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 explain how one should approach and validate maxent models from the perspective of model selection.

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 or asymmetry 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." 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 be precise about our uncertainty.

The idea behind the maximum entropy, or maxent, approach is to build a model that is consistent with the data but otherwise as structureless as possible [6, 7]. Using entropy as the measure of uncertainty, the formal notion is that we have a constrained maximization problem. From the data, we choose constraints indexed by k,

$$\langle f_k(s)\rangle_{\mathrm{data}} = \sum_{s\in\mathcal{S}} p_{\mathrm{data}}(\mathrm{s}) f_k(\mathrm{s}) \tag{2}$$

With K constraints, the maximum entropy problem can be solved by the method of Langrangian multipliers. We construct the Langrangian functional $\mathcal{L}$ by introducing the Langrangian multipliers λ_{k}.

$$\mathcal{L}[p] = -\sum_s p_s \log p_s - \sum_k^K \lambda_k \left[\langle f_k(s)\rangle - \langle f_k(s)\rangle_{\mathrm{data}}\right] \tag{3}$$

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) [19].

$$F = S - \langle E \rangle \tag{4}$$

$$E = -\sum_{\mathrm{k}}^{K} \lambda_{\mathrm{k}} f_{\mathrm{k}}(s) \tag{5}$$

This formulation makes clear the fundamental connection that statistical mechanics is an inference procedure using the maximum entropy principle [7]. Of course, the standard procedure is to solve for the fixed point by taking the derivative with respect to λ_{k}.

The resulting model is a Boltzmann distribution over states

$$p(s) = e^{-E(s)} \Big/ Z \tag{6}$$

with normalization, the partition function,

$$Z = \sum_{\mathrm{s}} e^{-E(s)} \tag{7}$$

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

$$D_{KL}(p_{\mathrm{data}} || p_{\mathrm{ME}}) = \sum_{s} p_{\mathrm{data}} \log \left(\frac{p_{\mathrm{data}}(s)}{p_{\mathrm{ME}}(s)} \right) \tag{8}$$

$$\frac{\partial D}{\partial \lambda_k} = \sum_{s} p_{\mathrm{data}}(s) \frac{\partial(-E - \log Z)}{\partial \lambda_k} \tag{9}$$

$$0 = \langle f_k \rangle_{\mathrm{data}} - \langle f_k \rangle_{\mathrm{ME}} \tag{10}$$

[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 [**?**]. 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 ($J_{\mathrm{ij}} > 0$) or against each other ($J_{\mathrm{ij}} < 0$). Typically, neighbors are defined as spins that interact with one another given by some underlying lattice structure as in Figure 1.

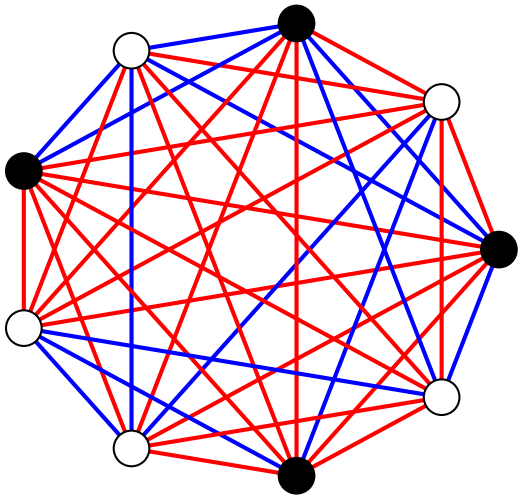

FIG. 1. Example of a fully connected Ising model with random couplings.

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

$$E = -\sum_{\langle \mathrm{ij} \rangle} J_{\mathrm{ij}} \sigma_{\mathrm{i}} \sigma_{\mathrm{j}} - \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}} \sigma_{\mathrm{i}} \tag{11}$$

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

$$\langle \sigma_{\mathrm{i}} \rangle_{\mathrm{data}} = \langle \sigma_{\mathrm{i}} \rangle \tag{12}$$

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

we go through the procedure of constructing the Langrangian from Eq 3

$$\mathcal{L}[p] = -\sum_{s} p(s) \log p(s) + \sum_{\langle \mathrm{ij} \rangle} J_{\mathrm{ij}} \langle \sigma_{\mathrm{i}} \sigma_{\mathrm{j}} \rangle + \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}} \langle \sigma_{\mathrm{i}} \rangle \tag{14}$$

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

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

$$p(s) = e^{-E(s)} \Big/ Z \tag{17}$$

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

$$Z = \sum_{s} e^{-E(s)} \tag{18}$$

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 [**?**]. 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 [**?**]. For a complete beginner, it may be useful to first get familiar with a slower introduction like in the Appendix of Ref [3], [**?**], or [**?**].

ENUMERATION

The naïve approach that only works for small systems is to write out the equations from Eq 10 and solve them numerically. After writing out all K equations,

$$\langle f_{\mathrm{k}}\rangle_{\mathrm{data}} = \langle f_{\mathrm{k}}\rangle \tag{19}$$

$$\langle f_{\mathrm{k}}\rangle_{\mathrm{data}} = -\frac{\partial \ln Z}{\partial \lambda_{\mathrm{k}}}, \tag{20}$$

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 10 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 is repeated til the stopping criterion is met. The magic behind MCMC approaches is that only relative probabilities of states need to be known to return a sample of the distribution. Thus, we do not need to calculate the partition function, but can instead compute the difference in energy between two states. This can be combined with a variety of stochastic gradient descent methods to reduce the number of sampling steps. The particular technique implemented in CONIII is the Monte Carlo Histogram method [11].

Since the sampling step is expensive, the idea behind Monte Carlo histogram 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 [11]. Given that we have a sample with probability distribution p_s generated with parameters λ_{k}, we would like to estimate the new distribution p'_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.

$$p'_s = \frac{p'_s}{p_s} p_s \tag{21}$$

$$= \frac{Z}{Z'} e^{\sum_k \Delta\lambda_k f_k(s)} p_s \tag{22}$$

To estimate the average,

$$\sum_s p'_s f_k(s) = \frac{Z}{Z'} \sum_s p_s e^{\sum_k \Delta\lambda_k f_k(s)} f_k(s) \tag{23}$$

To be explicit about the fact that we only have a sampled approximation to p,

$$\langle f_k\rangle' = \frac{Z}{Z'} \left\langle e^{\sum_k \Delta\lambda_k f_k(s)} f_k(s)\right\rangle_{\mathrm{sample}} \tag{24}$$

Likewise, the ratio of the partition function can be estimated

$$\frac{Z}{Z'} \approx 1 \Big/ \left\langle e^{\sum_k \Delta\lambda_k f_k(s)}\right\rangle_{\mathrm{sample}} \tag{25}$$

The main difficulty with MCH is choosing a learning rule for how to update the Lagrangian multipliers $\{\lambda_k\}$ at each iteration while being careful to stay within the bounds of a reasonable extrapolation. One suggestion is to update the parameters with some inertia

$$\Delta\lambda_k(t+1) = \Delta\lambda_k(t) + \eta\Delta\lambda_k(t-1) \tag{26}$$

$$\Delta\lambda_k(t) = \epsilon\left(\langle f_k\rangle' - \langle f_k\rangle\right) \tag{27}$$

This has the correct fixed points. One suggestion is to shrink ϵ exponentially with the number of iterations.

In practice, MCH can be difficult to tune properly and one must check in on the progress of the algorithm often. One issue is that parameters cannot be changed by too much 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 $\Delta\lambda_{\max}$ and restricting the

norm of the vector of change in parameters $\sum_k \sqrt{\Delta\lambda_{\rm k}^2}$. 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. To add further complications, for some sets of parameters samples are correlated over long times and alternative sampling methods must be used. We do not discuss these sampling details here, but see Ref [10] for examples.

The main computation cost for MCH is the sampling step. The runtime is proportional to the number of samples $n_{\rm sample}$, number of MCMC iterations n_{MC}, the number of constraints K: $\mathcal{O}(n_{\rm MC} n_{\rm sample} K)$, whereas the MCH estimate is relatively quick $\mathcal{O}(n_{\rm sample} n_{\rm MCH} K)$ because the number of MCH approximation steps is much smaller than the number of MCMC sampling iterations $n_{\rm MCH} << n_{\rm MC}$. 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 [12]. We maximize the conditional probability of each spin $s_{\rm i}$ given the rest of the system

$$p\left(s_{\rm i}|\mathbf{s}_{\backslash {\rm i}}\right) = \left(1 + e^{-2s_{\rm i}\left(h_{\rm i} + \sum_{j\neq i} J_{\rm ij} s_{\rm j}\right)}\right)^{-1} \tag{28}$$

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

$$f(h_{\rm i}, \boldsymbol{J}_{\rm i}) = \sum_{\rm r=1}^{R} \ln p\left(s_{\rm i}^{(r)} \middle| \boldsymbol{s}_{\backslash {\rm i}}^{(r)}\right) \tag{29}$$

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

$$f(h_{\rm i}, \boldsymbol{J}_{\rm i}) = \sum_{\boldsymbol{s}} \ln p\left(s_{\rm i}^{(r)} \middle| \boldsymbol{s}_{\backslash {\rm i}}^{(r)}\right) p(\boldsymbol{s}; h, J) \tag{30}$$

At maximum likelihood,

$$\frac{\partial f}{\partial J_{\rm ij}} = \sum_{\boldsymbol{s}} \ln p\left(s_{\rm i}^{(r)} \middle| \boldsymbol{s}_{\backslash {\rm i}}^{(r)}\right) p(\boldsymbol{s}; h, J) \tag{31}$$

$$0 = \tag{32}$$

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* [13]. 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 continuous state spaces and can directly take the derivative with respect to the states without specifying dynamics [14].

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.

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

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

Monte Carlo dynamics (satisfying ergodicity and detailed balance) would lead to equilibration of the two distributions. A simple transition matrix suggested in Ref [13] is

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

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

with transition probabilities $\Gamma_{\rm ss'}$ from state s$'$ to state s. The connectivity matrix $g_{\rm ss'}$ specifies whether there is edge between states s and s$'$ such that probability can flow between them. By choosing a sparse $g_{\rm ss'}$ 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

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

MPF satisfies a number of useful properties:

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

MPF is implemented in the `MPF` class.

REGULARIZATION TO AVOID OVERFITTING

Besides the computational complexity of locating best-fit parameters, a fundamental problem in any model inference method is that uncertainty coming from the finiteness of data translates into uncertainties in parameters. 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.

The danger in naively searching for a best-fit solution in a data-poor situation is overfitting, or erroneously finding patterns in the finite-sample noise. One way to avoid overfitting is regularization: restricting 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.

We will motivate the next two approaches to maximum entropy fitting in terms of regularization.

[This section might be tightened a bit more. For one, I am somewhat confused as to whether the entire maximum entropy approach might be motivated in this way—starting with few constraints and adding the possibility of higher order constraints until the model fits well enough.]

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: [15]

$$J_{k\ell}^{\text{mean-field}} = -\frac{(C^{-1})_{k\ell}}{\sqrt{p_k(1-p_k)p_\ell(1-p_\ell)}}, \tag{38}$$

where

$$C_{k\ell} = \frac{p_{k\ell} - p_k p_\ell}{\sqrt{p_k(1-p_k)p_\ell(1-p_\ell)}}, \tag{39}$$

and where p_k corresponds to the frequency of individual k being in the active (+1) state and $p_{k\ell}$ is the frequency of the pair k and ℓ 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:

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

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. [16, 17].

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 [16].

This is implemented in `RegularizedMeanField`.

CLUSTER EXPANSIONS

Adaptive cluster expansion [15, 17, 18] 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. [**?**] 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.

INCOMPLETE DATA

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

DISCUSSION

Validation

The first step in validating a maxent model is to confirm that it matches the given constraints. In the cases where we use an approximate technique or regularize the problem, we must make sure that it matches the data within reasonable error bars given by 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.

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

Since the model is only fit to the given constraints, it is not given that it will be able to fit other features of the data that have not been specified. For the Ising model, a good fit to any higher order feature of the probability distribution is in this sense nontrivial.

Basic checks would be to compare against higher order correlations in the data. In Figure ??, we show an example of a pairwise maxent model tested against higher order correlations. 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. For each problem, there are specific features that are more relevant to the question at hand, which would be important to check. 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 [**?**].

Model complexity

How do we know if we need to add more parameters? To answer this question, we need a way of quantifying the tradeoff between the complexity and fit of the model. The idea is that a more complicated model gives us more information about the data set that we are fitting, but specifying the parameters also requires information. This is tradeoff between the information gain and the information cost. Roughly speaking, we can imagine that specifying the K parameters is localizing a region in a high-dimensional space. The volume of the "ball" grows like l^K, so that the information cost goes like $K \log(l)$. This is the picture formalized by quantities like the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) []. An accessible description of model selection with maximum entropy is described in Ref [3].

Choosing which parameters to impose is an important question. Typically, the approach has been 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. In statistical physics, the observation that highly correlated behavior across space and time can emerge from pairwise interactions has even stronger justification from renormalization group theory []. In fact, the ability to model a system well with only pairwise interactions may be surprising [**?**]. From the model fitting perspective, however, we might choose to constrain other parts of the probability distribution. In Ref [**?**], they explore choosing correlations to constrain depending on whether or not they are signicantly large. In Ref [**?**], 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.

The microcanonical ensemble and maximum entropy

The conventional textbook in statistical mechanics first introduces the concept of entropy as a way of counting the phase volume available to the system at a given energy

$$S(E) = k_B \log \Omega(E) \tag{43}$$

where k_B is Boltzmann's constant and Ω the number of states between E and $E + \delta E$. Temperature is defined as

$$\frac{1}{T} = \frac{\partial S}{\partial E} \tag{44}$$

begins with the concept of a small system coupled to a heat bath. In this limit, we can linearly expand thermodynamic quantities about the energy of the bath $E_{\text{bath}} = E - E_s$

$$S_{\text{bath}}(E - E_s) \approx S_{\text{bath}}(E) - E_s \left.\frac{\partial S}{\partial E_s}\right|_{E-E_s} \tag{45}$$

$$= S_{\text{bath}}(E) - \frac{E_s}{T} \tag{46}$$

$$e^{S_{\text{bath}}(E-E_s)/k_B} = e^{S_{\text{bath}}(E)/k_B} e^{-E_s/k_B T} \tag{47}$$

Since the entropy is proportional to the density of states at a particular energy E_s, Eq ?? corresponds to

$$p(E_s) = e^{-E_s/k_B T}/Z \tag{48}$$

With partition function Z. In other words, the Gibbs measure takes exponential form. Eq ?? can be rewritten in terms of free energy

$$\log p(E_s) = -E_s/k_B T - \log(Z) \tag{49}$$

Averaging both sides over all energy configurations and rearranging

$$-k_B T \log Z = \sum_s p(E_s) E_s + k_B T \sum_s p(E_s) \log p(E_s) \tag{50}$$

Remembering the fundamental postulate of statistical mechanics that all states are equally likely

$$-k_B T \log Z = \langle E_s \rangle - TS \tag{51}$$

$$F = \langle E_s \rangle - TS \tag{52}$$

This equation tells us how the Helmholtz free energy is related to the internal energy and the entropy of the system.

When we say that free energy is minimized for a system with Hamiltonian E_s at equilibrium, we are equivalently saying that entropy is maximized. Entropy maximization is the crux of statistical physics models.

Notes on Monte Carlo Markov Chain sampling

The key points behind MCMC sampling is ergodicity and detailed balance. Ergodicity just means that we can get from one state to another in a finite number of steps, and this ensures that we don't get stuck in a few states. Detailed balance is a sufficient but not necessary condition (stronger than needed) for ensuring that the equilibrium distribution matches the distribution that we seek. There are some specialized algorithms that don't satisfy detailed balance but do produce the desired distribution.

To check whether an algorithm works, one must prove that both these conditions are satisfied. Ergodicity is usually trivial. Typically, we write down the condition for detailed balance for any two states a and b,

$$p(a|b)p(b) = p(b|a)p(a) \tag{53}$$

$$p(b)/p(a) = p(b|a)/p(a|b) \tag{54}$$

$$e^{-(E_b - E_a)} = \tag{55}$$

A reasonable way to do this would be to take $p(b|a) = e^{-(E_b - E_a)}$ and $p(a|b) = 1$ when $E_b > E_a$.

$$p_a \, T(a \to b) A(a \to b) = p_b \, T(b \to a) A(b \to a) \tag{56}$$

$$p_a/p_b = T(b \to a)A(b \to a)/T(a \to b)A(a \to b) \tag{57}$$

where T is the transition probability and A is the acceptance probability. For a Boltzmann type model, this means that this ratio must be equal to $\exp(-\beta(E_a - E_b))$.

Swendsen-Wang

As an example, we consider the Swendsen-Wang cluster algorithm. In this algorithm, the first step is to form bonds between like spins (to grow clusters) then we flip to any configuration permitted by the current bond structure which is uniform. If you work through the calculations (GNB V pg 81), you will find that the probability of transitioning between states depend on the bonds that could have formed between like spins but didn't, and this is equal to the energy difference between the states.

Wolff

Another example is the Wolff algorithm (see GNB V pg 83) which is like the SW algorithm except that only a single cluster is build and flipped at a time with probability 1, i.e. the acceptance probability is 1 in the original algorithm. We start with any random site as the initial spin then we build a cluster spreading out from that spin to its nearest neighbors where the choose the probability of choosing a neighbor to be $1 - \exp(-2J_{ij})$.

The random fields can be accounted for in the acceptance probabilities (so the probability of a cluster flipping is no longer 1). Working through the math, you will find that the cluster growth accounts for the ratio of the energies that come from the couplings but to account for the fields, the probability of a particular orientation of the clusters has to be

$$p(\Sigma_n) \propto \exp\left(\sum_i h_i \sigma_i\right) \tag{58}$$

which we can easily simulate using something like the Metropolis algorithm. Obviously, this will increase the correlation time between samples because we will not always accept a cluster flip. This can be especially problematic when the clusters become large and have similar fields, but as long as the sum of the distribution of local fields for spins in a cluster is symmetric about 0, this will not be too slow. Certainly, it is faster than any local flip algorithm.

A simple example that I worked through is with 4 spins and the transition probability between two different states. Here, we can easily enumerate all possible ways of transitioning between these two states using the Wolff algorithm.

Replica exchange Monte Carlo (REMC)

REMC simulates multiple replicas of the system at different temperatures simulaneously and allow states to be exchanged between them. The idea is that an energy barrier may be very difficult to cross below a certain critical temperature, but there may be a trajectory allowed by diffusing through higher temperatures to cross the boundary.

Instead of considering the ensemble of a single system, we consider the ensemble of multiple independent systems. Thus, the joint probability distribution on the extended state space is

$$p(\sigma, \beta_n). \tag{59}$$

where $\beta_n = 1/T_n$ is the inverse temperature.

The simplest possible assumption for the shape of this distribution would be that the probability of occupying any particular temperature β_n is uniform $p(n) = 1/N_T$. It turns out that this is optimal according to some criterion (see citation in Kerler and Rehberg). Note that this is not the same as just simulating a set of replicas for which the partition funtion would be

$$Z = \prod_m \sum_\sigma \exp\left[-\beta_m H(\sigma)\right] \tag{60}$$

This is for the obvious reason that the average energy is a function of T so a simulation of Eq 60 would spend much more time exploring lower temperatures than higher temperatures. This makes it difficult to have the system use higher energy states to cross energy barriers.

So instead of Eq 60, we can include a term g_n to compensate

$$Z = \prod_m \sum_\sigma \exp\left[-\beta_m H(\sigma) + g_m\right] \tag{61}$$

How do we find g_n?

Under the assumption that the probability $p(n)$ is a constant,

$$p(n) = e^{g_n} \sum_\sigma \exp\left[-\beta_n H(\sigma)\right] / Z \tag{62}$$

$$g_n = \log[p(n)\, Z] + \tilde{Z}_n \tag{63}$$

$$g_n \to \tilde{Z}_n \tag{64}$$

(remembering that constants don't matter in the relation between energy and probability) where

$$\tilde{Z}_n = \sum_\sigma \exp\left[-\beta_n H(\sigma)\right] \tag{65}$$

One way to find the g_n that satisfy this condition is to come up with an iterative algorithm that converges to a fixed point. One suggestion is to estimate g_n by reweighting the distributions from adjacent replicas.

Now, it remains to find a set of β_n that is optimal for simulating relaxing into equilibrium quickly. One suggestion to make the algorithm spend an equal amount of time at each temperature. Kerler and Rehberg suggest a method.

[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.

[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*. 2006.

[9] F Reif. *Fundamentals of Statistical and Thermal Physics*. Waveland Press, 2009.

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

[11] 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.

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

[13] 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.

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

[15] S Cocco and R Monasson. Adaptive cluster expansion for the inverse Ising problem: convergence, algorithm and tests. *Journal of Statistical Physics*, pages 1–57, 2012.

[16] Bryan C. Daniels, David C. Krakauer, and Jessica C. Flack. Control of finite critical behaviour in a small-scale social system. *Nature Communications*, 8:14301, 2017.

[17] John Barton and Simona Cocco. Ising models for neural activity inferred via selective cluster expansion: structural and coding properties. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03):P03002, mar 2013.

[18] S. Cocco, R. Monasson, and V. Sessak. High-dimensional inference with the generalized Hopfield model: Principal component analysis and corrections. *Physical Review E*, 83(5), may 2011.

[19] 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 [17].