CONvenient Interface to Inverse Ising (CONIII): A Python package for solving maximum entropy models

[1]Edward D Lee, [2]Bryan C Daniels
[1]*Department of Physics, 132A Clark Hall, Cornell University, Ithaca NY 14850,*
[2]*ASU–SFI Center for Biosocial Complex Systems, Arizona State University, Tempe, AZ 85287*

CONIII is an open-source Python project for providing a simple interface to solving maximum entropy models with a focus on the Ising model. We describe the maximum entropy problem and give an overview of the algorithms that are implemented as part of CONIII (`https://github.com/bcdaniels/coniii`) including a regularized mean field method, Monte Carlo histogram, pseudolikelihood, and minimum probability flow. We briefly explain how one should approach and validate maxent models from the perspective of model selection. Our goal is to make a variety of maximum entropy techniques accessible to those unfamiliar with the techniques and accelerate workflow for users.

INTRODUCTION

Many biological and social systems are characterized by collective behavior whether it is the correlated pattern of neuron firing [1], protein diversity in the immune system, conflict participation in monkeys, flocking in birds [2], statistics of letters in words, or consensus voting in the US Supreme Court [3]. Statistical physics is a natural approach to probing such systems precisely because they are collective [4]. Recently, with the advent of numerical, analytic, and computation tools, it has become possible to solve for the statistical physics model that corresponds to a particular system, an approach known as the inverse problem. This is in contrast with the typical problem in statistical physics where one postulates the Hamiltonian and works out the physical behavior of the system. In the *inverse* problem, we find the parameters that correspond to observed behavior of a known system. In many cases, this is a very difficult problem to solve and does not have an analytical solution, and we must rely on analytic approximation and numerical techniques to estimate the parameters.

The Ising model has been of particular interest because of its simplicity and generality. A variety of algorithms have been proposed to solve the inverse Ising problem, but different approaches are disparately available on separate code bases in different coding languages, which makes comparison difficult and pedagogy more complicated. CONIII (Convenient Interface to Inverse Ising) is a Python project intended to provide a centralized resource for the inverse Ising problem and provide a base for the addition of more maximum entropy problems in the future. With CONIII, it is possible to solve the inverse Ising problem with a variety of algorithms in just a few lines of code.

WHAT IS MAXIMUM ENTROPY?

Shannon introduced the concept of information entropy in his seminal paper about communication over a noisy channel [5]. Information entropy is the unique measure of uncertainty that follows from insisting on some elementary principles of consistency. According to Shannon, information entropy, hereon just "entropy," over the probability distribution of possible discrete configurations s of a system is

$$S[p] = -\sum_{\mathrm{s}\in\mathcal{S}} p(\mathrm{s}) \log p(\mathrm{s}) \tag{1}$$

These configurations could be firing on-off patterns in neurons, a the arrangement of 4 letters in a word, or the orientation of spins in a material.

When there is no structure in the distribution, meaning that the probability is uniform $p_s = p_{s'}$, entropy is at a maximum. In the context of communication theory as Shannon first discussed, this means that there is no structure to exploit to make a prediction about the next part of an incoming message; thus, maximum entropy means that each new part of the message is maximally "surprising." At the other extreme, when the message consists of the same bit over and over again, we can always guess at the following part of the message and the signal has zero entropy. In the context of modeling, we use entropy not to refer to the difficulty of the message, but to our state of knowledge about it. When we are uncertain as to the content of the message, we should make guesses that incorporate our uncertainty about the presence of structure in the message. Entropy is the precise quantity for measuring our uncertainty.

Maximum entropy, or maxent, is the formal framework for building models that are consistent with the data but otherwise as structureless as possible [6, 7]. This is a constrained maximization problem. From the data set, we compute some useful feature $f_{\mathrm{k}}(\mathrm{s})$ over all the observations in the data set $\mathrm{s} \in \mathcal{D}$. In a stochastic setting, this value will not always be the same, so we compute

the average of the feature,

$$\langle f_k \rangle_{\text{data}} = \frac{1}{R} \sum_{s \in \mathcal{D}} f_k(\text{s}) \tag{2}$$

According to the model in which each observation s occurs with some probability $p(\text{s})$, the same average is calculated over all possible states

$$\langle f_{\text{k}} \rangle = \sum_{\text{s} \in \mathcal{S}} p(\text{s}) f_{\text{k}}(\text{s}) \tag{3}$$

We assert that the model should fit the K features while maximizing entropy. The standard procedure is to solve this by the method of Langrangian multipliers. We construct the Langrangian functional $\mathcal{L}$ by introducing the multipliers λ_{k}.

$$\mathcal{L}[p] = -\sum_{\text{s}} p(\text{s}) \log p(\text{s}) - \sum_{k}^{K} \lambda_k \left(\langle f_{\text{k}} \rangle - \langle f_{\text{k}} \rangle_{\text{data}} \right) \tag{4}$$

In the notation of statistical physics, the Langrangian is the Helmholtz free energy describing the competition between entropy and the structure described in the Hamiltonian E in equilibrium (See Appendix) [1].

$$F = S - \langle E \rangle \tag{5}$$

$$E = -\sum_{\text{k}}^{K} \lambda_{\text{k}} f_{\text{k}}(s) \tag{6}$$

This formulation makes clear the fundamental connection that statistical mechanics is an inference procedure using the maximum entropy principle [7].

Then, we solve for the fixed point by taking the derivative with respect to λ_{k}. The resulting model is a Boltzmann distribution over states

$$p(s) = e^{-E(s)} \Big/ Z \tag{7}$$

with normalization, the partition function,

$$Z = \sum_{\text{s}} e^{-E(s)} \tag{8}$$

Finding the parameters that match the constraints is equivalent to minimizing the Kullback-Leibler divergence between the model and the data [8]

$$D_{KL}(p_{\text{data}} || p_{\text{ME}}) = \sum_{s} p_{\text{data}} \log \left(\frac{p_{\text{data}}(s)}{p_{\text{ME}}(s)} \right) \tag{9}$$

$$\frac{\partial D}{\partial \lambda_k} = \sum_{s} p_{\text{data}}(s) \frac{\partial (-E - \log Z)}{\partial \lambda_k} \tag{10}$$

$$0 = \langle f_k \rangle_{\text{data}} - \langle f_k \rangle_{\text{ME}} \tag{11}$$

[1] If we fix the average energy of the system, we get the microcanonical ensemble (See Appendix). which is equivalent to the axiom that all microstates have equal probability (that is the maxent distribution).

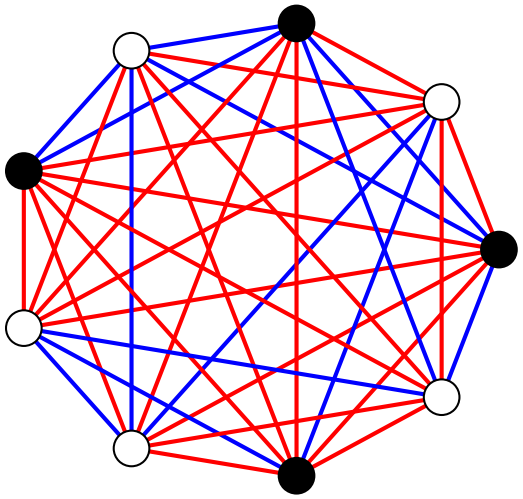

FIG. 1. Example of a fully connected Ising model with random couplings. Each spin (circle) can take one of two states $\sigma_{\text{i}} \in \{-1, 1\}$ and is connected to every other spin in the system with a positive or negative coupling.

[Of course, we also have to show that the problem is convex.] In other words, the parameters of the maximum entropy model are the ones that minimize the information theoretic "distance" to the distribution of the data. Note that these parameters are given by the data and so there is no search for the best parameters in the conventional sense.

Ising model

The Ising model is a statistical physics model of magnetism [9]. It consists of a set of spins σ_{i} with 2 possible orientations (up and down), each coupled to an external magnetic field h_{i} and coupled to each other with couplings J_{ij}. The strength of the magnetic field determines the tendency of each of the spins to orient in a particular direction and the couplings J_{ij} determine whether the spins tend to point together ($J_{\text{ij}} > 0$) or against each other ($J_{\text{ij}} < 0$). Typically, neighbors are defined as spins that interact with one another given by some underlying lattice structure as in Figure 1.

The energy, or Hamiltonian, of each configuration determines its probability.

$$E = -\sum_{\langle \text{ij} \rangle} J_{\text{ij}} \sigma_{\text{i}} \sigma_{\text{j}} - \sum_{\text{i}=1}^{N} h_{\text{i}} \sigma_{\text{i}} \tag{12}$$

We can derive the Ising model from the perspective maximum entropy. Fixing the the means and pairwise correlations to those observed in the data

$$\langle \sigma_{\text{i}} \rangle_{\text{data}} = \langle \sigma_{\text{i}} \rangle \tag{13}$$

$$\langle \sigma_{\text{i}} \sigma_{\text{j}} \rangle_{\text{data}} = \langle \sigma_{\text{i}} \sigma_{\text{j}} \rangle \tag{14}$$

we go through the procedure of constructing the Langrangian from Eq 4

$$\mathcal{L}[p] = -\sum_s p(s)\log p(s) + \sum_{\langle \mathrm{ij}\rangle} J_{\mathrm{ij}}\left\langle \sigma_{\mathrm{i}}\sigma_{\mathrm{j}}\right\rangle + \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}}\left\langle \sigma_{\mathrm{i}}\right\rangle \tag{15}$$

$$\frac{\partial \mathcal{L}[p]}{\partial p(s)} = -\log p(s) - 1 + \sum_{\langle \mathrm{ij}\rangle} J_{\mathrm{ij}}\sigma_{\mathrm{i}}\sigma_{\mathrm{j}} + \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}}\sigma_{\mathrm{i}} \tag{16}$$

$$\log p(s) = -1 + \sum_{\langle \mathrm{ij}\rangle} J_{\mathrm{ij}}\sigma_{\mathrm{i}}\sigma_{\mathrm{j}} + \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}}\sigma_{\mathrm{i}} \tag{17}$$

$$p(s) = e^{-E(s)} \Big/ Z \tag{18}$$

where to enforce normalization of the probability distribution $\sum_s p(s) = 1$,

$$Z = \sum_s e^{-E(s)} \tag{19}$$

Thus, the resulting model is exactly the Ising model mentioned earlier.

Despite the simplicity of the Ising model, the structure imposed by the discrete nature of the spins means that finding the parameters is challenging analytically and computationally. In the last few years, numerous techniques have been suggested for solving the inverse Ising problem exactly or approximately [10]. We have made a few of them part of our Python package CONIII to provide a convenient interface for using algorithms already part of the package or adding new algorithms. Here, we briefly describe the algorithms that are part of the first official version of the package. The goal is to give the user a sense (or reminder) of how they work without bogging him or her down in heavy detail. For more detail, we suggest perusing the papers referenced in each section or the review [10]. For a complete beginner, it may be useful to first get familiar with a slower introduction like in the Appendix of Ref [3], Ref [11], or Ref [6].

ENUMERATION

The naïve approach that only works for small systems is to write out the equations from Eq 11 and solve them numerically. After writing out all K equations,

$$\left\langle f_{\mathrm{k}}\right\rangle_{\mathrm{data}} = \left\langle f_{\mathrm{k}}\right\rangle \tag{20}$$

$$\left\langle f_{\mathrm{k}}\right\rangle_{\mathrm{data}} = -\frac{\partial \ln Z}{\partial \lambda_{\mathrm{k}}}, \tag{21}$$

we can use any standard optimization algorithm to find the parameters λ_{k}. This approach, however, involves enumerating all terms in the partition function Z whose number grows exponentially with system size.

For the Ising model, the first step in the algorithm for writing down the equations is $\mathcal{O}(K^2 2^N)$ where K is the number of constraints and N the number of spins. In the second step, each evaluation of the objective in the minimization algorithm will be of the same order. For relatively small systems $n \leq 15$, however, this approach is feasible on a typical desktop computer and is a good way to test the results of a more complicated algorithm.

This approach is part of the `Exact` class that contains code for writing Eqs 11 into a file and solving them with the `scipy.optimize` library.

MONTE CARLO METHOD

Perhaps the most straightforward and most expensive computational approach is to use Monte Carlo Markov Chain (MCMC) sampling to approximate the distribution and adjust the parameters appropriately after each step. The parameters are adjusted using a learning rule, and both sampling and learning are repeated til the stopping criterion is met. This can be combined with a variety of approximate gradient descent methods to reduce the number of sampling steps. The particular technique implemented in CONIII is the Monte Carlo Histogram (MCH) method [12].

Since the sampling step is expensive, the idea behind MCH is to reuse a sample for more than one gradient descent step because we can predict how the distribution will change if we modify the parameters slightly [12]. Given that we have a sample with probability distribution $p(\mathrm{s})$ generated with parameters λ_{k}, we would like to estimate the new distribution $p'(\mathrm{s})$ from adjusting our parameters $\lambda'_{\mathrm{k}} = \lambda_{\mathrm{k}} + \Delta\lambda_{\mathrm{k}}$. We can leverage our current sample to make this extrapolation.

$$p' = \frac{p'}{p}p \tag{22}$$

$$p'(\mathrm{s}) = \frac{Z}{Z'}e^{\sum_{\mathrm{k}} \Delta\lambda_k f_{\mathrm{k}}(\mathrm{s})}p(\mathrm{s}) \tag{23}$$

To estimate the average,

$$\sum_s p'(\mathrm{s})f_{\mathrm{k}}(\mathrm{s}) = \frac{Z}{Z'}\sum_s p(\mathrm{s})e^{\sum_{\mathrm{k}} \Delta\lambda_{\mathrm{k}} f_{\mathrm{k}}(\mathrm{s})}f_{\mathrm{k}}(\mathrm{s}) \tag{24}$$

To be explicit about the fact that we only have a sampled approximation to p, we replace p with the data distribution.

$$\left\langle f_{\mathrm{k}}\right\rangle' = \frac{Z}{Z'}\left\langle e^{\sum_{\mathrm{k}} \Delta\lambda_{\mathrm{k}} f_{\mathrm{k}}(\mathrm{s})}f_{\mathrm{k}}(\mathrm{s})\right\rangle_{\mathrm{sample}} \tag{25}$$

Likewise, the ratio of the partition function can be estimated

$$\frac{Z}{Z'} \approx 1 \Big/ \left\langle e^{\sum_{\mathrm{k}} \Delta\lambda_{\mathrm{k}} f_{\mathrm{k}}(\mathrm{s})}\right\rangle_{\mathrm{sample}} \tag{26}$$

At each step, we update the Lagrangian multipliers $\{\lambda_\mathrm{k}\}$ while being careful to stay within the bounds of a reasonable extrapolation. One suggestion is to update the parameters with some inertia

$$\Delta\lambda_\mathrm{k}(t+1) = \Delta\lambda_\mathrm{k}(t) + \epsilon\Delta\lambda_\mathrm{k}(t-1) \tag{27}$$

$$\Delta\lambda_\mathrm{k}(t) = \eta\left(\langle f_\mathrm{k}\rangle' - \langle f_\mathrm{k}\rangle\right) \tag{28}$$

This has the correct fixed points.

In practice, MCH can be difficult to tune properly and one must check in on the progress of the algorithm often. One issue is choosing how to set the learning rule parameters η and ϵ. One suggestion for η is to shrink it as the inverse of the number of iterations [13]. Another issue is that parameters cannot be changed by too much when using the MCH approximation step or the extrapolation to λ'_k will be inaccurate and the algorithm will fail to converge. In CONIII, this can be controlled by setting a bound on the maximum possible change in each parameter $\Delta\lambda_\mathrm{max}$ and restricting the norm of the vector of change in parameters $\sum_k \sqrt{\Delta\lambda_\mathrm{k}^2}$. Another issue is setting the parameters of the MCMC sampling routine. Both the burn time (the number of iterations before starting to sample) and sampling iterations (number of iterations between samples) must be large enough that we are sampling from the equilibrium distribution. Typically, these are found by looking at the decorrelation time in the energy or correlations as a function of MCMC iterations made. The parameter may need to be updated during the course of MCH because the sampling parameters may need to change with the estimated parameters of the model. For some regime of parameter space, samples are correlated over long times and alternative sampling methods like Wolff or Swendsen-Wang should be used. We do not discuss these sampling details here, but see Ref [14, 15] for examples.

The main computation cost for MCH is the sampling step. The runtime is proportional to the number of samples n_sample, number of MCMC iterations n_{MC}, the number of constraints K: $\mathcal{O}(n_\mathrm{MC} n_\mathrm{sample} K)$, whereas the MCH estimate is relatively quick $\mathcal{O}(n_\mathrm{sample} n_\mathrm{MCH} K)$ because the number of MCH approximation steps is much smaller than the number of MCMC sampling iterations $n_\mathrm{MCH} << n_\mathrm{MC}$. For the Ising model, $K \sim N^2$, the system size squared.

MCH is implemented in the `MCH` class.

PSEUDOLIKELIHOOD

The pseudolikelihood approach is an analytic approximation to the likelihood that drastically reduces the computational complexity of the problem and is exact in the thermodynamic limit [16]. We maximize the conditional probability of each spin s_i given the rest of the system

$$p\left(s_\mathrm{i}|\mathbf{s}_{\setminus\mathrm{i}}\right) = \left(1 + e^{-2s_\mathrm{i}\left(h_\mathrm{i} + \sum_{j\neq i} J_\mathrm{ij} s_\mathrm{j}\right)}\right)^{-1} \tag{29}$$

Taking the logarithm and summing over all spins, we define the approximate likelihood to be summed over all data points indexed by r.

$$f(h_\mathrm{i}, \boldsymbol{J}_\mathrm{i}) = \sum_{\mathrm{r}=1}^{R} \ln p\left(s_\mathrm{i}^{(r)} \middle| \boldsymbol{s}_{\setminus\mathrm{i}}^{(r)}\right) \tag{30}$$

In the limit where the ensemble is well sampled, the average over the data can be replaced by an average over the ensemble

$$f(h_\mathrm{i}, \boldsymbol{J}_\mathrm{i}) = \sum_{\boldsymbol{s}} \ln p\left(s_\mathrm{i}^{(r)} \middle| \boldsymbol{s}_{\setminus\mathrm{i}}^{(r)}\right) p(\boldsymbol{s}; h, J) \tag{31}$$

At maximum likelihood,

$$\frac{\partial f}{\partial J_\mathrm{ij}} = \sum_{\boldsymbol{s}} \ln p\left(s_\mathrm{i}^{(r)} \middle| \boldsymbol{s}_{\setminus\mathrm{i}}^{(r)}\right) p(\boldsymbol{s}; h, J) = 0 \tag{32}$$

Pseudolikelihood is extremely fast. Each iteration only is $O(RN^2)$ and often surprisingly accurate.

We have implemented pseudolikelihood for the Ising model in `Pseudo`.

MINIMUM PROBABILITY FLOW

Minimum probability flow involves analytically approximating how the probability distribution *changes* as we modify the *configurations* [17, 18]. In the methods so far mentioned, the approach has been to maximize the objective (the likelihood function) by immediately taking the derivative with respect to the parameters. With MPF, we first posit a set of dynamics that will lead the data distribution to equilibrate to that of the model. When these distributions are equivalent, then there is no "probability flow" between them. This technique is closely related to score matching where instead we have a continuous state space and can directly take the derivative with respect to the states without specifying dynamics [19].

As before, we start with minimizing the Kullback-Leibler divergence, but instead of taking the derivative with respect to the parameters, we first ask how the probability flows between the model and the states in the data $\mathcal{D}$ if the dynamics are run for an infinitesimal amount of time ϵ, the idea being that the relative difference between the probability distributions are minimized with optimal parameters.

$$\partial_t D_{KL}(p^{(0)}||p^{(t)}(\{\lambda_k\})) = \sum_{\mathrm{s}\notin\mathcal{D}} \dot{p}_\mathrm{s}(\lambda_k) \tag{33}$$

$$K(\{\lambda_\mathrm{k}\}) = \sum_{\mathrm{s}\notin\mathcal{D}} \dot{p}_\mathrm{s}(\lambda_k) \tag{34}$$

Monte Carlo dynamics (satisfying ergodicity and detailed balance) would lead to equilibration of the two distributions. A simple transition matrix suggested in Ref [18] is

$$\dot{p}_{\rm s} = \sum_{s' \neq s} \Gamma_{{\rm s}{\rm s}'} p_{{\rm s}'} - \sum_{s' \neq s} \Gamma_{{\rm s}'{\rm s}} p_{\rm s} \tag{35}$$

$$\Gamma_{{\rm s}{\rm s}'} = g_{{\rm s}{\rm s}'} \exp\left[\frac{1}{2}\left(E_{{\rm s}'} - E_{\rm s}\right)\right] \tag{36}$$

with transition probabilities $\Gamma_{{\rm s}{\rm s}'}$ from state s′ to state s. The connectivity matrix $g_{{\rm s}{\rm s}'}$ specifies whether there is edge between states s and s′ such that probability can flow between them. By choosing a sparse $g_{{\rm s}{\rm s}'}$ while not breaking ergodicity, we drastically reduce the computational cost of calculating the objective function.

Finally, we must find the minimum of the objective function

$$K(\{\lambda_{\rm k}\}) = \sum_{{\rm s} \notin \mathcal{D}} \dot{p}_{\rm s}(\lambda_k) \tag{37}$$

At each step of the algorithm for the Ising model, the runtime is $O(RN^2)$.

MPF is implemented in the `MPF` class.

MEAN-FIELD METHOD

One attractively simple and efficient version of the regularized approach starts with mean-field theory. In the inverse Ising problem, mean-field theory is equivalent to treating each binary individual as instead having a continuously varying state (corresponding to its mean value). The inverse problem then turns into simply inverting the correlation matrix C [**?**]:

$$J_{\rm ij}^{\rm mean-field} = -\frac{(C^{-1})_{\rm ij}}{\sqrt{p_{\rm i}(1-p_{\rm i})p_{\rm j}(1-p_{\rm j})}}, \tag{38}$$

where

$$C_{\rm ij} = \frac{p_{\rm ij} - p_{\rm i}p_{\rm j}}{\sqrt{p_{\rm i}(1-p_{\rm i})p_{\rm j}(1-p_{\rm j})}}, \tag{39}$$

and where $p_{\rm i}$ corresponds to the frequency of individual i being in the active (+1) state and $p_{\rm ij}$ is the frequency of the pair i and j being simultanously in the active state.

A simple regularization scheme in this case is to discourage large values in the interaction matrix J. This corresponds to putting more weight on solutions that are closer to the case with no interactions (independent individuals). A particularly convenient form adds the following term, quadratic in J, to the negative log-likelihood:

$$\gamma \sum_{\rm i} \sum_{\rm j>i} J_{\rm ij}^2 p_{\rm i}(1-p_{\rm i})p_{\rm j}(1-p_{\rm j}). \tag{40}$$

In this case, the regularized version of the mean-field solution in (38) can be solved analytically, with the slowest computational step coming from the inversion of the correlation matrix. For details, see Refs. [20**?**].

The idea is then to vary the regularization strength γ to move between the non-interacting case ($\gamma \to \infty$) and the naively calculated mean-field solution (38) ($\gamma \to 0$). While there is no guarantee that varying this one parameter will produce solutions that are good enough to "fit within error bars," this approach has been successful in at least one case of fitting social interactions [**?**].

This is implemented in `RegularizedMeanField`.

CLUSTER EXPANSIONS

Adaptive cluster expansion [21**?** **?**] iteratively calculates terms in the cluster expansion of the entropy S:

$$S = \sum_{\Gamma} \Delta S_{\Gamma}, \tag{41}$$

where the sum is over clusters Γ and in the exact case includes all $2^N - 1$ possible nonempty subsets of individuals in the system.[2] The inverse Ising problem is solved independently on each of the clusters, which can be done exactly when the clusters are small. These results are used to construct a full interaction matrix J. The expansion starts with small clusters and expands to use larger clusters, neglecting any clusters whose contribution ΔS_Γ to the entropy falls below a threshold. To find the best solution that does not overfit, the threshold is initially set at a large value and then lowered, gradually including more clusters in the expansion, until samples from the resulting J fit the desired statistics of the data sufficiently well.

In CONIII, the selective cluster expansion method is implemented in the `ClusterExpansion` class.

SAMPLERS

In CONIII, we have implemented two versions of the Metropolis algorithm. One is specific to the Ising model `MCIsing` and the other `MC` can sample a system as long as the function for calculating the energy is supplied by the user.

In some parameter regimes, where spins are tightly correlated, the Metropolis algorithm is very inefficient.

[2] In the simplest version of the expansion, one expands around $S = 0$. In some cases it can be more advantageous to write the expansion around $S - S_0$, where S_0 is a reference entropy corresponding to an easily calculated case such as the independent individual solution or one of the mean-field solutions described in the previous section [**?**].

Cluster sampling like Wolff or Swendsen-Wang are much more efficient.

MODEL FITTING

A fundamental problem in model inference is that uncertainty coming from the finiteness of data translates into uncertainties in parameters. In the exposition of the maxent formulation in Eqns ??, there is no fitting of the model parameters because they are given by the data once the constraints are specified. In reality, constraints are typically estimated from a finite sample, and they are noisy. The straightforward answer to this problem is to take more data—in a pairwise maximum entropy problem, we might insist that we have enough samples to well-constrain the correlation between every pair of individuals. But it is not always possible to take enough data. For instance, in a social system in which we are trying to measure stable social structure that lasts on the order of months, there are only a finite number of social interactions that occur over those months, which may not be enough to tightly constrain parameters. When we fit to the exactly measured constraints, we run the danger of overfitting and poor generalization in the limit of small data.

When we find the parameters that minimize the Kullback-Leilber divergence between the model and the data distributions, we are maximizing the likelihood of the data. Around the peak in likelihood, sample size fluctuations will determine some curvature of the likelihood curve around the maximum. This uncertainty is reflected in the sample size fluctuations we can easily calculate from the data. Assuming that the data is independently and identically distributed, the errors are given by the standard error of the mean of a binomial distribution $p_{\mathrm{ij...k}} = p(s_{\mathrm{i}} = s_{\mathrm{j}} = ... = s_{\mathrm{k}} = 1)$ with K data points.

$$\delta_{\mathrm{ij...k}} = \sqrt{(1 - p_{\mathrm{ij...k}})p_{\mathrm{ij...k}}/K} \tag{42}$$

In other words, we are not obliged to keep pushing the parameters to get closer to the data once we have gotten close enough, where close enough is given by the noise the data distribution.[3]

An alternative approach is to regularize the problem like with the regularized mean field or cluster expansion algorithms. Here, we restrict the search space in some principled way so that more complicated solutions are disallowed. We then check that the regularized solutions fit the data within expected statistical fluctuations. If not, a more lax regularization can be used to allow more complicated solutions that are able to fit the remaining signal in the data. In the Bayesian formulation, this approach is equivalent to including a prior distribution over the parameters

$$\log p(\mathrm{model}|\mathrm{data}) \propto \log p(\mathrm{data}|\mathrm{model}) + \log p(\mathrm{model}) \tag{43}$$

MODEL VALIDATION

Besides just fitting the model while accounting for the noisiness of the data, how do we know if we need a more complicated model? To answer this question, we need a way of quantifying the tradeoff between the complexity and fit of the model.

When we search for the peak in the posterior probability of the data, we must account for the balance between a good description of the data with the cost of describing the model. This tradeoff manifests as the competition between the likelihood and prior on the model's parameters in the Bayesian formulation of the problem in Eq 43. Roughly speaking, we can imagine that specifying the L parameters is localizing a region in a high-dimensional space where each dimension shrinks with the number of data points K as $\sim K^{-1/2}$. The volume of the "ball" grows like $K^{-L/2}$, so that the information cost goes like $-\frac{L}{2}\log(K)$ [3]. This is the picture formalized by quantities like the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) [22].

Although likelihood tells us how well the models are doing relative to one another, it does not tell us in detail how the model is fitting the data. Basic checks would be to compare against any other correlations in the data. Since these are quantities averaged over the joint probability distribution of many spins, a stricter check would be to compare the entire probability distribution of the pairwise maxent model with that of the data [3], but this is only feasible when the data set is reasonably large. A more specific test would be against features that are relevant to the question at hand. For conflict in monkeys, a coarse-grained feature is the distribution of conflict sizes which seems to have a characteristically long tail, and that is checked specifically [20].

A summary of how well the model captures the distribution across the entire probability distribution is the multi-information. If we take a maxent model and add further constraints, the models can be ordered in terms of entropy $S_1 > S_2 > ...S_{\mathrm{m}} > ... > S_{\mathrm{data}}$, where the minimum entropy the most constrained model could have is equal to the data. To measure how much correlation in the data our model has captured, we can calculate the multi-information $I_{\mathrm{m}} = S_1 - S_{\mathrm{m}}$, the amount of correlation captured by the model relative to the independent model. The fraction of multi-information captured is $F = I_{\mathrm{m}}/I_{\mathrm{data}}$. This is a measure of how much of the

[3] This is like early stopping [].

correlation in the data is captured by the model.[4]

Choosing which constraints to impose is an important question. Typically, the approach is to constrain the lowest order interactions that are sufficient to produce collective behavior. The intuition from physics is from the observation that many physical systems are extremely well (if not exactly) described by pairwise interactions.[5] In fact, just the observation that we can model a system well with only pairwise interactions may be surprising [23]. From the model fitting perspective, however, we might choose to constrain other parts of the probability distribution. In Ref [24], they explore choosing correlations to constrain depending on whether or not they are signicantly large. In Ref [25], they discuss how a pairwise model becomes an increasingly effective description of a system with higher order interactions as the system gets larger.

Conclusion

Science is a process of model selection. In the ideal picture, we start with the simplest models and the fewest constraints possible, and then we increase the complexity of the model til it is sufficiently so to make good predictions. In principle, we could add as many contraints as would allow us to fit the data well, but the idea is that complex models are not only "expensive" but they do not generalize well. Maxent is a principled framework for this picture of model building. The number of parameters and the order of the constraints we impose can be adjusted to test our hypotheses about what matters for the system. In this sense, the maxent approach is a useful "model" framework for thinking about statistical inference problems far beyond statistical physics. We build an open-source Python package that we hope will be accessible and useful for those unfamiliar with maxent approaches to experiment and perhaps apply this technique to their questions.

[1] E Schneidman, M J Berry, RS II, and W Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 2006.

[2] William Bialek, Andrea Cavagna, Irene Giardina, Thierry Mora, Edmondo Silvestri, Massimiliano Viale, and Aleksandra M Walczak. Statistical mechanics for natural flocks of birds. *PNAS*, 109(13):4786–4791, 2012.

[4] See Refs [11] and [3] for details and further references on how to estimate information quantities.

[5] In statistical physics, highly correlated behavior across space and time can emerge from pairwise interactions and renormalization group flow ensures they are the only relevant parameters in the thermodynamic limit [].

[3] Edward D Lee, Chase P Broedersz, and William Bialek. Statistical Mechanics of the US Supreme Court. *J Stat Phys*, pages 1–27, April 2015.

[4] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81(2):591–646, May 2009.

[5] Claude Elwood Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, July 1948.

[6] E T Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[7] E T Jaynes. Information Theory and Statistical Mechanics. *Phys. Rev.*, 106(4):620, May 1957.

[8] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, 2nd edition, 2006.

[9] E Ising. *Beitrag zur Theorie des Ferromagnetismus*. PhD thesis, December 1924.

[10] H Chau Nguyen, Riccardo Zecchina, and Johannes Berg. Inverse statistical problems: from the inverse Ising problem to data science. *arXiv*, February 2017.

[11] W S Bialek. *Biophysics: Searching for Principles*. Princeton University Press, 2012.

[12] Tamara Broderick, Miroslav Dudik, Gašper Tkačik, Robert E Schapire, and William Bialek. Faster solutions of the inverse pairwise Ising problem. *arXiv*, pages 1–8, December 2007.

[13] Gašper Tkačik, Elad Schneidman, Michael J Berry II, and William Bialek. Ising models for networks of real neurons. *arXiv*, November 2006.

[14] David J C MacKay. Information Theory, Inference and Learning Algorithms, September 2005.

[15] M E J Newman and G T Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, February 1999.

[16] Erik Aurell and Magnus Ekeberg. Inverse Ising Inference Using All the Data. *Physical Review Letters*, 108(9):090201, March 2012.

[17] Jascha Sohl-Dickstein, Peter Battaglino, and Michael R DeWeese. Minimum Probability Flow Learning. *arXiv*, June 2009.

[18] Jascha Sohl-Dickstein, Peter B Battaglino, and Michael R DeWeese. New Method for Parameter Estimation in Probabilistic Models: Minimum Probability Flow. *Physical Review Letters*, 107(22):220601, November 2011.

[19] Aapo Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, February 2007.

[20] Bryan C Daniels, David C Krakauer, and Jessica C Flack. Control of finite critical behaviour in a small-scale social system. *Nat Comms*, 8:1–8, 1.

[21] S Cocco and R Monasson. Adaptive Cluster Expansion for Inferring Boltzmann Machines with Noisy Data. *Physical Review Letters*, 106(9):090601, March 2011.

[22] Sadanori Konishi and Genshiro Kitagawa. Information Criteria and Statistical Modeling. Springer New York, New York, NY, 2008.

[23] William Bialek and Rama Ranganathan. Rediscovering the power of pairwise interactions. *arXiv*, pages 1–8, December 2007.

[24] Elad Ganmor, Ronen Segev, and Elad Schneidman. Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *PNAS*, 108(23):9679–9684, 2011.

[25] Lina Merchan and Ilya Nemenman. On the Sufficiency of Pairwise Interactions in Maximum Entropy Models of Networks. *J Stat Phys*, 162(5):1294–1308, February 2016.