Convolutional Dictionary Learning of Stimulus from Spiking Data

Summary

The recording of neural activity in response to repeated presentations of an external stimulus is an established experimental paradigm in the field of neuroscience. Generalized Linear Models (GLMs) are commonly used to describe how neurons encode external stimuli, by statistically characterizing the relationship between the covariates (the stimuli or their derived features) and neural activity. An important question becomes: how to choose appropriate covariates? We propose a data-driven answer to this question that learns the covariates from the neural spiking data, i.e. in an unsupervised manner, and requires minimal user intervention. Specifically, we cast the problem of learning the covariates (or templates) from the data as a Convolutional Dictionary Learning (CDL) problem, where the goal is to learn shift-invariant templates and the times when they occur. Our contribution is two-fold. First, we formulate an optimization objective with sparsity constraints, which accounts for the binary nature of the spikes. This aspect of the data poses an additional challenge as the observations cannot be assumed Gaussian, a common assumption in the CDL literature. Second, we propose iterative algorithms to solve this objective, with our key insight being that the observations need to be modified in a specific manner at each iteration. We apply our framework to neural spiking data recorded from the Barrel cortex of mice in response to periodic whisker deflections. Classical GLM analyses suggest that whisker velocity–obtained from the ideal whisker position programmed into the piezoelectrode used to move the whiskers–strongly modulates neural spiking. Our method obtains an estimate of whisker velocity, which suggests that during a deflection, whisker motion with respect to the piezoelectrode is highly variable. Moreover, when used as a covariate, this data-driven estimate of whisker velocity yields better goodness-of-fit, in terms of the Kolmogorov-Smirnov test, than GLMs.

Additional Details

Better goodness-of-fit for spike data from the Barrel Cortex We apply our framework, termed binomial CDL (BCDL) and described subsequently, to spiking data from the barrel cortex of mice recorded in response to periodic whisker deflections [1]. The dataset consists of spikes from $J = 10$ barrel cortex neurons. Within each trial, a periodic deflection (Fig. 1(a)) lasting 125 ms is applied to a whisker by a piezoelectrode using an *ideal* position waveform, eliciting a response (Fig. 1(b)). Previous GLM analyses [2] demonstrated that neurons respond strongly, and homogeneously across deflections, to ideal whisker velocity, computed as the first difference of the stimulus (Fig. 1(c) blue curve.)

The green curve from Fig. 1(c) depicts the estimate of whisker velocity computed from the neural spiking data using BCDL. The figure demonstrates that, within one period of whisker motion, whisker velocity, and therefore position, is similar to but deviates from the ideal motion programmed into the piezoelectric device. The presence of peaks around 25, 60, and 100 ms suggest that, within one period, the whisker moved multiple times upwards, likely due to whisker motion *with respect to* the piezoelectrode. Fig. 1(d) depicts the 16 non-zero codes that accurately capture the onset of the stimulus in each of the 16 deflection periods. The heterogeneity of amplitudes estimated by BCDL is indicative of the variability of the stimulus component of neural response across deflections. This is in sharp contrast to the GLM, which uses the ideal whisker velocity as a covariate and assumes that neural response to whisker deflections is constant across deflections. In Fig. 1(e), we use the Kolmogorov-Smirnov (KS) test to compare the goodness-of-fit of trained GLM and BCDL to a representative neuron. KS plots are a visualization of the KS test for assessing the
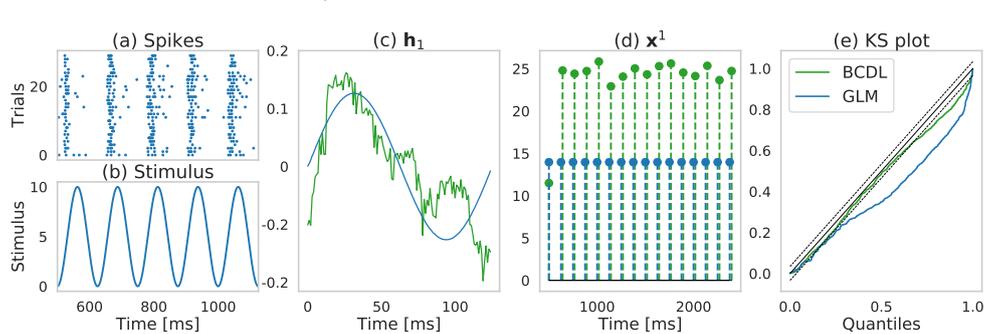
Figure 1: Real data results for a representative neuron. (a) Raster plot for one neuron. (b) The stimulus. (c) Whisker velocity covariate (blue) used in GLM analysis and the templates learned by BCDL (green). (d) The estimated codes (onset of deflection). (e) The goodness-of-fit, with the dotted lines representing 95% confidence intervals.

model's goodness-of-fit to point-process data. The figure shows that BCDL is a much better fit for the data than the GLM. The figure demonstrates that the BCDL framework, which estimates the effect of stimuli from the data in an *unsupervised fashion*, is superior to parametric GLM.

BCDL algorithm We now explain the technical aspects of BCDL. Let $\mathbf{y}^{j,m} \in \{0,1\}^N$ be a binary vector of length N comprising the neural activity from neuron j during trial m. Let $\mathbf{h}_c \in \mathbb{R}^K$ ($K \ll N$) be the template that modulates the neural activity, and $\mathbf{x}_c^j \in \mathbb{R}^{N-K+1}$ be the code vector whose non-zero values give the times when template c effects neuron j, and the amplitude of the effect. The generative model is $\mathbf{y}^j \sim \text{Binomial}(M_j, \boldsymbol{\mu}_j) \in \{0,1\}^{M \times N}$, where $\boldsymbol{\mu}_j = a_j + \sum_{c=1}^C \mathbf{h}_c * \mathbf{x}_c^j$ with a_j denoting the baseline rate of neuron j. Our goal is to learn both $\{\mathbf{h}_c\}_{c=1}^C$ and $\{\mathbf{x}_c^j\}_{j,c=1}^{J,C}$. Due to the ill-posed nature of the problem, we regularize the objective with sparsity constraints, either $\|\mathbf{x}_c^j\|_0$ or $\|\mathbf{x}_c^j\|_1$, as

$$\min_{\{\mathbf{h}_c\}_{c=1}^C, \{\mathbf{x}_c^j\}_{j,c=1}^{J,C}} \sum_{j=1}^J \sum_{m=1}^{M_j} -\left(a_j + \sum_{c=1}^C \mathbf{h}_c * \mathbf{x}_c^j\right)^{\text{T}} \mathbf{y}^{j,m} + \mathbf{1}_N^{\text{T}} \log\left(1 + \exp\left(a_j + \sum_{c=1}^C \mathbf{h}_c * \mathbf{x}_c^j\right)\right), \quad (1)$$

$$\text{such that } \mathbf{x}_c^j \geq 0, \|\mathbf{h}_c\|_2 = 1, \ \|\mathbf{x}_c^j\|_0 \leq \beta_0 \text{ or } \|\mathbf{x}_c^j\|_1 \leq \beta_1.$$

The BCDL framework solves Eq. (1) iteratively until convergence, by combining the alternating-minimization (AM) algorithm of classical CDL [3] and Iteratively Reweighted Least Squares (IRLS) for GLMs. At BCDL iteration g, we first minimize Eq. (1) with respect to $\mathbf{x}_c^{j,(g)}$, with $\mathbf{h}_c^{(g)}$ fixed.

Binary convolutional sparse coding For the ℓ_0 constraint, we use a greedy pursuit method, whereas for the ℓ_1 constraint, we use a proximal gradient mapping. In both of these approaches, $\mathbf{x}_c^{j,(g+1)}$ is estimated iteratively. One of our key insight is that, at each iteration, $\mathbf{y}^{j,m}$ needs to be modified as follows: $\widetilde{\mathbf{y}}_t^{j,m} = \mathbf{y}^{j,m} - \left(1 + \exp(-a_j - \sum_{c=1}^C \mathbf{h}_c^{(g)} * \mathbf{x}_{c,t-1}^j)\right)^{-1}$ for iteration $t = 1, \ldots, T$. The correction term arises from the gradient of the log-likelihood in Eq. (1), and ensures accurate estimation of $\mathbf{x}_c^{j,(g+1)} = \mathbf{x}_{c,T}^j$. Note that a similar correction occurs in the IRLS algorithms for fitting GLMs [2].

Dictionary update To estimate $\mathbf{h}_c^{(g+1)}$, we fix $\mathbf{x}_c^{(g+1)}$ and solve the constrained optimization problem.

Importance of a proper generative model and the accuracy of learned dictionary through simulation In addition to real data, we apply BCDL to simulated data to demonstrate that 1) multiple templates ($C > 1$) can be learned simultaneously and that 2) correct model specification for binary data is important. Fig. 2 summarizes the results, where we simulated $J = 1,000$ neurons, with $M_j = 30$ trials for each neuron and $\boldsymbol{\mu}_j$ is the sum of time-shifted templates $\{\mathbf{h}_c\}_{c=1}^3 \in \mathbb{R}^{50}$ each occurring twice (Fig. 2(a)) at random times. The figure compares the accuracy of the learned templates, $\{\widehat{\mathbf{h}}_c\}_{c=1}^3$, for Gaussian CDL and BCDL, using the metric $\text{err}(\mathbf{h}_c, \widehat{\mathbf{h}}_c) = \sqrt{1 - \langle \mathbf{h}_c, \widehat{\mathbf{h}}_c \rangle^2}$, for $\|\mathbf{h}_c\| = \|\widehat{\mathbf{h}}_c\| = 1$. Gaussian CDL models the observed spikes as Gaussian . Fig. 2(b) demonstrates that, unlike Gaussian CDL, BCDL initialized with random Gaussian templates is able to learn the true dictionary elements. This demonstrates the importance of specifying a proper generative model for the spike data.
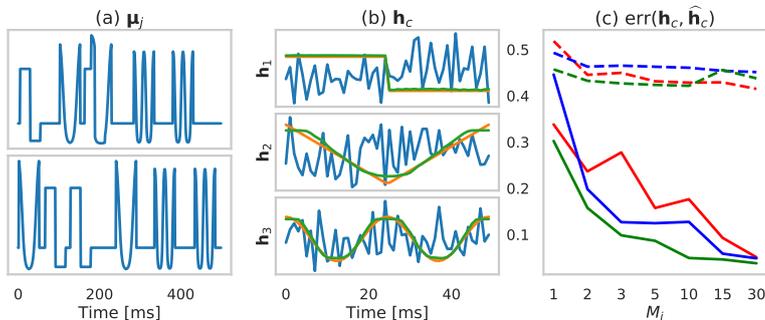

Figure 2: Simulation results. (a) Two examples of $\boldsymbol{\mu}_j$ from which $\mathbf{y}^{j,m}$ are generated. (b) True (orange), initial (blue), and learned (green) templates. (c) Error metric, $\text{err}(\mathbf{h}_c, \widehat{\mathbf{h}}_c) \in [0,1]$ (lower value means the learned templates are closer to the truth) as a function of M_j for BCDL (solid) and Gaussian CDL (dashed).

Despite increasing the number of trials M_j, which provides more information about the templates and the corresponding codes, Gaussian CDL converges, at best, to an error equal to 0.4. The binomial generative model results in a much lower error (0.05).

[1] S. Temereanca et al., Rapid changes in thalamic firing synchrony during repetitive whisker stimulation. *Journal of Neuroscience*, vol. 28, no. 44, pp. 11153-64, 2008.

[2] D. Ba et al., Algorithms for the analysis of ensemble neural spiking activity using simultaneous-event multivariate point-process models. *Frontiers in Computational Neuroscience*, vol. 8, pp. 1-6, 2014.

[3] C. Garcia-Cardona and B.Wohlberg, Convolutional dictionary learning: A comparative review and new algorithms. *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366-381, 2018.