS, Dana Foundation, Lady Davis and L. and L. Richmond research funds for financial support.

References

- van den Berg E, Friedlander MP. Probing the Pareto Frontier for Basis Pursuit Solutions. *SIAM Journal on Scientific Computing*. 2008; 31(2):890–912.
- van den Berg, E.; Friedlander, MP. SGPL1: A solver for large-scale sparse reconstruction. 2007. Available at: <http://www.cs.ubc.ca/labs/scl/spgl1/>
- Birgin EG, Martínez JM, Raydan M. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis*. 2003; 23(4):539–559.

- Birgin EG, Martínez JM, Raydan M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*. 2000; 10(4):1196–1211.
- de Brecht M, Yamagishi N. Combining sparseness and smoothness improves classification accuracy and interpretability. *NeuroImage*. 2012; 60(2):1550–1561. [PubMed: 22261376]
- Brockwell AE, Kass RE, Schwartz AB. Statistical Signal Processing and the Motor Cortex. *Proceedings of the IEEE*. 2007; 95(5):881–898. [PubMed: 21765538]
- Candes EJ, Wakin MB. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*. 2008; 25(2):21–30.
- Carmena JM, et al. Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. *PLoS Biology*. 2003; 1(2):193–208.
- Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer; 2010.
- Eldar, YC.; Kutyniok, G. *Compressed Sensing: Theory and Applications*. Cambridge University Press; 2012.
- Fraser GW, et al. Control of a brain–computer interface without spike sorting. *Journal of Neural Engineering*. 2009; 6(5):055004. [PubMed: 19721186]
- Ganguly K, Carmena JM. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biology*. 2009; 7(7):e1000153. [PubMed: 19621062]
- Georgopoulos AP, Schwartz AB, Kettner RE. Neuronal population coding of movement direction. *Science*. 1986; 233(4771):1416–1419. [PubMed: 3749885]
- Gerwinn S, Macke J, Bethge M. Bayesian population decoding of spiking neurons. *Frontiers in Computational Neuroscience*. 2009; 3:21. [PubMed: 20011217]
- Graham, DJ.; Field, DJ. Sparse coding in the neocortex. In: Kaas, JH., et al., editors. *Evolution of nervous systems*. Academic Press; 2006. p. 181–187.
- Greene G, et al. Sparse coding of birdsong and receptive field structure in songbirds. *Network*. 2009; 20(3):162–177. [PubMed: 19731147]
- Gupta R, Ashe J. Offline decoding of end-point forces using neural ensembles: application to a brain–machine interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2009; 17(3):254–262. [PubMed: 19497832]
- Héliot R, et al. Learning in closed-loop brain–machine interfaces: modeling and experimental validation. *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics*. 2010; 40(5):1387–1397.
- Herzfeld DJ, Beardsley SA. Improved multi-unit decoding at the brain–machine interface using population temporal linear filtering. *Journal of Neural Engineering*. 2010; 7(4):046012. [PubMed: 20644245]
- Hochberg LR, et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*. 2006; 442(13):164–171. [PubMed: 16838014]
- Humphrey DR, Schmidt EM, Thompson WD. Predicting measures of motor performance from multiple cortical spike trains. *Science*. 1970; 170(959):758–762. [PubMed: 4991377]
- Jortner RA, Farivar SS, Laurent G. A simple connectivity scheme for sparse coding in an olfactory system. *Journal of Neuroscience*. 2007; 27(7):1659–1669. [PubMed: 17301174]
- Kim S-P, et al. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of Neural Engineering*. 2008; 5(4):455–476. [PubMed: 19015583]
- Koyama S, et al. Comparison of brain–computer interface decoding algorithms in open-loop and closed-loop control. *Journal of Computational Neuroscience*. 2010; 29(1-2):73–87. [PubMed: 19904595]
- Lin L, Osan R, Tsien JZ. Organizing principles of real-time memory encoding: neural clique assemblies and universal neural codes. *Trends in Neurosciences*. 2006; 29(1):48–57. [PubMed: 16325278]
- Li Y, et al. Reproducibility and discriminability of brain patterns of semantic categories enhanced by congruent audiovisual stimuli. *PLoS ONE*. 2011; 6(6):e20801. [PubMed: 21750692]

- Li Y, et al. Voxel selection in fMRI data analysis based on sparse representation. *IEEE Transactions on Biomedical Engineering*. 2009a; 56(10):2439–2451. [PubMed: 19567340]
- Li Z, et al. Unscented Kalman filter for brain-machine interfaces. *PloS One*. 2009b; 4(7):e6243. [PubMed: 19603074]
- McCormick LM, et al. Anterior cingulate cortex: An MRI-based parcellation method. *NeuroImage*. 2006; 32:1167–1175. [PubMed: 16859929]
- Moran DW, Schwartz AB. Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology*. 1999; 82(5):2676–2692. [PubMed: 10561437]
- Ohiorehnuan IE, et al. Sparse coding and high-order correlations in fine-scale cortical networks. *Nature*. 2010; 466(7306):617–621. [PubMed: 20601940]
- O an R, et al. Subspace projection approaches to classification and visualization of neural network-level encoding patterns. *PloS One*. 2007; 2(5):e404. [PubMed: 17476326]
- Paninski L, Pillow J, Lewi J. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in Brain Research*. 2007; 165:493–507. [PubMed: 17925266]
- Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max- dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005; 27(8):1226–1238. [PubMed: 16119262]
- Quian-Quiroga R, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*. 2004; 16:1661–1687. [PubMed: 15228749]
- Quian Quiroga R, Panzeri S. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*. 2009; 10(3):173–185.
- Sanchez JC, et al. Ascertaining the importance of neurons to develop better brain-machine interfaces. *IEEE Transactions on Biomedical Engineering*. 2004; 51(6):943–953. [PubMed: 15188862]
- Sanger TD. Probability density estimation for the interpretation of neural population codes. *Journal of Neurophysiology*. 1996; 76(4):2790–2793. [PubMed: 8899646]
- Shi, J., et al. Perceptual decision making investigated via sparse decoding of a spiking neuron model of V1. *International IEEE EMBS Conference on Neural Engineering*; Antalya, Turkey. 2009. p. 558-561.
- Shin H-C, et al. Neural decoding of finger movements using Skellam-based maximum-likelihood decoding. *IEEE Transactions on Biomedical Engineering*. 2010; 57(3):754–760. [PubMed: 19403361]
- Shoham, S. Ph D dissertation. The University of Utah; 2001. Advances towards an implantable motor cortical interface.
- Shoham S, et al. Statistical encoding model for a primary motor cortical brain-machine interface. *IEEE Transactions on Biomedical Engineering*. 2005; 52(7):1312–1322. [PubMed: 16041995]
- Singhal G, et al. Ensemble fractional sensitivity: a quantitative approach to neuron selection for decoding motor tasks. *Computational Intelligence and Neuroscience*, 2010. 2010:648202.
- Spector AC, Travers SP. The representation of taste quality in the mammalian nervous system. *Behavioral and Cognitive Neuroscience Reviews*. 2005; 4(3):143–191. [PubMed: 16510892]
- Tankus A, Fried I, Shoham S. Structured neuronal encoding and decoding of human speech features. *Nature Communications*. 2012 Submitted.
- Tankus A, Yeshurun Y, Fried I. An automatic measure for classifying clusters of suspected spikes into single cells versus multiunits. *Journal of Neural Engineering*. 2009; 6(5):056001. [PubMed: 19667458]
- Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B (Methodological)*. 1996:267–288.
- Ting J-A, et al. Efficient learning and feature selection in high-dimensional regression. *Neural Computation*. 2009; 22(4):831–886. [PubMed: 20028222]
- Vargas-Irwin CE, et al. Decoding complete reach and grasp actions from local primary motor cortex populations. *Journal of Neuroscience*. 2010; 30(29):9659–9669. [PubMed: 20660249]
- Velliste M, et al. Cortical control of a prosthetic arm for self-feeding. *Nature*. 2008; 453:1098–1101. [PubMed: 18509337]

- Wagenaar JB, Ventura V, Weber DJ. State-space decoding of primary afferent neuron firing rates. *Journal of Neural Engineering*. 2011; 8(1):016002. [PubMed: 21245525]
- Wang Y, et al. Sequential Monte Carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces. *Neural Computation*. 2009a; 21(10):2894–2930. [PubMed: 19548797]
- Wang Y, Principe JC, Sanchez JC. Ascertaining neuron importance by information theoretical analysis in motor brain-machine interfaces. *Neural Networks*. 2009b; 22(5-6):781–790. [PubMed: 19615852]
- Wessberg J, et al. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*. 2000; 408:361–365. [PubMed: 11099043]
- Wu W, et al. A hierarchical Bayesian approach for learning sparse spatio-temporal decompositions of multichannel EEG. *NeuroImage*. 2011; 56(4):1929–1945. [PubMed: 21420499]
- Wu W, et al. Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Transactions on Biomedical Engineering*. 2004; 51(6):933–942. [PubMed: 15188861]
- Wu W, et al. Neural decoding of hand motion using a linear state-space model with hidden states. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2009; 17(4):370–378. [PubMed: 19497822]
- Yamashita O, et al. Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns. *NeuroImage*. 2008; 42(4):1414–1429. [PubMed: 18598768]
- Yu, BM., et al. *Neural Information Processing (ICONIP)*. LNCS. Springer; 2008. Neural decoding of movements: from linear to nonlinear trajectory models; p. 586-595.
- Zacksenhouse M, et al. Robust satisficing linear regression: performance/robustness trade-off and consistency criterion. *Mechanical Systems and Signal Processing*. 2009; 23(6):1954–1964. [PubMed: 20161510]
- Zacksenhouse, M.; Nemets, S. Strategies for neural ensemble data analysis for brain-machine interface (BMI) applications. In: Nicolelis, MAL., editor. *Methods for Neural Ensemble Recordings*. Boca Raton, FL: CRC Press; 2008.

Figure 1.

Explicit neuron-selection scheme vs. sparse decomposition scheme. (a) Of the n implanted electrodes, only a subset are task-relevant (channels marked by "1"; (a1)). An explicit selection scheme would ideally isolate these channels (3 in the example; (a2)). Each observation in the task-relevant channels is a point in low dimensional space. The example assumes that in each channel the feature is one dimensional, so the features of the task-relevant population form points in three dimensional space (matrix A ($n \times 3$); (a3)). Let us assume the data consists of two categories (marked as red and blue points). To decode, linear classifiers seek a linear transformation (matrix X (3×1)) that projects the training set (3D) into an even lower dimensional space (1D in the example), in a way that the projected points are easily separable. In the example, red point are ideally projected onto the 1D point: -0.5, whereas blue points, onto 1.5 (a4). These target points, taken for each neuron, form matrix B ($n \times 1$). Matrix X is then the decoder, serving to decode the test set as well. (b)

Selection based on sparsity constraint projects high dimensional feature points from all channels directly onto the target low dimensional (1D) space. The projection in this case is not optimal linear per se, but weighs down task-irrelevant channels.

Figure 2.

Example of a sparse speech decoder, showing the weight matrix (“ $\mathbf{x}$ ”) of a linear speech decoder obtained under sparsity constraint. The weight matrix maps the 1011 neuronal features to a 5-dimensional label space, where each vowel is ideally labeled by: i: (1, 0, 0, 0, 0); e: (0, 1, 0, 0, 0); a: (0, 0, 1, 0, 0); o: (0, 0, 0, 1, 0); u: (0, 0, 0, 0, 1). For 776 of the 1011 features (77%), all 5 weights (columns) were below 0.05% of the maximal weight, and are therefore of little importance to classification and can be neglected.

Figure 3.

Simulation results. (a) Comparison of the error rate of 6 algorithms. Sparse decomposition obtained lowest error rate. (b) Comparison of the CPU time of the algorithms. Sparse decomposition is two orders of magnitude faster than the next fastest method. (c) Comparison of the error rate as a function of sparsity, expressed by the total percentage of neurons responsive to any condition. The error rate decreases as a larger percentage of neurons are responsive. For all sparsity levels, the sparse decomposition method yielded significantly lower error rates than “add correlated” (see text). (d) Comparison of the error rate as a function of the overlap between the groups of neurons responsive to adjacent conditions. The higher the overlap, the higher the error rate. Sparse decomposition resulted in significantly lower error rates for all levels of overlap (see text).

Figure 4.

Decoding of movement direction – comparison of algorithms. (a) Comparison of the average error rates of the algorithms in classification of the 4 movement directions (Left, Right, Up, Down) in a Center-Out task from neural activity. Average is on cross-validation runs. (b) Comparison of the average CPU time consumed by each of the algorithm. Sparse decomposition obtained the lowest error rate at minimal runtime (3 orders of magnitude faster than the next algorithm). Neuron dropping with naive Bayes decoder was ran only on the first cross-validation iteration due to its impractical runtime (>1 CPU week).

Figure 5.

Speech decoding – comparison of algorithms. The figure is organized similarly to figure 4, with classification of neural activity to predict the 5 vowels: a, e, i, o, and u. Sparse decomposition obtained the lowest error rate at minimal runtime (2 orders of magnitude faster than the next algorithm).