The Evolution of Markov Chain Monte Carlo Methods

Matthew Richey

1. INTRODUCTION. There is an algorithm which is powerful, easy to implement, and so versatile it warrants the label "universal." It is flexible enough to solve otherwise intractable problems in physics, applied mathematics, computer science, and statistics. It works in both probabilistic and deterministic situations. Best of all, because it was inspired by Nature, it is blessed with extreme elegance.

This algorithm is actually a collection of related algorithms—Metropolis-Hastings, simulated annealing, and Gibbs sampling—together known as *Markov chain Monte Carlo* (MCMC) methods. The original MCMC method, the Metropolis algorithm, arose in physics, and now its most current variants are central to computational statistics. Along the way from physics to statistics the algorithm appeared in—and was transformed by—applied mathematics and computer science. Perhaps no other algorithm has been used in such a range of areas. Even before its wondrous utility had been revealed, its discovers knew they had found

> . . . a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. [**48**]

This is the story of the evolution of MCMC methods. It begins with a single paper, one with no antecedent. The original idea required the right combination of place, people, and perspective. The place was Los Alamos right after World War II. The people included the familiar—von Neumann, Ulam, Teller—along with several less familiar. The perspective was that randomness and sampling could be used to circumvent insurmountable analytic roadblocks. There was also one last necessary ingredient present: a computer.

The evolution of MCMC methods is marked by creative insights by individuals from seemingly disparate disciplines. At each important juncture, a definitive paper signaled an expansion of the algorithm into new territory. Our story will follow the chronological order of these papers.

1. *Equations of State Calculations by Fast Computing Machines*, 1953, by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller [**48**], which introduced the Metropolis algorithm.
2. *Optimization by Simulated Annealing*, 1983, by Kirkpatrick, Gelatt, and Vecchi [**45**], which brought simulated annealing to applied mathematics.
3. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, 1984, by Geman and Geman [**28**], which introduced Gibbs sampling.

doi:10.4169/000298910X485923

4. *Sampling-Based Approaches to Calculating Marginal Densities*, 1990, by Gelfand and Smith [**25**], which brought the power of MCMC methods to the statistics community.

It is difficult to overstate the impact of these papers and the importance of MCMC methods in modern applied mathematics. Collectively these four papers have been cited almost 40,000 times. The original Metropolis algorithm has been called one of the ten most important algorithms of the twentieth century [**19**].[1]

The goal here is not to provide a tutorial on how to use MCMC methods; there are many resources for this purpose [**8, 54, 27, 29, 10, 65, 61**]. Rather, the goal is to tell the story of how MCMC methods evolved from physics into applied mathematics and statistics. Parts of this story have already been told. Much has been written about the research at Los Alamos that led to the Metropolis algorithm; see, for example, the Los Alamos publications [**2, 46, 47, 32**]. A very nice overview of the connections between the Metropolis algorithm and modern statistics can be found in [**36, 10, 63**]. An unpublished work by Robert and Casella [**55**] also focuses on the history of MCMC methods, mostly from a statistician's perspective. Less has been said about the history of simulated annealing and the role of MCMC methods in image restoration.

2. THE BEGINNING: METROPOLIS ET AL., 1953.

Our story begins with the Metropolis algorithm, the original Markov chain Monte Carlo method. But first we take a brief look at the history of Monte Carlo methods and also recall some facts about Markov chains.

2.1. Monte Carlo Methods.

Shortly after World War II, Los Alamos was a hotbed of applied mathematics and theoretical physics. Much of the work was motivated by the intense focus on developing nuclear weapons. One particularly difficult problem was to estimate the behavior of large (e.g., 10^{23}) collections of atomic particles. The physical laws governing their behavior—thermodynamics, statistical physics, and quantum mechanics—were inherently probabilistic and so complicated that traditional methods were not sufficient for the sort of detailed analysis needed. In this setting a new idea took hold; instead of searching for closed form, analytic solutions, one could *simulate* the behavior of the system in order to estimate the desired solution. Producing simulations was a challenge. Before the late 1940s no device existed that could quickly and accurately carry out large-scale random simulations. By the end of World War II, things were different. Researchers at Los Alamos had access to such a device, the ENIAC (Electronic Numerical Integrator and Computer) at the University of Pennsylvania.

The use of probabilistic simulations predated the existence of a computer. The (perhaps apocryphal) case of the 18th century Buffon's needle is one example of how a "manual" simulation can be used to estimate a parameter of interest, in this case π. A more interesting, and better documented, example of simulation appears in Lord Kelvin's 1901 paper *Nineteenth Century Clouds Over the Dynamical Theory of Heat and Light*[2] [**43**], in which he described a method to estimate velocity distributions using simulated values obtained from a carefully constructed set of cards. There is also

[1] Others listed include the QR algorithm for eigenvalue calculation, the simplex method, quicksort, and the fast Fourier transform.

[2] Aside from describing Monte Carlo-like simulations, this paper had a significant role in the history of modern physics. Kelvin outlines the central problems of physics at the beginning of the twentieth century, namely the so-called "ultraviolet catastrophe" and the Michelson-Morely "anomaly" regarding the speed of light.

evidence that Enrico Fermi [**46, 58**] used manual Monte Carlo-like methods during the 1930s in his early work on nuclear fission.[3]

At Los Alamos credit for introducing probabilistic simulation goes to Stanislaw Ulam. As the story goes [**20, 64**], in 1946 Ulam was confined to bed to recover from an illness. To pass the time he played Canfield Solitaire, a form of solitaire for which the outcome is determined once the cards are shuffled and dealt. As he played, Ulam wondered how to determine the probability of winning a game. Clearly, this was an intractable calculation, but he imagined programming the ENIAC to simulate a random shuffle and then apply the rules of the game to determine the outcome. Repeating this a large number of times would give an empirical estimate of the probability of winning. Analyzing solitaire did not justify using Los Alamos's precious computing resources, but Ulam saw that this new approach could be used in realistic and important settings. He conveyed the idea to his friend John von Neumann.

In 1947, von Neumann and others were working on methods to estimate neutron diffusion and multiplication rates in fission devices (i.e., nuclear bombs) [**20**]. Following Ulam's suggestion, von Neumann proposed a simple plan: create a relatively large number of "virtual" neutrons and use the computer to randomly simulate their evolution through the fissionable material. When finished, count the number of neutrons remaining to estimate the desired rates. In modern terms, the scale was extremely modest: a simulation of just 100 neutrons with 100 collisions each required about five hours of computing time on the ENIAC. Nonetheless, the utility of this approach was immediately apparent. From this point forward, randomized simulations—soon to be called *Monte Carlo methods*—were an important technique in physics.

Apparently, Nicholas Metropolis was responsible for the name *Monte Carlo methods*.

> . . . I suggested an obvious name for the statistical method—a suggestion not unrelated to the fact that Stan (Ulam) had an uncle who would borrow money from relatives just because he had to go to Monte Carlo. The name seems to have endured. [**46**]

This new approach first appeared in Ulam and Metropolis's 1949 paper *The Monte Carlo Method* [**49**].

> We want to now point out that modern computing machines are extremely well suited to perform the procedures described. In practice, a set of values of parameters characterizing a particle is represented, for example, by a set of numbers punched on a card. . . . It may seem strange that the machine can simulate the production of a series of random numbers, but this is indeed possible. In fact, it suffices to produce a sequence of numbers between 0 and 1 which have a uniform distribution . . . [4]

[3] In 1955, fellow Nobel laureate Emilio Segrè recalled

> . . . Fermi acquired, by the combined means of empirical experience, Monte Carlo calculation, and more formal theory, that extraordinary feeling for the behavior of slow neutrons . . . [**58**]

[4] A standard approach of the era was the so-called "middle third" method. Let r_k be an n-digit random number. Square it and extract the middle n digits to form r_{k+1}. Linear congruential methods would be developed shortly thereafter by Lehmer and others.

From our perspective, it is perhaps difficult to appreciate the revolutionary nature of simulation as an alternative to analytical methods. But at the time, few mathematicians or physicists had any experience with the computer, much less simulation.

In addition to sampling from the uniform distribution, there soon emerged ways to sample from other probability distributions. For many of the standard distributions (e.g., the normal distribution), mathematical transformations of the uniform distribution sufficed. For more general distributions, in particular ones arising from physical models, more sophisticated techniques were needed.[5] One early method, also due to von Neumann, became what we now call *acceptance-rejection* sampling. These methods were far from universal and not well suited for higher-dimensional probability distributions. MCMC methods overcame these limitations. The key was the use of a Markov chain, which we now briefly review.

2.2. Markov Chains. Given a finite state (configuration) space $\mathbb{S} = \{1, 2, \ldots, N\}$, a *Markov chain* is a stochastic process defined by a sequence of random variables, $X_i \in \mathbb{S}$, for $i = 1, 2, \ldots$ such that

$$\text{Prob}(X_{k+1} = x_{k+1} \mid X_1 = x_1, \ldots, X_k = x_k) = \text{Prob}(X_{k+1} = x_{k+1} \mid X_k = x_k).$$

In other words, the probability of being in a particular state at the $(k + 1)$st step only depends on the state at the kth step. We only consider Markov chains for which this dependence is independent of k (that is, time-homogeneous Markov chains). This gives an $N \times N$ *transition matrix* $\mathbf{P} = (\mathbf{p}_{ij})$ defined by

$$\mathbf{p}_{ij} = \text{Prob}(X_{k+1} = j \mid X_k = i).$$

Note that for $i = 1, 2, \ldots, N$,

$$\sum_{j=1}^{N} \mathbf{p}_{ij} = 1.$$

The (i, j)-entry of the Kth power of $\mathbf{P}$ gives the probability of transitioning from state i to state j in K steps.

Two desirable properties of a Markov chain are:

- It is *irreducible*: for all states i and j, there exists K such that $(P^K)_{i,j} \neq 0$.
- It is *aperiodic*: for all states i and j, $\gcd\{K : (P^K)_{i,j} > 0\} = 1$.

An irreducible, aperiodic Markov chain must have a unique distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ on the state space $\mathbb{S}$ (π_i = the probability of state i) with the property that

$$\pi = \pi \mathbf{P}.$$

We say that the Markov chain is *stable on the distribution* π, or that π is the *stable distribution* for the Markov chain.

MCMC methods depend on the observation:

> If π is the stable distribution for an irreducible, aperiodic Markov chain, then *we can use the Markov chain to sample from* π.

[5] See [**15**] for a thorough overview of modern sampling methods.

To obtain a sample, select $s_1 \in \mathbb{S}$ arbitrarily. Then for any $k > 1$, if $s_{k-1} = i$, select $s_k = j$ with probability $\mathbf{p}_{ij}$. The resulting sequence $s_1, s_2, \ldots$ has the property that as $M \to \infty$,

$$\frac{|\{k : k \leq M \text{ and } s_k = j\}|}{M} \to \pi_j \tag{1}$$

with probability one.

Any large (but finite) portion of this sequence approximates a sample from π. Often, one discards the first m terms of the sequence, and uses the "tail" of the sequence

$$s_{m+1}, s_{m+2}, \ldots, s_M$$

as the sample.

However they are obtained, samples from π provide a way to approximate properties of π. For example, suppose f is any real-valued function on the state space $\mathbb{S}$ and we wish to approximate the expected value

$$E[f] = \sum_{i=1}^{N} f(i)\pi_i.$$

To do so, select a sample $s_1, s_2, \ldots, s_M$ from π and the ergodic theorem guarantees that

$$\frac{1}{M}\sum_{i=1}^{M} f(s_i) \to E[f] \tag{2}$$

as $M \to \infty$ with the convergence $O(M^{-1/2})$ [**34**].

Given the transition matrix for an irreducible, aperiodic Markov chain, it is a standard exercise to determine its stable distribution. We are keenly interested in the inverse problem:

> Given a distribution π on a finite state space, find an irreducible, aperiodic Markov chain which is stable on π.

The solution is the Metropolis algorithm.

2.3. Statistical Mechanics and the Boltzmann Distribution. The Metropolis algorithm was motivated by the desire to discern properties of the *Boltzmann distribution* from statistical mechanics, the branch of physics concerned with the average behavior of large systems of interacting particles. Let us briefly develop some of the fundamental ideas behind the Boltzmann distribution.

A state of the particles is described by a *configuration* ω taken from the *configuration space* Ω. The configuration space can be infinite or finite, continuous or discrete. For example, we might start with N interacting particles, each described by its position and velocity in three-dimensional space. In this case Ω is an infinite, continuous subset of $\mathbb{R}^{6N}$. Alternatively, Ω could be described by taking a bounded subset, Λ, of the integer lattice in the plane and to each site attaching a value, say ± 1. The value at a site might indicate the presence of a particle there, or it might indicate an orientation (or "spin") of a particle at the site. If $|\Lambda| = N$, then the configuration space consists of all 2^N possible assignments of values to sites in Λ.

The physics of a configuration space Ω is described by an *energy function* $E : \Omega \to \mathbb{R}^+$. We say that $E(\omega)$ is the energy of a configuration ω. For the continuous example above, energy could reflect the sum of gravitational potential energies. For the discrete example, the energy could reflect the total influence that neighboring particles exert on each other, as in the Ising model, which we will look at shortly.

A fundamental principle of statistical physics is that Nature seeks low-energy configurations. The random organization of molecules in a room is governed by this principle. Rarely observed configurations (e.g., all of the molecules gathering in a corner of the room) have high energies and hence very low probabilities. Common configurations (e.g., molecules isotropically distributed throughout the room) have low energies and much higher probabilities, high enough so that they are essentially the only configurations ever observed.

For a system at equilibrium, the relative frequency of a configuration ω is given by its *Boltzmann weight*,

$$e^{-E(\omega)/kT}, \tag{3}$$

where T is the temperature and k is Boltzmann's constant.

For any $\omega \in \Omega$, its *Boltzmann probability*, Boltz(ω), is

$$\text{Boltz}(\omega) = \frac{e^{-E(\omega)/kT}}{Z}. \tag{4}$$

The denominator

$$Z = \sum_{\omega' \in \Omega} e^{-E(\omega')/kT}$$

is called the *partition function*. In any realistic setting, the partition function is analytically and computationally intractable. This intractability single-handedly accounts for the dearth of analytic, closed-form results in statistical mechanics.

The relationship between energy and probability leads to expressions for many interesting physical quantities. For example, the total energy of the system, $\langle E \rangle$, is the expected value of the energy function $E(\omega)$ and is defined by

$$\langle E \rangle = \sum_{\omega \in \Omega} E(\omega)\text{Boltz}(\omega) = \frac{\sum_{\omega \in \Omega} E(\omega) e^{-E(\omega)/kT}}{Z}. \tag{5}$$

Many other physical quantities are defined similarly. In each case there is no avoiding the partition function Z.

Expressions such as (5) could be naively approximated using Monte Carlo sampling. To do so, generate a sample $\omega_1, \omega_2, \ldots, \omega_M$ *uniformly* from Ω, and estimate both the numerator and denominator of (5) separately, resulting in

$$\langle E \rangle \approx \frac{\sum_{i=1}^{M} E(\omega_i) e^{-E(\omega_i)/kT}}{\sum_{i=1}^{M} e^{-E(\omega_i)/kT}}.$$

Metropolis et al. understood the limitations of sampling uniformly from the configuration space and proposed an alternative approach.

> This method is not practical ... since with high probability we choose a configuration where $\exp(-E/kT)$ is very small; hence a configuration with very

> low weight. The method we employ is actually a modified Monte Carlo scheme where, instead of choosing configurations randomly, then weighting them with $\exp(-E/kT)$, we choose configurations with probability $\exp(-E/kT)$ and weight them evenly. [**48**]

In other words, it would be much better to sample from Ω so that ω is selected with probability $\mathrm{Boltz}(\omega)$. If this can be done, then for any such sample $\omega_1, \omega_2, \ldots, \omega_M$,

$$\frac{1}{M}\sum_{i=1}^{M} E(\omega_i) \to \langle E \rangle$$

with, as noted earlier, $O(M^{-1/2})$ convergence. The challenge is to sample from the Boltzmann distribution.

2.4. The Metropolis Algorithm. The genius of the Metropolis algorithm is that it creates an easily computed Markov chain which is stable on the Boltzmann distribution. Using this Markov chain, a sample from the Boltzmann distribution is easily obtained. The construction requires only the Boltzmann weights (3), not the full probabilities (4), hence avoiding the dreaded partition function. To appreciate the motivation for the Metropolis algorithm, let's recreate Metropolis et al.'s argument from their 1953 paper.

The setting for the Metropolis algorithm includes a large but finite configuration space Ω, an energy function E, and a fixed temperature T. Let $\tilde{\Omega}$ be any sample of configurations selected *with replacement* from Ω. It is possible, even desirable, to allow $\tilde{\Omega}$ to be larger than Ω. By adding and removing configurations, we want to modify $\tilde{\Omega}$ so that it becomes (approximately) a sample from the Boltzmann distribution. Suppose $|\tilde{\Omega}| = \tilde{N}$ and let N_ω denote the number of occurrences of ω in $\tilde{\Omega}$. To say that the sample perfectly reflects the Boltzmann distribution means

$$\frac{N_\omega}{\tilde{N}} \propto e^{-E(\omega)/kT},$$

or equivalently, for any two configurations ω and ω',

$$\frac{N_{\omega'}}{N_\omega} = \frac{e^{-E(\omega')/kT}}{e^{-E(\omega)/kT}} = e^{-\Delta E/kT}, \tag{6}$$

where $\Delta E = E(\omega') - E(\omega)$. Notice that this ratio does not depend on the partition function.

To get from an arbitrary distribution of energies to the desired Boltzmann distribution, imagine applying our yet-to-be-discovered Markov chain on Ω to all of the configurations in $\tilde{\Omega}$ simultaneously. Start by selecting a *proposal transition*: any irreducible, aperiodic Markov chain on Ω. Denote the probability of transitioning from a configuration ω to a configuration ω' by $P_{\omega,\omega'}$. As well, assume that the proposal transition is symmetric, that is, $P_{\omega,\omega'} = P_{\omega',\omega}$.

Consider configurations ω and ω' where $E(\omega) < E(\omega')$. Allow transitions from configurations with high energy $E(\omega')$ to configurations with low energy $E(\omega)$ whenever they are proposed; the number of times this occurs is simply

$$P_{\omega',\omega} N_{\omega'}.$$

By itself, this is just a randomized version of the steepest descent algorithm; any "downhill" transition is allowed.

In order to have any hope of reaching equilibrium, we must occasionally allow "uphill" transitions from configurations with low energy $E(\omega)$ to ones with high energy $E(\omega')$, that is, with some probability $\text{Prob}(\omega \to \omega')$. The number of times such a move is proposed is $P_{\omega,\omega'}N_{\omega}$ and hence the number of moves that actually occur is

$$P_{\omega,\omega'}N_{\omega}\text{Prob}(\omega \to \omega').$$

Since $P_{\omega,\omega'} = P_{\omega',\omega}$, the net flux between configurations with energy $E(\omega)$ and those with energy $E(\omega')$ is

$$P_{\omega,\omega'}\big[N_{\omega'} - N_{\omega}\text{Prob}(\omega \to \omega')\big]. \tag{7}$$

If (6) holds, that is, if the distribution of energies in $\tilde{\Omega}$ perfectly reflects the Boltzmann distribution, then the flux (7) should be zero. The result is what physicists call the *detailed balance*. This implies that the uphill probability must be

$$\text{Prob}(\omega \to \omega') = e^{-\Delta E/kT}.$$

This choice of occasional "uphill" transitions provides the magic of the Metropolis algorithm.

This process will also drive an arbitrary distribution of energies toward the Boltzmann distribution. Suppose there are too many configurations with high energy $E(\omega')$ relative to configurations with the low energy $E(\omega)$, that is,

$$\frac{N_{\omega'}}{N_{\omega}} > e^{-\Delta E/kT}.$$

In this case, the flux (7) is positive and there will be more transitions from configurations with energy $E(\omega)$ to those with energy $E(\omega')$ than in the other direction. Accordingly, the distribution of energies in $\tilde{\Omega}$ will move toward the Boltzmann distribution. Repeating this process a large number of times will produce a set of configurations whose distribution of energies approximates the Boltzmann distribution.

Based on this argument and physical intuition, Metropolis et al. were satisfied that their algorithm would produce samples from the Boltzmann distribution. More mathematically rigorous proofs of the convergence to the stable distribution would soon appear [**34, 35**]. Other important practical considerations, particularly understanding the rate of convergence, would have to wait longer.[6]

We now formally state the Metropolis algorithm. Assume a suitable proposal transition has been selected. For an arbitrary $\omega \in \Omega$ define the transition to a configuration ω^* as follows.

Step 1. Select ω' according to the proposal transition.
Step 2A. If $E(\omega') \leq E(\omega)$, or equivalently, $\text{Boltz}(\omega') \geq \text{Boltz}(\omega)$, let $\omega^* = \omega'$. In other words, always move to lower energy (higher probability) configurations.
Step 2B. If $E(\omega') > E(\omega)$, or equivalently, $\text{Boltz}(\omega') < \text{Boltz}(\omega)$, let $\omega^* = \omega'$ with probability

$$\frac{\text{Boltz}(\omega')}{\text{Boltz}(\omega)} = e^{-\Delta E/kT}. \tag{8}$$

Otherwise, $\omega^* = \omega$.

[6] Metropolis et al. knew the rate of convergence was an open question: "[Our] argument does not, of course, specify how rapidly the canonical distribution is approached." [**48**]

Several observations are in order:

- This process defines an irreducible, aperiodic Markov chain on the configuration space Ω.
- The ratio (8) is crucial to the computational utility of the Metropolis algorithm in that it avoids the intractable partition function.
- The steps in the chain are easily computable, or at least as easily computable as the proposal transition, $E(\omega)$, and, most importantly, $\Delta E = E(\omega') - E(\omega)$. In many settings, ΔE is extremely simple to compute; often it is independent of $|\Omega|$.
- The Markov chain defined by the Metropolis algorithm can be implemented without knowing the entire transition matrix.

The first application of the Metropolis algorithm in [**48**] was to analyze the so-called *hard spheres* model, a simple model of nonoverlapping molecules (e.g., a gas). Despite its apparent simplicity, the hard spheres model has proven to be a rich source of insight for statistical physicists. Using their new algorithm on 224 two-dimensional discs, Metropolis et al. allowed the system to evolve from an ordered state to a state close to equilibrium. The results were encouraging; the physical values they estimated agreed nicely with estimates obtained by traditional analytic methods. Best of all, the calculation times were reasonable. A single data point (of which there were hundreds) on a curve representing information about the hard spheres model only took about four or five hours of computing time on Los Alamos's MANIAC (Mathematical Analyzer, Numerator, Integrator, and Calculator).

2.5. The Metropolis Algorithm and the Ising Model. For a more illustrative application of the Metropolis algorithm, consider the two-dimensional *Ising model*. The Ising model has been extensively studied in both physics and mathematics. For more on the history and features of the Ising model, see [**6, 11**]. In addition to illustrating the effectiveness of the Metropolis algorithm, the Ising model plays a fundamental role in Geman and Geman's work on image reconstruction.

The Ising model can be thought of as a simple model of a ferromagnet in that it captures the tendency for neighboring sites to align with each other or with an external magnetic field. Formally, the two-dimensional Ising model is defined on a bounded planar lattice with N sites. At each lattice site, there is a "spin" represented by ± 1. A configuration is given by $\omega = (\omega_1, \omega_2, \ldots, \omega_N)$, where $\omega_i = \pm 1$ is the spin at the ith site; hence $|\Omega| = 2^N$. The energy of a configuration is defined as

$$E_{\text{ising}}(\omega) = -J \sum_{\langle i,j \rangle} \omega_i \omega_j - H \sum_{i=1}^{N} \omega_i \tag{9}$$

where $J > 0$ represents the nearest-neighbor affinity, $H > 0$ represents the external field, and $\langle i, j \rangle$ indicates that sites i and j are nearest neighbors, that is, sites that share either a horizontal or vertical bond. We will assume there is no external field (i.e., $H = 0$) and that $J = 1$.

One reason that the Ising model has long interested statistical physicists is that it exhibits a *phase transition*. Mathematically, a phase transition occurs when a quantity undergoes a dramatic change as a parameter passes through a *critical value*. The most familiar example of a phase transition occurs in water as it freezes or boils; in this case, the density of water changes dramatically as the temperature T passes through the critical value of $T_c = 0$ (or $T_c = 100$).

An important phase transition for the two-dimensional Ising model occurs in the *magnetization*. For a configuration ω, define

$$M(\omega) = \sum_{i=1}^{N} \omega_i.$$

The magnetization $\langle M \rangle_T$ at a temperature T is the expected value of $M(\omega)$:

$$\begin{aligned}\langle M \rangle_T &= \sum_{\omega \in \Omega} M(\omega)\mathrm{Boltz}(\omega) \\ &= \frac{1}{Z} \sum_{\omega \in \Omega} M(\omega) e^{-E_{\text{ising}}(\omega)/kT}. \end{aligned} \tag{10}$$

At high temperatures, states are essentially uniformly distributed and hence $\langle M \rangle_T$ is zero; in particular, there is almost no correlation between sites. However, as the temperature is lowered, *spontaneous magnetization* occurs: there is a critical temperature, T_c, below which sites influence each other at long ranges. One of the most celebrated results of statistical physics is Osager's exact calculation of the critical temperature for the two-dimensional Ising model:[7]

$$kT_c/J = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269.$$

Let's use the Metropolis algorithm to visualize the phase transition in the magnetization for an Ising lattice with N sites. To implement **Step 1**, we need a proposal transition process between configurations ω and ω'. A simple way to do this is to pick a lattice site i uniformly from $1, 2, \ldots, N$. At site i, with probability $1/2$, flip the spin ω_i to its opposite value; otherwise keep its current value. Notice that $\omega'_j = \omega_j$ for all $j \neq i$; that is, only the one site, ω_i, is affected. This proposal transition between configurations is irreducible, aperiodic, and symmetric.

For **Step 2**, we must decide whether to keep the proposed ω'. The important quantity is the change in energy:

$$\begin{aligned}\Delta E &= E_{\text{ising}}(\omega') - E_{\text{ising}}(\omega) \\ &= (\omega'_i - \omega_i) \sum_{\langle i,j \rangle} \omega_j, \end{aligned}$$

where the sum is over the four nearest neighbors of the ith site. Hence ΔE only depends on *the spins at the four sites neighboring the affected site* and therefore the computational cost of updating a site is both small and independent of the size of the lattice. This dependence on the local structure, the so-called *local characteristics*, is a recurring part of the Metropolis algorithm and MCMC methods in general.

Another recurring—but vexing—theme of MCMC methods is convergence. In general, it is extremely hard to determine how many iterations of the algorithm are required to reasonably approximate the target distribution. Also, an unavoidable feature of a Markov chain is sequential correlation between samples. This means it can take

[7] Surprisingly, there is no phase transition for the Ising model in one dimension. For a purely combinatorial argument for the existence of a phase transition for the two-dimensional Ising model, see [**44**].

a long time to traverse the configuration space, especially near the critical temperature where things are most interesting.[8] See Diaconis [**17**] for a survey of convergence results related to Ising-like models both near and far from the critical temperature.

2.6. An Application of the Metropolis Algorithm. Figure 1 shows two snapshots of a 200 × 200 Ising lattice; black indicates a spin of $+1$ and white a spin of -1. The lattice on the left is above the critical temperature for a phase transition, while the lattice on the right is below it. In each case, the Metropolis algorithm ran long enough so that the resulting sequence of states represented a sample from the Boltzmann distribution. On the left it is visually evident that there is little correlation of spin values of sites located some distance from each other. On the right there is a clear long-range correlation between spins. This qualitative difference reflects two distinct phases of the Ising model.

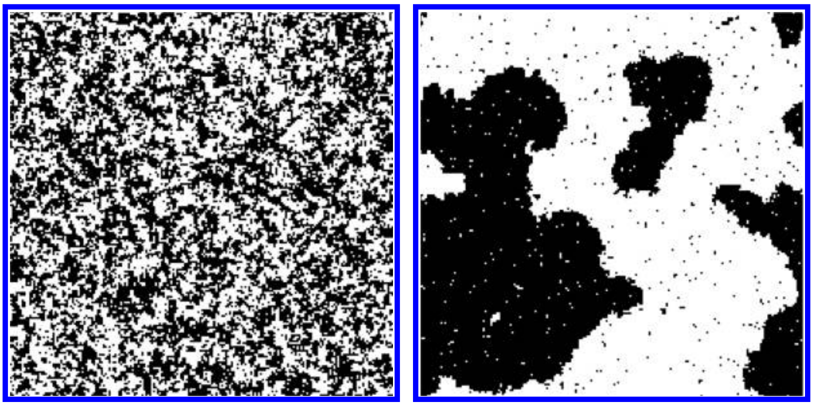

Figure 1. A 200 × 200 Ising model simulated using the Metropolis algorithm. The image on the left is above $kT_c/J \approx 2.269$ and exhibits very little long-range correlation between sites. The image on the right is below $kT_c/J \approx 2.269$ and shows a clear long-range correlation between sites.

2.7. Attribution. So who is responsible for the Metropolis algorithm? That there are five co-authors of [**48**] (including two husband-and-wife teams) makes any claim of "ownership" murky. Given the passage of time, the answer will probably never be known. Perhaps the final word on the subject belongs to Marshall Rosenbluth, the last survivor of the five co-authors, who commented on this question a few months before his death in 2003. At a conference celebrating the 50th anniversary of the Metropolis algorithm, he stated "Metropolis played no role in the development other than providing computer time" [**32**]. Rosenbluth went on to say that the idea to use Monte Carlo methods came from conversations with Ulam and von Neumann and that Teller made useful suggestions that led to replacing ordinary averages with averages weighted by the Boltzmann distribution. As well, he claimed that he and his wife and co-author, Arianna, did the important work.

There is evidence that essentially the same idea was independently proposed by Bernie Alder, Stan Frankel, S. G. Lewinson, and J. G. Kirkwood working at California

[8] The Swendsen-Wang algorithm [**60**] provides a significant, and elegant, solution to the second problem. This version of the Metropolis algorithm updates entire clusters of like spins. As a result, it traverses the configuration space much more rapidly, even at and below the critical temperature. The Swendsen-Wang algorithm is particularly interesting in that it introduces "bond" variables, an idea that appears in the image reconstruction work of Geman and Geman and also in Bayesian hierarchical models in statistics.

Institute of Technology and the Lawrence Livermore National Laboratories in the late 1940s. When asked, Alder is quite clear about their role.

> My guess is we did it first at Cal Tech. It's not that difficult to come up with that algorithm, which, by the way, I think is one of, if not THE, most powerful algorithms. . . . The fact is, we never published—you can't publish something your boss doesn't believe in! [**50**]

Support from their boss was not the only thing Alder and his colleagues lacked. They also did not have easy access to the tool most crucial to an implementation of the algorithm: a computer.

Interestingly, in 1947 Frankel and Metropolis had co-authored one of the first papers demonstrating the usefulness of the computer to perform numerical (not Monte Carlo) integration [**22**]. No doubt, the interplay between all these individuals during this era makes a strong argument that credit for what we now call the Metropolis algorithm should be shared among many.

2.8. Interlude. From the 1950s to the 1980s, most of the interest in the Metropolis algorithm came from the physics community. One exception was Hammersley and Handscomb's 1964 classic *Monte Carlo Methods* [**34**]. This delightful—and still relevant—monograph describes that era's state of the art of Monte Carlo methods, including a short survey of Markov chain Monte Carlo methods in statistical mechanics.

Physicists of this era were busy developing generalizations of the Metropolis algorithm, many of which were applied to spin models such as the Ising model. One of the first of these was the "heat bath" proposed by Glauber[9] in 1963 [**31**]. Glauber's algorithm moves through the lattice sites sequentially. At the ith site, the spin ω_i is assigned according to the local Boltzmann weight

$$\text{Prob}(\omega_i = s) = e^{-\left(s \sum_{\langle i,j \rangle} \omega_j\right)/kT},$$

where the sum, as usual, is over the nearest neighbor sites of i. Interestingly, Glauber's motivation was to understand analytical properties of the time-dependent (nonequilibrium) dynamics of spin models, not to develop a new computational tool. A decade later, Flinn [**21**] described a similar "spin-exchange" algorithm to computationally investigate phase transitions in the Ising model. Creutz in 1979 [**13**] showed how single-site updates could be applied to $SU(2)$ (special unitary group of degree 2) gauge theories.

Another generalization appeared in 1965 when A. A. Barker [**3**] introduced an alternative to the Metropolis construction, resulting in one more Markov chain with the Boltzmann distribution as its stable distribution. The existence of these variants raised the questions: How many Metropolis-like algorithms are there? Among all the variants, which one, if any, is best?

Answers to these questions appeared in the 1970s, starting with the work of the statistician W. K. Hastings. He was the first to see that the Metropolis algorithm was a (perhaps, *the*) general-purpose sampling algorithm. In an interview, Hastings recalled how he came across the algorithm.

[9] Glauber is also known for his contributions to the quantum theory of optical coherence, work for which he shared the 2005 Nobel Prize.

> [The chemists] were using Metropolis's method to estimate the mean energy of a system of particles in a defined potential field. With six coordinates per particle, a system of just 100 particles involved a dimension of 600. When I learned how easy it was to generate samples from high dimensional distributions using Markov chains, I realised how important this was for Statistics and I devoted all my time to this method and its variants which resulted in the 1970 paper. [**57**]

This 1970 paper was *Monte Carlo Sampling Methods using Markov Chains and Their Applications* [**35**] in which Hastings was able to distill the Metropolis algorithm down to its mathematical essentials. He also demonstrated how to use the Metropolis algorithm to generate random samples from a variety of standard probability distributions, as well as in other settings, such as from the group of orthogonal matrices. The importance of this paper was not immediately understood; recognition would have to wait for Gelfand and Smith's work twenty years later. In statistical circles, the Metropolis algorithm is now often referred to as the *Metropolis-Hastings algorithm.*

Let's briefly look at Hastings's generalization of the Metropolis algorithm. Given a distribution π from which we want to sample, select any Metropolis-like proposal transition, $\mathbf{Q} = (\mathbf{q}_{ij})$, on the state space $\mathbb{S}$. Unlike in the original Metropolis algorithm, *it does not need to be symmetric*. Define the transition matrix $\mathbf{P} = (\mathbf{p}_{ij})$ by

$$\mathbf{p}_{ij} = \begin{cases} \mathbf{q}_{ij}\alpha_{ij} & \text{if } i \neq j, \\ 1 - \sum_{k\neq i} \mathbf{p}_{ik} & \text{if } i = j, \end{cases} \tag{11}$$

where α_{ij} is given by

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i}{\pi_j}\frac{q_{ij}}{q_{ji}}}.$$

The values s_{ij} can be quite general, so long as (i) $s_{ij} = s_{ji}$ for all i, j and (ii) $\alpha_{ij} \in [0, 1]$. For any such choice of s_{ij}, it is easy to verify that π is the unique stable distribution for $\mathbf{P}$. For a symmetric $\mathbf{Q}$, a simple choice of s_{ij} recovers the original Metropolis algorithm.

For a given distribution π, different choices of the s_{ij} lead to qualitatively different Metropolis-like algorithms, all of which produce a Markov chain stable on π. Why does only the original Metropolis(-Hasting) algorithm live on? The reason was provided by Hastings's student, P. H. Peskun. Peskun [**52**] showed that among all choices of the s_{ij}, the variance of the estimate given in (2) is asymptotically minimal for the choice that leads to the Metropolis algorithm. Whether by luck or intuition, the first example of a Markov chain Monte Carlo method proved to be the best.

3. SIMULATED ANNEALING AND COMBINATORIAL OPTIMIZATION: KIRKPATRICK ET AL., 1983. Despite the popularity of the Metropolis algorithm in statistical physics and Hastings's observation of its potential as a general-purpose sampling tool, before 1980 the algorithm was little known in other circles. The situation changed with the appearance of *Optimization by Simulated Annealing* [**45**] by Scott Kirkpatrick, C. D. Gelatt, and M. P. Vecchi in 1983. At almost the same time V. Černý, a Czech applied mathematician, independently developed equivalent ideas in his 1985 paper *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm* [**9**]. Kirkpatrick et al.'s work is better known and rightfully so; they did more to develop the mathematics of annealing and applied it to a larger collection of problems. Although we will focus primarily on their work, Černý's paper is significant in its own right and deserves to be more widely read.

Kirkpatrick et al. were part of IBM's Thomas Watson Research Center. They were working on problems in *combinatorial optimization*, a type of deterministic problem for which the Metropolis algorithm was unexpectedly effective. Combinatorial optimization problems share two features:

1. An objective (cost) function for which a global minimum value is sought.
2. A discrete (often finite, but large) search space in which one looks for the global minimum. In practice, approximations to the global minimum are the best one can expect.

A standard example of a combinatorial optimization problem is the *traveling salesman problem* (TSP) where the goal is to minimize the distance of a tour through a set of vertices. The search space consists of possible tours and the objective function is the total distance of a tour. Like many combinatorial optimization problems, the TSP (when recast as a decision problem) is NP-complete.

Kirkpatrick and the other authors were trained as statistical physicists, so it was natural for them to think of the objective function as an energy function. Knowing that Nature seeks low energy configurations, they considered ways to use the Metropolis algorithm to select low energy configurations from the search space. The challenge, they discovered, was to find a way to properly utilize temperature T, a quantity for which there is no natural analog in a combinatorial optimization setting. For large values of T, the Metropolis algorithm produced an essentially uniform distribution, hence was nothing more than a random search. For small values of T, the Metropolis algorithm was susceptible to becoming trapped near local minima far removed from the desired global minimum. An understanding of how to properly utilize T required insights from statistical mechanics. We will construct Kirkpatrick et al.'s original argument to see how this is done.

3.1. Circuit Design and Spin Glasses. The motivating question for Kirkpatrick et al. was not the TSP, but how to place circuits (i.e., transistors) on computer chips efficiently. Circuits on the same chip communicate easily, but there is a substantial communication penalty for signals connecting circuits on different chips. The goal is to place the circuits in a way that minimizes the total communication cost with the constraint that there must be a (roughly) equal number of circuits on each chip.

To formulate this problem mathematically, suppose there are N circuits that must be placed on two separate chips. A configuration ω is given by the N-tuple

$$\omega = (\omega_1, \omega_2, \ldots, \omega_N),$$

where $\omega_i = \pm 1$ indicates the chip on which the ith circuit is placed. The value a_{ij} indicates the number of signals (connections) between circuits i and j.

Following Kirkpatrick et al., represent the between-chip communication cost as

$$\sum_{i>j} \frac{a_{ij}}{4}(\omega_i - \omega_j)^2. \tag{12}$$

The cost of an imbalance between the number of circuits on each of the two chips can be expressed as

$$\lambda \left(\sum_i \omega_i \right)^2, \tag{13}$$

where $\lambda > 0$ is the imbalance "penalty."

Expanding (12), combining it with (13), and dropping all constant terms results in an objective function

$$C(\omega) = \sum_{i>j} \left(\lambda - \frac{a_{ij}}{2} \right) \omega_i \omega_j. \tag{14}$$

By the early 1980s, researchers at IBM had developed various algorithms to (approximately) minimize $C(\omega)$. As the number of transistors N grew from several hundred to thousands (and beyond), these methods were proving less viable. As Kirkpatrick recalls,

> Previous methods were arcane, if you looked at them carefully they involved solving for conflicting objectives one after another. We knew we could do better than that. (Scott Kirpatrick, personal communication)

Fortunately, Kirkpatrick et al. knew of a model in statistical mechanics whose energy function bore a striking resemblance to (14). This model is called a *spin glass*.

Spin glasses are much like the Ising model but with a slightly different energy function

$$E(\omega) = \sum_{i>j} \left(U - U_{ij} \right) \omega_i \omega_j.$$

The analogy to (14) is immediate. The values of U_{ij} represent local attractive (ferromagnetic) forces between neighboring states. These are in competition with long-range repulsive (anti-ferromagnetic) interactions represented by U. Spin glasses are called *frustrated* because they cannot have configurations which simultaneously satisfy both the attractive and repulsive requirements. As a result, the low energy ground states do not have extended regions of pure symmetry.

For spin glasses and other frustrated systems, Kirkpatrick knew that the Metropolis algorithm had to be carefully applied in order to identify low-temperature ground states. If the system is *quenched*, that is, the temperature is lowered too quickly, then it can settle into a state other than a ground state. A better approach is to *anneal*, that is, to slowly lower the temperature so the system can evolve gently to a ground state. This observation led Kirkpatrick et al. to *simulated annealing*.

> Using the cost function in place of the energy and defining configurations by a set of parameters $\{x_{ij}\}$, it is straightforward with the Metropolis procedure to generate a population of configurations of a given optimization problem at some effective temperature. This temperature is simply a control parameter in the same units as the cost function. The simulated annealing process consists of first "melting" the system being optimized at a high effective temperature, then lowering the temperature by slow stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach steady state. The sequence of temperatures and number of rearrangements of the $\{x_{ij}\}$ attempted to reach equilibrium at each temperature can be considered an annealing schedule. [**45**]

Kirkpatrick et al. applied this new technique to several realistic problems in circuit design, along with a demonstration of its effectiveness on the TSP. The results were impressive—clearly simulated annealing worked. As well, around the same time Černý produced similar results applying his version of simulated annealing to the TSP.

3.2. After Kirkpatrick et al. Since 1983 simulated annealing has become a standard technique in the applied mathematician's toolbox. The range of problems to which it has been applied is staggering. It works, to some degree, in both discrete and continuous settings. It has been used in almost every area of applied mathematics, including operations research, biology, economics, and electrical engineering. Combinatorial optimization is replete with algorithms that solve particular problems—or even special cases of particular problems—quite well. However, most are customized to fit the particular nuances of the problem at hand. Simulated annealing's popularity is due to a combination of its effectiveness and ease of implementation: given an objective function and a proposal transition, one can almost always apply simulated annealing.

Perhaps because of the lack of a strong mathematical framework, it took some time for simulated annealing to become accepted in the applied mathematics community. The first thorough empirical analysis of simulated annealing appeared in 1989 in a series of papers by Johnson, Aragon, McGeoch, and Schevon [**41, 42**]. See [**66, 65**] for an excellent discussion of some of the theoretical issues, a survey of the applications of simulated annealing (especially of the type considered by Kirkpatrick et al.), and more analysis of its performance relative to other algorithms. For an accessible description, along with a simple example of an application of simulated annealing, see [**1**].

The computational efficiency of simulated annealing depends on the relationship between the proposal transition and the energy function. A good proposal transition changes the energy function as little as possible, that is, ΔE is easily computed, often in a manner that is independent of the problem size. In the original circuit design problem the proposal transition consists of randomly selecting a circuit and moving it to the other chip. The cost of computing the change in energy is therefore independent of the problem size. The advantage of these efficient, local changes was demonstrated in the work of Geman and Geman, who used ideas from both the Metropolis algorithm and simulated annealing to attack problems in digital image reconstruction.

3.3. An Application of Simulated Annealing. The importance of local changes can be seen in an application of simulated annealing to the traveling salesman problem. In this setting a configuration ω is a tour of the n vertices and is specified by a permutation of $(1, 2, \dots, n)$.

A simple proposal transition is defined by randomly selecting two vertices $1 \le i < j \le n$ and reversing the direction of the path between them. This means if

$$\omega = (a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{j-1}, a_j, a_{j+1}, \dots, a_n)$$

then

$$\omega' = (a_1, \dots, a_{i-1}, a_j, a_{j-1}, \dots, a_{i+1}, a_i, a_{j+1}, \dots, a_n).$$

The change in distance (energy) is easily computed:

$$\Delta E = (|a_{i-1} - a_j| + |a_i - a_{j+1}|) - (|a_{i-1} - a_i| + |a_j - a_{j+1}|).$$

Figure 2 illustrates this implementation of simulated annealing on a TSP graph consisting of 500 vertices which were randomly placed in the unit square.

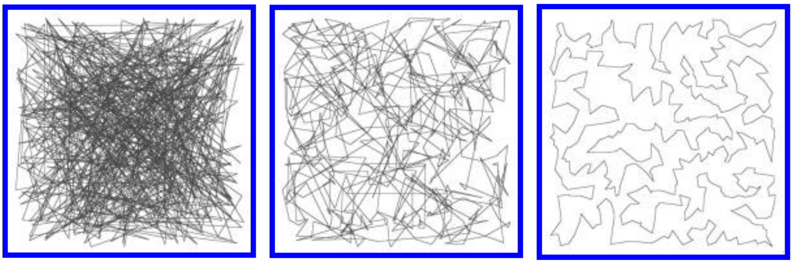

Figure 2. Simulated annealing applied to a traveling salesman problem with 500 randomly placed vertices. The figure on the left is the initial ($T = 4.0$) configuration with total distance of approximately 253. The middle configuration is after several hundred iterations ($T = 0.04$) and has a total length of about 50. The configuration on the right is near the global minimum of 17 ($T = 0.002$).

4. GIBBS SAMPLING AND A BAYESIAN PERSPECTIVE: GEMAN AND GEMAN, 1984.

The next chapter in our story takes place in 1984 when the brothers Donald and Stuart Geman, in *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images* [**28**], demonstrated that a variant of the Metropolis algorithm could be applied to problems in digital image restoration. This paper introduced a new MCMC method, *Gibbs sampling*.

4.1. Bayesian Digital Images.

A simple model of a digital image consists of pixel elements arranged on a rectangular lattice with N sites. Each pixel takes on a value from a set $\mathcal{S} = \{1, 2, \dots, K\}$ of *intensity levels* (e.g., grayscale or color levels). A configuration (image) $\omega \in \Omega$ is an assignment of an intensity level to each of the N sites, that is, $\omega_i \in \mathcal{S}$ for $i = 1, 2, \dots, N$. Even modestly sized images result in immensely large configuration spaces; for a 100×100 binary image, $|\Omega| = 2^{10000}$.

Images can be degraded in many different ways. The model Geman and Geman considered consisted of a combination of blurring, nonlinear transformations, and noise. We will focus on only additive noise, which can be modeled with N independently and identically distributed random values $\mathcal{N} = \{\eta_1, \eta_2, \dots, \eta_N\}$. “White noise” is common; in this case the η_i are normally distributed with mean 0 and variance σ^2 (i.e., $\eta_i \sim N(0, \sigma^2)$). Letting ω^{blurred} indicate the degraded (noisy) image, we have

$$\omega^{\text{blurred}} = \omega + \mathcal{N}.$$

Note that the values $\omega_i^{\text{blurred}}$ are real numbers; the resulting image is determined by rounding each value to the nearest value in $\mathcal{S}$. The two images in Figure 3 show a two-color 200×200 image and a version degraded by the addition of $N(0, 1.5^2)$ white noise.

The relationship between the original image and its degraded version is inherently probabilistic; given any ω, there is some probability that a particular ω^{blurred} is the degraded version of ω. Image reconstruction looks at the problem the other way around; given ω^{blurred}, there is some probability that ω is the original image. This leads to an application of Bayes' rule[10] and the formulation of the so-called *posterior distribution* for ω conditioned on ω^{blurred}:

$$\text{Prob}(\omega \mid \omega^{\text{blurred}}) = \frac{\text{Prob}(\omega^{\text{blurred}} \mid \omega)\text{Prob}(\omega)}{\text{Prob}(\omega^{\text{blurred}})}. \tag{15}$$

[10] We will interpret probability functions and probability density functions interchangeably. Hopefully, this will not cause confusion.

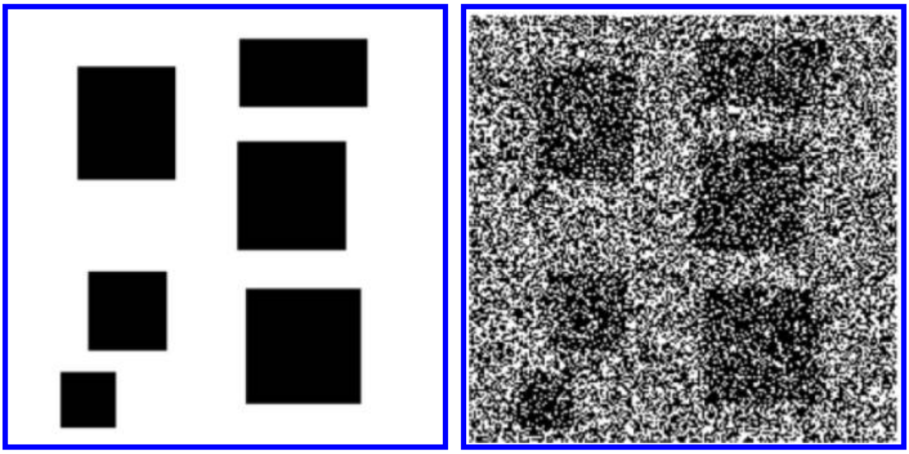

Figure 3. A two-color 200×200 image, ω, and a degraded version, ω^{blurred}, obtained by the addition of $N(0, 1.5^2)$ noise.

The goal is to find the configuration ω which maximizes $\text{Prob}(\omega \mid \omega^{\text{blurred}})$, often called the *maximum a posteriori estimate*.

Despite the mathematical elegance of the Bayesian formalism, finding the optimal $\omega \in \Omega$ is an extremely challenging computational problem, reminiscent of difficulties encountered in statistical physics and combinatorial optimization. Geman and Geman's response was to formulate a new version of the Metropolis algorithm—Gibbs sampling.

To understand Gibbs sampling and its connection to statistical mechanics, let's look at how Geman and Geman constructed the posterior distribution (15). Consider the three probabilities on the right-hand side of (15).

- $\text{Prob}(\omega^{\text{blurred}} \mid \omega)$: the *likelihood* function.
- $\text{Prob}(\omega)$: the so-called *prior* distribution for ω.
- $\text{Prob}(\omega^{\text{blurred}})$: the denominator.

The denominator is given by

$$\text{Prob}(\omega^{\text{blurred}}) = \int_{\omega \in \Omega} \text{Prob}(\omega^{\text{blurred}} \mid \omega) \text{Prob}(\omega)\, d\omega,$$

where the integral (or sum) is over all values of $\omega \in \Omega$ and hence is independent of ω. Remembering the partition function and what we know about the Metropolis algorithm (and simulated annealing), it is not surprising that we can safely ignore it.

The likelihood function, $\text{Prob}(\omega^{\text{blurred}} \mid \omega)$, is easily handled. Since

$$\omega_i^{\text{blurred}} = \omega_i + \eta_i,$$

where $\eta_i \sim N(0, \sigma^2)$, it follows that

$$\text{Prob}(\omega^{\text{blurred}} \mid \omega) \propto \prod_{i=1}^{N} e^{-\frac{(\omega_i^{\text{blurred}} - \omega_i)^2}{2\sigma^2}} = e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2},$$

where any constant of proportionality will be absorbed into the denominator above.

Determining Prob(ω), the prior distribution, is a delicate issue and the heart of the Bayesian approach. Probabilistically it asks: *What is an image?* Any collection of pixel values could be an image, but some are more likely to be interpreted as images than others. Broadly speaking, images have patterns, i.e., contiguous regions of similar pixel values. On the other hand, if neighboring values are uncorrelated, then the result is the visual equivalent of white noise. This brings to mind the Ising model. Consider, for example, the Ising lattice at equilibrium in Figure 1. The image on the left is too "noisy" to be considered a prototype of an image. In contrast, the image on the right has image-like features, namely a high degree of long-range correlation between pixels.

This observation suggested to Geman and Geman that the Boltzmann probability (4), using the Ising model energy function (9), could serve as the prior distribution on images, that is,

$$\mathrm{Prob}(\omega) \propto e^{-E_{\mathrm{ising}}(\omega)/kT}.$$

To retain the idea of correlated pixel values, they let $kT/J = 1 < T_c$, the critical temperature below which a phase transition occurs.

Putting all the parts of (15) together, the posterior distribution, $\mathrm{Prob}(\omega \mid \omega^{\mathrm{blurred}})$, can be written as

$$\begin{aligned}\mathrm{Prob}(\omega \mid \omega^{\mathrm{blurred}}) &\propto e^{\frac{1}{2\sigma^2}\sum_{i=1}^{N}(\omega_i^{\mathrm{blurred}}-\omega_i)^2}\, e^{-E_{\mathrm{ising}}(\omega)} \\ &\propto e^{-\left[\frac{1}{2\sigma^2}\sum_{i=1}^{N}(\omega_i^{\mathrm{blurred}}-\omega_i)^2+E_{\mathrm{ising}}(\omega)\right]}.\end{aligned} \tag{16}$$

Viewing (16) from a statistical mechanics perspective leads to an analog of an energy function

$$\begin{aligned}E_{\mathrm{image}}(\omega \mid \omega^{\mathrm{blurred}}) &= \frac{1}{2\sigma^2}\sum_{i=1}^{N}(\omega_i^{\mathrm{blurred}} - \omega_i)^2 + E_{\mathrm{ising}}(\omega) \\ &= \frac{1}{2\sigma^2}\sum_{i=1}^{N}(\omega_i^{\mathrm{blurred}} - \omega_i)^2 + \sum_{\langle i,j\rangle}\omega_i\omega_j,\end{aligned} \tag{17}$$

where $\langle i, j\rangle$ indicates that sites i and j are nearest neighbors.

Finding the most probable original image ω given $\omega^{\mathrm{blurred}}$ is thus equivalent to minimizing $E_{\mathrm{image}}(\omega \mid \omega^{\mathrm{blurred}})$. The first term of (17) is a penalty for straying too far from the data, $\omega^{\mathrm{blurred}}$, while the second term represents the desire to align neighboring pixel values, that is, making them conform to the prior notion of a generic image. The optimal solution balances the tension between these two conflicting constraints.

It is interesting to compare the approach of Kirpatrick et al. to that of Geman and Geman, both of whom borrowed ideas from statistical mechanics. Kirpatrick et al. started with the objective function and interpreted it as energy, eventually using the physicist's notion of probability. Geman and Geman started with a probabilistic situation and introduced a Bayesian structure, eventually leading to an energy function.

4.2. Gibbs Sampling. Gibbs sampling is Geman and Geman's version of the Metropolis algorithm. For an image site ω_i, we identify its *neighborhood system*, namely, its nearest neighbors. The probability of ω_i conditioned on all the other sites depends on only the sites in the neighborhood system. Suppressing the conditional dependence on

ω^{blurred}, this means that

$$\begin{aligned}\text{Prob}(\omega_i \mid \omega_j \text{ such that } j \neq i) &= \text{Prob}(\omega_i \mid \omega_j \text{ such that } \langle i, j\rangle)\\ &\propto e^{-E_i(\omega_i|\omega_j \text{ such that } \langle i,j\rangle)}\\ E_i(\omega_i \mid \omega_j \text{ such that } \langle i, j\rangle) &= \frac{1}{2\sigma^2}(\omega_i^{\text{blurred}} - \omega_i)^2 + \sum_{\langle i,j\rangle} \omega_i\omega_j.\end{aligned}$$

In general, a probability distribution whose conditional probabilities depend on only the values in a neighborhood system is called a *Gibbs distribution* and is part of a structure called a *Markov random field*, a notion introduced by the Russian statistical physicist R. L. Dobrushin [**18**]. The fact that the probability of the state of site i conditioned on all the other sites depends on only the sites in a small neighborhood is crucial to the computational efficiency of Gibbs sampling.

To implement Gibbs sampling, use any method that guarantees all sites are visited infinitely often. For example, a sequence of raster scans (in order, by rows and columns) of the sites will suffice. At a selected site i, select $\omega_i = k$ with probability

$$\text{Prob}(\omega_i = k) \propto e^{-\frac{1}{2\sigma^2}(k-\omega_i^{\text{blurred}})^2 - k\sum_{\langle i,j\rangle}\omega_j}. \tag{18}$$

This is Gibbs sampling. Repeating this for a large number of raster scans will result in a sequence of images that approximates a sample from the posterior distribution (16).

This method seems to differ from the Metropolis(-Hastings) algorithm in that there is no proposal transition. Actually, Gibbs sampling fits nicely into the Hastings generalization of the Metropolis algorithm where the proposal transition probabilities are given by (18). In the second step of the Metropolis-Hastings algorithm, the probabilities α_{ij} of (11) are all equal to one. Therefore, Gibbs sampling will produce a sequence representing a sample from $\text{Prob}(\omega \mid \omega^{\text{blurred}})$.[11]

Bayesian formulations of image degradation predate Geman and Geman's work. For example, in 1972 Richardson [**53**] used Bayes' rule to define an alternative to the standard Fourier transform methods of image reconstruction. His approach foreshadowed the iterative ideas that Geman and Geman developed more fully. Around the same time others [**38, 33, 51**] used the Bayesian perspective in a variety of ways. As well, earlier we noted that by 1980 Gibbs-like algorithms (often called "spin-flip" algorithms) were a standard tool in statistical mechanics. Around this time, the Metropolis algorithm was being used to generate digital textures [**14**].

Geman and Geman's insight was to merge a Bayesian formulation—which provided a richer model for describing the relationship between the original and the degraded image—with the power of the Metropolis algorithm. The key to the computational efficiency was the local neighborhood system, that is, the local characteristics. This meant that calculation of ΔE was independent of the image size.

> . . . the computational problem is overcome by exploiting the pivotal observation that the posterior distribution is again Gibbsian with approximately the same local neighborhood system as the original image . . . [**28**]

There is a second part to Geman and Geman's Gibbs sampling that has been mostly lost to history. They included a "temperature" parameter T and used a simple form of

[11] Because the transitions are no longer time-independent (as they depend on the site choice), the proof is somewhat more involved than the straightforward original algebraic treatments given by Hastings and others. See [**34, 25, 27, 8**] for proofs of the convergence of this form of Gibbs sampling.

simulated annealing.[12] The resulting probability distribution is

$$\text{Prob}(\omega \mid \omega^{\text{blurred}}) \propto e^{-E(\omega|\omega^{\text{blurred}})/T}.$$

Geman and Geman's annealing schedule used just one Gibbs sampling step at each value of T. They not only rigorously demonstrated that this algorithm converges to the maximum of the posterior distribution (15) as $T \to 0$, but they also provided the first quantitative results concerning the rate of convergence of an MCMC method.

Theorem 4.1 (Geman and Geman [28]). *Consider an image with N pixels. Let T_k be any decreasing sequence of temperatures such that*

- $T_k \to 0$ *as* $k \to \infty$.
- $T_k \geq N\Delta/\ln k$ *for all sufficiently large* k *and constant* Δ.

Then, starting at $\omega^0 = \omega^{\text{blurred}}$ the Gibbs sampling sequence ω^k for $k = 0, 1, \ldots$ converges in distribution to the distribution which is uniform on the minimum values of $E_{\text{image}}(\omega)$ and zero elsewhere.

In other words, following the prescribed annealing schedule, Gibbs sampling must, in theory, produce a *maximum a posterori estimate* of $\text{Prob}(\omega \mid \omega^{\text{blurred}})$.

Even though this result guarantees convergence to the most likely image, the rate of convergence is excruciatingly slow. For a 100 × 100 lattice ($N = 10^4$ sites), using the theorem requires e^{20000} steps to go from $T = 4$ to $T = 0.5$. In practice, Geman and Geman found that 300–1000 raster scans would produce acceptable results.

An application of Gibbs sampling is seen in the two-color images shown in Figure 4. For a two-color image Gibbs sampling (with annealing) is straightforward to implement. At the pixel ω_i, define

$$E^k = \frac{1}{2\sigma^2}(k - \omega_i^{\text{blurred}})^2 + k\sum_{\langle i,j\rangle}\omega_j. \tag{19}$$

Set $\omega_i = k$ with probability

$$\frac{e^{-E^k/T}}{e^{-E^0/kT} + e^{-E^1/kT}}.$$

In Figure 4, on the left is the original two-color 200 × 200 image. The center image is the result of adding $N(0, 1.5^2)$ noise. The rightmost image is the result of applying Gibbs sampling with the annealing schedule defined by $T_k = 3/\ln(1 + k)$ for $k = 1$ to $k = 300$. There was one complete raster scan of the image for each temperature in the annealing schedule.

Despite the discrepancy between the theory and practice of convergence rates, Gibbs sampling had arrived. Its effectiveness and ease of implementation within a solid theoretical framework certainly hastened the acceptance of MCMC methods into other areas of applied mathematics, especially computational statistics.

Geman and Geman also considered models with edges between contiguous regions of the same color. As opposed to "observable" pixels, edges are "unobservable" quantities. These sorts of models appealed to Bayesian statisticians because they are related to *hierarchical models*, an idea we will describe in more depth in the next section.

[12] Geman and Geman were aware of the idea of simulated annealing independently of the work of Kirkpatrick et al. (Stuart Geman, personal communication).

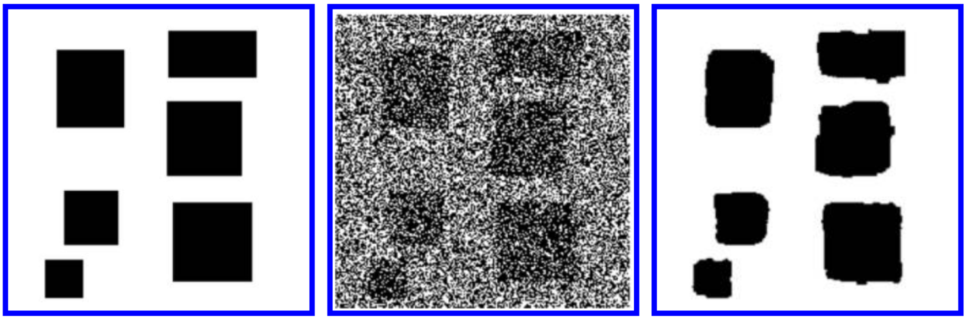

Figure 4. Image restoration of a two-color 200 × 200 image using Gibbs sampling on an image degraded with $N(0, 1.5^2)$ noise. The leftmost image is the original, the rightmost image is the restoration.

5. MCMC METHODS ENTER STATISTICS: GELFAND AND SMITH, 1990.

The final chapter of our story takes place in 1990 with the appearance of Alan Gelfand and Adrian Smith's *Sampling-Based Approaches to Calculating Marginal Densities* [**25**], which heralded the arrival of MCMC methods in statistics. Ostensibly, Gelfand and Smith's goal was to compare three different sampling-based approaches: Gibbs sampling, Tanner and Wong's data augmentation,[13] and Rubin's importance-sampling algorithm (originally proposed in the discussion of Tanner and Wong's work) [**62**]. Even though in 1970, Hastings saw that the Metropolis algorithm was a general purpose sampling tool, it wasn't until two decades later that Gelfand and Smith convinced the statistics community of the power of MCMC methods.

5.1. From Gibbs to Gelfand.

The emergence of Gibbs sampling in the statistics community can be traced to a single conference—the 1986 meeting of the Royal Statistical Society. At this meeting, Julian Besag presented the suggestively entitled paper *On the Statistical Analysis of Dirty Pictures* [**4**], in which he discussed the state of the art in the reconstruction of degraded images (i.e., of dirty pictures) to an audience of leading statisticians, including many leading Bayesians, along with the Geman brothers. He described not only the virtues of Gibbs sampling, but also how it compared to other techniques for image reconstruction. Interestingly, Besag appreciated Gibbs sampling for its simplicity and effectiveness, but was critical of its computational demands.

In the vigorous discussion that followed the paper, it was clear that a new era in computational statistics was about to begin. As one participant said, " . . . we are offered no alternative, as statisticians, to the route of the vast computing power being pioneered by the Gemans" [**4**]. Stuart Geman made a particularly poignant comment about the universality of Gibbs sampling, one reminiscent of Metropolis et al.'s earlier comment about the potential universality of their original algorithm.

> We are able to apply a single computer program to every new problem by merely changing the subroutine that computes the energy function in the Gibbs representation of the posterior distribution. [**4**]

Around this time, personal computing was becoming available and the effects of Moore's law on computational speed and memory were apparent. Besag's concerns about the computational demands of Gibbs sampling would soon be swept aside.

[13] Tanner and Wong's work also played an important role in the introduction of Gibbs sampling to the statistics community. In [**62**] they identified Gibbs sampling as a means of data augmentation but failed to realize the full potential of its use as a way of generating samples from arbitrary probability distributions.

Although not at the 1986 meeting, Gelfand and Smith soon became aware of Geman and Geman's work. They saw that Gibbs sampling could be "mapped" onto many probability distributions common to statistical models, especially those arising from a Bayesian approach. Their work was mostly a rearticulation of Geman and Geman's Gibbs sampling, although without annealing. The importance of their paper was how clearly they demonstrated the effectiveness of Gibbs sampling as a statistical technique.

Gelfand and Smith's version of Gibbs sampling in statistics can be described as follows. Start with a joint probability distribution $f(x_1, x_2, \dots, x_N)$ in which the variables represent parameters of a statistical model. The goal is to obtain (point and interval) estimates for these parameters.

To fit this into the Gibbs sampling framework, assume that all the single-variable conditional probability densities

$$f(x_i \mid x_j,\ j \neq i)$$

are *available*, that is, are a type for which samples can be obtained using standard algorithms. Examples of available distributions include the uniform, the normal, the gamma, the Poisson, and any finite distribution. From Geman and Geman's perspective, these are the Gibbs distributions, though with potentially large local neighborhoods. To generate a sequence of samples, select $\vec{x}^{\,0} = (x_1^0, x_2^0, \dots, x_N^0)$ arbitrarily and then create $\vec{x}^{\,1} = (x_1^1, x_2^1, \dots, x_N^1)$ as follows.

Generate a sample x_1^1 from $f(x_1 \mid x_2^0, x_3^0, \dots, x_N^0)$.
Generate a sample x_2^1 from $f(x_2 \mid x_1^1, x_3^0, x_4^0, \dots, x_N^0)$.
Generate a sample x_3^1 from $f(x_3 \mid x_1^1, x_2^1, x_4^0, x_5^0, \dots, x_N^0)$.
$\vdots$
Finally, generate a sample x_N^1 from $f(x_N \mid x_1^1, x_2^1, \dots, x_{N-1}^1)$.

One cycle (a raster scan of an image) produces a new value $\vec{x}^{\,1}$. Repeating this process M times produces

$$\vec{x}^{\,0}, \vec{x}^{\,1}, \vec{x}^{\,2}, \dots, \vec{x}^{\,M}$$

which, as usual, approximates a sample from the joint probability distribution $f(x_1, x_2, \dots, x_N)$.

Using this sample, almost any property of the probability distribution can be investigated. For example, focusing on only the first component of each $\vec{x}^k$ produces a sample

$$x_1^0, x_1^1, x_1^2, \dots, x_1^M \tag{20}$$

from the marginal probability distribution of the first component, formally given by the integral

$$f(x_1) = \int_{x_2} \cdots \int_{x_N} f(x_1, x_2, \dots, x_N)\, dx_N \cdots dx_2.$$

In this light, Gibbs sampling can be thought of as a multi-dimensional numerical integration algorithm. The expected value of the first component x_1,

$$E[x_1] = \int_{x_1} x_1 f(x_1)\, dx_1,$$

is estimated by the average of the sample (20). A 95% confidence interval for x_1 can be taken directly from the sample.

Gelfand and Smith, in both [**25**] and its immediate follow-up [**24**], applied Gibbs sampling to a rich collection of statistical models. Most of these were Bayesian *hierarchical* models, structurally similar to Geman and Geman's Bayesian image models with edge weights.

A simple three-level hierarchical model uses Bayes' rule to bind together data, X, a parameter to be estimated, λ, and an additional *hyper-parameter*, β. Both λ and β can be vectors.

- At the first level, X is described by its likelihood function $f(X \mid \lambda)$, i.e., the probability of observing X conditioned on λ.
- At the next level, λ is modeled by a probability density function, $g(\lambda \mid \beta)$, conditioned on the parameter β.
- At the third level, the hyper-parameter β is modeled with another density function $h(\beta)$. The choice of $h(\beta)$ reflects the modeler's prior beliefs about likely values of β.

The three density functions are stitched together with Bayes' rule, producing a probability density function for λ and β conditioned on the data X:

$$F(\lambda, \beta \mid X) \propto f(X \mid \lambda) g(\lambda \mid \beta) h(\beta). \tag{21}$$

The constant of proportionality is the reciprocal of

$$\int_\lambda \int_\beta f(X \mid \lambda) g(\lambda \mid \beta) h(\beta)\, d\beta\, d\lambda, \tag{22}$$

which is independent of the parameters λ and β, though dependent on the data X. The integrals (or sums, in the case of discrete distributions) are over all values of λ and β. In most cases (22) is impossible to evaluate. Recalling what we've seen so far, it is not surprising that we can comfortably ignore this intimidating-looking expression.

Hierarchical Bayesian models were known to statisticians before 1990. They naturally describe the subtle connections between data, observed parameters, and other unobserved parameters (sometimes called *latent variables*). Their utility was limited by their analytic intractability. Even if a hierarchical model accurately describes the interplay of data and parameters, it is usually extremely difficult, if not impossible, to obtain analytical expressions for important quantities such as point or interval estimates. Gelfand and Smith showed that many of these hierarchical models fit nicely into a form suitable for Gibbs sampling.

To see Gibbs sampling in action, let's consider a model of water pump failure rates originally described by Gaver and O'Muircheartaigh [**23**] and used by Gelfand and Smith in [**25**]. The data, X, are given by pairs (s_i, t_i) for $i = 1, 2, \ldots, 10$. Each pair represents failure information for an individual pump. For each pump, assume that the number of failures s_i in time t_i is given by a Poisson($\lambda_i t_i$) distribution, that is,

$$f_i(s_i \mid \lambda_i) = \frac{(\lambda_i t_i)^{s_i} e^{-\lambda_i t_i}}{s_i!}, \quad i = 1, 2 \ldots, 10.$$

Assuming the failures occur independently, the product gives the likelihood function for $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_{10})$:

$$f(X \mid \vec{\lambda}) = \prod_{i=1}^{10} \frac{(\lambda_i t_i)^{s_i} e^{-\lambda_i t_i}}{s_i!}.$$

The traditional "frequentist" approach is to use $\bar{\lambda}_i = s_i/t_i$ as the point estimate of λ_i for $i = 1, 2, \dots, 10$. The Bayesian approach is to assume that the individual λ_i's are linked together by a common distribution. A natural choice in this case, and the one used by Gelfand and Smith, is a gamma distribution with parameters α and β, so that the density for the ith parameter is

$$g_i(\lambda_i \mid \alpha, \beta) = \frac{\lambda_i^{\alpha-1} e^{-\lambda_i/\beta}}{\beta^\alpha \Gamma(\alpha)}, \quad i = 1, 2, \dots, 10.$$

Gelfand and Smith estimated the "shape" parameter α from the data using the method of moments and considered β as the hyper-parameter. The product of the $g_i(\lambda_i \mid \alpha, \beta)$ for $i = 1, 2, \dots, 10$ gives the second-level density in the hierarchy:

$$g(\vec{\lambda} \mid \beta) = \prod_{i=1}^{10} \frac{\lambda_i^{\alpha-1} e^{-\lambda_i/\beta}}{\beta^\alpha \Gamma(\alpha)}.$$

The remaining hyper-parameter β is described by an inverse gamma distribution with parameters γ and δ, so that

$$h(\beta) = \frac{\delta^\gamma e^{-\delta/\beta}}{\beta^{\gamma+1}\Gamma(\gamma)}.$$

The parameters γ and δ are selected so as to make the top-level inverse gamma reasonably diffuse.[14]

The resulting posterior joint density (21) for the parameters $\lambda_1, \lambda_2, \dots, \lambda_{10}$ along with the scale parameter β is

$$F(\lambda_1, \dots, \lambda_{10}, \beta \mid X) \propto \left[\prod_{i=1}^{10} \frac{(\lambda_i t_i) e^{-\lambda_i t_i}}{s_i!}\right] \left[\prod_{i=1}^{10} \frac{\lambda_i^{\alpha-1} e^{-\lambda_i/\beta}}{\beta^\alpha \Gamma(\alpha)}\right] \left[\frac{\delta^\gamma e^{-\delta/\beta}}{\beta^{\delta+1}\Gamma(\gamma)}\right]. \quad (23)$$

This complicated-looking joint density possesses the necessary structure for applying Gibbs sampling. For $i = 1, 2, \dots 10$, the density for λ_i conditioned on the other parameters is proportional to

$$\lambda_i^{s_i+\alpha-1} e^{-\lambda_i(t_i+1/\beta)}. \quad (24)$$

The constant of proportionality is obtained by absorbing all factors independent of λ_i. The form of (24) shows that $\text{Prob}(\lambda_i \mid \lambda_j,\ j \neq i, X, \beta)$ is a gamma distribution with parameters $s_i + \alpha - 1$ and $1/(t_i + 1/\beta)$. Since the gamma distribution is available, Gibbs sampling can be applied at this step.

[14] A *diffuse* distribution tries to convey as little prior information as possible about the parameters. In the extreme case, a distribution can be "noninformative." A common example of such a distribution is the uniform distribution on the parameter space.

The density for β, conditioned on the other parameters, is proportional to

$$\frac{e^{(\sum_{i=1}^{10} \lambda_i + \delta)/\beta}}{\beta^{10\alpha+\gamma+1}},$$

showing that $\text{Prob}(\beta \mid \lambda_1, \ldots, \lambda_{10}, X)$ is an inverse gamma distribution with parameters $\gamma + 10\alpha$ and $\sum_{i=1}^{10} \lambda_i + \delta$. Once again, this is an available distribution.

Gelfand and Smith applied Gibbs sampling to the posterior distribution in the pumps model and obtained marginal posterior distributions for all the λ_i's and for β. The results were impressive: in relatively few iterations, the posterior samples recreated results obtained from other more involved integration methods.

Some of the results of using Gibbs sampling with the pumps model are shown in Figure 5. The histograms show samples for λ_2 ($s_2 = 1$ and $t_2 = 15.72$, $\bar{\lambda}_2 = s_2/t_2 = 0.0636$) and λ_8 ($s_8 = 1$ and $t_8 = 1.048$, $\bar{\lambda}_8 = s_8/t_8 = 0.9542$). From the samples we can estimate the means and 95% confidence intervals. For λ_2 the estimate of the mean is 0.1541 and the 95% confidence interval is (0.0294, 0.3762). For λ_8 these are 0.8246 and (0.1459, 2.1453).

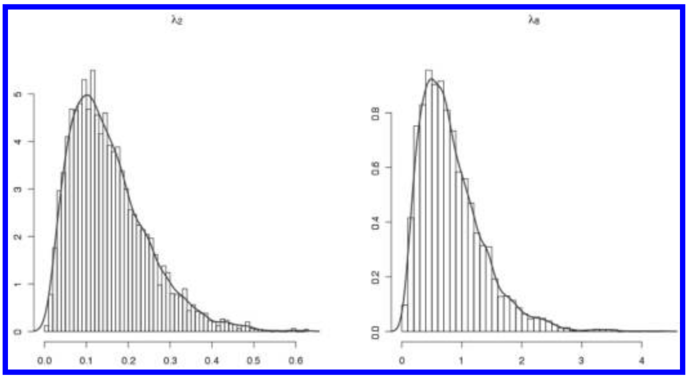

Figure 5. Histograms of samples for λ_2 and λ_8 from the Gelfand and Smith pumps model. From these samples, estimates of the means and 95% confidence intervals are easily obtained. For λ_2 the estimate of the mean is 0.1541 and the 95% confidence interval is (0.0294, 0.3762). For λ_8 these are 0.8246 and (0.1459, 2.1453).

The pumps model is an example of a *conjugate* hierarchical model, that is, one whose intermediate distributions (in this case, those for the λ_i and β) are similar to the original distributions in the hierarchical model. This fits with the Gibbs sampling requirement that these distributions be available. Bayesians had already identified a large number of conjugate models, all of which were candidates for Gibbs sampling.

Gelfand and Smith also applied Gibbs sampling (along with the other two algorithms they studied) to other settings, including a multivariate normal model, a variance component model, and a normal means model. Their conclusions "primed the pump" for Gibbs sampling to enter computational statistics.

> [These algorithms] . . . are all straightforward to implement in several frequently occurring practical situations, thus avoiding complicated numerical or analytical approximation exercises (often necessitating intricate attention to reparametrization and other subtleties requiring case-by-case consideration). For this latter reason if no other the techniques deserve to be better known and experimented with for a wide range of problems. [**25**]

Gelfand and Smith concluded that Gibbs sampling and Tanner and Wong's data augmentation worked better than Rubin's importance sampling algorithm. Because of evidence that data augmentation could be more computationally efficient, they did not settle on Gibbs sampling as the overall favorite. Their follow-up paper made a much stronger statement about the efficacy of Gibbs sampling.

> In the previous article, we entered caveats regarding the computational efficiency of such sampling-based approaches, but our continuing investigations have shown that adaptive, iterative sampling achieved through the Gibbs sampler (Geman and Geman, 1984) is, in fact, surprisingly efficient, converging remarkably quickly for a wide range of problems. [**24**]

In the years to come, this “surprisingly efficient” algorithm was made even more so by the remarkable advances in computational power.

An important impact of Gibbs sampling was that it brought Bayesian methods into mainstream statistics. At last, it was possible to handle the elegant, but analytically intractable, Bayesian posterior distributions. Gelfand and Smith (and others) showed that many Bayesian hierarchical models fit perfectly into the Gibbs sampling framework. Even when the conditional distributions were not available, all was not lost. In these cases, statisticians soon discovered that the more general Metropolis-Hastings algorithm worked wonderfully. Soon statisticians were applying MCMC methods to a wide range of important problems.

5.2. MCMC in Statistics after Gelfand and Smith. It is impossible to do justice to the depth, breadth, and quality of work done with Gibbs sampling and MCMC methods in statistics since 1990. A good place to start is the set of three discussion papers in *The Journal of the Royal Statistical Society, Series B* **55**, No. 1, 1993 by Smith and Roberts [**59**], Besag and Green [**5**], and Gilks et al. [**29**]. Together, these articles attempt to summarize the impact of Gibbs sampling on the statistics community in the short time since the appearance of Gelfand and Smith’s paper.

Parallel to the development of applications and the expansion of the theory, there were numerous papers that focused on simply explaining this “new” theory. Noteworthy early expository articles on Gibbs sampling are Gelfand and Smith’s *Bayesians Statistics Without Tears: A Sampling-Resampling Approach* [**26**] and Casella and George’s *Explaining the Gibbs Sampler* [**8**]. A very nice overview of the Metropolis-Hastings algorithm is Greenberg and Chib’s *Understanding the Metropolis-Hastings Algorithm* [**10**], which has an excellent discussion of both the algorithmic and theoretical aspects of Metropolis-Hastings. For a more general, if somewhat idiosyncratic, overview of Metropolis-Hastings, see Diaconis and Saloff-Coste’s *What Do We Know About the Metropolis Algorithm?* [**17**]. All of these articles contain ample references for anyone interested in reading more about MCMC methods in statistics.

The 1990s also saw the appearance of a number of books dealing with MCMC methods and Bayesian statistics. One of the best early overviews is *Markov Chain Monte Carlo Methods in Practice* [**30**], edited by Gilks et al., which contains numerous examples of MCMC methods applied to problems in areas such as hierarchical modeling, image analysis, longitudinal modeling, genetics, and archaeology, along with discussions of some of the theoretical issues. A more recent resource is Casella and George’s *Monte Carlo Statistical Methods* [**54**]. Tanner’s *Tools for Statistical Inference* [**61**] does an excellent job of presenting and applying MCMC methods (along with expectation maximization and data augmentation). For an overview of Bayesian methods, including MCMC methods, see Gelman et al.’s *Bayesian Data Analysis* [**27**] and Carlin and Louis’s *Bayes and Empirical Bayes Methods for Data Analysis* [**7**].

6. EPILOG.

Since the mid-1990s, MCMC methods have appeared in almost every area of natural and social science, as well proving to be intrinsically interesting to mathematicians, especially probabilists. Innovative applications of MCMC methods appear regularly in computer science, biology (especially genetics), chemistry, physics, psychology, and neuroscience, as well as in economics, political science, sociology, and almost any area one can think of.[15] There are even applications to pure mathematics—for example, sampling from the symmetric group—for which MCMC methods work well.

Along with this explosion of applications, there have been theoretical advances in the understanding of convergence of MCMC methods. Tierney's 1994 paper *Markov Chains for Exploring Posterior Distributions* [**63**] had the most early influence on the statistics community's understanding of convergence. In this, he provides a strong theoretical framework for MCMC methods, including Gibbs sampling, Metropolis-Hastings, and even hybrid methods (some steps using Gibbs sampling, others using Metropolis-Hastings). Tierney addressed a wide range of questions, including the effects of different types of proposal transitions in Metropolis-Hastings. He also proved several strong results related to the ergodic nature of the chains, in particular results that help the practitioner determine run lengths of the Markov chains. Around the same time, Rosenthal [**56**] described a method (called *minorization*) which gives explicit *a priori* polynomial bounds on the number of iterations needed to ensure satisfactory convergence.

There has been much work within the computer science community to understand the convergence properties of MCMC methods. Of particular note is Jerrum and Sinclair's results on polynomial-time bounds for mixing times for the Metropolis algorithm applied to the Ising model [**39**], counting [**37**], and permanents of matrices [**40**]. The fundamental nature of their work was recognized by the 1996 Gödel prize in computer science.

Diaconis [**16**] provides an up-to-date survey of the state of affairs regarding his own work and the work of others on the convergence of the Metropolis algorithm in a variety of settings. For example, Diaconis generalizes the original Metropolis algorithm applied to hard spheres to higher-dimensional Lipschitz domains. In this case, for a target distribution $p(x) = \bar{p}(x)/Z$ on a domain $\Omega \subset R^d$, the Metropolis algorithm defines a transition kernel $P(x, dy)$ which is a bounded, self-adjoint operator on $L^2(p)$. For a maximal step size of h, he shows that

$$\left| P_x^k(A) - \int_A p(y)\, dy \right| \leq c_1 e^{c_2 k h^2}$$

uniformly in $x \in \Omega$ and $A \subset \Omega$. The constant c_1 is given explicitly.

Despite the emergence of these sorts of theoretical results, many applications of MCMC methods do not lend themselves to *a priori* estimates of convergence time. In statistics, for example, the focus is on *diagnostics* of convergence, that is, methods that help determine if a particular MCMC-generated sequence has come sufficiently close to the target distribution. Often, these diagnostics are built on the idea of running the algorithm a number of times with different initial conditions and then checking if the output is consistent across runs. For more information on these methods, see Cowles and Carlin's survey work on MCMC diagnostics [**12**].

In applications outside of statistics (of which there are many), there is even less understanding of convergence. As Diaconis notes:

[15] A simple Google search of the form *MCMC "area of interest"* will undoubtedly return hundreds of results.

> I believe you can take any area of science, from hard to social, and find a burgeoning MCMC literature specifically tailored to that area. I note that *essentially none* of these applications are accompanied by any kind of practically useful running time analysis. [**16**]

7. CONCLUSION. We've arrived at a good place to conclude the story of the evolution of Markov chain Monte Carlo methods. It is difficult not to wax poetic about the algorithm. It is as close to universal as anything in mathematics. It is elegant and efficient. It arose almost spontaneously in the deserts of New Mexico due to the fortunate confluence of people, a problem, and a machine. It grew up hand-in-hand with advances in computing and made substantial impacts across the mathematical and natural sciences. There are still questions about why it works and predicting ahead of time how long it will take to work. Nonetheless, it does work and it works well. After observing the effectiveness of simulated annealing on the traveling salesman problem, perhaps Čerńy said it best.

> It might be surprising that our simple algorithm worked so well in the examples described above. We believe that this is caused by the fact that our algorithm simulates what Nature does in looking for the equilibrium of complex systems. And Nature often does its job quite efficiently. [**9**]

ACKNOWLEDGMENTS. I want to thank Michael Kahn for introducing me to Gibbs sampling. I am grateful to Bernie Alder, Scott Kirkpatrick, Stuart Geman, and Andrew Gelfand for taking the time to share their thoughts on the history of MCMC methods. Lastly, I want to thank Suzanne Dorée for assistance and support throughout the process of writing this paper.

REFERENCES

1. B. Albright, An introduction to simulated annealing, *College Math. J.* **38** (2007) 37–42.
2. H. Anderson, Metropolis, Monte Carlo and the MANIAC, *Los Alamos Science* **14** (1986) 96–108.
3. A. A. Barker, Monte Carlo calculations of the radial distribution functions of the protonelectron plasma, *Australian Journal of Physics* **18** (1969) 119–133.
4. J. Besag, On the statistical analysis of dirty pictures, *J. Roy. Statist. Soc. Ser. B* **48** (1986) 259–302.
5. ———, Spatial statistics and Bayesian computation, *J. Roy. Statist. Soc. Ser. B* **55** (1993) 25–37.
6. S. Brush, History of the Lenz-Ising model, *Reviews of Modern Physics* **39** (1967) 883–893. doi:10.1103/RevModPhys.39.883
7. B. Carlin and T. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*, Chapman & Hall/CRC, Boca Raton, FL, 1996.
8. G. Casella and E. I. George, Explaining the Gibbs sampler, *Amer. Statist.* **46** (1992) 167–174. doi:10.2307/2685208
9. A. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *J. Opt. Theory Appl.* **45** (1985) 41–51. doi:10.1007/BF00940812
10. S. Chib and E. Greenberg, Understanding the Metropolis-Hastings algorithm, *Amer. Statist.* **49** (1995) 327–335. doi:10.2307/2684568
11. B. Cipra, An introduction to the Ising model, *Amer. Math. Monthly* **94** (1987) 937–959. doi:10.2307/2322600
12. M. K. Cowles and B. Carlin, Markov chain Monte Carlo convergence diagnostics: A comparative review, *J. Amer. Statist. Assoc.* **91** (1996) 883–904. doi:10.2307/2291683
13. M. Creutz, Confinement and critical dimensionality in space time, *Phys. Rev. Lett.* **43** (1979) 553–556. doi:10.1103/PhysRevLett.43.553
14. G. Cross and A. Jain, Markov random field texture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5** (1983) 25–39. doi:10.1109/TPAMI.1983.4767341
15. L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.
16. P. Diaconis, The Markov chain Monte Carlo revolution, *Bull. Amer. Math. Soc.* **46** (2009) 179–205. doi:10.1090/S0273-0979-08-01238-X

17. P. Diaconis and L. Saloff, What do we know about the Metropolis algorithm? *J. Comput. System Sci.* **57** (1998) 20–36. doi:10.1006/jcss.1998.1576

18. R. L. Dobrushin, The description of a random field by means of conditional probabilities and conditions of its regularity, *Journal of Probability and Applications* **13** (1968) 197–224. doi:10.1137/1113026

19. J. Dongarra and F. Sullivan, Top ten algorithms of the century, *Computing in Science and Engineering* **2** (2000) 22–23. doi:10.1109/MCISE.2000.814652

20. R. Eckhardt, Stan Ulam, John von Neumann, and the Monte Carlo method, *Los Alamos Science* **Special Issue** (1987) 131–137.

21. P. M. Flinn, Monte Carlo calculations of phase separation in a 2-dimensional Ising system, *J. Stat. Phys.* **10** (1974) 89–97. doi:10.1007/BF01011718

22. S. P. Frankel and N. Metropolis, Calculations in the liquid-drop model of fission, *Phys. Rev.* **72** (1947) 914–925. doi:10.1103/PhysRev.72.914

23. D. P. Gaver and I. G. O'Muircheartaigh, Robust empirical Bayes analyses of event rates, *Technometrics* **29** (1987) 1–15. doi:10.2307/1269878

24. A. E. Gelfand, S. E. Hills, A. Racine, and A. F. M. Smith, Illustration of Bayesian inference in normal data models using Gibbs sampling, *J. Amer. Statist. Assoc.* **85** (1990) 972–985. doi:10.2307/2289594

25. A. E. Gelfand and A. F. M. Smith, Sampling-based approaches to calculating marginal densities, *J. Amer. Statist. Assoc.* **85** (1990) 398–409. doi:10.2307/2289776

26. ———, Bayesian statistics without tears: A sampling-resampling perspective, *Amer. Statist.* **46** (1992) 85–88.

27. A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, Boca Raton, FL, 1995.

28. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6** (1984) 721–741. doi:10.1109/TPAMI.1984.4767596

29. W. R. Gilks, D. G. Clayton, D. J. Spiegelhalter, N. G. Best, A. J. McNeil, L. D. Sharples, and A. J. Kirby, Gibbs sampling in medicine, *J. Roy. Statist. Soc. Ser. B* **55** (1993) 39–52.

30. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC, London, 1996.

31. R. Glauber, Time dependent statistics of the Ising model, *J. Math. Phys.* **4** (1963) 294–307. doi:10.1063/1.1703954

32. J. E. Gubernatis, Marshall Rosenbluth and the Metropolis algorithm, *Phys. Plasmas* **12** (2005) 1–5. doi:10.1063/1.1887186

33. A. Habibi, Two-dimensional Bayesian estimate of images, *Proceedings of the IEEE* **60** (1972) 878–883. doi:10.1109/PROC.1972.8787

34. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London, 1964.

35. W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57** (1970) 97–108. doi:10.1093/biomet/57.1.97

36. D. B. Hitchcock, A history of the Metropolis-Hastings algorithm, *Amer. Statist.* **57** (2003) 254–257. doi:10.1198/0003130032413

37. D. Hochbaum, ed., *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, Boston, 1997.

38. B. R. Hunt, Bayesian methods in nonlinear digital image restoration, *IEEE Trans. Comput.* **C-26** (1977) 219–229. doi:10.1109/TC.1977.1674810

39. M. Jerrum and A. Sinclair, Polynomial-time approximations algorithms for the Ising model, *SIAM J. Comput.* **22** (1993) 1087–1116. doi:10.1137/0222066

40. M. Jerrum, A. Sinclair, and E. Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries, *J. ACM* **51** (2004) 671–697. doi:10.1145/1008731.1008738

41. D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, Optimization by simulated annealing: An experimental evaluation; Part I, Graph partitioning, *Oper. Res.* **37** (1989) 865–892. doi:10.1287/opre.37.6.865

42. ———, Optimization by simulated annealing: An experimental evaluation; Part II, Graph coloring and number partitioning, *Oper. Res.* **39** (1991) 378–406. doi:10.1287/opre.39.3.378

43. L. Kelvin, Nineteenth century clouds over the dynamical theory of heat and light, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2** (1901) 1–40.

44. R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, American Mathematical Society, Providence, RI, 1980.

45. S. Kirkpatrick, C. D. Gelett, and M. P. Vecchi, Optimization by simulated annealing, *Science* **220** (1983) 671–680. doi:10.1126/science.220.4598.671

46. N. Metropolis, The beginning of the Monte Carlo method, *Los Alamos Science* **15** (1987) 125–130.

47. N. Metropolis and F. H. Harlow, Computing and computers: Weapons simulation leads to the computer era, *Los Alamos Science* **12** (1983) 132–141.

48. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *Journal of Chemical Physics* **21** (1953) 1087–1092. doi:10.1063/1.1699114
49. N. Metropolis and S. Ulam, The Monte Carlo method, *J. Amer. Statist. Assoc.* **44** (1949) 335–341. doi:10.2307/2280232
50. G. Michael, An interview with Bernie Alder (1997), available at http://www.computer-history.info/Page1.dir/pages/Alder.html.
51. N. E. Nahi and T. Assefi, Bayesian recursive image estimation, *IEEE Trans. Comput.* **C-21** (1972) 734–738.
52. P. H. Peskun, Optimum Monte-Carlo sampling using Markov chains, *Biometrika* **60** (1973) 607–612. doi:10.1093/biomet/60.3.607
53. W. Richardson, Bayesian-based iterative method of image restoration, *Journal of the Optical Society of America* **62** (1972) 55–59. doi:10.1364/JOSA.62.000055
54. C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, Berlin, 1999.
55. ———, A History of Markov chain Monte Carlo—Subjective recollections from incomplete data (2008), available at http://arxiv.org/abs/0808.2902v1.
56. J. Rosenthal, Minorization conditions and convergence rates for Markov chain Monte Carlo, *J. Amer. Statist. Assoc.* **90** (1995) 558–566. doi:10.2307/2291067
57. J. Rosenthal and W. K. Hastings, statistician and developer of the Metropolis-Hastings algorithm (2004), available at http://probability.ca/hastings/.
58. E. Segré, Fermi and neutron physics, *Rev. Modern Phys.* **27** (1955) 257–263. doi:10.1103/RevModPhys.27.257
59. A. F. Smith and G. O. Roberts, Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, *J. Roy. Statist. Soc. Ser. B* **55** (1993) 3–23.
60. R. Swendsen and J. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.* **58** (1987) 86–88. doi:10.1103/PhysRevLett.58.86
61. M. Tanner, *Tools for Statistical Inference: Methods for Exploration of Posterior Distributions and Likelihood Functions*, Springer-Verlag, New York, 1993.
62. M. Tanner and W. Wong, The calculation of posterior distributions by data augmentation, *J. Amer. Statist. Assoc.* **82** (1987) 528–540. doi:10.2307/2289457
63. L. Tierney, Markov chains for exploring posterior distributions, *Ann. Statist.* **22** (1994) 1701–1728. doi:10.1214/aos/1176325750
64. S. M. Ulam, *Adventures of a Mathematician*, Scribner, New York, 1976.
65. P. J. M. van Laarhoven, *Theoretical and Computational Aspects of Simulated Annealing*, Stichting Mathematisch Centrum, Amsterdam, 1988.
66. P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing, Dordrecht, Holland, 1987.

MATTHEW RICHEY hails from Kentucky and then earned his B.A. from Kenyon College in 1981 and his Ph.D. from Dartmouth College in 1985. He is currently a Professor of Mathematics, Statistics, and Computer Science and the Associate Dean for Natural Sciences and Mathematics at St. Olaf College. His research interests are computational and applied mathematics. In his spare time, Matt enjoys music, reading, and sports (he and his students have used MCMC methods to search for evidence of clutch hitting in baseball; none was found). Ten years ago he took up golf and has tormented himself with that ridiculous pursuit ever since.

Department of Mathematics, Statistics, and Computer Science, St. Olaf College, Northfield, MN 55057
richeym@stolaf.edu