Inferring Network Structure from Observation I: Binary Neural Networks

Peiran Gao

December 11, 2011

1 Motivations

Inferring network structure from observed data is a useful procedure to study the relation between structure and function networks. For networks with observable dynamics but hidden structure, inference gives the best guess of the underlying connectivity that explains the observed data. For networks with known structure and observable dynamics, inference helps to separate parts of the network that directly contribute to the dynamics and function and those that don't. Both aspects of network inference may find their uses in neuroscience in the future: inferring connectivities of hundreds of interconnected neurons recorded concurrently by microscopic imaging, or deconstructing the functional connections in a heuristically trained artificial neural network for example.

In this study, we give theoretical answers to two important questions of network structure inference. First, using binary neural networks as an example of single-step, discrete-time network, we study and characterize the constraint on individual node's dynamics for the network inference problem to be solvable using convex optimization. Second, as $\mathcal{L}_1$ regularization is often used to solve large-scale sparse problems the case for most real networks, we derive and verify a closely form way of calculating the corresponding regularization parameter given either the prior knowledge or the estimated value for sparsity.

This study was first motivated by *On the Convexity of Latent Social Network Inference* by S.A. Myers and J. Leskovec 2010 in which they used convex optimization to infer structures of information diffusion networks. The binary neural network was modified from the network analyzed in *Real-Time Computation at the Edge of Chaos in Recurrent Neural Networks* by N. Bertschinger and T. Natschläger in 2004. Instead of individual nodes having binary values of -1 and 1, we use 0 and 1. The theoretical approach to find the optimal regularization parameter was inspired by *Bayesian Regularization and Pruning Using a Laplace Prior* by P.M. Williams in 1995, in which a heuristic choice of regularization parameter was given to cases with no prior sparsity knowledge.

2 Binary Neural Networks

In this work, we consider a discrete-time binary neural network consisting of probabilistic cells. Each network consist of N cells, whose states are denoted by $x_i, 0 \leq i < N$ that takes the value of either 0 or 1. Connectivities of such network is parameterized by a connection weight matrix W, where each entry $W_{ij} \in \mathbb{R}$ of the matrix represents the connection weights from cell i to j. At each time step t, a cell first compute it's net input $s_{i,t}$,

$$s_i(t) = \sum_j W_{ij} x_j(t) + u_i(t) \tag{1}$$

where $u_i(t)$ is an optional external input to cell i at time t. To update the cell's state at the next time step $x_i(t+1)$, the value of $s_i(t)$ is passed through an activation function $h(s_i(t))$ common for all cells in the network. The activation function's output then corresponds to the probability that at the next time step, $x_i(t+1) = 1$,

$$P(x_i(t+1) = 1) = h(s_i(t))$$

Note that $1 - P(\cdot)$ conversely the probability of $x_i(t+1) = 0$. While a binary neural network constructed in this fashion is probabilistic in nature, limiting forms of $h(\cdot)$ readily give rises to deterministic networks. For example, one may increase the slope of a soft-thresholding sigmoid activation function to approach a step function, making the network's elements equivalent to deterministic thresholding units.

The subject of this paper is to study how the structure for an instant of such neural networks can be inferred from observations of their state evolutions (why should anyone care?). While the Markovian nature of the binary neural network may suggest a transition matrix approach to the problem, the matrix's 2^{2N} elements make it unpractical. Instead, we formulate the problem of structure inference as a maximum likelihood estimation (MLE) problem.

3 Convexity of Structure Inference

Mathematically, we take a network whose structure is defined by W, and denote one trajectory of its evolution as a function $\vec{x}$, where $\vec{x}(t)$ is a binary vector describing the states for all nodes in the network at time t, $|\vec{x}(t)| = N$. Similarly, we denote the input sequence as $\vec{u}$ with $|\vec{u}(t)| = N$ as well. Note that values of $\vec{u}(t)$ are not necessarily restricted to be binary by definition. Since more than one trajectory may be available, the structure inference problems takes as input a set of trajectories and inputs of the network, $X = \{(\vec{x}, \vec{u})\}$, and outputs the best estimate, $\hat{W}$, of W.

To solve the problem, we formulate it as a maximum likelihood estimation problem,

$$\hat{W} = \arg\max_W L(W; X) = \arg\max_W \prod_{\vec{x} \in X} \prod_t \prod_i P(x_i(t+1) | \vec{x}(t), W) \tag{2}$$

where $P(x_i(t+1)|\vec{x}(t), W)$ is the probability of observing $x_i(t+1)$ given the state of the network in the previous time step,

$$P(x_i(t+1)|\vec{x}(t), W) = x_i(t+1)h(s_i(t)) + (1 - x_i(t+1))(1 - h(s_i(t))) \tag{3}$$

Furthermore, since the rows of W are do not interact in computing the transition probability, the overall likelihood is maximized if the likelihood for each row is maximized,

$$\hat{W}_{i*} = \arg\max_{W_{i*}} L_i(W_{i*}; X) = \arg\max_{W_{i*}} \prod_{\vec{x} \in X} \prod_t P(x_{i,t+1} | \vec{x}_t, W_{i*})$$

Or, equivalently, to avoid computing products of small probabilities, we maximize the likelihood by minimizing the negative log likelihood for each row, L_i,

$$\hat{W}_{i*} = \arg\min_{W_{i*}} [-\log L_i(W_{i*}; X)] = \arg\min_{W_{i*}} - \sum_{\vec{x} \in X} \sum_t \log P(x_{i,t+1} | \vec{x}_t, W_{i*}) \tag{4}$$

3.1 Convexity Constraint on the Activation Function

The maximum likelihood estimation of $\hat{W}_{i*}$ is in general difficult to solve. However, if the log likelihood L_i over the W space is convex everywhere, the inference problem can be solved efficiently using convex optimization techniques. We therefore proceed to derive how the convexity property of the likelihood function constrains the construction of binary neural networks.

Beginning with the derivation, we first introduce a few shorthands: we use p as shorthand for $P(x_i(t+1)|\vec{x}(t), W_{i*})$, s as shorthand for $s_i(t)$, and h as shorthand for $h(s_i(t))$. Furthermore, we abbreviate the first and second derivatives of h with respect to $s_i(t)$ as h' and h''. We take the second derivative of the log likelihood with respective two W_{i*},

$$-\frac{\partial^2 \log L_i(W_{i*}; X)}{\partial W_{ij_1} \partial W_{ij_2}} = \sum_{\vec{x} \in X} \sum_t \frac{1}{p^2} \left(\frac{\partial p}{\partial W_{ij_1}} \frac{\partial p}{\partial W_{ij_2}} - \frac{p \partial^2 p}{\partial W_{ij_1} W_{ij_2}} \right)$$

Convexity requires the second derivative to be non-negative for any choice of indices $j_{1,2}$. Furthermore, since convexity shouldn't depend on the specific trajectory $\vec{x}$ or time t, convexity of the log likelihood translates to convexity of what is inside the summation operator. As $1/p^2 \geq 0$, we enforce non-negativity inside the parentheses,

$$\frac{\partial p}{\partial W_{ij_1}} \frac{\partial p}{\partial W_{ij_2}} \geq \frac{p \partial^2 p}{\partial W_{ij_1} W_{ij_2}}$$

Expanding the derivatives, we obtain,

$$\begin{aligned}
\frac{\partial p}{\partial W_{ij}} &= (2x_i(t+1)-1)h'\frac{\partial s}{\partial W_{ij}} \\
\frac{\partial^2 p}{\partial W_{ij_1}\partial W_{ij_2}} &= (2x_i(t+1)-1)h''\frac{\partial^2 s}{\partial W_{ij_1}\partial W_{ij_2}} \\
\frac{\partial p}{\partial W_{ij_1}}\frac{\partial p}{\partial W_{ij_2}} &= (2x_i(t+1)-1)^2 h'^2 x_{j_1}(t)x_{j_2}(t) \\
\frac{\partial^2 p}{\partial W_{ij_1}W_{ij_2}}p &= (2x_i(t+1)-1)^2 hh'' x_{j_1}(t)x_{j_2}(t) + \\
&\quad (2x_i(t+1)-1)(1-x_i(t+1))h'' x_{j_1}(t)x_{j_2}(t)
\end{aligned}$$

Note that we only have to check the case in which $x_{j_1}(t) = x_{j_2}(t) = 1$ as the other three cases result in $x_{j_1}(t)x_{j_2}(t) = 0$. Furthermore, we know that $(2x_i(t+1)-1)^2 \equiv 1$ and $(2x_i(t+1)-1)(1-x_i(t+1)) = 0$ or -1. Then for convexity of the maximum likelihood estimation, we have only a simple constraint on the activation function,

$$\frac{1}{h-1} \leq \frac{h''}{h'^2} \leq \frac{1}{h} \tag{5}$$

or, alternatively,

$$h'^2 - \max(h''h, h''h - h'') \geq 0$$

A perhaps more intriguing interpretation of the result is with respect to carrying out computations using cellular automata: *for a given set of trajectories X that phenomenologically describe some computation, as long as the activation function $h(s_{i,t})$ complies with the convexity constraint, 1. there exists a unique globally optimal network structure that best implements the computation; 2. this optimal network structure can be found efficiently using convex optimization.*

4 Space of Convex Activation Functions

The function space of activation functions that satisfy the convexity constraint are solutions to the ordinary differential equation

$$h'' = f(h)h'^2$$

where $f(h)$ is defined on the interval $[0, 1]$ and obeys,

$$\frac{1}{h-1} \leq f(h) \leq \frac{1}{h}$$

In this section, we first show that some widely used activations functions satisfy the convexity constraint. We then proceed to prove a set of invariance transformations that generalize individual examples to function spaces.

4.1 Example Activation Functions

We start from the anti-symmetric sigmoidal activation function often used in Boltzmann machines:

$$h(s) = \frac{1}{1+e^{-s}} \tag{6}$$

It's not hard to show that,

$$\begin{aligned}
h'^2 - hh'' &= \frac{e^{-s}}{(1+e^{-s})^4} \geq 0 \\
h'^2 - hh'' + h'' &= \frac{e^{-3s}}{(1+e^{-s})^4} \geq 0
\end{aligned}$$

A similar anti-symmetric function activation function

$$h(s) = \frac{\tanh s + 1}{2} \tag{7}$$

also satisfies the constraint, because

$$\begin{aligned} h'^2 - hh'' &= \frac{1}{2}(\tanh s + 1)^2(1 - \tanh s) \geq 0 \\ h'^2 - hh'' + h'' &= \frac{1}{2}(1 - \tanh s)^2(\tanh s + 1) \geq 0 \end{aligned}$$

Next, instead of giving one example activation function, we consider a class of symmetric activation function taking on the form,

$$h(s) = e^{-s^{2k}}, k \geq 1 \tag{8}$$

When $k = 1$, the activation function is Gaussian shaped. As k increases, $h(s)$ morphs into a uniform square between $-1 \leq s \leq 1$. We check the convexity constraint,

$$\begin{aligned} h'^2 - hh'' &= 2k(2k-1)s^{2k-2}e^{-2s^{2k}} \geq 0 \\ h'^2 - hh'' + h'' &= 2k(2k-1)s^{2k-2}e^{-s^{2k}}(e^{-s^{2k}} - 1) + 4k^2 s^{4k-2}e^{-s^{2k}} \geq 0 \end{aligned}$$

To check the second inequality expression, we note that the expression is convex with both its value and its first derivative equal to 0 at $s = 0$.

4.2 Invariance to Linear Transformation of Input

Let $s_T = T(s)$ where T is a linear transformation of s. Note that $T''(s) = 0$. Let $\dot{h}s_T$ be the derivative of $h(s_T)$ with respect to s,

$$\begin{aligned} \dot{h}(s_T)^2 &= (T'(s))^2 h'(s_T)^2 \\ &\geq (T'(s))^2 \max\left[h(s_T)h''(s_T), (h(s_T) - 1)h''(s_T)\right] \\ &= \max\left[h(s_T)\ddot{h}(s_T), (h(s_T) - 1)\ddot{h}(s_T)\right] \end{aligned}$$

4.3 Invariance to Exponentiation

Let $g(s) = h(s)^k$ where $k \geq 1$ (works with $0 < k < 1$ for gaussian and sigmoid, but I have no proof). We will show that

$$\begin{aligned} & h'^2 \geq \max(hh'', hh'' - h'') \\ \Rightarrow \quad & g'^2 \geq \max(gg'', gg'' - g'') \end{aligned}$$

To start, we first write down the derivatives of g

$$\begin{aligned} g' &= kh^{k-1}h' \\ g'' &= k(k-1)h^{k-2}h'^2 + kh^{k-1}h'' \end{aligned}$$

For the first entry in the $\max$ operator, we work our way backwards by first substituting all gs with hs

$$\begin{aligned} g'^2 &\geq gg'' \\ \Leftrightarrow k^2 h^{2k-2}h'^2 &\geq k(k-1)h^{2k-2}h'^2 + kh^{2k-1}h'' \end{aligned}$$

Dividing both sides by kh^{2k-2} and regrouping the terms, we have

$$h'^2 \geq hh''$$

which is indeed the case.

For the second entry in the $\max$ operator, we similarly carry out the substitution and scaling,

$$\begin{aligned} g'^2 &\geq gg'' - g'' \\ \Leftrightarrow (h^k + k - 1)h'^2 &\geq (h^k - 1)hh'' \end{aligned}$$

To prove the last inequality, we equivalently have to show that

$$\mathcal{H}(k) = (h^k - 1)(h'^2 - hh'') + kh'^2 \geq 0$$

To show the above, we start from $\mathcal{H}(0) = 0$ and $\mathcal{H}(1) \geq 0$. Then, we take the second derivative of $\mathcal{H}$ with respect to k,

$$\frac{\partial^2 \mathcal{H}(k)}{\partial k^2} = \log(k)^2 h^k (h'^2 - hh'') \geq 0$$

and find that $\mathcal{H}(k)$ is convex, implying that $\mathcal{H}(k) \geq 0$ for $k \geq 1$.

5 Structure Inference Accuracy

5.1 Estimation Error and the Cramér-Rao Inequality

The Cramér-Rao inequality bounds the mean-squared error of any MLE of by the reciprocal of the Fisher information. When subsequent observations of the system are independent, the inequality has the matrix inequality,

$$\Sigma \geq \frac{1}{|X|} I^{-1} \tag{9}$$

where Σ is the covariance matrix of the estimation error $\hat{W}_{i*} - W_{i*}$, $|X|$ is the total number of independent observations and I is the Fisher information matrix. The inequality is defined in the positive semi-definite sense. A MLE is called "efficient" if the above expression becomes an equality. We note that the covariance roughly scales inversely with the amount of independent observations available.

To come up with bounds on our inference problem, we derive the element-wise expression for the Fisher information matrix,

$$\begin{aligned} I_{j_1 j_2} &= \mathbf{E}\left[\frac{1}{p^2}\frac{\partial p}{\partial W_{ij_1}}\frac{\partial p}{\partial W_{ij_2}}\right] \\ &= \mathbf{E}\left[\frac{h'^2}{h - h^2} x_{j_1} x_{j_2}\right] \\ \lim_{N\to\infty} I_{j_1 j_2} &= \mathbf{E}\left[\frac{h'^2}{h - h^2}\right] \mathbf{E}\left[x_{j_1} x_{j_2}\right] \end{aligned} \tag{10}$$

Note that we are still using the shorthand h for $h(s_i)$. We calculate the value of $I_{j_1 j_2}$ assuming the system's size N is very large. We then show actual data that the results apply well to small Ns. With the large N assumption, s_i and the product $x_{j_1} x_{j_2}$ become independent; and the expectation becomes a product of two expectations on s_i and $x_{j_1} x_{j_2}$, respectively.

The marginal distributions for s_i and $x_{j_1} x_{j_2}$ depend on the actual dynamics of the network. To keep the problem tractable this section, we shall make two assumptions to simplify the problem and leave the dynamics' impact on inference for later considerations. First, we make the thermodynamic assumption that all possible states of the network are equally likely to be visited. Second, we assume that entries of W_{i*} are distributed with mean 0 and standard

deviation σ_W. Note that we don't have to assume independence among the entries as MLE assumes independent uniform distribution of the quantity being inferred. The 0 mean assumption isn't absolutely necessary as any bias in the weight distribution can be effectively cancelled out by a bias in the input $\vec{u}$.

The thermodynamic assumption allows us to easily compute the second expectation of $x_{j_1}x_{j_2}$ to be $1/2$ when $j_1 = j_2$ and $1/4$ otherwise. Since $h(\cdot)$ is a function of s_i, the first expression is a measure over s_i's marginal distribution. With the assumption that all states are equally likely, half of the x_js are 1 with the other half 0. Then s_i is simply the sum of $N/2$ independent drawing from a distribution with mean 0 and standard deviation σ_W. The law of large numbers then implies that $s_i \sim \mathcal{N}(0, N\sigma_W^2/2)$. Simple integration yields

$$\begin{aligned}
\mathbf{E}\left[\frac{h'^2}{h-h^2}\right] &= \int_{-\infty}^{\infty} ds \frac{h'^2}{h-h^2} \frac{1}{\sqrt{\pi N}\sigma_W} e^{-s^2/(N\sigma_W^2)} \\
I_{\text{diag}} &= \frac{1}{2}\mathbf{E}\left[\frac{h'^2}{h-h^2}\right] \\
I_{\text{off}} &= \frac{1}{4}\mathbf{E}\left[\frac{h'^2}{h-h^2}\right]
\end{aligned} \tag{11}$$

5.2 Inference Accuracy on Random Networks

In this section, we compare the performance of the proposed MLE against the theoretical limit predicted by the Cramér-Rao inequality on networks constructed using random weight matrices with i.i.d. entries with sigmoidal activation function (Eq 6). Note that with the sigmoid activation function, the inference problem becomes equivalent to logistic regression with zero bias. We quantify quality of the inference by normalizing the values of the inference error covariance matrix Σ relative to the weight matrices' variance σ_W^2. Inverting the Fisher information matrix (Eq 11) and numerically integrating the expectation over s (Eq 11), the Cramér-Rao bound of an efficient MLE becomes,

$$\begin{aligned}
\frac{\Sigma_{\text{diag}}}{\sigma_W^2} &= \frac{4N}{\sigma_W^2|X|(N+1)\mathbf{E}\left[\frac{h'^2}{h-h^2}\right]} \approx \frac{17.8}{\sigma_W^2|X|} \\
\frac{\Sigma_{\text{off}}}{\sigma_W^2} &= \frac{-4}{\sigma_W^2|X|(N+1)\mathbf{E}\left[\frac{h'^2}{h-h^2}\right]} = \frac{-17.8}{\sigma_W^2|X|(N+1)}
\end{aligned} \tag{12}$$

The first approximation sign comes from $N/(N+1) \approx 1$. The correlation terms in the covariance matrix vanishes as N gets large.

To keep the spectra of W constant with different sizes, we scale σ_W by $1/\sqrt{N}$ such that W_{ij} drawn i.i.d. from $\mathcal{N}(0, 1/N)$. (have to prove the underlying network is ergodic?) In this case, the relative inference error scales linearly with the ratio $N/|X|$. Using observations from random network simulation of sizes 20 and 50 with varying durations, we use the CVXOPT software package to solve the convex MLE problem in parallel on a computer with Intel Core i7 CPU at 4GHz with 6 cores. The MLE algorithm performs as expected, giving solutions that are close to the ground truth with relative small residuals (Fig 1).

We then compute the normalized diagonal entires of Σ, equivalent to the normalized the mean squared error (MSE), for the two sizes at different values of $|X|$, and compare them against the Cramér-Rao bound (Fig 2). The convex optimization algorithm solves the structure inference problem efficiently with accuracies that tightly follows the optimal bound given by the Cramér-Rao inequality. For MSE within 10% of σ_W^2, one needs approximate 3000 and 8000 observations of the system's states for network sizes of 20 and 50 respectively. Furthermore, the convex optimization algorithm showed a runtime that scales linearly with the amount of data (Fig 2 insets).

6 Sparse Connectivity and the $\mathcal{L}_1$ Regularization

Real world networks are often sparse. We call a connectivity matrix k-sparse, $0 \leq k \leq 1$, if only kN^2 of entries in W are populated. If one has a prior knowledge or estimation of what the k is for the network to be inferred, the inference procedure can perceivably perform better than a fully connected network given the same amount of observations. The

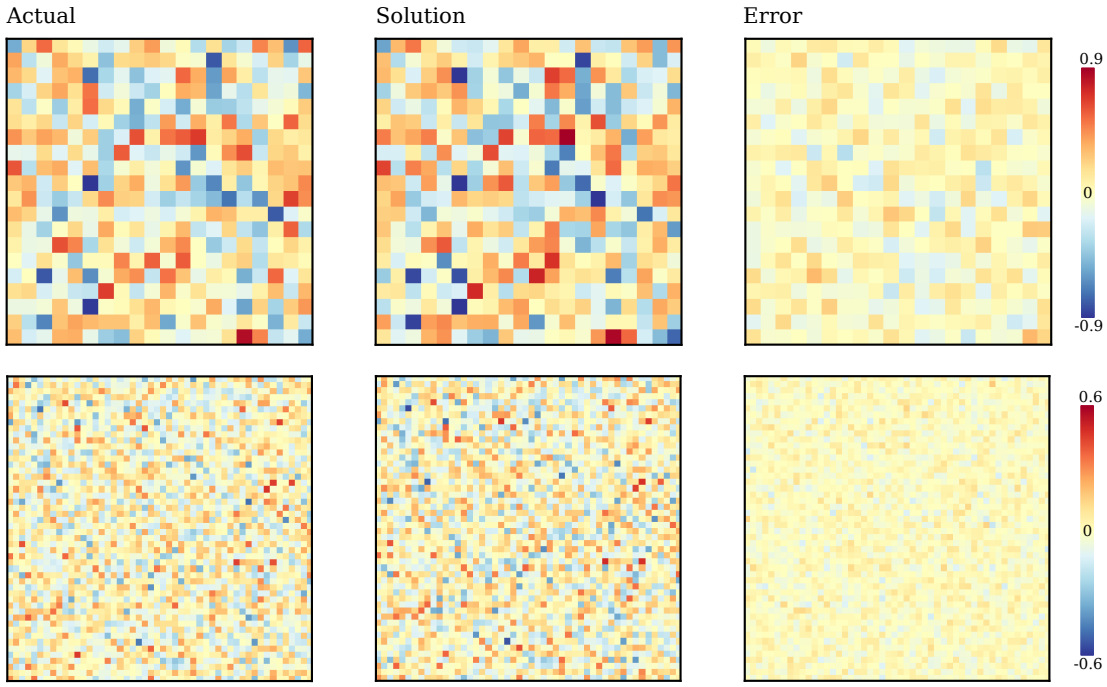

Figure 1: Example inference results at network sizes of 20 (top) and 50 (bottom). The numbers of observation where 3,000 and 8,000 for each case respectively. From left to right, the three columns of images are the ground truth, inference results and inference error of the connectivity matrix.

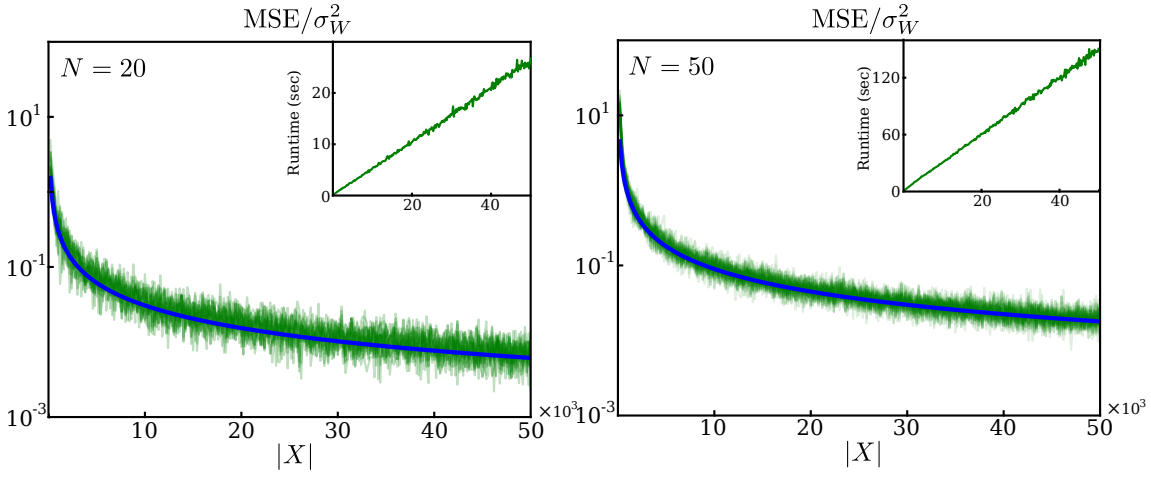

Figure 2: Inference MSE normalized by σ_W^2 with different amount of observations with network sizes of 20 and 50. The theoretical Cramér-Rao bounds are plotted in blue. The actual values, green lines, are calculated for each element of the W matrix, and tightly follow the theoretical lower bound. Insets show the linear scaling of inference runtime with respect to the number of observations $|X|$.

intuition is information theoretical in the sense that the saving in the amount of observations needed to infer kN^2 instead of N^2 entries exceeds the extra amount of observations needed to determine which kN^2 entries are populated.

To add in a sparsity constraint while keeping the objective function (Eq 4) convex, the favored approach has been to add an $\mathcal{L}_1$ norm regularization term to the objective function

$$\hat{W}_{i*} = \arg\min_{W_{i*}} \left[-\log L_i(W_{i*}; X) + \lambda ||W_{i*}||_1 \right]$$

where larger regularization parameter λ constrains the optimization to give more sparse solutions.

6.1 Eliminating the Regularization Parameter with Prior Knowledge of k

In the absence of prior knowledge of k, the standard practice of finding the best λ has been cross-validation. Observations are divided to training and testing sets. The optimal value of λ is the one that maximize the likelihood when trained on the training set and tested on the testing set. For network structure inference, however, this may not be necessary, as estimating the value of k is a convenient byproduct to sampling a network's degree distribution–a much easier task. We then set out to find the relation between the optimal λ given k.

Since the objective function is convex, we first linearize it into a quadratic form around the unregularized solution $\hat{W}_{i*}$ to obtain

$$\frac{1}{2}(W_{i*} - \hat{W}_{i*})^T \Sigma^{-1} (W_{i*} - \hat{W}_{i*}) + \lambda ||W_{i*}||_1$$

where Σ is the error covariance matrix (Eq 12). Since as N gets large, Σ becomes diagonal. We can write the above expression for each element W_{ij}

$$\frac{1}{2\Sigma_{\text{diag}}}(W_{ij} - \hat{W}_{ij})^2 + \lambda |W_{ij}|$$

Given the unregularized solution $\hat{W}_{ij}$, the new objective function minimizes the above equation and computes the regularized solution $\hat{W}^\dagger_{ij}$ according to,

$$\hat{W}^\dagger_{ij} = \begin{cases} \hat{W}_{ij} - \lambda \Sigma_{j_1=j_2} & (\hat{W}_{ij} \geq \lambda \Sigma_{j_1=j_2}) \\ \hat{W}_{ij} + \lambda \Sigma_{j_1=j_2} & (\hat{W}_{ij} \leq -\lambda \Sigma_{j_1=j_2}) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The optimal choice of λ is the one that minimizes the mean squared error ($\text{MSE}_{\mathcal{L}_1}$) between the regularized solution and the real answer conditioned on k (alternatively, can one optimize for the expected sparsity?),

$$\lambda_{\text{opt}} = \arg\min_{\lambda} \text{MSE}_{\mathcal{L}_1} = \arg\min_{\lambda} \frac{1}{N} \sum_j \mathbf{E}\left[(\hat{W}^\dagger_{ij} - W_{ij})^2 | k \right] \tag{14}$$

In Appendix A, by assuming a Gaussian prior for the non-zero elements, we derive the analytical expression for the above expectation by separating it into a "Benefit" function, B, that represents the improved estimation of W_{i*}'s zero elements, and a "Cost" function, C, that represents the degraded estimation of non-zero elements due to regularization. We introduce two parameter substitutions to keep the expression concise

$$\alpha = \lambda \sqrt{\Sigma_{\text{diag}}/2},\ \gamma = \sqrt{1 + \sigma_W^2 / \Sigma_{\text{diag}}}$$

where σ_W is the standard deviation of the non-zero elements of W which can be estimated from the unregularized solution.We normalize $\text{MSE}_{\mathcal{L}_1}$ by MSE for the unregularized solution,

$$\begin{aligned} \frac{\text{MSE}_{\mathcal{L}_1}}{\text{MSE}} &= \frac{\sum_j \mathbf{E}\left[(\hat{W}^\dagger_{ij} - W_{ij})^2 | k \right]}{N \Sigma_{\text{diag}}} = (1-k)\text{B}(\alpha) + k\text{C}(\alpha, \gamma) \\ \text{B}(\alpha) &= (1 + 2\alpha^2)\text{Erfc}(\alpha) - \frac{2}{\sqrt{\pi}} \alpha e^{-\alpha^2} \\ \text{C}(\alpha, \gamma) &= (1 + 2\alpha^2)\text{Erfc}\left(\alpha/\gamma\right) + (\gamma^2 - 1)\text{Erf}\left(\alpha/\gamma\right) - \frac{2}{\sqrt{\pi}} \alpha\gamma e^{-\alpha^2/\gamma^2} \end{aligned} \tag{15}$$

The optimal λ can now be found by setting the derivative of the expected error to zero and solving for α,

$$(1-k)\left(\sqrt{\pi}\alpha_{\text{opt}}\text{Erfc}(\alpha_{\text{opt}}) - e^{-\alpha_{\text{opt}}^2}\right) + k\left(\sqrt{\pi}\alpha_{\text{opt}}\text{Erfc}(\frac{\alpha_{\text{opt}}}{\gamma}) - \frac{1}{\gamma}e^{-\alpha_{\text{opt}}^2/\gamma^2}\right) = 0 \tag{16}$$

6.2 Experimental Verification

To verify that the above calculation indeed yields λ_{opt}, we simulated random networks with $N = 20, k = 0.2$ to obtain between 2000 and 10000 observations. The non-zero element of W are distributed according to $\mathcal{N}(0, 1/kN)$ to again keep the same eigenvalue spectrum. For each $|X|$, we solve for λ_{opt} numerically (Eq 16). We then run the optimization algorithm with λ between 0 and $2\lambda_{\text{opt}}$, and compute the fraction of the mean squared error of the regularized solution to that of the unregularized solution (Eq 15). This gives the ground truth of the optimal λ which we compare against theoretical predictions.

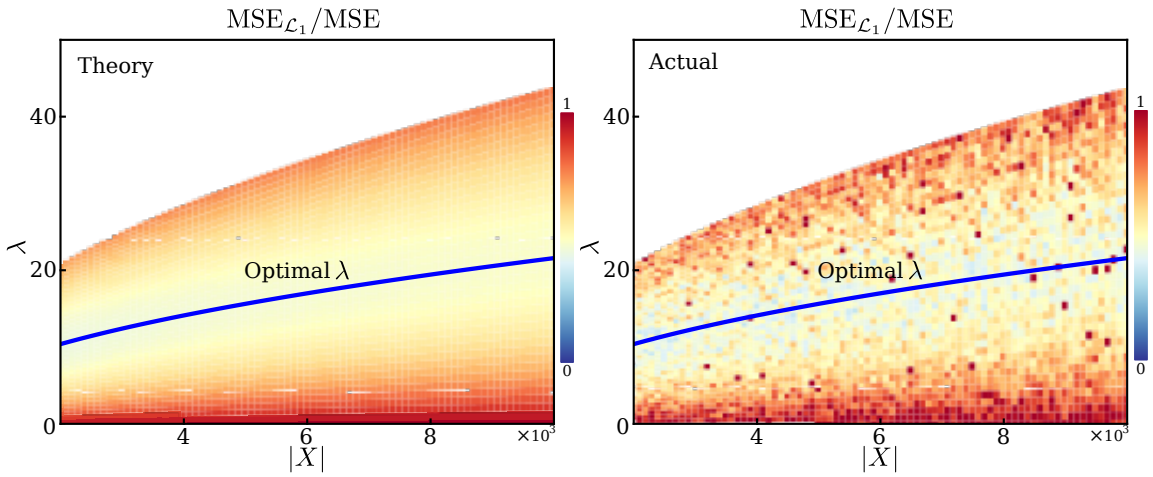

Figure 3: Optimal λ for the $\mathcal{L}_1$ regularization. Networks with $N = 20, k = 0.2$ were simulated to give observation quantities $|X|$ between 2,000 and 10,000. For each value of $|X|$, structural inference was carried out with λ between 0 and $2\lambda_{\text{opt}}$. The resulting MSE$_{\mathcal{L}_1}$, theory (left) and data (right), were then compared against the unregularized MSE ($\lambda = 0$). The calculated λ_{opt} (blue lines) does correspond to the valley of minimal ratio, or maximal $\mathcal{L}_1$ improvement.

A total of 3,200 data points were collected using PiCloud's computing service with Amazon's c1 CPUs over 600 CPU hours. The actual ratios between regularized mean squared error to the unregularized matches theoretical prediction well (Fig 3 pixel values left and right panels). However, actual data's values are noisier than theoretical calculation, which may be attributed to the small network size of 20. More importantly, the theoretically predicted λ_{opt} (Fig 3 blue line) follows the valley of the minimum ratio, or best $\mathcal{L}_1$ performance, for different values of $|X|$, verifying the correctness of our calculations.

6.3 Limiting Behaviors of λ_{opt}

While solving for α_{opt} is a numerical task, its limiting cases, however, are informative and analytically tractable. We first consider the condition under which $\alpha_{\text{opt}} = 0$. This corresponds to the boundary in the k, γ plane beyond which regularization is unnecessary. The boundary has the form

$$k - 1 - \frac{k}{\gamma} = 0$$

which is only satisfied when both $k = 1$ and $\gamma \to \infty$. Therefore, even in the fully connected case, unless infinite amount of observation is available, regularization still helps.

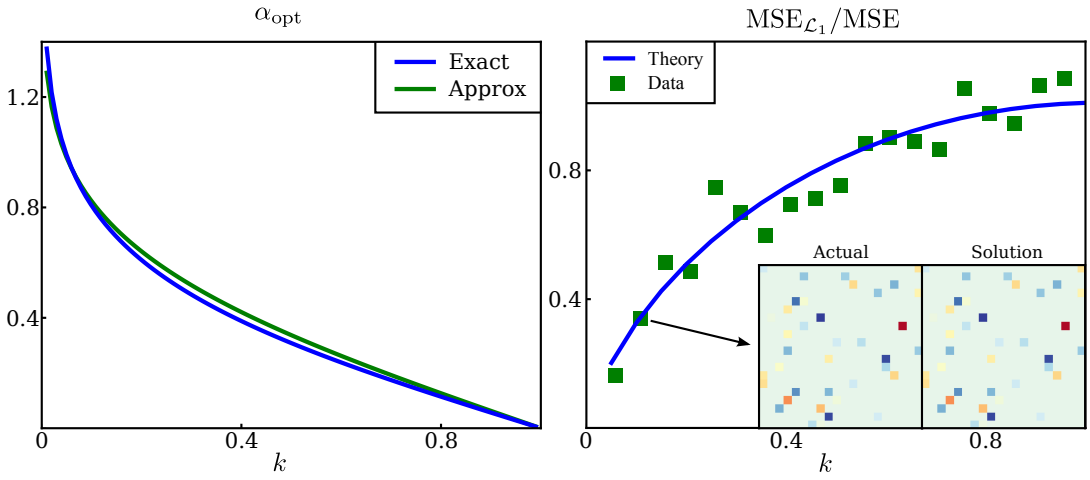

Figure 4: $\mathcal{L}_1$ regularization in the ample data limit ($\gamma \gg 1$). Left panel shows the exact (blue) and inverse error function approximated (green) optimal value of α as a function of sparsity k. Right panel shows the calculated (blue) and actual (green) fraction of regularized solution's MSE to that of the unregularized solution at various k. The actual data was obtained with $N = 20, |X| = 20,000$, thus $\gamma > 56$. Inset shows example solution at $k = 0.1$.

Another interesting limiting case is when $\gamma \gg 1$, or $\Sigma_{\text{diag}} \ll 1$, or when ample amount of observations are available and the unregularized optimization is performing well. Under such condition, the optimal α is only a function of k and the solution to

$$(1-k)\left(\sqrt{\pi}\alpha_{\text{opt}}\text{Erfc}(\alpha_{\text{opt}}) - e^{-\alpha_{\text{opt}}^2}\right) + k\sqrt{\pi}\alpha_{\text{opt}} = 0$$

which is reasonably approximated using the inverse error function as

$$\alpha_{\text{opt}}(k) \approx \frac{\text{InvErf}(1-k)}{\sqrt{2}},\ \lambda_{\text{opt}} \approx \frac{\text{InvErf}(1-k)}{\sqrt{\Sigma_{\text{diag}}}} \tag{17}$$

which does not depend on σ_W.

The value of α_{opt} tends to infinity as k tends to zero, and approximately follows the linear relation $\sqrt{\pi/8}(1-k)$ as k tends to 1 (Fig 4 left panel). We then compare the calculated performance under such α_{opt} against actual values obtained from data (Fig 4 right panel) in the form of MSE ratios, which again showed excellent agreement. As an example, regularized solution at $k = 0.1$ shows perfect recovery of zero elements (Fig 4 inset).

7 Discussion

While we solved the convexity constraint and the optimal regularization parameter problem in this study, more could be desired before this approach is applied to real data. First, we have to study the case in which the observed data is noise, incomplete or corrupt. This is especially important if one desires to infer connectivities in the brain as all existing methods of measuring neuron activities involve dramatic subsampling. Second, we have to understand how network's dynamics introduce biases into the observation of its states, this is especially important if the network is doing something useful as its state trajectory may not be ergotic. Related, when the network is only working with a reduced state space, there may be feedback mechanisms to send input back into the network to make its states more random, thus improving inference accuracy. Last but not least, the method have to be extended to real-time spiking neurons. Given more computation power, it would be fun to try all the results on much larger networks.

A Cost and Benefit of $\mathcal{L}_1$ Regularization

In this section, we compute the expected inference error with $\mathcal{L}_1$ regularization (Eq 14) as a function of the sparsity parameter k and regularization parameter λ. First, we know that each element of the unregularized solution $\hat{W}_{ij}$ is normally distributed around the true solution W_{ij} with variance Σ_{diag} (Eq 12), or

$$P(\hat{W}_{ij}|W_{ij}) = \frac{1}{\sqrt{2\pi\Sigma_{\text{diag}}}} \exp\left(-\frac{(\hat{W}_{ij} - W_{ij})^2}{2\Sigma_{\text{diag}}}\right)$$

Next, we note that there are kN elements of the true W_{i*} that are nonzero and $(1-k)N$ elements have true values at exactly zero. Starting with simpler calculation for the zero elements, we first make a parameter substitutions for convenience: $\alpha = \lambda\sqrt{\Sigma_{\text{diag}}/2}, \beta = \sqrt{\Sigma_{\text{diag}}}$. The expected error for the zero elements, averaged and normalized by Σ_{diag}, is,

$$\begin{aligned}
\frac{\sum_{\{j:W_{ij}=0\}} \mathbf{E}\left[(\hat{W}^{\dagger}_{ij} - W_{ij})^2|k\right]}{(1-k)N\beta^2} &= \frac{1}{\beta^2}\int_{\sqrt{2}\alpha\beta}^{\infty} d\hat{W}_{ij}\ (\hat{W}_{ij} - \sqrt{2}\alpha\beta)^2 P(\hat{W}_{ij}|W_{ij}=0) \\
&\quad + \frac{1}{\beta^2}\int_{-\infty}^{-\sqrt{2}\alpha\beta} d\hat{W}_{ij}\ (\hat{W}_{ij} + \sqrt{2}\alpha\beta)^2 P(\hat{W}_{ij}|W_{ij}=0) \\
&= (1+2\alpha^2)\text{Erfc}(\alpha) - \frac{2}{\sqrt{\pi}}\alpha e^{-\alpha^2} \\
&= \text{B}(\alpha)
\end{aligned}$$

Intuitively, this term represents the improved performance in estimating the zero elements of W_{i*} with regularization, or the benefit function $\text{B}(\alpha)$. It's not hard to see that if one lets λ, or α, to go to infinity, there will be no error in the estimation of the zero elements.

To calculate the expected error for the non-zero elements of W_{ij}, we have to assume a prior, which for the sake of calculation we treat as normally distributed $P(W_{ij}|W_{ij} \neq 0) \sim \mathcal{N}(0, \sigma_W^2)$. Introducing another substitution for convenience $\gamma = \sqrt{1 + \sigma_W^2/\Sigma_{\text{diag}}}$, we expand the expectation using the expression for the regularized solution (Eq 13),

$$\begin{aligned}
\frac{\sum_{\{j:W_{ij}\neq 0\}} \mathbf{E}\left[(\hat{W}^{\dagger}_{ij} - W_{ij})^2|k, W_{ij}\right]}{kN\beta^2} &= \frac{1}{\beta^2}\int_{-\infty}^{\infty} dW_{ij} P(W_{ij}|W_{ij}\neq 0) \\
&\quad \times \left(\int_{-\sqrt{2}\alpha\beta}^{\sqrt{2}\alpha\beta} d\hat{W}_{ij}\ W_{ij}^2 P(\hat{W}_{ij}|W_{ij})\right. \\
&\quad + \int_{\sqrt{2}\alpha\beta}^{\infty} d\hat{W}_{ij}\ (\hat{W}_{ij} - \sqrt{2}\alpha\beta - W_{ij})^2 P(\hat{W}_{ij}|W_{ij}) \\
&\quad \left. + \int_{-\infty}^{-\sqrt{2}\alpha\beta} d\hat{W}_{ij}\ (\hat{W}_{ij} + \sqrt{2}\alpha\beta - W_{ij})^2 P(\hat{W}_{ij}|W_{ij})\right) \\
&= (1+2\alpha^2)\text{Erfc}\left(\alpha/\gamma\right) + (\gamma^2-1)\text{Erf}\left(\alpha/\gamma\right) - \frac{2}{\sqrt{\pi}}\alpha\gamma e^{-\alpha^2/\gamma^2} \\
&= \text{C}(\alpha,\gamma)
\end{aligned}$$

Intuitively, this term represents the additional error introduced to estimation of W_{i*}'s the non-zero elements by regularization, or the cost function $\text{C}(\alpha,\gamma)$.