Published in final edited form as:
IEEE Trans Neural Syst Rehabil Eng. 2011 April ; 19(2): 121–135. doi:10.1109/TNSRE.2010.2086079.

Statistical Inference for Assessing Functional Connectivity of Neuronal Ensembles with Sparse Spiking Data

Zhe Chen[Senior Member, IEEE],
Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

David F. Putrino,
Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Soumya Ghosh,
Centre for Neuromuscular & Neurological Disorders, University of Western Australia, QEII Medical Centre, Nedlands, Western Australia, Australia

Riccardo Barbieri[Senior Member, IEEE], and
Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Emery N. Brown[Fellow, IEEE]
Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Harvard-MIT Division of Health Science and Technology, and the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Zhe Chen: zhechen@neurostat.mit.edu; David F. Putrino: trinod01@neurostat.mit.edu; Soumya Ghosh: sghosh@cyllene.uwa.edu.au; Riccardo Barbieri: barbieri@neurostat.mit.edu; Emery N. Brown: enb@neurostat.mit.edu

Abstract

The ability to accurately infer functional connectivity between ensemble neurons using experimentally acquired spike train data is currently an important research objective in computational neuroscience. Point process generalized linear models and maximum likelihood estimation have been proposed as effective methods for the identification of spiking dependency between neurons. However, unfavorable experimental conditions occasionally results in insufficient data collection due to factors such as low neuronal firing rates or brief recording periods, and in these cases, the standard maximum likelihood estimate becomes unreliable. The present studies compares the performance of different statistical inference procedures when applied to the estimation of functional connectivity in neuronal assemblies with sparse spiking data. Four inference methods were compared: maximum likelihood estimation, penalized maximum likelihood estimation, using either ℓ_2 or ℓ_1 regularization, and hierarchical Bayesian estimation based on a variational Bayes algorithm. Algorithmic performances were compared using well-established goodness-of-fit measures in benchmark simulation studies, and the hierarchical Bayesian approach performed favorably when compared with the other algorithms, and this approach was then successfully applied to real spiking data recorded from the cat motor

Correspondence to: Zhe Chen, zhechen@neurostat.mit.edu.

cortex. The identification of spiking dependencies in physiologically acquired data was encouraging, since their sparse nature would have previously precluded them from successful analysis using traditional methods.

Index Terms

Functional connectivity; neuronal interactions; point process generalized linear model; maximum likelihood estimate (MLE); penalized maximum likelihood; ℓ_2 regularization; ℓ_1 regularization; conjugate gradient; interior-point method; variational Bayes; time-rescaling theorem

I. Introduction

Identifying the functional connectivity of a neuronal system using simultaneously recorded neural spike trains has provided valuable implications for understanding the system from a statistical perspective [6], [25]. Statistical modeling of neuronal data has been used for establishing statistical associations or causality between neurons, finding spatiotemporal correlations, or studying the functional connectivity in neuronal networks [10], [3], [50], [32], [31], [41], [14]. This analysis has many functional applications such as neural decoding, and assisting attempts to understand the collective dynamics of coordinated spiking cortical networks [49]. A statistical tool for analyzing multiple spike trains is the theory of random point processes. Statistical inference for point process observations often starts with a certain class of statistical (parametric or nonparametric) model, followed by parameter estimation using a statistical (either maximum likelihood or Bayesian) inference procedure [42], [5], [47]. To date, a number of statistical tools and models have been used to identify functional connectivity between ensemble neurons. The cross-correlogram and joint peri-stimulus time histogram (JPSTH) are standard (and possibly simplest) nonparametric methods for analyzing the interactions between pairwise neurons [35], [20], [1]. However, these tools have serious drawbacks: correlation-based analysis is limited to second-order spike count statistics, which is inadequate for neuronal spike trains. Further, these methods are nonparametric and there is no model validation or goodness-of-fit tests for the data. Recently, point process generalized linear models (GLMs) have been widely used for characterizing functional (spiking) dependence among ensemble neurons [10], [32], [47]. Specifically, the spiking probability of a particular neuron may be modeled as a function of the spiking history of concurrently recorded ensemble neurons (and possibly, a function of the input of the other stimuli as well), and the corresponding parameters of the point process GLM are inferred by maximum likelihood estimation. Bayesian inference has also been recently proposed for GLM inference on neural spike train data [36], [39], [48], [21], [42], [9]. Various approximation procedures have been developed, based on either Laplace approximation, expectation propagation (EP), Markov chain Monte Carlo (MCMC) sampling, or variational approximation.

To date, spike train modeling based upon the point process GLM has been used as a statistical tool in an effective manner to infer functional connections between groups of simultaneously recorded neurons [32]. However, physiological experiments that aim to record neural activity from awake, behaving animals are technically difficult to perform, and at times, the quantities of data that are collected during recording periods can be less than ideal. In addition, neurons that are recorded from animals or human subjects in anesthetized state often fire at very low spiking rates. Most traditional methods that analyze pairwise relationships between neurons simply cannot be used to reliably infer interactions when neural firing rates are low, or the number of trials of a behavioral task that is being studied are low. This presents a rather important problem in the field of neuroscience, as many sets of carefully acquired experimental data are then considered unusable for analysis (i.e.,

because insufficient trials of a difficult behavioral task were acquired, or recorded neurons are firing at an extremely low rate). Theoretically, the more recently developed model-based methods have the ability to reliably perform this analysis, but their effectiveness at being applied to this problem has not been examined. The present paper evaluates the ability of two different approaches to address this problem based on the point process GLM framework: the first one is penalized maximum likelihood estimation that uses ℓ_2 or ℓ_1 regularization, which aims to improve the generalization of the model while reducing the variance of the estimate; the second is hierarchical Bayesian estimation, which uses an efficient variational approximation technique that allows deterministic inference (without resorting to random MCMC sampling). The statistical algorithms under investigation are all capable of handling large-scale problems, but the present paper focuses on relatively small-scale data sets. The current paper focuses on the investigation of different inference algorithms rather than on different modeling paradigms for characterizing functional connectivity. It is worth noting that, in addition to the point process GLMs, other statistical models such as the maximum entropy model [38], [40] and the dynamical Bayesian network [15], are also useful complementary tools for inferring the spiking dependence among ensemble neurons.

II. Point Process Generalized Linear Model

A point process is a stochastic process with 0 and 1 observations [5], [12]. Let $c = 1, \cdots, C$ denote the index of a multivariate (C-dimensional) point process. For the cth point process, let $\boldsymbol{y}^c_{1:T}=(y^c_1,\ldots,y^c_T)$ denote the observed response variables during a (discretized) time interval $[1, T]$, where y^c_t is an indicator variable that equals to 1 if there is a spike at time t and 0 otherwise. Therefore, multiple neural spike train data are completely characterized by a multivariate point process $\{\boldsymbol{y}^c_{1:T}\}^C_{c=1}$. Mathematical backgrounds on the point process theory can be found in [12], [7].

A. Exponential Family and Generalized Linear Models

In the framework of GLM [29], we assume that the observations $\{y_{1:T}\}$ follow an exponential family distribution with the form:

$$p_\theta(y_t|\theta_t)=\exp(y_t\theta_t - b(\theta_t)+c(y_t)), \tag{1}$$

where θ denotes the canonical parameter, and $c(y_t)$ is a normalizing constant. Assume that $b(\theta_t)$ is twice differentiable, then $\mu_t \equiv \mathbb{E}_{y|\theta}[y_t]= \dot{b}(\theta_t)=\frac{\partial b(\theta_t)}{\partial \theta_t}$, $\mathrm{Var}[y_t]=\ddot{b}(\theta_t)=\frac{\partial^2 b(\theta_t)}{\partial \theta_t \partial \theta_t^\top}$ (where $^\top$ denotes the transpose). Moreover, the mean μ_t is related to the linear predictor via a link function g:

$$g(\mu_t)=\eta_t=\beta \boldsymbol{x}_t \tag{2}$$

where $\boldsymbol{x}_t$ denotes the input covariate at time t. Using a canonical link function, the natural parameter relates to the linear predictor by $\theta_t = \eta_t = \boldsymbol{\beta x}_t$,Table I lists two probability distributions of exponential family (in a canonical form) for modeling point process data. In the case of Bernoulli distribution, the link function is a logit function ($\mathrm{logit}(\pi)=\log\frac{\pi}{1-\pi}$); in the case of Poisson distribution, the link function is a log function. Consequently, the point process GLMs based on either logistic regression or Poisson regression can be used to model neural spike trains [47]. The difference between these two models is that in Poisson regression, the generalized "rate" (or conditional intensity function) λ is estimated, whereas in logistic regression, the spiking probability π is directly estimated. When the bin size Δ of

the spike trains is sufficiently small, we can approximate $\pi = \lambda\Delta$ and the difference of using these two models is small. In the present paper, we use the (Binomial) logistic regression GLM for the illustration purpose.

To model the spike train point process data, we use the following logistic regression model with the logit link function.[1] Specifically, let c be the index of target neuron, and let $i = 1, \ldots C$ be the indices of trigger neurons. The Bernoulli (binomial) logistic regression GLM is written as:

$$\text{logit}(\pi_t) = \beta_c \boldsymbol{x}_t = \sum_{j=0}^{d} \beta_j^c x_{j,t} = \beta_0^c + \sum_{i=1}^{C} \sum_{k=1}^{K} \beta_{i,k}^c x_{i,t-k} \tag{3}$$

where $dim(\boldsymbol{\beta}_c) = d + 1$ (where $d = C \times K$) denotes total number of parameters in the augmented parameter vector $\beta_c = \{\beta_0^c, \beta_{i,k}^c\}$, and $\boldsymbol{x}(t) = \{x_0, x_{i,t-k}\}$. Here, $x_0 \equiv 1$ and $x_{i,t-k}$ denotes the spike count from neuron i at the kth time-lag history window. The spike count is nonnegative, therefore $x_{i,t-k} \geq 0$. Alternatively, we can rewrite (3) as

$$\pi_t = \frac{\exp(\beta_c \boldsymbol{x}_t)}{1 + \exp(\beta_c \boldsymbol{x}_t)} = \frac{\exp(\beta_0^c + \sum_{j=1}^{d} \beta_j^c x_{j,t})}{1 + \exp(\beta_0^c + \sum_{j=1}^{d} \beta_j^c x_{j,t})} \tag{4}$$

which yields the probability of a spiking event at time t. It is seen from (4) that the spiking probability π_t is a logistic sigmoid function of $\boldsymbol{\beta}_c \boldsymbol{x}(t)$; when the linear regressor $\boldsymbol{\beta}_c \boldsymbol{x}(t) = 0$, $\pi_t = 0.5$. Note that $\boldsymbol{\beta}_c \boldsymbol{x}(t) = 0$ defines a $(d + 1)$-dimensional hyperplane that determines the decision favoring either $\pi_t > 0.5$ or $\pi_t < 0.5$.

Equation (3) essentially defines a spiking probability model for neuron c based on its own spiking history, and that of the other neurons in the ensemble. It has been shown that such a simple spiking model is a powerful tool for the inference of functional connectivity between ensemble neurons [32], and in predicting single neuronal spikes based on collective population neuronal dynamics [49]. Here, $\exp(\beta_0^c)$ can be interpreted as the baseline firing probability of neuron c. Depending on the algebraic (positive or negative) sign of coefficient $\beta_{i,k}^c$, $\exp(\beta_{i,k}^c)$ can be viewed as a "gain" factor (dimensionless, > 1 or < 1) that influences the current firing probability of neuron c from another neuron i at the previous kth time lag. Therefore, a negative value of $\beta_{i,k}^c$ will strengthen the inhibitory effect and move π_t towards the negative side of the hyperplance; a positive value of $\beta_{i,k}^c$ will enhance the excitatory effect, and thereby moving π_t towards the positive side of the hyperplance. In our paper, two neurons are said to be functionally connected if any of their pairwise connections is nonzero (or the statistical estimate is significantly nonzero).

For the cth spike train point process data, we can write down the log-likelihood function:

$$L(\beta_c) = \sum_{t=1}^{T} [y_t^c \log \pi_t(\beta_c) + (1 - y_t^c) \log \pi_t(1 - \beta_c)] \tag{5}$$

[1] In practice, the value of K (or d) needs to be determined using a statistical procedure, such as the *Akaike information criterion* (AIC) or *Bayesian information criterion* (BIC).

Let $\boldsymbol{\theta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_C\}$ be the ensemble parameter vector, where $dim(\boldsymbol{\theta}) = C(1 + d)$. By assuming that the spike trains of ensemble neurons are mutually *conditionally independent*, the network log-likelihood of C-dimensional spike train data is written as [32]:

$$L(\theta)=\sum_{c=1}^{C} L(\beta_c). \tag{6}$$

Note that the index c is uncoupled from each other in the network log-likelihood function, which implies that we can optimize the function $L(\boldsymbol{\beta}_c)$ separately for individual spike train observations $\boldsymbol{y}_{1:T}^{c}$. For simplicity, from now on we will drop off the index c at notations y_t^c and $\boldsymbol{\beta}_c$ when no confusion occurs.

III. Maximum Likelihood Estimation and Regularization

The objective of the standard maximum likelihood estimation is to maximize (6) given all spike train data. It is known that when the data sample is sufficiently large, the maximum likelihood estimate (m.l.e.) is asymptotically unbiased, consistent, and efficient. However, when the number of samples is small, or the neural spiking rate is small (i.e., the number of '1's in the observation $\boldsymbol{y}$ is sparse), many empirical observations have indicated that m.l.e. produces either wrong or unreliable estimates. The error in the m.l.e. is typically related to two factors: bias and variance, and thus the ultimate goal of statistical estimation is to produce an unbiased minimum variance estimate. The issue of bias and large variance becomes more severe when a data set with a small sample size is encountered, and the size of the parameter space is relatively large. One way to reduce variance is through regularization, which aims to improve the generalization ability of the model (on new data) while fitting finite observed training data. The idea of regularization is to impose certain prior knowledge (such as sparsity) or physiologically plausible constraint (such as temporal smoothness) on the parameters [46], [23]. Furthermore, regularization can be interpreted as imposing a prior on the parameter space from an empirical Bayesian perspective, and the penalized log-likelihood will be interpreted as the log posterior density of the parameters [42], [8]. Therefore, penalized maximum likelihood estimation seeks to maximize a regularized log-likelihood function, which consists of a log-likelihood function plus a penalty function weighted by a regularization parameter. The resultant penalized m.l.e. can be viewed as a *maximum a posteriori* (MAP) estimate.

A. ℓ_2 Regularization

First, let us consider the following penalized log-likelihood function using ℓ_2-regularization:

$$L_2(\beta)=L(\beta) - \rho\beta^{\top}\boldsymbol{Q}\beta \tag{7}$$

where $\rho > 0$ denotes the regularization parameter, and $\boldsymbol{Q}$ denotes a user-defined positive semidefinite matrix. The use of the quadratic term $\boldsymbol{\beta}^{\top}\boldsymbol{Q}\boldsymbol{\beta}$ brings to the name of ℓ_2 regularization. Different choices of matrix $\boldsymbol{Q}$ lead to different regularization solutions (see Appendix A for more discussions on the choices of matrix $\boldsymbol{Q}$). As a special case when $\boldsymbol{Q} = \mathbf{I}$ (identity matrix), the standard "ridge regression" is recovered:

$$L_2(\beta)=L(\beta) - \rho\left\|\beta\right\|_2. \tag{8}$$

Note that equations (7) and (8) are concave function of the parameter vector $\boldsymbol{\beta}$, and minimizing the negative penalized log-likelihood estimation is a convex optimization problem.

Once the regularization parameter ρ is determined (e.g., via cross-validation or regularization path), the optimization problem reduces to maximize a concave function of $\boldsymbol{\beta}$. A standard approach to minimize a convex function is through the Newton method. Specifically, let $\mathbf{H}(\boldsymbol{\beta})$ and $\mathbf{g}(\boldsymbol{\beta})$ denote the Hessian matrix and gradient vector of the parameter vector $\boldsymbol{\beta}$ computed from (7), respectively. Denote $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{T \times (d+1)}$ and $\boldsymbol{y} = [y_1, \ldots, y_T] \in \mathbb{R}^T$, the iterative Newton update equation (at the nth iteration) is given by [16], [34]:

$$\begin{aligned}\beta_{n+1} &= \beta_n - \mathbf{H}^{-1}(\beta_n)\mathbf{g}(\beta_n) \\ &= \beta_n + [\boldsymbol{X}^\top \boldsymbol{W}(\beta_n)\boldsymbol{X} + \rho \boldsymbol{Q}]^{-1} \boldsymbol{X}^\top (\boldsymbol{y} - \widehat{\boldsymbol{y}}(\beta_n))\end{aligned} \tag{9}$$

where $\hat{\boldsymbol{y}}(\boldsymbol{\beta}_n) = [\hat{\pi}_1(\boldsymbol{\beta}_n), \ldots, \hat{\pi}_T(\boldsymbol{\beta}_n)]$, $\boldsymbol{W}(\boldsymbol{\beta}) = \mathrm{diag}\{w_1, \ldots, w_T\}$ is a $T \times T$ diagonal weighting matrix, with diagonal entry $w_t = \pi_t(\boldsymbol{\beta}_n)(1 - \pi_t(\boldsymbol{\beta}_n))$. Equation (9) can also be formulated as iteratively solving a linear quadratic system:

$$[\boldsymbol{X}^\top \boldsymbol{W}(\beta_n)\boldsymbol{X} + \rho \boldsymbol{Q}]\beta_{n+1} = \boldsymbol{X}^\top \boldsymbol{W}(\beta_n)\boldsymbol{b}, \tag{10}$$

where $\boldsymbol{b} = \boldsymbol{X}\boldsymbol{\beta}_n + \boldsymbol{W}^{-1}(\boldsymbol{\beta}_n)(\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{\beta}_n))$. For such a convex optimization problem, efficient iterative algorithms such as the *iteratively reweighted least squares* (IRWLS) [34] or *conjugate gradient* (CG) [28] can be used. For a large-scale data, or a large-size parameter estimation problem, the CG method presents a more computationally efficient solution. The CG algorithm is known to be highly efficient (with a linear complexity proportional to $dim(\boldsymbol{\beta})$), especially when the matrix $\boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X}$ is sparse [28]. Since the optimization problem is convex, the solution (from either IRWLS or CG) will identify the global optimum. The convergence criterion is determined such that the iteration stops when the log-likelihood change in two subsequent updates is less than 10^{-4}.

B. ℓ_1 Regularization

Another popular regularization scheme is through ℓ_1-regularization, which penalizes the ℓ_1 norm of the solution [44]. Unlike ℓ_2 regularization, ℓ_1 regularization favors the sparse solution (i.e., many coefficients in $\hat{\boldsymbol{\beta}}$ are zeros). From a decision-theoretic perspective, ℓ_2-norm is a result of penalizing the mean of a Gaussian prior of the unknown variables, while an ℓ_1 norm penalizes the median of a Laplace prior, which has heavier tails in its distribution shape. Specifically, the penalized log-likelihood function with ℓ_1 norm regularization is written as

$$L_1(\beta) = L(\beta) - \rho \left\| \beta \right\|_1 \tag{11}$$

which is a concave function of $\boldsymbol{\beta}$, but is not twice differentiable with respect to $\boldsymbol{\beta}$ (therefore, the Hessian matrix cannot be computed). Recently, many convex optimization procedures have been proposed for the ℓ_1 regularized GLM [45], [26], [27], [33], [37], [17]. Although individual algorithms differ in their own implementations, the common optimization goal is to seek a sparse solution that simultaneously satisfies the data fitting constraints. We shall briefly describe one efficient and state-of-the-art algorithm based on an interior-point method [27], which will be used for benchmark comparison in the Results section.

The interior-point method for maximizing $L_1(\boldsymbol{\beta})$ in (11) aims to solve an equivalent optimization problem [27]:

$$\begin{aligned} &\text{minimize} && -L(\beta)+\rho\mathbf{1}^\top \boldsymbol{u} \\ &\text{subject to} && -u_j \le \beta_j \le u_j, \quad j=1,\ldots,d, \end{aligned}$$

with variables $\boldsymbol{u} \in \mathbb{R}^d$. The logarithmic barrier for the bound constraints $-u_j \le \beta_j \le u_j$ is

$$\Phi(\beta, \boldsymbol{u}) = -\sum_{j=1}^{d} \log(u_j^2 - \beta_j^2), \tag{12}$$

with domain $\mathrm{dom}\Phi = \{(\boldsymbol{\beta}, \boldsymbol{u}) \in \mathbb{R}^{d+1} \times \mathbb{R}^d | |\beta_j| < u_j, j = 1, \ldots, d\}$. The new weighted objective function augmented by the logarithmic barrier is further written as [27]:

$$E(\beta, \boldsymbol{u}) = -\kappa L(\beta)+\kappa\rho\mathbf{1}^\top \boldsymbol{u}+\Phi(\beta, \boldsymbol{u}) \tag{13}$$

where $\kappa > 0$ is a scalar parameter that defines the central path of a curve of $E(\boldsymbol{\beta}, \boldsymbol{u})$. The new function defined in (13) is smooth and strictly convex, and it can be optimized using the Newton or CG method. Increasing the values of κ leads to a sequence of points on the central path, which ultimately leads to a suboptimal estimate $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{u}})$ [27].

C. Identifying functional connectivity

Upon completing the standard or penalized likelihood inference, we obtain the parameter estimate $\hat{\boldsymbol{\beta}}$. Let

$$\sum \approx I(\beta)^{-1} = -\mathbb{E}\left[\frac{\partial^2 L}{\partial\widehat{\beta}\partial\widehat{\beta}^\top}\right]^{-1} \tag{14}$$

denote the inverse of the negative Hessian matrix of the log-likelihood estimated from the ensemble samples. From the property of m.l.e. it is known that $\boldsymbol{\Sigma}$ approximates the inverse of the Fisher information matrix $I(\boldsymbol{\beta})$; in addition, under the regularity condition and large sample assumption, the m.l.e. $\hat{\boldsymbol{\beta}}$ *asymptotically* follows a multivariate Gaussian distribution [34], [5], with the mean as the true parameter $\boldsymbol{\beta}$ and the covariance matrix given in (14): $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, from which we can further derive the 95% Wald confidence bounds of each element in $\boldsymbol{\beta}$ as $\widehat{\beta}_j \pm 1.96\sum_{jj}^{1/2}$. Provided any of the coefficients are significantly different from zero, or their 95% Wald confidence intervals are not overlapping with 0, we conclude that the "directional connection" (at a certain time lag) between the trigger neuron(s) to target neuron is either excitatory (positive) or inhibitory (negative).

To estimate the matrix $\boldsymbol{\Sigma}$ in (14), in the case of standard maximum likelihood estimation for the GLM, upon convergence we can derive that $\boldsymbol{\Sigma} = (\boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}$; in the case of ℓ_2 penalized maximum likelihood (PML), we have $\boldsymbol{\Sigma} = (\boldsymbol{X}^\top \boldsymbol{W}\boldsymbol{X} + \rho\boldsymbol{Q})^{-1}$: the derivation follows a regularized IRWLS algorithm. In the case of ℓ_1-PML, let $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_-)$ and $\boldsymbol{X} = (\mathbf{1}, \boldsymbol{X}_-)$, upon convergence for the augmented vector $(\beta_0, \boldsymbol{\beta}_-, \boldsymbol{u})$, the Hessian matrix computed from the objective function (13) is given by [27]:

$$\mathbf{H}=\begin{bmatrix} \kappa\mathbf{1}^\top W\mathbf{1} & \kappa\mathbf{1}^\top W X_- & \mathbf{0} \\ \kappa X_-^\top W\mathbf{1} & \kappa X_-^\top W X_- + D_1 & D_2 \\ \mathbf{0} & D_2 & D_1 \end{bmatrix} \in \mathbb{R}^{(2d+1)\times(2d+1)} \tag{15}$$

where $D_1=\text{diag}\{\frac{2(u_1^2+\beta_1^2)}{(u_1^2-\beta_1^2)^2},\ldots,\frac{2(u_d^2+\beta_d^2)}{(u_d^2-\beta_d^2)^2}\}$, and $D_2=\text{diag}\{\frac{-4u_1\beta_1}{(u_1^2-\beta_1^2)^2},\ldots,\frac{-4u_d\beta_d}{(u_d^2-\beta_d^2)^2}\}$. In light of the Schur complement, we obtain

$$\sum=\begin{bmatrix} \kappa\mathbf{1}^\top W\mathbf{1} & \kappa\mathbf{1}^\top W X_- \\ \kappa X_-^\top W\mathbf{1} & \kappa X_-^\top W X_- + D_3 \end{bmatrix}^{-1} \tag{16}$$

where $D_3=D_1 - D_2 D_1^{-1} D_2$.

To quantify the connectivity among C neurons, from (3) we define the mean connectivity ratio as follows:

$$ratio=\frac{1}{K}\sum_{c=1}^{C}\sum_{i=1}^{C}\sum_{k=1}^{K}\frac{\#\{|\beta_{i,k}^c| \gg 0\}}{C(C-1)}. \tag{17}$$

where $\#\{|\beta_{i,k}^c| \gg 0\}$ denotes that the number of the coefficient $\beta_{i,k}^c$ whose statistical estimates are significantly nonzero. Note that the spiking dependence is directional and asymmetric in our statistical model, the spiking dependence between A→B and B→A is not necessarily the same. Therefore, for a total of C ensemble neurons, there are possibly ($C^2 - C$) directions (excluding C self-connection coefficients) between all neuron pairs.

IV. Bayesian Inference and Variational Bayes Method

In addition to the maximum likelihood estimation, another appealing statistical inference tool is Bayesian estimation. The goal of Bayesian inference is to estimate the parameter posterior $p(\boldsymbol{\beta}|\boldsymbol{y})$ given a specific parameter prior $p(\boldsymbol{\beta})$. Normally, because the posterior is analytically non-trackable, we will need to resort to strategies for approximation. These methods include the Laplace approximation for log-posterior [18], [2], expectation propagation (EP) for moment matching [39], [21], and MCMC sampling [36], [18]. In comparison amongst these approximation methods, the Laplace and EP approximations are less accurate (especially when the posterior has multiple modes or the mode is not near the majority of the probability mass); MCMC methods are more general, but have high computational demands, and experience difficulties with assessing the convergence of Markov chains. As an alternative Bayesian inference procedure, variational Bayesian (VB) methods attempt to maximize the lower bound of the marginal likelihood (a.k.a. *evidence*) or the marginal log-likelihood [2]. Unlike Laplace and EP approximation, MCMC and VB methods allow for a fully hierarchical Bayesian inference. Furthermore, the VB method is deterministic, and thereby more computationally efficient, while MCMC methods are prohibitive for large-scaled problems and require careful convergence diagnosis. Here we use the hierarchical variational Bayesian (HVB) algorithm for inferring the parameters in the point process GLM [9].[2]

Specifically, let $\boldsymbol{\alpha}$ denote the hyperparameter set, and we can derive

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \log \int\int p(\boldsymbol{y}|\boldsymbol{x},\beta)p(\beta|\alpha)p(\alpha)\mathrm{d}\beta\mathrm{d}\alpha \geq \int\int q(\beta,\alpha)\log\frac{p(\boldsymbol{y}|\boldsymbol{x},\beta)p(\beta|\alpha)p(\alpha)}{q(\beta,\alpha)}\mathrm{d}\beta\mathrm{d}\alpha \equiv \tilde{L}, \quad (18)$$

where $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$ denotes the prior distribution of $\boldsymbol{\beta}$, specified by the hyperparameter $\boldsymbol{\alpha}$. The variational distribution has a factorial form such that $q(\boldsymbol{\beta}, \boldsymbol{\alpha}) = q(\boldsymbol{\beta})q(\boldsymbol{\alpha})$, which attempts to approximate the posterior $p(\boldsymbol{\beta}, \boldsymbol{\alpha}|\boldsymbol{y})$. This approximation leads to an analytical posterior form if the distributions are conjugate-exponential. The use of hyper-parameters within the hierarchical Bayesian estimation framework provides a modeling advantage compared to the empirical Bayesian approach, since the hierarchical Bayesian modeling employs a fully Bayesian inference procedure that makes the parameter estimate less sensitive to the fixed prior (as in the empirical Bayesian approaches). It is emphasized that the variational log-likelihood $\tilde{L}$ is indeed a *functional*—the function of two variational distributions (or pdfs) $q(\boldsymbol{\beta})$ and $q(\boldsymbol{\alpha})$.

A variational approximation algorithm for logistic regression has been developed in the field of machine learning [24], and it can be easily extended to the Bayesian setting [2]. The basic idea of variational approximation is to derive a variational lower bound for the marginal log-likelihood function. However, the hyperparameters used in [24] are fixed a priori, so their model is empirical Bayesian. Here, we extend the model with hierarchical Bayesian modeling using *automatic relevance determination* (ARD) [30] for the purpose of variable selection. Such a fully Bayesian inference integrated with ARD allows us to design a separate prior for each element β_j in the vector $\boldsymbol{\beta}$ and to set a conjugate prior $p(\boldsymbol{\alpha})$ for the hyperparameters using a common gamma hyperprior. Our prior distributions are set up as follows:

$$p(\beta|\alpha) \sim \mathcal{N}(\beta|\mu_0, \boldsymbol{A}^{-1}) \propto \exp\left(-\tfrac{1}{2}(\beta-\mu_0)^\top \boldsymbol{A}(\beta-\mu_0)\right),$$
$$p(\alpha) = \prod_{j=0}^{d} \mathrm{Gamma}(\alpha_j|a_0, b_0),$$

where $\boldsymbol{A} = \mathrm{diag}\{\boldsymbol{\alpha}\} \equiv \mathrm{diag}\{\alpha_0, \ldots, \alpha_d\}$ (a non-ARD prior is equivalent to setting $\boldsymbol{A} = \alpha\mathbf{I}$ as a special case, where α is a global hyperparameter), and $\mathrm{Gamma}(\alpha_j|a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \alpha_j^{a_0-1} e^{-b_0\alpha_j}$. Here, we assume that the mean hyperparameter is fixed (e.g., $\boldsymbol{\mu}_0 = \mathbf{0}$).

Let $\boldsymbol{\xi} = \{\xi_t\}$ denote the data-dependent variational parameters (that are dependent on the input variables $\{\boldsymbol{x}_t\}$). In light of the variational approximation principle [24], one can derive a tight lower bound for the logistic regression likelihood, which will be used in the VB inference. Specifically, applying the VB inference yields the variational posteriors $q(\boldsymbol{\beta}|\boldsymbol{y})$ and $q(\boldsymbol{\alpha}|\boldsymbol{y})$:

$$\log q(\beta|\boldsymbol{y}) = \log \tilde{p}(\beta, \boldsymbol{\xi}) + \mathbb{E}_{q(\alpha)}[\log p(\beta|\alpha)] = \log \mathcal{N}(\beta|\mu_T, \textstyle\sum_T), \quad (19)$$

[2]The Bayesian logistic regression algorithm [19], which uses a cyclic coordinate descent algorithm for either Gaussian or Laplace prior, is de facto an empirical Bayes algorithm, since the hyperparameter's probability distribution was not modeled and the hyperparameter was selected by heuristics.

$$\begin{aligned}\log q(\alpha|\boldsymbol{y}) &= \mathbb{E}_{q(\beta)}[\log p(\beta|\alpha)] + \log p(\alpha) \\ &= \log\left\{\prod_{j=0}^{d} \text{Gamma}(\alpha_j|a_T, b_{j,T})\right\},\end{aligned} \tag{20}$$

which follow from updates from conjugate priors and posteriors for the exponential family (Gaussian and Gamma distributions). The term $\tilde{p}(\boldsymbol{\beta}, \boldsymbol{\xi})$ appearing in (19) denotes the variational likelihood bound for logistic regression:

$$\begin{aligned}\log p(\beta, \boldsymbol{\xi}) &\geq \log \tilde{p}(\beta, \boldsymbol{\xi}) \\ &= \sum_{t=1}^{T}\left(\log\sigma(\xi_t) - \frac{\xi_t}{2} + \varphi(\xi_t)\xi_t^2\right) - [\beta^\top(\varphi(\xi_t)\boldsymbol{x}_t\boldsymbol{x}_t^\top)\beta] + \beta^\top\left(\boldsymbol{x}_t y_t - \tfrac{1}{2}\boldsymbol{x}_t\right),\end{aligned} \tag{21}$$

where $\sigma(\cdot)$ is a logistic sigmoid function. The variational likelihood bound at the right-hand side of (21) has a quadratic form in terms of $\boldsymbol{\beta}$, and therefore it can be approximated by a Gaussian likelihood (which results in equation 19). The other terms appearing in (19) through (21) are defined by

$$\begin{aligned}\xi_t &= \sqrt{\boldsymbol{x}_t^\top(\Sigma_T + \mu_T\mu_T^\top)\boldsymbol{x}_t} \\ \varphi(\xi_t) &= \frac{\tanh(\xi_t/2)}{4\xi_t} \\ \Sigma_T^{-1} &= \mathbb{E}_{q(\alpha)}[\boldsymbol{A}] + 2\sum_{t=1}^{T}\varphi(\xi_t)\boldsymbol{x}_t\boldsymbol{x}_t^\top \\ \mu_T &= \Sigma_T\left(\mathbb{E}_{q(\alpha)}[\boldsymbol{A}]\mu_0 + \sum_{t=1}^{T}(y_t - 0.5)\boldsymbol{x}_t\right) \\ \mathbb{E}_{q(\alpha)}[\boldsymbol{A}] &= \text{diag}\{a_T/b_{j,T}\} \equiv \boldsymbol{A}_T \\ a_T &= a_0 + 0.5 \\ b_{j,T} &= b_0 + 0.5[(\mu_T)_j^2 + (\Sigma_T)_{jj}],\end{aligned}$$

where the subscript T in the updated parameters represents the fact that the parameters and hyperparameters are updated after passing a total of T samples. Finally, we can derive the variational lower bound of marginal log-likelihood (Appendix B):

$$\begin{aligned}\tilde{L} = \frac{1}{2}&\left\{\mu_T^\top\Sigma_T^{-1}\mu_T + \log|\Sigma_T| + \sum_{t=1}^{T}\left(2\log\sigma(\xi_t) - \xi_t + 2\varphi(\xi_t)\xi_t^2\right)\right\} \\ &+ \sum_{j=0}^{d}\left\{-\log\Gamma(a_0) + a_0\log b_0 - b_0\frac{a_T}{b_{j,T}} - a_T\log b_{j,T} + \log\Gamma(a_T) + a_T\right\}.\end{aligned} \tag{22}$$

The VB inference alternatingly updates (19) and (20) to monotonically increase $\tilde{L}$. The criterion for algorithmic convergence is set until the consecutive change of (22) is sufficiently small (say 10^{-4}). Upon completing the VB inference, the confidence bounds of the estimates can be derived from the posterior mean and the posterior variance [2].

It is noted that due to the assumed factorial form of posterior distribution, the variance of the estimates is relatively underestimated [2]. However, this will have little effect on the identification result. While using the ARD for variable selection, a nonsignificant coefficient is said to be pruned if its mean and variance estimates are both small (close to 0). Therefore,

even if the variance is slightly underestimated, provided that the mean estimate value is is relatively large (or the solution is non-sparse), it will not change the inferred result.

V. Results

Data simulations were used to compare the performance of different statistical inference procedures: (i) the standard ML method, (ii) penalized ML (PML) with ℓ_2 regularization, (iii) PML with ℓ_1 regularization, and (iv) hierarchical VB (HVB) method. A summary of these methods is given in Table II. Based on the performance of these methods with the simulated data, the optimal statistical inference method was also applied to real spike train data. All of the custom statistical inference algorithms were written in MATLAB© (MathWorks, Natick, MA) and can be accessed online: http://neurostat.mit.edu/software. The software of the ℓ_1 regularized logistic regression [27] was accessible from http://www.stanford.edu/~boyd/l1_logreg/.

A. Goodness-of-fit and Performance Metrics

The goodness-of-fit of the point process models estimated from all algorithms is evaluated based on the *Time-Rescaling Theorem* and *Kolmogorov-Smirnov* (KS) test [4], [5].[3] Assuming that a univariate point process specified by J discrete events: $0 < u_1 < \cdots < u_J < T$, defines the random variables $z_j = \sum_{\tau=u_{j-1}}^{u_j} \pi_\tau$ for $j = 1, 2, \ldots, J-1$. Thus, the random variables z_js are independent, and unit-mean exponentially distributed. By introducing the variable of transformation $v_j = 1 - \exp(-z_j)$, then v_js are independent, uniformly distributed within the region [0, 1]. Let $r_j = F^{-1}(v_j)$ (where $F(\cdot)$ denotes the cumulative distribution function (cdf) of the standard Gaussian distribution), then r_js will be independent standard Gaussian random variables. Furthermore, the standard KS test is used to compare the cdf of v_j against that of the random variables uniformly distributed within [0, 1], and the KS statistic measures the maximum deviation of the empirical cdf from the uniform cdf. The KS statistics will be computed for both simulated and real spike trains.

In simulation studies, in addition to the KS test, we also compute the mis-identification error rate, which is the sum of the the false positive (FP) and false negative (FN) rates. By false positive, it is meant that the true connection coefficient (only known in simulations) is zero, but its statistical estimate from the algorithm is mistakenly identified as being significantly nonzero. By false negative, it is meant that the true connection coefficient is nonzero, but the statistical estimate from the algorithm is mistakenly identified as being zero. In simulation studies, we also compute the mean-squared error (MSE) and the normalized MSE (NMSE) of the estimate, which are defined as

$$\mathrm{MSE} = \frac{1}{C}\sum_{c=1}^{C} \left\| \beta_c - \widehat{\beta}_c \right\|_2, \qquad \mathrm{NMSE} = \frac{1}{C}\sum_{c=1}^{C} \frac{\left\| \beta_c - \widehat{\beta}_c \right\|_2}{\left\| \beta_c - \overline{\beta}_c \right\|_2},$$

where $\hat{\boldsymbol{\beta}}_c$ denotes the estimate of the true (simulated) $\boldsymbol{\beta}_c$, and $\overline{\beta_c}$ denotes the mean value of the vector $\boldsymbol{\beta}_c$.

Above all, we like to compare the *bias* and *variance* of the estimates computed from different statistical inference algorithms. Roughly speaking, MSE and RMSE provide two

[3] In the case of fitting low-firing rate neural spike trains, when the lengths of inter-spike intervals are comparable to the length of the observation window, a modified KS test that considers censoring can be used [51].

measures of the estimate's bias. The KS statistics on the testing or validation data as well as the mis-identification rate provide essential information about the estimate's variance. As far as functional connectivity is concerned in the present study, the FP and FN rates are particularly relevant. Ideally, a low mis-identification error rate and a lower KS statistics would be the most important criterion for choosing the best algorithm.

B. Simulation Studies

With the simulation data, a systematic investigation of algorithmic performance was conducted under different conditions. We considered a relatively small network that consisted of 10 simulated neurons with varying connectivity ratios (five scales: 0.1, 0.2, 0.3, 0.4, 0.5). All neurons are assumed to have roughly equal baseline firing rates (four scales: 5, 10, 15, 20 spikes/s), and the nonzero history-dependent coefficients were uniformly randomly generated between $[-h, h]$ (where h was set to be inversely proportional to the baseline firing rate). To create a small spike train data set with short recordings, we simulated multivariate spike trains under the same condition for 8 independent trials, each trial lasting 1000 ms. The input covariates consisted of spike counts from previous temporal windows (equal size of 5 ms) up to 80 ms from all 10 neurons. Thus, there were 16 spike counts from each neuron, totaling $d = 16 \times 10 = 160$ covariates ($dim(\boldsymbol{\beta}_C) = 161$ because of an additional baseline shift parameter). Under each condition, we simulated 20 Monte Carlo runs with varying random seeds. In each condition, we also generated an independent set of trials, which are reserved for testing (or cross-validation). The KS statistics for fitting the training (KStrain) and testing (KStest) data were both computed.

Since the simulated coefficients are random and have no smoothness structure (which may also be the case in real biological systems), we only used the standard ℓ_2 regularization by setting $\boldsymbol{Q} = \mathbf{I}$. The optimal regularization parameters were chosen by cross-validation or leave-one-out procedure. In the ℓ_1 regularization, there are two options for choosing regularization parameters, one based on fixed value followed by cross-validation, and the other based on regularization path [27]. The regularization parameter that resulted in the best KS statistic during cross-validation was selected. We run experiments for 20 Monte Carlo runs using random generated data from 10 virtual neurons and compared the algorithmic performance. The averaged results of the simulation study are summarized in Figs. 1,2,3,4. Note that all mean statistics are averaged over 10 neurons and 20 independent Monte Carlo runs.

The following observations summarize our findings:

1. Testing the algorithms began with the training data, and the KS statistics for each algorithm's performance using the same training data under the different baseline firing rate conditions are displayed in Fig. 1. Across all algorithms, differences in KS statistic were only mildly affected by the varying connectivity ratio in this case. As expected, the KS statistics do decrease as the baseline firing rates of the virtual neurons increase (since more spikes were generated). Interestingly, when the baseline rate was at 5 or 10 Hz, the standard ML algorithm had the highest KS statistics in the training data, but when the baseline rate was increased (at 15 and 20 Hz), its performance became comparable to that of the other two penalized ML algorithms. In all cases, the HVB algorithm performed better, and the majority of the neuron fits fall within the 95% confidence bounds. The HVB's superior KS statistics are more apparent with higher firing rate than with lower firing rates. Because all neurons have similar baseline firing rate, the KS scores from all neurons are also very close in value.
2. The algorithms were then applied to the testing data, and the KS statistics were computed for testing the generalization ability on unseen data from each estimate

(Fig. 2). The ML and HVB algorithms seem to be more robust regardless of the connectivity ratio, while the KS statistics of two penalized ML algorithms appeared to be affected by changes of the connectivity ratio. However, no general trend was observed. Again, the HVB algorithm exhibited the best performance in all of the tested conditions, followed by the ℓ_1-PML algorithm. As a general observation, the two PML algorithms and HVB algorithm all showed noticeable advantages over the standard ML inference, but this was expected, as it was anticipated that the ML estimate would lose its desirable asymptotic properties in a "small data set" scenario. However, while the two PML algorithms clearly outperform the standard ML inference only under low baseline firing rate conditions, the HVB has showed significantly improved performances regardless of the baseline rate. The results also confirm that when the firing rate is high, and the number of '1s' in the point process observations is large, the KS statistics from all (standard or regularized) ML algorithms are similar for both training and testing data sets. Summarizing Figs. 1 and 2, we have seen HVB has better performance than ML and PML methods in both fitting accuracy (training data) and generalization (testing data), under varying connectivity ratios and varying baseline firing rates. Specifically, the advantage of HVB is most prominent in the low connectivity ratio and/or the medium-to-high firing condition.

3. We next considered how the mis-identification (FP+FN) error rate changes as the connectivity ratio increases under different baseline firing rate conditions (Fig. 3). Once again, the overall best performance was produced by use of the HVB algorithm, followed by the ℓ_2-PML algorithm, and finally the standard ML algorithm (the ℓ_1 PML algorithm was excluded from the comparison here because the codes provided in [27] do not produce the confidence bound of the estimates). Under the low baseline firing rate (5 Hz), the PML and HVB had almost identical mis-identification error rates, however, as the baseline firing rate gradually increased, the gap between the PML and HVB algorithms also increased. In our simulations, it was also found that there is an inverse relationship between FP and FN error: an algorithm that had a low FP error rate typically had a relatively greater FN error. A relationship also existed between connectivity ratio and (FP+FN) error rate that occurred regardless of the baseline firing rate, as the connectivity ratio increases to 0.5, the (FP+FN) error rate reaches an asymptotic level for all tested algorithms. This implies that there is a "bottleneck" limit for identifying the true functional connectivity for all algorithms. Therefore, in the worst case, the (FP +FN) error rate can be close to a random guess (50%). The high FN error rate may be due to the fact that many of the "weak connections" in the simulations are mistakenly identified as not showing connections.[4] In the case where a sparse solution is favored (i.e., in ℓ_1-PML and HVB algorithms), weak connectivity coefficients will be pushed towards zero, while only the stronger connections (which matter more) will reach significance. However, as seen in simulations, at about 10–15 Hz baseline rate and with 0.3 connectivity ratio, the mis-identification error rate is still relatively high (~30%) even for the HVB algorithm. Therefore, it remains a challenging task to identify the weak connections (especially weak negative connections where most mis-identification errors occur) in statistical inference.

[4]The seemingly high misidentification error was partially due to the setup of our simulations. While simulating the connection weights from a uniform distribution, we counted all weak connections (even very small values) as evidence of true connectivity. In the case of high connectivity ratio, there will be proportionally more weak connections. Consequently, the tested algorithms often failed to detect the "weak connections", thereby causing a high FN error (see Fig. 3 and Table III). As expected, if in simulations we only maintain the strong connections, then the resultant mis-identification error rate would significantly reduce.

4. Our final comparison using the simulated data involved measuring the bias associated with each algorithm's estimate using MSE and NMSE metrics (Fig. 4). The PML and HVB approaches showed much better performance in MSE and NMSE than the standard ML method, however, the HVB algorithm again performed the best. In most of the testing conditions (except for 20 Hz baseline rate), ℓ_2-PML is better than ℓ_1-PML. This is because ℓ_1 regularization favors a sparse solution, which forces many small or weak connection coefficients' estimates towards zero, possibly causing an increase in the bias of the estimate. As also expected, ℓ_1 regularization had a smaller variance than ℓ_2 regularization (see Fig. 2 on KStest statistics).

5. Overall, the HVB method achieved the best performance in all categories: KS statistics (in both training and testing spike train data sets), MSE and NMSE, as well as the mis-identification error rate. Moreover, in terms of computational speed (to achieve algorithmic convergence), it is observed that the standard ML and ℓ_2-PML algorithms have the fastest convergence, while the ℓ_1-PML and HVB algorithms have comparable convergence rates, roughly $3 \sim 6$ longer (depending on specific simulation) than the other two algorithms. In our experiments, all algorithms converged in all simulated conditions. Generally, when the connectivity ratio is higher or the spiking rate is lower, the convergence speed of ℓ_1-PML is slower; but the convergence speed of HVB remains very similar regardless of the connectivity ratio and spiking rate. Furthermore, when *N*-fold cross-validation was used to select the suboptimal regularization parameter in the ℓ_1 and ℓ_2 PML algorithms, the total *N* -run computational cost and time could be much greater when compared with the single-run HVB algorithm. Interestingly, ℓ_2 PML achieved similar performance to HVB in the mis-identification error, especially in the low firing rate condition. However, comparing the KStrain and KStest statistics and MSE/NMSE indicates that HVB has significantly better performance. This might suggest that PML is effective in finding a "yes/no" solution, but not an accurate solution; in statistical jargon, the PML methods have a good "discrimination" but a relatively poor "calibration". However, the discrimination ability of PML gradually decreases as the firing rate increases.

Additional issues of interest are discussed below.

a) Data Size Analysis—The effect of the data size on algorithmic performance was also examined. This was accomplished by keeping the size of the parameter space intact, but doubling and tripling the size of the training data set, and comparing the performance of the different algorithms. To illustrate the method, we have used a medium (0.3) connectivity ratio, and computed the KS statistics (only on testing data), mis-identification error rate, and MSE for all of the tested algorithms. The results are summarized in Table III, and mixed results amongst the different algorithms are evident. For the standard ML method, increasing data size improves the KS statistic and MSE, but not necessarily the mis-identification error rate. For the penalized ML methods, increasing data size either mildly improves or does not change the MSE or KS statistic, and has a mixed effect on mis-identification error rate. For the HVB method, increasing data size improves the KS statistic but has very little effect on the MSE and mis-identification error rate. These observations suggest that the results obtained using the HVB method are very robust with regard to the data sample size, which is particularly appealing for the small data set problem.

b) Sensitivity Analysis—Except for the standard ML algorithm, the other three algorithms have some additional free parameters (see the last column in Table II). The regularization parameter ρ needs to be selected from cross-validation. The κ parameter is set

in a way that it is gradually increased in the interior-point method in a systematic way [27]. In HVB, the two hyperprior parameters a_0 and b_0 control the shape of the gamma prior (which influences the sparsity of the solution). A close examination using simulation data further revealed that the KS statistics are somewhat insensitive to the choices of a_0 and b_0, although their values may change the respective FP or FN error rate. However, given a wide range of values for (a_0, b_0), the total (FP+FN) error rate remains roughly stable. This suggests that changing the hyperprior parameters of the HVB algorithm will potentially change the desirable sparsity of the solution, which will further affect the trade-off between the FP and FN error. As an illustration, Figure 5 presents the Monte Carlo averaged (across 10 independent runs) performances on the KS statistics and mis-identification error by varying different values of a_0 and b_0 in the HVB algorithm. In these simulations, the connectivity ratio was chosen to be 0.3 and the baseline firing rate was fixed at 10 Hz. As seen from Fig. 5, the performance of the HVB algorithm is relatively robust to a variety range of the hyperprior parameters. In this example, according to the averaged KStrain statistics (minimum 0.110), the optimal set up is (a_0, b_0) = (10^{-3}, 10^{-3}); according to the averaged KStest statistics (minimum 0.146), the optimal setup is (a_0, b_0) = (10^{-4}, 10^{-4}); according to the averaged (FP+FN) error rate (minimum 24.5%), the optimal setup is (a_0, b_0) = (10^{-2}, 10^{-4}). In practice, we have found that the range (a_0, b_0) $\in [10^{-4}, 10^{-2}]$ consistently achieved good performance.

Overall, it was found in our simulations (with various setup conditions) that in the presence of small connectivity ratio, high spiking rate and a large number of spiking data, all tested algorithms produce similar KS statistics and mis-identification error. In the other conditions, the HVB algorithm always has an obviously superior margin

C. Real Spike Train Analysis

The best statistical inference method, the HVB algorithm, was then applied to real spike train data. This experimental data featured groups of neurons that were simultaneously recorded from the primary motor cortex of an awake, behaving cat sitting quietly in a Faraday cage. The experimental protocol for data acquisition, and the behavioral paradigm has been previously reported in detail [22]. In the current study, real neural data is used purely for demonstration purposes, and thus we used recordings from only one animal during a period of motor inactivity (i.e., baseline firing activity) so that neural firing rates were as low as possible (to purposely create sparse spiking data sets). Specific physiological details of the data used for this analysis are provided in Table IV.

Three independent recording sessions were used in this analysis. The first, second and third data sets consist of 13, 15 and 15 neurons, and 18, 18 and 17 trials, respectively, and each trial was 3000 ms in length. The M1 neurons in these data sets were classified as either regular-spiking (RS) or fast-spiking (FS) neurons based upon extracellular firing features. Many of the neurons in these data sets had very low firing rates during the trial periods (Table IV), and the short data recordings and low firing rate fit the two key criteria of the “small data problem”. In Fig. 6, we show representative spike rasters and inter-spike interval (ISI) histograms from one RS and one FS neuron. To estimate the functional connectivity amongst the ensemble neurons, we have assumed that the spiking dependence among the cells remain unchanged across different trials.

We binned the spike trains with 1 ms temporal resolution, and obtained multivariate point process observations. From the observed ISI histograms, we chose the spiking history up to 100 ms and selected 8 history bins (unit: ms) with the following setup:

$$[1 \sim 3, 4 \sim 10, 11 \sim 20, 21 \sim 30, 31 \sim 40, 41 \sim 60, 61 \sim 80, 81 \sim 100]$$

The first spiking history window is chosen to capture the refractory period of the spiking property. For a total of C neurons, the size of parameters for fitting each neuron's spike train recordings is $dim(\boldsymbol{\beta}) = 8C + 1$. For the present problem, $dim(\boldsymbol{\beta}) = 105$ (for $C = 13$) or $dim(\boldsymbol{\beta}) = 121$ (for $C = 15$). For the inference of the HVB algorithm, the initial parameters were set as follows: $a_0 = b_0 = 10^{-3}$, and $\boldsymbol{\mu}_0 = \mathbf{0}$, although it was found that the results are insensitive to these values. Upon fitting all 43 neurons, 30 neurons' KS plots (Dataset-1: 8/13; Dataset-2: 12/15; Dataset-3: 10/15) are completely within the 95% confidence bounds, and 41 neurons' KS plots are within the 90% confidence bounds. These highly accurate fits indicate that the statistical model (3) can satisfactorily characterize the spiking activity of individual neurons. Using the HVB algorithm, the inferred (averaged) network connectivity ratios from 3 data sets are 0.27, 0.21, and 0.13, respectively. Amongst all three data sets, it was found that the fraction of neural interactions is the highest amongst the FS-FS cell-pairing, followed by FS-RS and RS-RS groups (Table IV), which supports previous studies investigating the network properties of M1 neurons [8]. The relatively lower connectivity ratio in Dataset-3 is due to a lower ratio of FS/RS neurons (Table IV). These results suggest that during periods of motor inactivity, FS neurons (inhibitory interneurons) are involved in more functional connectivity than RS neurons (pyramidal cells) in M1. Furthermore, a close examination of the neural interactions inferred by the HVB algorithm also reveals that many FS neurons have inhibitory effects on the spiking activity of RS neurons. Figure 7 illustrates an example of such spiking dependence between one FS and one RS neuron—note that the inhibitory spiking dependence at a fine timescale was not detectable by a more traditional method (the JPSTH). Finally, the strengths of (absolute value of $\beta_{i,k}^{c}$ averaged over K time lags) neuronal interactions inferred from 3 data sets are shown in Fig. 8. In each plot, the size of the circle is proportional to the relative (normalized) strength respective to all neurons (including self-interactions).

VI. Discussion and Conclusion

Inferring functional connectivity of a neuronal network using simultaneously recorded spike trains is an important task in computational neuroscience, and the point process GLM provides a principled framework for statistical estimation. However, in addition to employing a favorable statistical model, the selection of an appropriate statistical inference algorithm is also a crucial undertaking. The present study aimed to solve a problem that has been present in the practice of experimental brain recording for many years: the reliability of sparse data sets for analysis [43]. Thus, this paper investigates several statistical inference procedures, while also applying these methods to the sparse spiking data. Many sets of experimental data are not appropriate for analysis of spiking dependence either because they feature neurons that are spiking at very low frequencies, or because during a difficult behavioral experiment, too few trials of the desired behavior are secured. Essentially, improving the reliability of the statistical estimates was the focus of our study, and simulated spike train data were used to compare different statistical inference algorithms. The four algorithms that were tested were the standard ML method, the ℓ_2 and ℓ_1 PML methods, and the HVB method. Systematic investigations were conducted to compare the performance of the algorithms (in terms of the KS statistics, MSE, and mis-identification error) under different conditions (with varying baseline firing rate and network connectivity ratio). From the Monte Carlo estimation results we conclude that (i) the HVB algorithm performed the best amongst all tested algorithms, and (ii) regularization is very important for the maximum likelihood estimation, especially in the presence of neurons with low firing rate. As an illustration, we apply the HVB algorithm to real spike train recordings from ensemble M1 neurons.

The hierarchical Bayesian method has been shown to be a powerful tool for statistical inference [18], whose applications have gone far beyond the point process GLM. The VB

inference is appealing for Bayesian inference from the perspective of computational efficiency, with the goal of maximizing the lower bound of the marginal log-likelihood of observed data. The ARD principle employed in the Bayesian inference procedure provides a natural way for variable selection in that redundant or insignificant features will be shown to have smaller weights (close to zeros), thereby favoring a sparse solution. The sparsity of the solution is controlled by the hyperprior parameters, which can be set to be non-informative. Finally, the full posteriors of the parameters can be obtained, which can be used to compute a predictive distribution of unseen data [2].

The framework of the point process GLM and the HVB method provide a new way to investigate the neural interactions of ensemble neurons using simultaneously recorded spike trains. The point process GLM using the collective neuronal firing history has been shown to be effective in predicting single neuron spikes from humans and monkeys [49], which has potential applications for neural prosthetic devices. In addition, our proposed methodology can be used for assessing neural interactions. Since the point process GLM using a network likelihood function enabled us to assess spiking dependencies in populations of simultaneously recorded neurons, our approach is favorable when compared with traditional techniques (e.g., cross-correlation or JPSTH), as it may be used to examine functional connectivity as it occurs in multiple neurons simultaneously, compared with only being able to perform pairwise analysis. This is appealing for examining neural interactions at different regions of the brains, or for conducting quantitative comparison during different behaviors or task performances. The findings of our study indicates that the proposed HVB method provides a satisfactory solution to the “sparse spiking data problem” faced by many neuroscience researchers, and this method appears to outperform other contemporary statistical inference procedures in the assessment of functional connectivity in sets of spike train data where sparsity is not an issue.

Similar to other contemporary statistical approaches, our method for inferring the functional connectivity of ensemble neurons relies on certain statistical assumptions. For example, the stationarity of neuronal data during short durations of trials as well as across trials. Whilst this assumption is not always valid between trials, the non-stationarity issue across trials can be addressed by considering a random-effects GLM [11], and maximum likelihood and Bayesian inference procedures can be developed accordingly.

Acknowledgments

This work was supported by the National Institutes of Health (NIH) under Grant DP1-OD003646, Grant R01-DA015644, and Grant R01-HL084502.

References

1. Aertsen A, Gerstein G, Habib MK, Palm G. Dynamics of neuronal firing correlation: modulation of ‘effective connectivity’. J Neurophysiol. 1989; 61:900–917. [PubMed: 2723733]
2. Bishop, CM. Pattern Recognition and Machine Learning. New York: Springer; 2006.
3. Brillinger, DR.; Villa, EP. Assessing connections in networks of biological neurons. In: Brillinger, DR.; Fernholz, LT.; Morgenthaler, S., editors. The Practice of Data Analysis: Essays in Honor of John W Turkey. 1997. p. 77-92.
4. Brown EN, Barbieri R, Ventura V, Kass RE, Frank LM. The time-rescaling theorem and its application to neural spike data analysis. Neural Computat. 2002; 14(2):325–346.
5. Brown, EN.; Barbieri, R.; Eden, UT.; Frank, LM. Likelihood methods for neural data analysis. In: Feng, J., editor. Computational Neuroscience: A Comprehensive Approach. CRC Press; 2003. p. 253-286.
6. Brown EN, Kass RE, Mitra PP. Multiple neural spike train data analysis: state-of -the-art and future challenges. Nat Neurosci. 2004; 7(5):456–461. [PubMed: 15114358]

7. Brown, EN. Theory of point processes for neural systems. In: Chow, CC., et al., editors. Methods and Models in Neurophysics. Elsevier; 2005. p. 691-727.
8. Chen, Z.; Putrino, DF.; Ba, DE.; Ghosh, S.; Barbieri, R.; Brown, EN. A regularized point process generalized linear model for assessing the functional connectivity in the cat motor cortex. Proc. IEEE EMBC'09; Minneapolis, MN. 2009. p. 5006-5009.
9. Chen, Z.; Kloosterman, F.; Wilson, MA.; Brown, EN. Variational Bayesian inference for point process generalized linear models in neural spike trains analysis. Proc. IEEE ICASSP'10; Dallas, TX. 2010. p. 2086-2089.
10. Chornoboy E, Schramm L, Karr A. Maximum likelihood identication of neural point process systems. Biol Cybern. 1988; 59:265–275. [PubMed: 3196770]
11. Czanner G, Eden UT, Wirth S, Yanike M, Suzuki WA, Brown EN. Analysis of between-trial and within-trial neural spiking dynamics. J Neurophy. 2008; 99:2672–2693.
12. Daley, DJ.; Vere-Jones, D. An Introduction to the Theory of Point Processes. 2. New York: Springer; 2003.
13. Efron B, Hastie T, Johnstone I, Tibshirani R. Least-angle regression. Ann Stat. 2004; 32(2):407–499.
14. Eldawlatly S, Jin R, Oweiss K. Identifying functional connectivity in large scale neural ensemble recordings: A multiscale data mining approach. Neural Computat. 2009; 21:450–477.
15. Eldawlatly S, Zhou Y, Jin R, Oweiss K. On the use of dynamic Bayesian networks in reconstructing functional neuronal networks from spike train ensembles. Neural Computat. 2010; 22:158–189.
16. Fahrmeir, L.; Tutz, G. Multivariate Statistical Modelling Based on Generalized Linear Models. 2. New York: Springer; 2001.
17. Friedman J, Hastie T, Tibshirani R. Regularized paths for generalized linear models via coordinate descent. J Statistical Software. 2010; 33(1)
18. Gelman, A.; Carlin, JB.; Stern, HS.; Rubin, DB. Bayesian Data Analysis. 2. Chapman & Hall/CRC; 2004.
19. Genkin, A.; Lewis, DD.; Madigan, D. Tech Rep. Rutgers Univ; 2004. Large-scale Bayesian logistic regression for text categorization. http://www.stat.rutgers.edu/~madigan/BBR/
20. Gerstein GL, Perkel DH. Simultaneous recorded trains of action potentials: analysis and functional interpretation. Science. 1969; 164:828–830. [PubMed: 5767782]
21. Gerwinn, S.; Macke, JH.; Seeger, M.; Bethge, M. Bayesian inference for spiking neuron models with a sparsity prior. In: Platt, JC.; Koller, D.; Singer, Y.; Roweis, S., editors. Adv Neural Info Proc Syst (NIPS). Vol. 20. Cambridge, MA: MIT Press; 2008. p. 529-536.
22. Ghosh S, Putrino DF, Burro B, Ring A. Patterns of spatio-temporal correlations in the neural activity of the cat motor cortex during trained forelimb movements. Somatosensory and Motor Research. 2009; 26:31–49. [PubMed: 19697261]
23. Hebiri, M. Regularization with the smooth-lasso procedure. 2008. http://arxiv.org/abs/0803.0668
24. Jaakkola TS, Jordan MI. Bayesian parameter estimation via variational methods. Statist Comput. 2000; 10:25–37.
25. Kass RE, Ventura V, Brown EN. Statistical issues in the analysis of neuronal data. J Neurophysiol. 2005; 94:8–25. [PubMed: 15985692]
26. Krishnapuram B, Carin L, Figueiredo MAT, Hartemink AJ. Sparse multinomial logistic regression: fast algorithms and generalization bounds. IEEE Trans Patt Anal Mach Intell. 2005; 27(6):957–968.
27. Koh K, Kim SJ, Boyd S. An interior-point method for large-scale ℓ_1-regularized logistic regression. J Machine Learning Res. 2007; 8:1519–1555.
28. Komarek, P. PhD thesis. Carnegie Mellon University; 2004. Logistic regression for data minding and high-dimensional classification.
29. McCullagh, P.; Nelder, J. Generalized Linear Models. 2. London: Chapman & Hall; 1989.
30. Neal, R. Bayesian Learning for Neural Networks. New York: Springer; 1996.
31. Nykamp D. A mathematical framework for inferring connectivity in probabilistic neuronal networks. Mathematical Biosciences. 2007; 205:204–251. [PubMed: 17070863]

32. Okatan M, Wilson MA, Brown EN. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. Neural Computat. 2005; 17:1927–1961.
33. Park MY, Hastie T. An ℓ_1 regularization-path algorithm for generalized linear models. J Roy Stat Soc B. 2007; 69(4):659–677.
34. Pawitan, Y. In All Likelihood: Statistical Modelling and Inference Using Likelihood. New York: Oxford Univ. Press; 2001.
35. Perkel DH, Gerstein GL, Moore GP. Neuronal spike trains and stochastic point processes. II. simultaneous spike trains. Biophy J. 1967; 7:419–440.
36. Rigat F, de Gunst M, van Pelt J. Bayesian modelling and analysis of spatio-temporal neuronal networks. Bayesian Analysis. 2006; 1(4):733–764.
37. Schmidt, M.; Fung, G.; Rosaless, R. Tech Rep. Dept. Computer Sciences, Univ. Wisconsin; 2009. Optimization methods for ℓ_1-regularization. URL: http://pages.cs.wisc.edu/~gfung/GeneralL1/
38. Schneidman E, Berry M, Segev R, Bialek W. Weak pair-wise correlations imply strongly correlated network states in a neural population. Nature. 2006; 440:10007–10212.
39. Seeger M, Gerwinn S, Bethge M. Bayesian inference for sparse generalized linear models. Proc ECML'07. 2007:298–309.
40. Shlens J, Field GD, Gauthier JL, et al. The structure of multi-neuron firing patterns in primate retina. J Neurosci. 2006; 26:8254–8266. [PubMed: 16899720]
41. Stevenson IH, Rebesco JM, Miller LE, Körding KP. Inferring functional connections between neurons. Curr Opin Neurobiol. 2008; 18:582–588. [PubMed: 19081241]
42. Stevenson IH, Rebesco JM, Hatsopoulos NG, Haga Z, Miller LE, Körding KP. Bayesian inference of functional connectivity and network structure from spikes. IEEE Trans Neural Syst Rehab Engr. 2009; 17:203–213.
43. Strangman G. Detecting synchronous cell assemblies with limited data and overlapping assemblies. Neural Computat. 1997; 9:51–76.
44. Tibshirani R. Regression shrinkage and selection via the Lasso. J Roy Stat Soc B. 1996; 58(1): 267–288.
45. Tibshirani R. The Lasso for variable selection in the Cox model. Statistics in Medicine. 1997; 16:385–395. [PubMed: 9044528]
46. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. J Roy Stat Soc B. 2005; 67:91–108.
47. Truccolo W, Eden UT, Fellow M, Donoghue JD, Brown EN. A point process framework for relating neural spiking activity to spiking history, neural ensemble and covariate effects. J Neurophysiol. 2005; 93:1074–1089. [PubMed: 15356183]
48. Truccolo W, Donoghue JD. Nonparametric modeling of neural point processes via stochastic gradient boosting regression. Neural Computation. 2007; 19(3):672–705. [PubMed: 17298229]
49. Truccolo W, Hochberg LR, Donoghue JP. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. Nat Neurosci. 2010; 13:105–111. [PubMed: 19966837]
50. Utikal KJ. A new method for detecting neural interconnectivity. Biol Cybern. 1997; 76:459–470.
51. Wiener MC. An adjustment to the time-rescaling method for application to short-trial spike train data. Neural Computat. 2003; 15:2565–2576.

Appendix A: Design of Desirable Positive-Definite Matrix Q

Assuming that the spiking history dependent coefficients change smoothly between the neighboring temporal windows), we may impose a "local smoothness" constraint on the parameters. Heuristically, when the parameter sequences $\{\beta_{c,k}\}$ are temporally smooth for any index c, the local variance will be relatively small. Let $\bar{\beta}_{c,k}$ denote the corresponding short-term exponentially weighted average of $\beta_{c,k}$:

$$\bar{\beta}_{c,k}=\gamma\bar{\beta}_{c,k-1}+(1-\gamma)\beta_{c,k} \tag{23}$$

where $0 < \gamma < 1$ is a forgetting factor that determines the range of local smoothness. The role of γ is to act like a low-pass filter: the smaller the value of γ, the more emphasis is placed on the $\beta_{c,k}$, and a smaller smoothing effect emerges. Let us define a new quadratic penalty function from (23):

$$\sum_c \sum_k (\beta_{c,k} - \bar{\beta}_{c,k})^2 = \sum_c \sum_k \gamma (\beta_{c,k} - \bar{\beta}_{c,k-1})^2, \tag{24}$$

which penalizes the local variance of $\{\beta_{c,k}\}$. Let $\bar{\boldsymbol{\beta}}_c$ denote the short-term average vector for the corresponding parameter $\boldsymbol{\beta}_c = [\beta_{c,1}, \ldots, \beta_{c,K}]$, then we further have

$$\left\| \beta_c - \bar{\beta}_c \right\|^2 = \left\| \beta_c - \mathbf{S}\beta_c \right\|^2, \tag{25}$$

where a smoothing matrix $\mathbf{S}$ is introduced to represent $\bar{\boldsymbol{\beta}}_c$ in terms of $\boldsymbol{\beta}_c$. Note that the exponentially moving average $\bar{\beta}_{c,k}$ can be viewed as a *convolution product* between the sequences $\{\beta_{c,k}\}$ and a template. Suppose the template vector has an exponential-decay property with length 4, such that *template* = $[(1-\gamma), \gamma(1-\gamma), \gamma^2(1-\gamma), \gamma^3(1-\gamma)]$. Note that the convolution smoothing operation can also be conveniently expressed as a matrix product operation: $\bar{\boldsymbol{\beta}}_c = \mathbf{S}\boldsymbol{\beta}_c$, where $\mathbf{S}$ is a Toeplitz matrix with the right-shifted template appearing at each row given as follows:

$$\mathbf{S} = \begin{bmatrix} 1-\gamma & 0 & 0 & 0 & \cdots & 0 \\ \gamma(1-\gamma) & 1-\gamma & 0 & 0 & \cdots & 0 \\ \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & 0 & \cdots & 0 \\ \gamma^3(1-\gamma) & \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & 0 & \cdots \\ 0 & \gamma^3(1-\gamma) & \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & \cdots \\ & & \cdots & 0 & \ddots & \vdots \end{bmatrix}$$

Finally, we obtain the regularization matrix $\boldsymbol{Q} = \mathbf{P}^\top \mathbf{P}$, where the matrix $\mathbf{P}$ has a block-Toeplitz structure:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{P}_2 & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{P}_C \end{bmatrix} = \begin{bmatrix} \mathbf{I}-\mathbf{S} & 0 & \cdots & 0 \\ 0 & \mathbf{I}-\mathbf{S} & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{I}-\mathbf{S} \end{bmatrix}. \tag{26}$$

where $dim(\mathbf{I} - \mathbf{S}) = K$, $dim(\boldsymbol{Q}) = KC$, and the number of blocks is equal to C. It is worth commenting that our smoothing operator can be seen as an extension of the contingent smoothness operator [46], where the term $(\beta_{c,k} - \bar{\beta}_{c,k})^2$ in equation (24) is replaced by $(\beta_{c,k} - \beta_{c,k-1})^2$ (i.e., the local mean $\bar{\beta}_{c,k}$ is replaced by its intermediate neighbor $\beta_{c,k-1}$ without using any moving averaging). Nevertheless, our regularization operator is more general and also accommodates [23] as a special case. Like ours, the regularization matrix $\boldsymbol{Q}$ in [23] also has a block-Toeplitz structure. Note that when $\gamma = 1$, $\mathbf{S}$ will be an all-zeros matrix, $\mathbf{P}$ and $\boldsymbol{Q}$ will become identity matrices, and our smoothed regularization will reduce to the standard "ridge regularization"; on the other hand, when $\gamma = 0$, $\mathbf{S}$ will be an identity matrix, $\mathbf{P}$ and $\boldsymbol{Q}$ will become all-zeros matrices, therefore no regularization is imposed. Hence, our approach

($0 < \gamma < 1$) can be viewed as a *quantitative* choice between two extrema of no regularization ($\gamma = 0$) and ridge regularization ($\gamma = 1$).

As already mentioned, we may use the contingent smoothness operator as in [46], in which $\{\beta_{c,k}\}_{k=1}^{K}$ is viewed as a curve, and the first-order derivative of the curve is approximated by $\beta_{c,k} - \beta_{c,k-1}$. If we penalize the norm of the first-order derivative, the objective function is then written as

$$L_2(\beta_c) = L(\beta_c) - \rho \sum_k \left\| \beta_{c,k} - \beta_{c,k-1} \right\|^2, \tag{27}$$

and the ith block ($i = 1, \ldots, C$) of the block-diagonal matrix $\mathbf{P}$ in (27) is derived as

$$\mathbf{P}_i = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & -1 \end{bmatrix}, \quad \text{and} \quad \mathbf{P}_i^{\top}\mathbf{P}_i = \begin{bmatrix} -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{bmatrix},$$

which both have a banded-diagonal structure.

Appendix B: Derivation of Equation (22)

From equations (18) and (21), we can derive the variational lower bound of the marginal log-likelihood (for notation simplicity, the dependence of $\boldsymbol{y}$ in all posteriors is made implicit):

$$\tilde{L} = \mathbb{E}_{q(\beta)}[\log \tilde{p}(\boldsymbol{y}|\beta)] + \mathbb{E}_{q(\beta)q(\alpha)}[\log p(\beta|\alpha)] - \mathbb{E}_{q(\beta)}[\log q(\beta)] + \mathbb{E}_{q(\alpha)}[\log p(\alpha)] - \mathbb{E}_{q(\alpha)}[\log q(\alpha)]. \tag{28}$$

where the individual terms in (28) are given by

$$\mathbb{E}_{q(\beta)}[\log \tilde{p}(\boldsymbol{y}|\beta)] = \frac{1}{2}\mu_T^{\top}\sum_T^{-1}\mu_T - \frac{d+1}{2} + \frac{1}{2}\left(\mu_T^{\top}\boldsymbol{A}_T\mu_T + \mathrm{tr}\left(\sum_T \boldsymbol{A}_T\right)\right) + \frac{1}{2}\sum_{t=1}^{T}\left(2\log\sigma(\xi_t) - \xi_t + 2\varphi(\xi_t)\xi_t^2\right) \tag{29}$$

$$\mathbb{E}_{q(\beta)a(\alpha)}[\log p(\beta|\alpha)] = -\frac{d+1}{2}\log(2\pi) + \sum_{j=0}^{d}\frac{1}{2}\left(\psi(a_T) - \log b_{j,T}\right) - \frac{1}{2}\left(\mu_T^{\top}\boldsymbol{A}_T\mu_T + \mathrm{tr}\left(\sum_T \boldsymbol{A}_T\right)\right) \tag{30}$$

$$\mathbb{E}_{q(\beta)}[\log q(\beta)] = -\frac{d+1}{2}(1+\log(2\pi)) + \frac{1}{2}\log\left|\sum_T\right| \tag{31}$$

$$\mathbb{E}_{q(\alpha)}[\log p(\alpha)] = \sum_{j=0}^{d} \left(a_0 \log b_0 - \log\Gamma(a_0) + (a_0 - 1)(\psi(a_T) - \log b_{j,T}) - b_0 \frac{a_T}{b_{j,T}} \right) \tag{32}$$

$$\mathbb{E}_{q(\alpha)}[\log q(\alpha)] = \sum_{j=0}^{d} \left((a_T - 1)\psi(a_T) - \log\Gamma(a_T) + \log b_{j,T} - a_T \right) \tag{33}$$

where $\Gamma(\cdot)$ denotes the gamma function, and $\psi(\cdot)$ denotes the digamma function, which is defined as the logarithmic derivative of the gamma function. Other notations have been defined earlier following equation (21). Summarizing equations (29) through (33) yields (22). The pseudocode of the HVB algorithm is given below.

Algorithm 1

Pseudocode for the HVB algorithm.

Initialize the hyperprior parameters a_0 and b_0, and set the initial parameter value to 0 (i.e. $\boldsymbol{\beta} = \mathbf{0}$).

while convergence criteria not met **do**

Evaluate the data-dependent variational parameters $\boldsymbol{\xi} = \{\xi_t\}$ for each data points $\boldsymbol{x}_t$ ($t = 1, \ldots, T$).

Update the parameter variational posterior mean μ_T and variance $\boldsymbol{\Sigma}_T$.

Update the noise precision hyperparameter $\mathbb{E}_{q(\alpha)}[\boldsymbol{A}]$.

Update the hyperprior parameters a_T and $b_{j,T}$ ($j = 0, \ldots, d$).

Compute the variational lower bound of marginal log-likelihood $\tilde{L}$ (Eq. 22).

end while

end

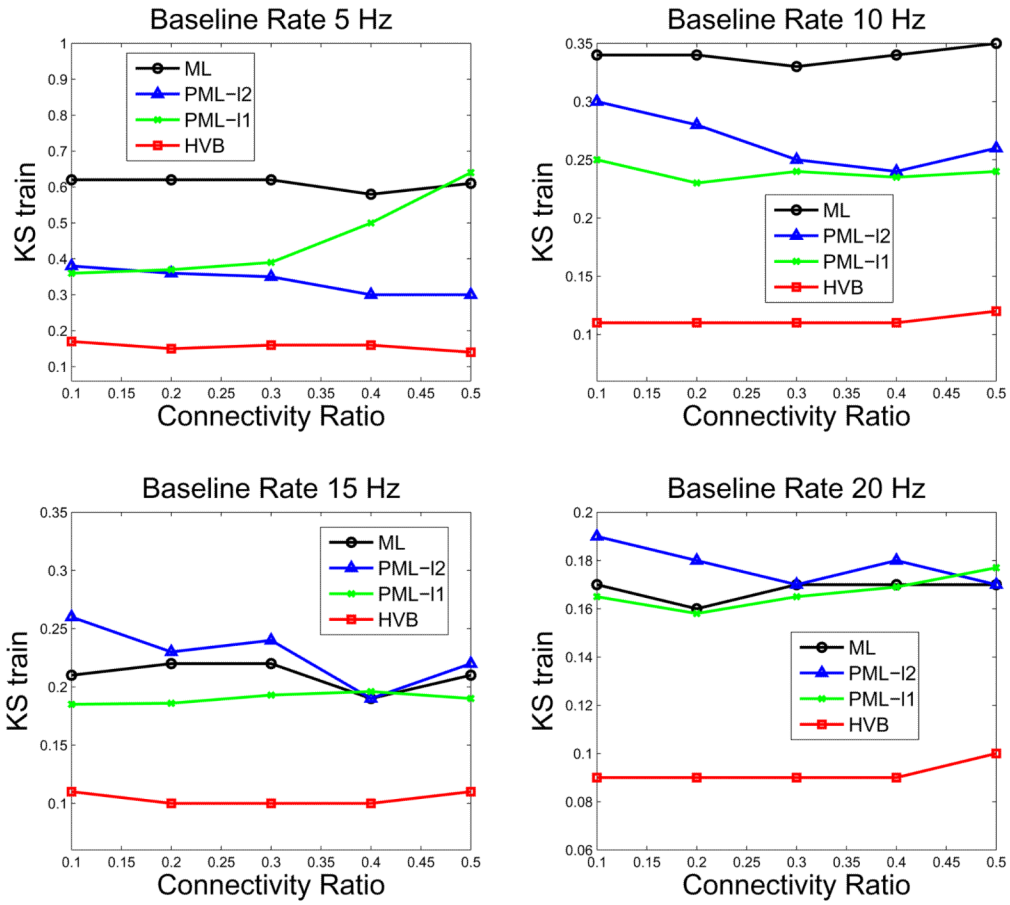

Figure 1.
Comparison of inference methods on KS statistics for the training set.

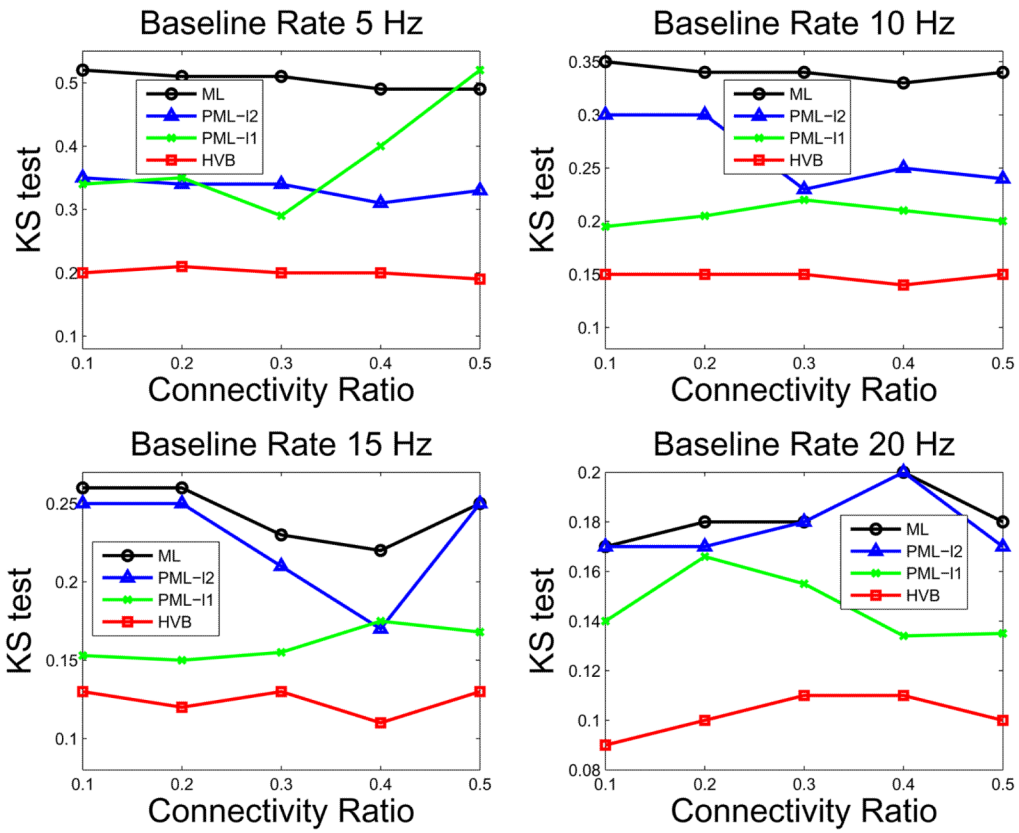

Figure 2.
Comparison of inference methods on KS statistics for the testing set.

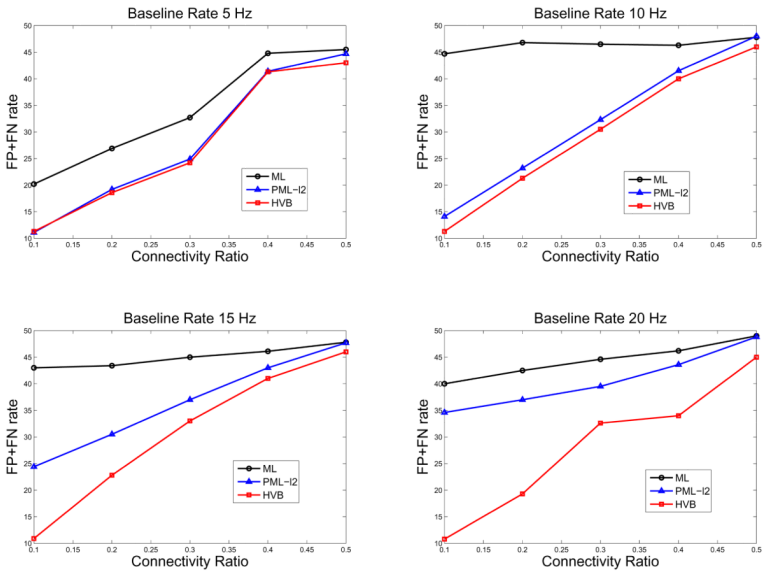

Figure 3.
Comparison of inference methods on mis-identification (FP+FN) error rate.

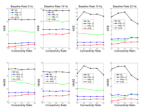

Figure 4.
Comparison of inference methods on MSE and NMSE statistics. Note that the y-axis in the first two columns are in log-scale.

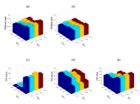

Figure 5.
Comparative performances (mean statistics averaged over 10 Monte Carlo runs) on the KS statistics (a, b), FP error (c), FN error (d), and (FP+FN) error (e) with varying values of hyperprior parameters a_0 and b_0 in the HVB algorithm.

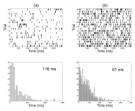

Figure 6.
Spike rasters and inter-spike interval (ISI) histograms of two representative M1 neurons (Dataset-1): (a) a RS neuron (mean firing rate: 3.7 Hz), (b) a FS neuron (mean firing rate: 13 Hz). The legends in the ISI histograms denote the median ISI values. One-sample KS test indicated that these two ISI samples are not exponential distributed ($P > 0.05$).

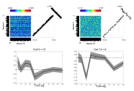

Figure 7.
Illustrations of pairwise neuronal spiking dependence between one RS neuron (#5) and one FS neuron (#10), using the raw (top left) and corrected (top right) joint peri-stimulus time histograms (JPSTHs) and the inferred bi-directional (bottom) spiking dependence based on the HVB algorithm. Here, A→B means that neuron A is a trigger cell, and neuron B is the target cell; the shaped area indicates the 95% confidence intervals of the estimates. An positive/negative value of the estimate implies an excitatory/inhibitory effect on the spiking of the target from the trigger cell. The corrected JPSTH (5 ms bin) did not detect any significant interactions between two neurons as the correlation coefficient is 0.029. In contrast, a significant excitatory/inhibitory RS→FS effect and a significant inhibitory FS→RS effect were detected by our method.

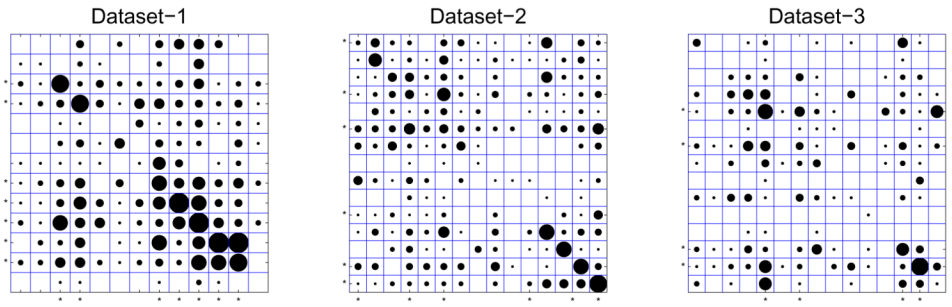

Figure 8.

Visualization of the strengths of neuronal interactions inferred from M1 neuronal assemblies during the baseline period. In each plot, the size of the circle is proportional to the relative strength (normalized such that the maximum strength is 1) respective to all neurons (including self-interactions); the symbol * along the axes in each plot marks the FS neuron.

TABLE I

Examples of exponential family in a canonical form.

prob. dist.	link func.	θ	$b(\theta)$	$\bar{b}(\theta)$	$\bar{b}'(\theta)$
Bernoulli(1, π)	logit	$\log \frac{\pi}{1-\pi}$	$\log(1+e^{\theta})$	π	$1-\pi$
Poisson(λ)	log	$\log \lambda$	$\exp(\theta)$	λ	λ

TABLE II

Comparison of statistical inference methods and algorithms.

Inference method	Algorithm	Free parameter(s)
maximum likelihood (ML)	IRWLS, CG	none
ℓ_2 penalized ML	IRWLS, CG	ρ
ℓ_1 penalized ML	interior-point method	ρ, κ
hierarchical Bayesian	variational Bayes	hyperpriors a_0, b_0

TABLE III

Performance comparison under different data size (with a fixed connectivity ratio of 0.3) in simulations. Mean statistics in the table are averaged over 10 Monte Carlo runs. Data size "1×" represents the standard setup described in the text.

Data size	**1 ×**				**2 ×**				**4 ×**			
					KStest statistics							
	5 Hz	**10 Hz**	**15 Hz**	**20 Hz**	**5 Hz**	**10 Hz**	**15 Hz**	**20 Hz**	**5 Hz**	**10 Hz**	**15 Hz**	**20 Hz**
ML	0.510	0.345	0.238	0.184	0.254	0.153	0.115	0.090	0.120	0.082	0.060	0.055
ℓ_2-PML	0.345	0.232	0.215	0.182	0.243	0.146	0.108	0.088	0.119	0.082	0.062	0.053
ℓ_1-PML	0.295	0.220	0.165	0.154	0.224	0.135	0.098	0.085	0.120	0.082	0.067	0.050
HVB	0.202	0.153	0.135	0.119	0.140	0.115	0.086	0.065	0.092	0.070	0.060	0.048
					MSE							
	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz
ML	137.2	74.1	17.1	7.4	54	10.5	5.5	4.1	20.4	5.3	3.4	2.4
ℓ_2-PML	2.3	2.4	2.7	3.0	2.2	2.5	2.6	2.6	2.4	2.3	2.2	2.2
ℓ_1-PML	18.7	11.7	6.9	2.5	15.6	7.6	4.3	3.1	12.9	5.0	3.2	2.3
HVB	1.5	1.1	1.3	1.8	1.3	1.1	1.2	1.1	1.4	1.0	1.0	0.9
					FP+FN error rate							
	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz
ML	32.7	46.5	45.0	44.6	38.9	43.1	42.4	44.0	41.7	44.5	41.0	42.5
ℓ_2-PML	24.9	32.3	37.0	39.5	32.3	35.3	38.9	45.9	31.9	41.0	41.5	45.5
HVB	24.2	30.5	33.0	32.6	32.2	29.5	27.7	33.4	31.1	36.8	28.4	32.2

TABLE IV

Summary of real spike train data from ensemble M1 neurons (during the baseline period).

Dataset	# trials	# neurons (RS+FS)	min/median firing rate (Hz)	# neurons with firing rate below 10 Hz	fraction of neuronal interactions		
					RS-RS	RS-FS	FS-FS
1	18	13 (6+7)	2.7/13.0	6	13/36	67/84	49/49
2	18	15 (9+6)	1.0/8.7	9	50/81	83/108	33/36
3	17	15 (11+4)	0.7/5.4	12	36/121	61/88	16/16