1 Finding good models

The notion of finding a good model—in our case, a good generative model of network structure—comes in two distinct flavors.

In the first, we specify a particular family of models $\mathcal{M}(\theta)$, where a particular choice of θ specifies which member of this family we are talking about. In this context, finding a good model is equivalent to searching over the various choices for θ to find those with large or maximal likelihood scores. For instance, with the stochastic block model, the maximum likelihood choice for the stochastic block matrix implies $\theta = z$, the partition vector, and finding a good model is equivalent to searching the space of all partitions for one or several that yields a good block structure on the edges.

There are two main approaches for searching over the parameter space: (i) optimization approaches and (ii) sampling approaches. Both approaches have advantages and disadvantages, and which you prefer depends on what you want. For instance, optimization is nearly always faster than sampling because it generally only produces a single choice of θ in the end, even if other choices are just or almost as good. Furthermore, optimization techniques are typically on guaranteed to converge on a local optimum; only if the likelihood function is convex will this outcome always coincide with the global optimum.[1] In contrast, sampling is nearly always slower than optimization because it considers many choices of θ, and furthermore is only guaranteed to find the global maximum in the infinite-time limit. The advantage of sampling techniques, like Markov chain Monte Carlo (MCMC) sampling or Gibbs sampling or parallel tempering, is that it allows us to compute distributions over good parameter choices, which can be a desirable outcome. We will return to this idea shortly.

In the second flavor of finding a good model, we have a set of distinct model families $\{\mathcal{M}_i(\theta_i)\}$, and we aim to choose a particular model that both fits the data well and fits the data better than the others. In this context, finding a good model requires comparing the models with each other (and potentially to a null alternative that all of the models are terrible). When models are "nested," i.e., when some $\theta_i \subset \theta_{j\neq i}$, this task can be complicated, because the "larger" model will always have a likelihood at least as large as that of the "smaller" model. Comparing non-nested models also presents some difficulties. In either case, we can in principle use a variety of model comparison techniques to sort among the models. We won't cover any of these techniques here.

1.1 Markov chain Monte Carlo (MCMC) sampling

Suppose we are given a particular network generative model $\Pr(G \,|\, \theta)$, and we now want to use this model to estimate the expected value of some quantity of interest that depends on the network structure G, e.g., the expected degree of any or some vertex, the expected propagation time

[1] Which is why so much effort is put into modifying likelihood functions so that they become convex. Convex optimization is a lot simpler than optimization on rugged "landscapes."

of a disease or meme spreading across the network, the expected centrality of some vertex, etc. Mathematically, each of these is computed as

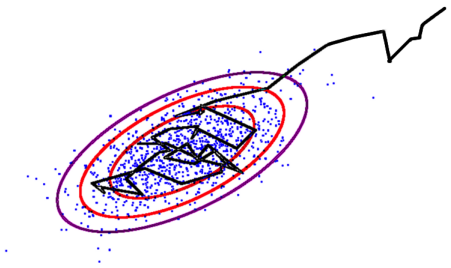
$$\langle x \rangle = \int_{\theta} x(G) \Pr(G \,|\, \theta) \, \mathrm{d}\theta \ , \tag{1}$$

where $x(G)$ represents the value of our quantity of interest on a particular graph G. That is, because $\Pr(G \,|\, \theta)$ defines a probability distribution over networks, the expected value for some quantity of interest may be computed by integrating that quantity over its probability distribution.[2]

When the generative model is simple, as in the case of a low-dimensional distribution like the normal or perhaps even the Erdős-Rényi random graph, this integral can often be carried out analytically. For more complicated models, however, the integral must be evaluated numerically. Numerical evaluation, however, presents a substantial problem for network generative models, as the parameter space θ is exponential in size. An elegant solution is to use Markov chain Monte Carlo sampling, or constrained versions like Gibbs sampling.

In MCMC, we construct a random walk on θ such that the limiting distribution of the walker's location converges on the target distribution, in this case $\Pr(G \,|\, \theta)$, the likelihood of G given the choice θ.[3] That is, each choice of θ will represent a unique state in a Markov chain, and we will define the transition probabilities between pairs of states s, t so that the random walker will visit each state s with a probability proportional to its likelihood $\Pr(G \,|\, \theta)$. Once the MCMC has converged on this distribution, we can carry out the integral in Eq. (1) directly.

For instance, the following figure shows this schematically, with the solid black line representing the trajectory of the random walk, and the concentric circles representing the quantiles of a 2-dimensional normal distribution over the data (blue dots).



[2] There are many good references on MCMC. Two good ones in particular are Newman and Barkema, *Monte Carlo Methods in Statistical Physics* (1999) and Walsh, "Markov Chain Monte Carlo and Gibbs Sampling" (2004).

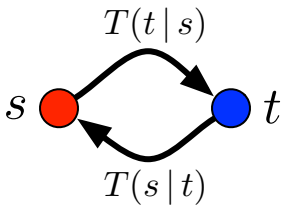
[3] The mathematics that allows us to do this is the same mathematics that allows us to calculating the limiting distribution of a random walker on a network, a la PageRank.

1.2 Constructing the MCMC

The key to constructing a Markov chain Monte Carlo sampler is in defining the transition probability $\Pr(s \to t)$ from some state s to some other state r. If the state space is discrete, then the set of these probabilities is sometimes called the *transition matrix* T, in which the s, t entry gives the probability $\Pr(s \to t)$. To guarantee that iterating T produces a random walker whose distribution converges on the target, T must satisfy two requirements[4] :

- *ergodicity*: every state s is reachable from every other state t with non-zero probability, and
- *detailed balance*: at equilibrium, the flux into a state s must equal the flux out of state s (which implies the absence of limit cycles in the state space).

Creating detailed balance.
Consider a pair of states s and t, and the corresponding transition probabilities between them:



The detailed balance requirement, that the flux into a state equal the flux out of the same state, implies the following constraint

$$p(s)T(t \mid s) = p(t)T(s \mid t) \ , \tag{2}$$

where $p(s)$ is the target distribution. (For now, we use the simpler notation $p(s)$ instead of $\Pr(G \mid \theta)$, but remember that they are the same.)

Without loss of generality, we can structure the random walker's movements in the following way. Given that the walker is at some state s, it will first *propose* a new state t with some probability and then independently choose to transition to that state with some other probability. Mathematically, we say

$$T(t \mid s) = g(t \mid s) \times a(t \mid s) \ , \tag{3}$$

where $g(t \mid s)$ is the probability that we propose t as the next state in the chain given that we are currently at s, and $a(t \mid s)$ is the probability that we accept this proposal. Substituting this

[4]These requirements go by other names in other fields, like irreducibility and aperiodicity. The names here are from physics.

expression into Eq. (2) yields

$$\frac{p(s)}{p(t)} = \frac{g(s\,|\,t) \times a(s\,|\,t)}{g(t\,|\,s) \times a(t\,|\,s)} \tag{4}$$

$$= \frac{a(s\,|\,t)}{a(t\,|\,s)} \ , \tag{5}$$

when we choose equal proposal probabilities, i.e., $g(s\,|\,t) = g(t\,|\,s)$. It thus only remains to choose the acceptance probabilities.

The Metropolis-Hastings choice.
If we choose the acceptance probabilities poorly, then the efficiency of the random walk will be low, i.e., it will sit at some state s rejecting many proposed transitions before continuing its walk. Thus, a reasonable choice for the acceptance probabilities is to maximize the forward progress of the chain, i.e., to accept every proposal if it improves the likelihood score. This choice is yields the Metropolis-Hastings algorithm after the physicist Nicholas Metropolis (1915–1999), who coauthored the original paper on MCMC in 1953, and W. K. Hastings (1930–) who extended its to more general cases in a 1973 paper.

For inference, where we prefer larger values of the likelihood,[5] implying that if $p(t) > p(s)$ then we want to maximize the probability that we accept this transition. No probability is greater than 1, thus, if the likelihood of the proposed state is better than our current state, we always accept the transition.

Using this choice, we can derive, from Eq. (5), an expression for the other acceptance probability, i.e, the probability that we accept a transition to a state with lower likelihood, i.e., $p(t) < p(s)$:

$$\begin{aligned} a(t\,|\,s) &= \frac{p(t)}{p(s)} \\ &= \mathrm{e}^{\ln p(t) - \ln p(s)} \\ &= \mathrm{e}^{\Delta \ln p} \ . \end{aligned} \tag{6}$$

Thus, the full Metropolis-Hastings acceptance probability for a proposal to move to state t from state s is

$$a(t\,|\,s) = \begin{cases} \mathrm{e}^{\Delta \ln \mathcal{L}} & \text{if } \mathcal{L}(t) < \mathcal{L}(s) \\ 1 & \text{otherwise.} \end{cases}$$

[5] That is, we like to maximize the likelihood. In physics, the aim is to minimize the energy of a system; the equations are identical except that the direction of the inequality in the condition is reversed.

1.3 Other sampling approaches

Sampling is a rich and active field. Below are some of the more common or more powerful techniques for estimating complicated distributions.

Gibbs sampling
This is a special or restricted version of MCMC in which θ is multi-dimensional. To perform Gibbs sampling, we iterate through each of the variables within θ. For a given choice θ_j, we hold the other variables $\theta_{i \neq j}$ fixed and allow the MCMC to reach equilibrium on θ_j. We then choose a different variable, hold the others fixed again, and allow the new variable to reach equilibrium.

This approach allows us to estimate the conditional or marginal distributions for each model variable independently, and in the long-time limit, the total distribution converges on the joint distribution, as in the MCMC described above. There are many tricks to make Gibbs sampling run faster, which you can find in the literature.

Importance sampling
In this approach, we have some other distribution $q(s)$ from which we can draw samples, which we can use to estimate the target distribution $p(s)$. In the end, we still want to compute an integral, and thus MCMC is often used when $q(s)$ is also of a complicated form.

Parallel tempering
This technique is a parallel form of MCMC, in which we have multiple walkers, each walking at a different "temperature." In the above analysis, there is no temperature parameter, but when MCMC is derived for simulating physical systems, such a parameter is included in Eq. (6) as a way to make accepting lower-quality states more or less likely than normal.

In parallel tempering, we create a set of MCMCs each of which has a different temperature parameter, and we then occasionally propose a move such that we swap the state variables of two of the MCMCs. This move can be made probabilistically rigorous so that the equilibrium distributions remain unchanged, but allowing it means that a particular walker can conduct a "tempered" walk, increasing and decreasing its preference for high-likelihood states.

2 At home

1. Read Chapter 11 (pages 523–555) in *Pattern Recognition*