suddenly shifted above their boiling temperature. Small bubbles would then nucleate along the ionization trail left by charged particles moving through the tank. The bubbles could be photographed and the tracks of the particles identified. Such detectors were called bubble chambers. This methodology has been largely abandoned in favor of electronic detectors. There is a limit to how far a system can be supercooled or superheated. The limit is easy to understand in the Ising model. If a system with a positive magnetization m is subject to a negative magnetic field of magnitude greater than zJm, then each individual spin will flip DOWN independent of its neighbors. This is the ultimate limit for nucleation kinetics.

1.6.9 *Connections between CA and the Ising model*

Our primary objective throughout this section is the investigation of the equilibrium properties of interacting systems. It is useful, once again, to consider the relationship between the equilibrium ensemble and the kinetic CA we considered in Section 1.5. When a deterministic CA evolves to a unique steady state independent of the initial conditions, we can identify the final state as the $T = 0$ equilibrium ensemble. This is, however, not the way we usually consider the relationship between a dynamic system and its equilibrium condition. Instead, the equilibrium state of a system is generally regarded as the time average over microscopic dynamics. Thus when we use the CA to represent a microscopic dynamics, we could also identify a long time average of a CA as the equilibrium ensemble. Alternatively, we can consider a stochastic CA that evolves to a unique steady-state distribution where the steady state is the equilibrium ensemble of a suitably defined energy function.

1.7 Computer Simulations (Monte Carlo, Simulated Annealing)

Computer simulations enable us to investigate the properties of dynamical systems by directly studying the properties of particular models. Originally, the introduction of computer simulation was viewed by many researchers as an undesirable adjunct to analytic theory. Currently, simulations play such an important role in scientific studies that many analytic results are not believed unless they are tested by computer simulation. In part, this reflects the understanding that analytic investigations often require approximations that are not necessary in computer simulations. When a series of approximations has been made as part of an analytic study, a computer simulation of the original problem can directly test the approximations. If the approximations are validated, the analytic results often generalize the simulation results. In many other cases, simulations can be used to investigate systems where analytic results are unknown.

1.7.1 *Molecular dynamics and deterministic simulations*

The simulation of systems composed of microscopic Newtonian particles that experience forces due to interparticle interactions and external fields is called molecular dynamics. The techniques of molecular dynamics simulations, which integrate

Newton's laws for individual particles,have been developed to optimize the efficiency of computer simulation and to take advantage of parallel computer architectures. Typically, these methods implement a discrete iterative map (Section 1.1) for the particle positions. The most common (Verlet) form is:

$$r(t) = 2r(t - \Delta t) - r(t - 2\Delta t) + \Delta t^2\, a(t - \Delta t) \tag{1.7.1}$$

where $a(t) = F(t)/m$ is the force on the particle calculated from models for interparticle and external forces. As in Section 1.1, time would be measured in units of the time interval Δt for convenience and efficiency of implementation. Eq. (1.7.1) is algebraically equivalent to the iterative map in Question 1.1.4, which is written as an update of both position and velocity:

$$\begin{aligned} r(t) &= r(t - \Delta t) + \Delta t\, v(t - \Delta t/2) \\ v(t + \Delta t/2) &= v(t - \Delta t/2) + \Delta t a(t) \end{aligned} \tag{1.7.2}$$

As indicated, the velocity is interpreted to be at half integral times, though this does not affect the result of the iterative map.

For most such simulations of physical systems,the accuracy is limited by the use of models for interatomic interactions. Modern efforts attempt to improve upon this approach by calculating forces from quantum mechanics. However, such simulations are very limited in the number of particles and the duration of a simulation.A useful measure of the extent of a simulation is the product $Nt_{\max}$ of the amount of physical time $t_{\max}$, and the number of particles that are simulated N. Even without quantum mechanical forces,molecular dynamics simulations are still far from being able to describe systems on a space and time scale comparable to human senses. However, there are many questions that can be addressed regarding microscopic properties of molecules and materials.

The development of appropriate simplified macroscopic descriptions of physical systems is an essential aspect of our understanding of these systems. These models may be based directly upon macroscopic phenomenology obtained from experiment. We may also make use of the microscopic information obtained from various sources, including both theory and experiment, to inform our choice of macroscopic models. It is more difficult, but important as a strategy for the description of both simple and complex systems, to develop systematic methods that enable macroscopic models to be obtained directly from microscopic models. The development of such methods is still in its infancy, and it is intimately related to the issues of emergent simplicity and complexity discussed in Chapter 8.

Abstract mathematical models that describe the deterministic dynamics for various systems, whether represented in the form of differential equations or deterministic cellular automata (CA, Section 1.5), enable computer simulation and study through integration of the differential equations or through simulation of the CA. The effects of external influences, not incorporated in the parameters of the model, may be modeled using stochastic variables (Section 1.2). Such models, whether of fluids or of galaxies, describe the macroscopic behavior of physical systems by assuming that the microscopic (e.g., molecular) motion is irrelevant to the macroscopic

phenomena being described. The microscopic behavior is summarized by parameters such as density, elasticity or viscosity. Such model simulations enable us to describe macroscopic phenomena on a large range of spatial and temporal scales.

1.7.2 *Monte Carlo simulations*

In our investigations of various systems, we are often interested in average quantities rather than a complete description of the dynamics. This was particularly apparent in Section 1.3, when equilibrium thermodynamic properties of systems were discussed. The ergodic theorem (Section 1.3.5) suggested that we can use an ensemble average instead of the space-time average of an experiment. The ensemble average enables us to treat problems analytically, when we cannot integrate the dynamics explicitly. For example, we studied equilibrium properties of the Ising model in Section 1.6 without reference to its dynamics. We were able to obtain estimates of its free energy, energy and magnetization by averaging various quantities using ensemble probabilities.

However, we also found that there were quite severe limits to our analytic capabilities even for the simplest Ising model. It was necessary to use the mean field approximation to obtain results analytically. The essential difficulty that we face in performing ensemble averages for complex systems,and even for the simple Ising model, is that the averages have to be performed over the many possible states of the system. For as few as one hundred spins,the number of possible states of the system—2^{100}—is so large that we cannot average over all of the possible states. This suggests that we consider approximate numerical techniques for studying the ensemble averages. In order to perform the averages without summing over all the states, we must find some way to select a representative sample of the possible states.

Monte Carlo simulations were developed to enable numerical averages to be performed efficiently. They play a central role in the use of computers in science. Monte Carlo can be thought of as a general way of estimating averages by selecting a limited sample of states of the system over which the averages are performed. In order to optimize convergence of the average, we take advantage of information that is known about the system to select the limited sample. As we will see, under some circumstances,the sequence of states selected in a Monte Carlo simulation may itself be used as a model of the dynamics of a system. Then,if we are careful about designing the Monte Carlo, we can separate the time scales of a system by treating the fast degrees of freedom using an ensemble average and still treat explicitly the dynamic degrees of freedom.

To introduce the concept of Monte Carlo simulation,we consider finding the average of a function $f(s)$, where the system variable s has the probability $P(s)$. For simplicity, we take s to be a single real variable in the range $[-1,+1]$. The average can be approximated by a sum over equally spaced values s_i:

$$< f(s) > = \int_{-1}^{1} f(s)P(s)ds \approx \sum_{s_i} f(s_i)P(s_i)\Delta s = \frac{1}{M}\sum_{n=-M}^{M} f(n/M)P(n/M) \qquad (1.7.3)$$

This formula works well if the functions $f(s)$ and $P(s)$ are reasonably smooth and uniform in magnitude. However, when they are not smooth,this sum can be a very inef-

ficient way to perform the integral. Consider this integral when $P(s)$ is a Gaussian,and $f(s)$ is a constant:

$$< f(s) > \int_{-1}^{1} e^{-s^2/2\sigma^2} ds \approx \frac{1}{M} \sum_{n=-M}^{M} e^{-(n/M)^2/2\sigma^2} \tag{1.7.4}$$

A plot of the integrand in Fig. 1.7.1 shows that for $\sigma<<1$ we are performing the integral by summing many values that are essentially zero. These values contribute nothing to the result and require as much computational effort as the comparatively few points that do contribute to the integral near $s = 0$, where the function is large. The few points near $s = 0$ will not give a very accurate estimate of the integral. Thus,most of the computational work is being wasted and the integral is not accurately evaluated. If we want to improve the accuracy of the sum, we have to increase the value of M. This means we will be summing many more points that are almost zero.

To avoid this problem, we would like to focus our attention on the region in Eq. (1.7.4) where the integrand is large. This can be done by changing how we select the points where we perform the average. Instead of picking the points at equal intervals along the line, we pick them with a probability given by $P(s)$. This is the same as saying that we have an ensemble representing the system with the state variable s. Then we perform the ensemble average:

$$< f(s) > = \int f(s)P(s)ds = \frac{1}{N} \sum_{s:P(s)}^{N} f(s) \tag{1.7.5}$$

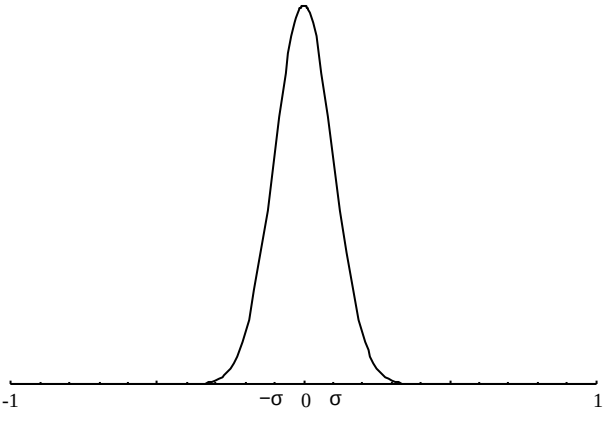

Figure 1.7.1 Plot of the Gaussian distribution illustrating that an integral that is performed by uniform sampling will use a lot of points to represent regions where the Gaussian is vanishingly small. The problem gets worse as σ becomes smaller compared to the region over which the integral must be performed. It is much worse in typical multidimensional averages where the Boltzmann probability is used. Monte Carlo simulations make such integrals computationally feasible by sampling the integrand in regions of high probability. ∎

The latter expression represents the sum over N values of s, where these values have the probability distribution $P(s)$. We have implicitly assumed that the function $f(s)$ is relatively smooth compared to $P(s)$. In Eq. (1.7.5) we have replaced the integral with a sum over an ensemble. The problem we now face is to obtain the members of the ensemble with probability $P(s)$. To do this we will invert the ergodic theorem of Section 1.3.5.

Since Section 1.3 we have described an ensemble as representing a system, if the dynamics of the system satisfied the ergodic theorem. We now turn this around and say that the ensemble sum in Eq.(1.7.5) can be represented by any dynamics that satisfies the ergodic theorem, and which has as its equilibrium probability $P(s)$. To do this we introduce a time variable t that, for our current purposes, just indicates the order of terms in the sum we are performing. The value of s appearing in the tth term would be $s(t)$. We then rewrite the ergodic theorem by considering the time average as an approximation to the ensemble average (rather than the opposite):

$$< f(s) > = \frac{1}{T} \sum_{t=1}^{T} f(s(t)) \tag{1.7.6}$$

The problem remains to sequentially generate the states $s(t)$, or, in other words, to specify the dynamics of the system. If we know the probability $P(s)$,and s is a few binary or real variables,this may be done directly with the assistance of a random number generator (Question 1.7.1). However, often the system coordinate s represents a large number of variables.A more serious problem is that for models of physical systems, we generally don't know the probability distribution explicitly.

Thermodynamic systems are described by the Boltzmann probability (Section 1.3):

$$\begin{aligned} P(\{x,p\}) &= \frac{1}{Z} e^{-E(\{x,p\})/kT} \\ Z &= \sum_{\{x,p\}} e^{-E(\{x,p\})/kT} \end{aligned} \tag{1.7.7}$$

where $\{x,p\}$ are the microscopic coordinates of the system,and $E(\{x,p\})$ is the microscopic energy. An example of a quantity we might want to calculate would be the average energy:

$$U = \frac{1}{Z} \sum_{\{x,p\}} E(\{x,p\}) e^{-E(\{x,p\})/kT} \tag{1.7.8}$$

In many cases,as discussed in Section 1.4, the quantity that we would like to find the average of depends only on the position of particles and not on their momenta. We then write more generally

$$\begin{aligned} P(s) &= \frac{1}{Z_s} e^{-F(s)/kT} \\ Z_s &= \sum_{s} e^{-F(s)/kT} \end{aligned} \tag{1.7.9}$$

where we use the system state variable s to represent the relevant coordinates of the system. We make no assumption about the dimensionality of the coordinate s which may, for example, be the coordinates $\{x\}$ of all of the particles. $F(s)$ is the free energy of the set of states associated with the coordinate s. A precise definition, which indicates both the variable s and its value s', is given in Eq. (1.4.27):

$$F_s(s') = -kT \ln\Big(\sum_{\{x,p\}} \delta_{s,s'}\, e^{-E(\{x,p\})/kT}\Big) \tag{1.7.10}$$

We note that Eq. (1.7.9) is often written using the notation $E(s)$ (the energy of s) instead of $F(s)$ (the free energy of s), though $F(s)$ is more correct. An average we might calculate, of a quantity $Q(s)$, would be:

$$U = \frac{1}{Z}\sum_s Q(s) e^{-F(s)/kT} \tag{1.7.11}$$

where $Q(s)$ is assumed to depend only on the variable s and not directly on $\{x,p\}$.

The problem with the evaluation of either Eq. (1.7.8) or Eq. (1.7.11) is that the Boltzmann probability does not explicitly give us the probability of a particular state. In order to find the actual probability, we need to find the partition function Z. To calculate Z we need to perform a sum over all states of the system, which is computationally impossible. Indeed, if we were able to calculate Z, then, as discussed in Section 1.3, we would know the free energy and all the other thermodynamic properties of the system. So a prescription that relies upon knowing the actual value of the probability doesn't help us. However, it turns out that we don't need to know the actual probability in order to construct a dynamics for the system, only the relative probabilities of particular states. The relative probability of two states, $P(s) / P(s')$, is directly given by the Boltzmann probability in terms of their relative energy:

$$P(s) / P(s') = e^{-(F(s)-F(s'))/kT} \tag{1.7.12}$$

This is the key to Monte Carlo simulations. It is also a natural result, since a system that is evolving in time does not know global properties that relate to all of its possible states. It only knows properties that are related to the energy it has, and how this energy changes with its configuration. In classical mechanics, the change of energy with configuration would be the force experienced by a particle.

Our task is to describe a dynamics that generates a sequence of states of a system $s(t)$ with the proper probability distribution, $P(s)$. The classical (Newtonian) approach to dynamics implies that a deterministic dynamics exists which is responsible for generating the sequence of states of a physical system. In order to generate the equilibrium ensemble, however, there must be contact with a thermal reservoir. Energy transfer between the system and the reservoir introduces an external interaction that disrupts the system's deterministic dynamics.

We will make our task simpler by allowing ourselves to consider a stochastic Markov chain (Section 1.2) as the dynamics of the system. The Markov chain is described by the probability $P_s(s'|s'')$ of the system in a state $s = s''$ making a transition

to the state $s = s'$. A particular sequence $s(t)$ is generated by starting from one configuration and choosing its successors using the transition probabilities.

The general formulation of a Markov chain includes the classical Newtonian dynamics and can also incorporate the effects of a thermal reservoir. However, it is generally convenient and useful to use a Monte Carlo simulation to evaluate averages that do not depend on the momenta,as in Eq.(1.7.11). There are some drawbacks to this approach. It limits the properties of the system whose averages can be evaluated. Systems where interactions between particles depend on their momenta cannot be easily included. Moreover, averages of quantities that depend on both the momentum and the position of particles cannot be performed. However, if the energy separates into potential and kinetic energies as follows:

$$E(\{x,p\}) = V(\{x\}) + \sum_i p_i^2/2m \tag{1.7.13}$$

then averages over all quantities that just depend on momenta (such as the kinetic energy) can be evaluated directly without need for numerical computation. These averages are the same as those of an ideal gas. Monte Carlo simulations can then be used to perform the average over quantities that depend only upon position $\{x\}$, or more generally, on position-related variables s. Thus,in the remainder of this section we focus on describing Markov chains for systems described only by position-related variables s.

As described in Section 1.2 we can think about the Markov dynamics as a dynamics of the probability rather than the dynamics of a system. Then the dynamics are specified by

$$P_s(s';t) = \sum_{s''} P_s(s'|s'')P_s(s'';t-1) \tag{1.7.14}$$

In order for the stochastic dynamics to represent the ensemble, we must have the time average over the probability distribution $P_s(s',t)$ equal to the ensemble probability. This is true for a long enough time average if the probability converges to the ensemble probability distribution, which is a steady-state distribution of the Markov chain:

$$P_s(s') = P_s(s';\infty) = \sum_{s''} P_s(s'|s'')P_s(s'';\infty) \tag{1.7.15}$$

Thermodynamics and stochastic Markov chains meet when we construct the Markov chain so that the Boltzmann probability, Eq. (1.7.9), is the limiting distribution.

We now make use of the Perron-Frobenius theorem (see Section 1.7.4 below), which says that a Markov chain governed by a set of transition probabilities $P_s(s'|s'')$ converges to a unique limiting probability distribution as long as it is irreducible and acyclic. Irreducible means that there exist possible paths between each state and all other possible states of the system. This does not mean that all states of the system are connected by nonzero transition probabilities. There can be transition probabilities that are zero. However, it must be impossible to separate the states into two sets for which there are no transitions from one set to the other. Acyclic means that the system is not ballistic—the states are not organized by the transition matrix into a ring

with a deterministic flow around it. There may be currents, but they must not be deterministic. It is sufficient for there to be a single state which has a nonzero probability of making a transition to itself for this condition to be satisfied,thus it is often assumed and unstated.

We can now summarize the problem of identifying the desired Markov chain. We must construct a matrix $P_s(s'|s'')$ that satisfies three properties.First,it must be an allowable transition matrix. This means that it must be nonnegative, $P_s(s''|s') \geq 0$, and satisfy the normalization condition (Eq (1.2.4)):

$$\sum_{s''} P_s(s''|s') = 1 \tag{1.7.16}$$

Second,it must have the desired probability distribution, Eq.(1.7.9),as a fixed point. Third, it must not be reducible—it is possible to construct a path between any two states of the system.

These conditions are sufficient to guarantee that a long enough Markov chain will be a good approximation to the desired ensemble. There is no guarantee that the convergence will be rapid. As we have seen in Section 1.4,in the case of the glass transition,the ergodic theorem may be violated on all practical time scales for systems that are following a particular dynamics. This applies to realistic or artificial dynamics. In general such violations of the ergodic theorem, or even just slow convergence of averages, are due to energy barriers or entropy "bottlenecks" that prevent the system from reaching all possible configurations of the system in any reasonable time. Such obstacles must be determined for each system that is studied, and are sometimes but not always apparent. It should be understood that different dynamics will satisfy the conditions of the ergodic theorem over very different time scales. The equivalence of results of an average performed using two distinct dynamics is only guaranteed ifthey are both simulated for long enough so that each satisfies the ergodic theorem.

Our discussion here also gives some additional insights into the conditions under which the ergodic theorem applies to the actual dynamics of physical systems. We note that any proof of the applicability of the ergodic theorem to a real system requires considering the actual dynamics rather than a model stochastic process. When the ergodic theorem does not apply to the actual dynamics, then the use of a Monte Carlo simulation for performing an average must be considered carefully. It will not give the same results if it satisfies the ergodic theorem while the real system does not.

We are still faced with the task of selecting values for the transition probabilities $P_s(s'|s'')$ that satisfy the three requirements given above. We can simplify our search for transition probabilities $P_s(s'|s'')$ for use in Monte Carlo simulations by imposing the additional constraint of microscopic reversibility, also known as detailed balance:

$$P_s(s''|s')P_s(s';\infty) = P_s(s'|s'')\,P_s(s'';\infty) \tag{1.7.17}$$

This equation implies that the transition currents between two states of the system are equal and therefore cancel in the steady state, Eq.(1.7.15). It corresponds to true equilibrium,as would be present in a physical system. Detailed balance implies the steady-state condition, but is not required by it.Steady state can also include currents that do

not change in time. We can prove that Eq.(1.7.17) implies Eq. (1.7.15) by summing over s':

$$\sum_{s'} P_s(s''|s')P_s(s';\infty) = \sum_{s'} P_s(s'|s'')P_s(s'';\infty) = P_s(s'';\infty) \tag{1.7.18}$$

We do not yet have an explicit prescription for $P_s(s'|s'')$. There is still a tremendous flexibility in determining the transition probabilities.One prescription that enables direct implementation, called Metropolis Monte Carlo, is:

$$\begin{aligned} P_s(s'|s'') &= \lambda(s'|s'') && P_s(s')/P_s(s'') \geq 1 \quad s'' \neq s' \\ P_s(s'|s'') &= \lambda(s'|s'')P_s(s')/P_s(s'') && P_s(s')/P_s(s'') < 1 \quad s'' \neq s' \\ P_s(s''|s'') &= 1 - \sum_{s' \neq s''} P_s(s'|s'') \end{aligned} \tag{1.7.19}$$

These expressions specify the transition probability $P_s(s'|s'')$ in terms of a symmetric stochastic matrix $\lambda(s'|s'')$. $\lambda(s'|s'')$ is independent of the limiting equilibrium distribution. The constraint associated with the limiting distribution has been incorporated explicitly into Eq. (1.7.19). It satisfies detailed balance by direct substitution in Eq. (1.7.17), since for $P_s(s') \geq P_s(s'')$ (similarly for the opposite) we have

$$\begin{aligned} P_s(s''|s')P_s(s') &= \lambda(s''|s')P_s(s') = \lambda(s'|s'')P_s(s') \\ &= \left(\lambda(s'|s'')P_s(s')/P_s(s'')\right)P_s(s'') = P_s(s'|s'')P_s(s'') \end{aligned} \tag{1.7.20}$$

The symmetry of the matrix $\lambda(s'|s'')$ is essential to the proof of detailed balance. One must often be careful in the design of specific algorithms to ensure this property. It is also important to note that the limiting probability appears in Eq. (1.7.19) only in the form of a ratio $P_s(s')/P_s(s'')$ which can be given directly by the Boltzmann distribution.

To understand Metropolis Monte Carlo, it is helpful to describe a few examples. We first describe the movement of the system in terms of the underlying stochastic process specified by $\lambda(s'|s'')$, which is independent of the limiting distribution. This means that the limiting distribution of the underlying process is uniform over the whole space of possible states.

A standard way to choose the matrix $\lambda(s'|s'')$ is to set it to be constant for a few states s' that are near s''. For example, the simplest random walk is such a case, since it allows a probability of 1/2 for the system to move to the right and to the left. If s is a continuous variable, we could choose a distance r_0 and allow the walker to take a step anywhere within the distance r_0 with equal probability. Both the discrete and continuous random walk have d-dimensional analogs or, for a system of interacting particles, N-dimensional analogs. When there is more than one dimension, we can choose to move in all dimensions simultaneously. Alternatively, we can choose to move in only one of the dimensions in each step. For an Ising model (Section 1.6), we could allow equal probability for any one of the spins to flip.

Once we have specified the underlying stochastic process, we generate the sequence of Monte Carlo steps by applying it. However, we must modify the probabilities according to Eq. (1.7.19). This takes the form of choosing a step, but sometimes rejecting it rather than taking it. When a step is rejected,the system does not change

its state. This gives rise to the third equation in Eq.(1.7.19) where the system does not move. Specifically, we can implement the Monte Carlo process according to the following prescription:

1. Pick one of the possible moves allowed by the underlying process. The selection is random from all of the possible moves. This guarantees that we are selecting it with the underlying probability $\lambda(s'|s'')$.
2. Calculate the ratio of probabilities between the location we are going to, compared to the location we are coming from

$$P_s(s'') \,/\, P_s(s') = e^{-(E(s')-E(s''))/kT} \tag{1.7.21}$$

If this ratio of probabilities is greater than one, which means the energy is lower where we are going, the step is accepted. This gives the probability for the process to occur as $\lambda(s'|s'')$, which agrees with the first line of Eq.(1.7.19). However, if this ratio is less than one, we accept it with a probability given by the ratio. For example,if the ratio is 0.6, we accept the move 60% of the time. If the move is rejected,the system stays in its original location. Thus,if the energy where we are trying to go is higher, we do not accept it all the time, only some of the time. The likelihood that we accept it decreases the higher the energy is.

The Metropolis Monte Carlo prescription makes logical sense. It tends to move the system to regions of lower energy. This must be the case in order for the final distribution to satisfy the Boltzmann probability. However, it also allows the system to climb up in energy so that it can reach, with a lower probability, states of higher energy. The ability to climb in energy also enables the system to get over barriers such as the one in the two-state system in Section 1.4.

For the Ising model, we can see that the Monte Carlo dynamics that uses all single spin flips as its underlying stochastic process is not the same as the Glauber dynamics (Section 1.6.7), but is similar. Both begin by selecting a particular spin. After selection of the spin, the Monte Carlo will set the spin to be the opposite with a probability:

$$\min(1, e^{-(E(1)-E(-1))/kT}) \tag{1.7.22}$$

This means that if the energy is lower for the spin to flip, it is flipped. If it is higher, it may still flip with the indicated probability. This is different from the Glauber prescription, which sets the selected spin to UP or DOWN according to its equilibrium probability (Eq. (1.6.61)–Eq. (1.6.63)). The difference between the two schemes can be shown by plotting the probability of a selected spin being UP as a function of the energy difference between UP and DOWN, $E_+ = E(1) - E(-1)$ (Fig. 1.7.2). The Glauber dynamics prescription is independent of the starting value of the spin. The Metropolis Monte Carlo prescription is not. The latter causes more changes, since the spin is more likely to flip. Unlike the Monte Carlo prescription, the Glauber dynamics explicitly requires knowledge of the probabilities themselves. For a single spin flip in an Ising system this is fine, because there are only two possible states and the probabilities depend only on E_+. However, this is difficult to generalize when a system has many more possible states.

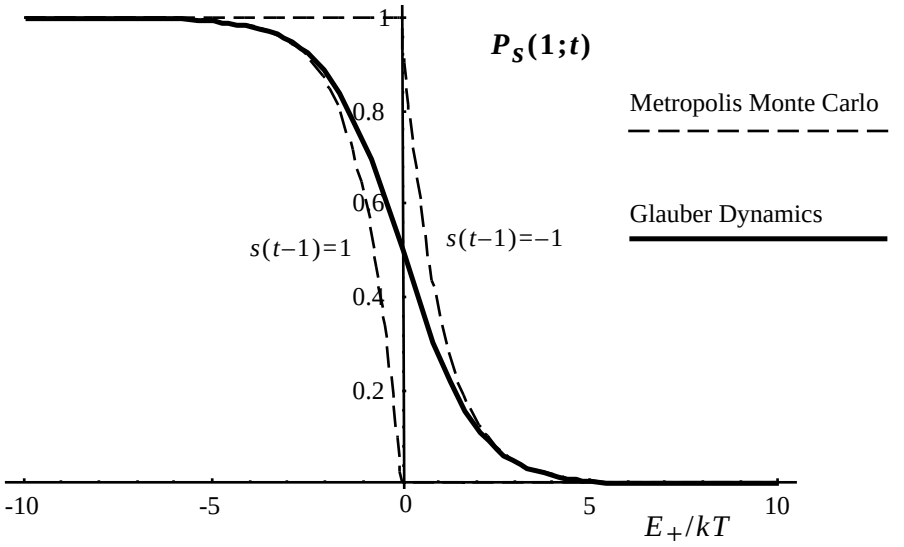

Figure 1.7.2 Illustration of the difference between Metropolis Monte Carlo and Glauber dynamics for the update of a spin in an Ising model. The plots show the probability $P_s(1;t)$ of a spin being UP at time t. The Glauber dynamics probability does not depend on the starting value of the spin. There are two curves for the Monte Carlo probability, for $s(t-1) = 1$ and $s(t-1) = -1$. ∎

There is a way to generalize further the use of Monte Carlo by recognizing that we do not even have to use the correct equilibrium probability distribution when generating the time series. The generalized expression for an arbitrary probability distribution $P'(s)$ is:

$$< f(s) >_{P(s)} = \int \frac{f(s)P(s)}{P'(s)} P'(s)ds = \frac{1}{N} \sum_{s:P'(s)}^{N} \frac{f(s)P(s)}{P'(s)} \tag{1.7.23}$$

The subscript $P(s)$ indicates that the average assumes that s has the probability distribution $P(s)$ rather than $P'(s)$. This equation generalizes Eq. (1.7.5). The problem with this expression is that it requires that we know explicitly the probabilities $P(s)$ and $P'(s)$. This can be remedied. We illustrate for a specific case, where we use the Boltzmann distribution at one temperature to evaluate the average at another temperature:

$$< f(s) >_{P(s)} = \frac{1}{N} \sum_{s:P'(s)}^{N} \frac{f(s)P(s)}{P'(s)} = \left(\frac{Z'}{Z}\right) \frac{1}{N} \sum_{s:P'(s)}^{N} f(s) e^{-E(s)(1/kT - 1/kT')} \tag{1.7.24}$$

The ratio of partition functions can be directly evaluated as an average:

$$\left(\frac{Z}{Z'}\right) = \frac{\sum_s e^{-E(s)/kT}}{\sum_s e^{-E(s)/kT'}} = \frac{\sum_s e^{-E(s)(1/kT-1/kT')} e^{-E(s)/kT'}}{\sum_s e^{-E(s)/kT'}}$$
$$= \left\langle e^{-E(s)(1/kT-1/kT')} \right\rangle_{P'(s)} = \frac{1}{N} \sum_{s:P'(s)}^{N} e^{-E(s)(1/kT-1/kT')} \tag{1.7.25}$$

Thus we have the expression:

$$< f(s) >_{P(s)} = \sum_{s:P'(s)}^{N} f(s) e^{-E(s)(1/kT-1/kT')} \Big/ \sum_{s:P'(s)}^{N} e^{-E(s)(1/kT-1/kT')} \tag{1.7.26}$$

This means that we can obtain the average at various temperatures using only a single Monte Carlo simulation. However, the whole point of using the ensemble average is to ensure that the average converges rapidly. This may not happen if the ensemble temperature T' is much different from the temperature T. On the other hand, there are circumstances where the function $f(s)$ may have an energy dependence that makes it better to perform the average using an ensemble that is not the equilibrium ensemble.

The approach of Monte Carlo simulations to the study of statistical averages ensures that we do not have to be concerned that the dynamics we are using for the system is a real dynamics. The result is the same for a broad class of artificial dynamics. The generality provides a great flexibility; however, this is also a limitation. We cannot use the Monte Carlo dynamics to study dynamics. We can only use it to perform statistical averages. Must we be resigned to this limitation? The answer, at least in part, is no. The reason is rooted in the central limit theorem. For example,the implementations of Metropolis Monte Carlo and the Glauber dynamics are quite different. We know that in the limit of long enough times, the distribution of configurations generated by both is the same. We expect that since each of them flips only one spin,if we are interested in changes in many spins,the two should give comparable results in the sense of the central limit theorem. This means that aside from an overall scale factor, the time evolution of the distribution of probabilities for long times is the same. Since we already know that the limiting distribution is the same in both cases, we are asserting that the approach to this limiting distribution, which is the long time dynamics, is the same.

The claim that for a large number of steps all dynamics is the same is not true about all possible Monte Carlo dynamics. If we allow all of the spins in an Ising model to change their values in one step of the underlying dynamics $\lambda(s'|s'')$, then this step would be equivalent to many steps in a dynamics that allows only one spin to flip at a time. In order for two different dynamics to give the same results,there are two types of constraints that are necessary. First, both must have similar kinds of allowed steps. Specifically, we define steps to the naturally proximate configurations as local moves. As long as the Monte Carlo allows only local moves, the long time dynamics should be the same. Such dynamics correspond to a local diffusion in the space of possible

configurations of the system. More generally, two different dynamics should be the same if configuration changes that require many steps in one also require many steps in the other. The second type of constraint is related to symmetries of the problem. A lack of bias in the random walk was necessary to guarantee that the Gaussian distribution resulted from a generalized random walk in Section 1.2. For systems with more than one dimension, we must also ensure that there is no relative bias between motion in different directions.

We can think about Monte Carlo dynamics as diffusive dynamics of a system that interacts frequently with a reservoir. There are properties of more realistic dynamics that are not reproduced by such configuration Monte Carlo simulations. Correlations between steps are not incorporated because of the assumptions underlying Markov chains. This rules out ballistic motion,and exact or approximate momentum conservation. Momentum conservation can be included if both position and momentum are included as system coordinates. The method called Brownian dynamics incorporates both ballistic and diffusive dynamics in the same simulation. However, if correlations in the dynamics of a system have a shorter range than the motion we are interested in, momentum conservation may not matter to results that are of interest, and conventional Monte Carlo simulations can be used directly.

In summary, Monte Carlo simulations are designed to reproduce an ensemble rather than the dynamics of a particular system. As such,they are ideally suited to investigating the equilibrium properties of thermodynamic systems. However, Monte Carlo dynamics with local moves often mimic the dynamics of real systems. Thus, Monte Carlo simulations may be used to investigate the dynamics of systems when they are appropriately designed. This property will be used in Chapter 5 to simulate the dynamics of long polymers.

There is a flip side to the design of Monte Carlo dynamics to simulate actual dynamics. If our objective is the traditional objective of a Monte Carlo simulation, of obtaining an ensemble average, then the ability to simulate dynamics may not be an advantage. In some systems, the real dynamics is slow and we would prefer to speed up the process. This can often be done by knowingly introducing nonlocal moves that displace the state of the system by large distances in the space of conformations. Such nonlocal Monte Carlo dynamics have been designed for various systems. In particular, both local and nonlocal Monte Carlo dynamics for the problem of polymer dynamics will be described in Chapter 5.

Question 1.7.1 In order to perform Monte Carlo simulations, we must be able to choose steps at random and accept or reject steps with a certain probability. These operations require the availability of random numbers. We might think of the source of these random numbers as a thermal reservoir. Computers are specifically deigned to be completely deterministic. This means that inherently there is no randomness in their operation. To obtain random numbers in a computer simulation requires a deterministic algorithm that generates a sequence of numbers that look random but are not random. Such sequences are called pseudo-random numbers. Random

numbers should not be correlated to each other. However, using pseudo-random numbers, if we start a program over again we must get exactly the same sequence of numbers. The difficulties associated with the generation of random numbers are central to performing Monte Carlo computer simulations. If we assume that we have random numbers, and they are not really uncorrelated, then our results may very well be incorrect. Nevertheless, pseudo-random numbers often give results that are consistent with those expected from random numbers.

There are a variety of techniques to generate pseudo-random numbers. Many of these pseudo-random number generators are designed to provide, with equal "probability," an integer between 0 and the maximal integer possible. The maximum integer used by a particular routine on a particular machine should be checked before using it in a simulation. Some use a standard short integer which is represented by 16 bits (2 bytes).One bit represents the unused sign of the integer. This leaves 15 bits for the magnitude of the number. The pseudo-random number thus ranges up to $2^{15} - 1 = 32767$. An example of a routine that provides pseudo-random integers is the subroutine `rand()` in the ANSI C library, which is executed using a line such as:

$$k = \texttt{rand()}; \tag{1.7.27}$$

The following three questions discuss how to use such a pseudo-random number generator. Assume that it provides a standard short integer.

1. Explain how to use a pseudo-random number generator to choose a move in a Metropolis Monte Carlo simulation, Eq. (1.7.19).
2. Explain how to use a pseudo-random number generator to accept or reject a move in a Metropolis Monte Carlo simulation, Eq. (1.7.19).
3. Explain how to use a pseudo-random number generator to provide values of x with a probability $P(x)$ for x in the interval [0,1]. Hint: Use two pseudo-random numbers every step.

Solution 1.7.1

1. Given the necessity of choosing one out of M possible moves, we create a one-to-one mapping between the M moves and the integers $\{0, \ldots, M-1\}$ If M is smaller than 2^{15} we can use the value of $k = \texttt{rand()}$ to determine which move is taken next. If k is larger than $M - 1$, we don't make any move. If M is much smaller than 2^{15} then we can use only some of the bits of k. This avoids making many unused calls to `rand()`. Fewer bits can be obtained using a modulo operation. For example, if $M = 10$ we might use k modulo 16. We could also ignore values above 32759,and use k modulo 10. This also causes each move to occur with equal frequency. However, a standard word of caution about using only a few bits is that we shouldn't use the lowest order bits (i.e., the units, twos and fours bits), because they tend to be more correlated than the

higher order bits. Thus it may be best first to divide k by a small number, like 8 (or equivalently to shift the bits to the right),if it is desired to use fewer bits. If M is larger than 2^{15} it is necessary to use more than one call to `rand()` (or a random number generator that provides a 4-byte integer) so that all possible moves are accounted for.

2. Given the necessity of determining whether to accept a move with the probability P, we compare $2^{15}\,P$ with a number given by k = `rand()`. If the former is bigger we accept the move, and if it is smaller we reject the move.
3. One way to do this is to generate two random numbers r_1 and r_2. Dividing both by 32767 (or 2^{15}), we use the first random number to be the location in the interval $x = r_1/32767$. However, we use this location only if the second random number $r_2/32767$ is smaller than $P(x)$. If the random number is not used, we generate two more and proceed. This means that we will use the position x with a probability $P(x)$ as desired. Because it is necessary to generate many random numbers that are rejected, this method for generating numbers for use in performing the integral Eq. (1.7.3) is only useful if evaluations of the function $f(x)$ are much more costly than random number generation. ∎

Question 1.7.2 To compare the errors that arise from conventional numerical integration and Monte Carlo sampling, we return to Eq.(1.7.4) and Eq. (1.7.5) in this and the following question. We choose two integrals that can be evaluated analytically and for which the errors can also be evaluated analytically.

Evaluate two examples of the integral $\int P(x)f(x)dx$ over the interval $x \in [1,1]$. For the first example (1) take $f(x) = 1$, and for the second (2) $f(x) = x$. In both cases assume the probability distribution is an exponential

$$P(x) = Ae^{-\lambda x} = \left[\frac{\lambda}{e^{\lambda} - e^{-\lambda}}\right] e^{-\lambda x} \tag{1.7.28}$$

where the normalization constant A is given by the expression in square brackets.

Calculate the two integrals exactly (analytically). Then evaluate approximations to the integrals using sums over N equally spaced points, Eq.(1.7.4). These sums can also be evaluated analytically. To improve the result of the sum, you can use Simpson's rule. This modifies Eq.(1.7.4) only by subtracting 1/2 of the value of the integrand at the first and last points. The errors in evaluation of the same integral by Monte Carlo simulation are to be calculated in Question 1.7.3.

Solution 1.7.2

1. The value of the integral of $P(x)$ is unity as required by normalization. If we use a sum over equally spaced points we would have:

$$A\int_{-1}^{1} dx e^{-\lambda x} \approx \frac{A}{M}\sum_{n=-M}^{M} e^{-\lambda(n/M)} = \frac{A}{M}\sum_{n=-M}^{M} a^n \tag{1.7.29}$$

where we used the temporary definition $a = e^{-\lambda/M}$ to obtain

$$A\int_{-1}^{1} dx e^{-\lambda x} \approx \frac{A}{M}\frac{(a^{M+1} - a^{-M})}{a-1} = \frac{A}{M}\frac{(e^{\lambda}e^{\lambda/M} - e^{-\lambda})}{e^{\lambda/M} - 1} \tag{1.7.30}$$

Expanding the answer in powers of λ/M gives:

$$\begin{aligned} A\int_{-1}^{1} dx e^{-\lambda x} \approx & A\frac{(e^{\lambda} - e^{-\lambda})}{\lambda} + A\frac{(e^{\lambda} + e^{-\lambda})}{2M} + A\frac{\lambda(e^{\lambda} - e^{-\lambda})}{2M^2} + \\ & = 1 + \frac{\lambda}{2M}\tanh(\lambda) + \frac{\lambda^2}{2M^2} + \end{aligned} \tag{1.7.31}$$

The second term can be eliminated by noting that the sum could be evaluated using Simpson's rule by subtracting 1/2 of the contribution of the end points. Then the third term gives an error of $\lambda^2/2M^2$. This is the error in the numerical approximation to the average of $f(x) = 1$.

2. For $f(x) = x$ the exact integral is:

$$\begin{aligned} A\int_{-1}^{1} dx\, x e^{-\lambda x} &= -A\frac{d}{d\lambda}\int_{-1}^{1} dx e^{-\lambda x} = -A\frac{d}{d\lambda}\left[\frac{e^{\lambda} - e^{-\lambda}}{\lambda}\right] \\ &= -\coth(\lambda) + (1/\lambda) \end{aligned} \tag{1.7.32}$$

while the sum is:

$$\begin{aligned} A\int_{-1}^{1} dx\, x e^{-\lambda x} &\approx \frac{A}{M^2}\sum_{n=-M}^{M} n e^{-\lambda(n/M)} = \frac{A}{M^2}\sum_{n=-M}^{M} n a^n \\ &= \frac{A}{M^2} a\frac{d}{da}\sum_{n=-M}^{M} a^n = \frac{A}{M^2} a\frac{d}{da}\frac{(a^{M+1} - a^{-M})}{a-1} \\ &= \frac{A}{M^2}\frac{((M+1)a^{M+1} + Ma^{-M})}{a-1} + \frac{A}{M^2}\frac{a(a^{M+1} - a^{-M})}{(a-1)^2} \\ &= \frac{A}{M}\frac{(e^{\lambda}e^{\lambda/M} + e^{-\lambda})}{e^{\lambda/M} - 1} + \frac{A}{M^2}\frac{(e^{\lambda}e^{\lambda/M})}{e^{\lambda/M} - 1} + \frac{A}{M^2}\frac{e^{\lambda/M}(e^{\lambda}e^{\lambda/M} - e^{-\lambda})}{(e^{\lambda/M} - 1)^2} \end{aligned} \tag{1.7.33}$$

With some assistance from Mathematica,the expansion to second order in λ/M is:

$$\begin{aligned} &= -A\frac{(e^{\lambda} - e^{-\lambda})}{\lambda^2} + A\frac{(e^{\lambda} + e^{-\lambda})}{\lambda} + \frac{A}{M}\frac{(e^{\lambda} - e^{-\lambda})}{2} + \frac{A}{M^2}\frac{11}{12}(e^{\lambda} - e^{-\lambda}) + \\ &= -1/\lambda + \coth(\lambda) + \frac{\lambda}{2M} + \frac{11}{12}\frac{\lambda}{M^2} + \end{aligned} \tag{1.7.34}$$

The first two terms are the correct result. The third term can be seen to be eliminated using Simpson's rule. The fourth term is the error. ∎

Question 1.7.3 Estimate the errors in performing the same integrals as in Question 1.7.2 using a Monte Carlo ensemble sampling with N terms as in Eq.(1.7.5). It is not necessary to evaluate the integrals to evaluate the errors.

Solution 1.7.3

1. The errors in performing the integral for $f(x) = 1$ are zero, since the Monte Carlo sampling would be given by the expression:

$$<1>_{P(s)} = \frac{1}{N}\sum_{s:P(s)}^{N} 1 = 1 \tag{1.7.35}$$

One way to think about this result is that Monte Carlo takes advantage of the normalization of the probability, which the technique of summing the integrand over equally spaced points cannot do. This knowledge makes this integral trivial, but it is also of use in performing other integrals.

2. To evaluate the error for the integral over $f(x) = x$ we use an argument based on the sampling error of different regions of the integral. We break up the domain $[-1,1]$ into q regions of size $\Delta x = 2/q$. Each region is assumed to have a significant number of samples. The number of these samples is approximately given by:

$$NP(x)\Delta x \tag{1.7.36}$$

If this were the exact number of samples as q increased,then the integral would be exact. However, since we are picking the points at random, there will be a deviation in the number of these from this ideal value. The typical deviation, according to the discussion in Section 1.2 of random walks, is the square root of this number. Thus the error in the sum

$$\sum_{s:P(s)}^{N} f(x) \tag{1.7.37}$$

from a particular interval Δx is

$$(NP(x)\Delta x)^{1/2} f(x) \tag{1.7.38}$$

Since this error could have either a positive or negative sign, we must take the square root of the sum of the squares of the error in each region to give us the total error:

$$\left|\int P(x)f(x) - \frac{1}{N}\sum_{s:P(s)}^{N} f(x)\right| \approx \frac{1}{N}\sqrt{\sum NP(x)\Delta x\, f(x)^2} \approx \frac{1}{\sqrt{N}}\sqrt{\int P(x)f(x)^2} \tag{1.7.39}$$

For $f(x) = x$ the integral in the square root is:

$$\int Ae^{-\lambda x} f(x)^2 dx = \int Ae^{-\lambda x} x^2 dx = A\frac{d^2}{d\lambda^2}\frac{(e^{\lambda} - e^{-\lambda})}{\lambda} = \frac{2}{\lambda^2} - \frac{2\coth(\lambda)}{\lambda} + 1 \tag{1.7.40}$$

The approach of Monte Carlo is useful when the exponential is rapidly decaying. In this case, $\lambda >> 1$, and we keep only the third term and have an error that is just of magnitude $1/\sqrt{N}$. Comparing with the sum over equally spaced points from Question 1.7.2, we see that the error in Monte Carlo is independent of λ for large λ, while it grows for the sum over equally spaced points. This is the crucial advantage of the Monte Carlo method. However, for a fixed value of λ we also see that the error is more slowly decreasing with N than the sum over equally spaced points. So when a large number of samples is possible, the sum over equally spaced points is more rapidly convergent. ∎

Question 1.7.4 How would the discrete nature of the integer random numbers described in Question 1.7.1 affect the ensemble sampling? Answer qualitatively. Is there a limit to the accuracy of the integral in this case?

Solution 1.7.4 The integer random numbers introduce two additional sources of error, one due to the sampling interval along the x axis and the other due to the imperfect approximation of $P(x)$. In the limit of a large number of samples, each of the possible values along the x axis would be sampled equally. Thus, the ensemble sum would reduce to a sum of the integrand over equally spaced points. The number of points is given by the largest integer used (e.g., 2^{15}). This limits the accuracy accordingly. ∎

1.7.3 *Perron-Frobenius theorem*

The Perron-Frobenius theorem is tied to our understanding of the ergodic theorem and the use of Monte Carlo simulations for the representation of ensemble averages. The theorem only applies to a system with a finite space of possible states. It says that a transition matrix that is irreducible must ultimately lead to a stable limiting probability distribution. This distribution is unique, and thus depends only on the transition matrix and not on the initial conditions. The Perron-Frobenius theorem assumes an irreducible matrix,so that starting from any state,there is some path by which it is possible to reach every other state of the system. If this is not the case,then the theorem can be applied to each subset of states whose transition matrix is irreducible.

In a more general form than we will discuss,the Perron-Frobenius theorem deals with the effect of matrix multiplication when all of the elements of a matrix are positive. We will consider it only for the case of a transition matrix in a Markov chain, which also satisfies the normalization condition, Eq. (1.7.16). In this case, the proof of the Perron-Frobenius theorem follows from the statement that there cannot be any eigenvalues of the transition matrix that are larger than one. Otherwise there would be a vector that would increase everywhere upon matrix multiplication. This is not

possible, because probability is conserved. Thus if the probability increases in one place it must decrease someplace else, and tend toward the limiting distribution.

A difficulty in the proof of the theorem arises from dealing with the case in which there are deterministic currents through the system: e.g., ballistic motion in a circular path. An example for a two-state system would be

$$\begin{aligned} P(1|1) = 0 \;\; P(1|-1) = 1 \\ P(-1|1) = 1 \; P(-1|-1) = 0 \end{aligned} \tag{1.7.41}$$

In this case, a system in the state $s = +1$, goes into $s = -1$, and a system in the state $s = -1$ goes into $s = +1$. The limiting behavior of this Markov chain is of two probabilities that alternate in position without ever settling down into a limiting distribution. An example with three states would be

$$\begin{aligned} &P(1|1) = 0 \quad P(1|2) = 1 \quad P(1|3) = 1 \\ &P(2|1) = .5 \quad P(2|2) = 0 \quad P(2|3) = 0 \\ &P(3|1) = .5 \quad P(3|2) = 0 \quad P(3|3) = 0 \end{aligned} \tag{1.7.42}$$

Half of the systems with $s = 1$ make transitions to $s = 2$ and half to $s = 3$. All systems with $s = 2$ and $s = 3$ make transitions to $s = 1$. In this case there is also a cyclical behavior that does not disappear over time. These examples are special cases, and the proof shows that they are special. It is sufficient, for example, for there to be a single state where there is some possibility of staying in the same state. Once this is true, these examples of cyclic currents do not apply and the system will settle down into a limiting distribution.

We will prove the Perron-Frobenius theorem in a few steps enumerated below. The proof is provided for completeness and reference, and can be skipped without significant loss for the purposes of this book. The proof relies upon properties of the eigenvectors and eigenvalues of the transition matrix. The eigenvectors need not always be positive, real or satisfy the normalization condition that is usually applied to probability distributions, $P(s)$. Thus we use $v(s)$ to indicate complex vectors that have a value at every possible state of the system.

Given an irreducible real nonnegative matrix ($P(s'|s) \geq 0$) satisfying

$$\sum_{s'} P(s'|s) = 1 \tag{1.7.43}$$

we have:

1. Applying $P(s'|s)$ cannot increase the value of all elements of a nonnegative vector, $v(s') \geq 0$:

$$\min_{s'} \left(\frac{1}{v(s')} \sum_{s} P(s'|s) v(s) \right) \leq 1 \tag{1.7.44}$$

To avoid infinities, we can assume that the minimization only includes s' such that $v(s') \neq 0$.

Proof: Assume that Eq. (1.7.44) is not true. In this case

$$\sum_s P(s'|s)v(s) > v(s') \tag{1.7.45}$$

for all $v(s') \neq 0$, which implies

$$\sum_{s'} \sum_s P(s'|s)v(s) > \sum_{s'} v(s') \tag{1.7.46}$$

Using Eq. (1.7.43), the left is the same as the right and the inequality is impossible.

2. The magnitude of eigenvalues of $P(s'|s)$ is not greater than one.

Proof: Let $v(s)$ be an eigenvector of $P(s'|s)$ with eigenvalue λ:

$$\sum_s P(s'|s)v(s) = \lambda v(s') \tag{1.7.47}$$

Then:

$$\sum_s P(s'|s)|v(s)| \geq |\lambda||v(s')| \tag{1.7.48}$$

This inequality follows because each term in the sum on the left has been made positive. If all terms started with the same phase, then equality holds. Otherwise, inequality holds. Comparing Eq. (1.7.48) with Eq. (1.7.44), we see that $|\lambda| \leq 1$.

If $|\lambda| = 1$,then equality must hold in Eq.(1.7.48), and this implies that $|v(s)|$, the vector whose elements are the magnitudes of $v(s)$,is an eigenvector with eigenvalue 1. Steps 3–5 show that there is one such vector which is strictly positive (greater than zero) everywhere.

3. $P(s'|s)$ has an eigenvector with eigenvalue $\lambda = 1$. We use the notation $v_1(s)$ for this vector.

Proof: The existence of such an eigenvector follows from the existence of an eigenvector of the transpose matrix with eigenvalue $\lambda = 1$. Eq.(1.7.43) implies that the vector $v(s) = 1$ (one everywhere) is an eigenvector of the transpose matrix with eigenvalue $\lambda = 1$. Thus $v_1(s)$ exists, and by step 2 we can take it to be real and nonnegative, $v_1(s) \geq 0$. We can, however, assume more, as the following shows.

4. An eigenvector of $P(s'|s)$ with eigenvalue 1 must be strictly positive, $v_1(s) > 0$.

Proof: Define a new Markov chain given by the transition matrix

$$Q(s'|s) = (P(s'|s) + \delta_{s,s'}) / 2 \tag{1.7.49}$$

Applying $Q(s'|s)$ $N - 1$ times to any vector $v_1(s) \geq 0$ must yield a vector that is strictly positive. This follows because $P(s'|s)$ is irreducible. Starting with unit probability at any one value of s, after $N - 1$ steps we will move some probability everywhere. Also, by the construction of $Q(s'|s)$, any s which has a nonzero probability at one time will continue to have a nonzero probability at all later times. By linear superposition,this applies to any initial probability distribution. It also applies to any unnormalized vector $v_1(s) \geq 0$. Moreover, if $v_1(s)$ is an eigenvector of $P(s'|s)$ with eigenvalue one,then it

is also an eigenvector of $Q(s'|s)$ with the same eigenvalue. Since applying $Q(s'|s)$ to $v_1(s)$ changes nothing, applying it $N-1$ times also changes nothing. We have just proven that $v_1(s)$ must be strictly positive.

5. There is only one linearly independent eigenvector of $P(s'|s)$ with eigenvalue $\lambda = 1$.

Proof: Assume there are two such eigenvectors: $v_1(s)$ and $v_2(s)$. Then we can make a linear combination $c_1 v_1(s) + c_2 v_2(s)$,so that at least one of the elements is zero and others are positive. This linear combination is also an eigenvector of $P(s'|s)$ with eigenvalue $\lambda = 1$, which violates step 4. Thus there is exactly one eigenvector of $P(s'|s)$ with eigenvalue $\lambda = 1$, $v_1(s)$:

$$\sum_s P(s'|s)v_1(s) = v_1(s') \tag{1.7.50}$$

6. Either $P(s'|s)$ has only one eigenvalue with $|\lambda| = 1$ (in which case $\lambda = 1$), or it can be written as a cyclical flow.

Proof: Steps 2 and 5 imply that all eigenvectors of $P(s'|s)$ with eigenvalues λ satisfying $|\lambda| = 1$ can be written as:

$$v_i(s) = D_i(s)v_1(s) = e^{i\phi_i(s)}v_1(s) \tag{1.7.51}$$

As indicated, $D_i(s)$ is a vector with elements of magnitude one, $|D_i(s)| = 1$. We can write

$$\sum_s P(s'|s)D_i(s)v_1(s) = \lambda_i D_i(s')v_1(s') \tag{1.7.52}$$

There cannot be any terms in the sum on the left of Eq.(1.7.52) that add terms of different phase. If there were, then we would have a smaller magnitude than adding the absolute values, which would not agree with Eq.(1.7.50). Thus we can assign all of the elements of $D_i(s)$ into groups that have the same phase. $P(s'|s)$ cannot allow transitions to occur from any two of these groups into the same group. Since $P(s'|s)$ is irreducible, the only remaining possibility is that the different groups are connected in a ring with the first mapped onto the second, and the second mapped onto the third, and so on until we return to the first group. In particular, if there are any transitions between a site and itself this would violate the requirements and we could have no complex eigenvalues.

7. A Markov chain governed by an irreducible transition matrix, which has only one eigenvector, $v_1(s)$ with $|\lambda| = 1$,has a limiting distribution over long enough times which is proportional to this eigenvector. Using $P^t(s'|s)$ to represent the effect of applying $P(s'|s)$ t times, we must prove that:

$$\lim_{t\to\infty} v(s;t) = \lim_{t\to\infty} \sum_s P^t(s'|s)v(s) = cv_1(s') \tag{1.7.53}$$

for $v(s) \geq 0$. The coefficient c depends on the normalization of $v(s)$ and $v_1(s)$. If both are normalized so that the total probability is one, then conservation of probability implies that $c = 1$.

Proof: We write the matrix $P(s'|s)$ in the Jordan normal form using a similarity transformation. In matrix notation:

$$\mathbf{P} = \mathbf{S}^{-1}\mathbf{J}\mathbf{S} \tag{1.7.54}$$

$\mathbf{J}$ consists of a block diagonal matrix.Each of the block matrices along the diagonal is of the form

$$\mathbf{N} = \begin{pmatrix} \lambda & 1 & 0 & 0 \\ 0 & \lambda & \ddots & 0 \\ 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \lambda \end{pmatrix} \tag{1.7.55}$$

where λ is an eigenvalue of $\mathbf{P}$. In this block the only nonzero elements are λs on the diagonal, and 1s just above the diagonal.

Since $\mathbf{P}^t = \mathbf{S}^{-1}\mathbf{J}^t\mathbf{S}$, we consider $\mathbf{J}^t$, which consists of diagonal blocks $\mathbf{N}^t$. We prove that $\mathbf{N}^t \to 0$ as $t \to \infty$ for $|\lambda| < 1$. This can be shown by evaluating explicitly the matrix elements. The qth element above the diagonal of $\mathbf{N}^t$ is:

$$\lambda^{t-q}\binom{t}{q} \tag{1.7.56}$$

which vanishes as $t \to \infty$.

Since 1 is an eigenvalue with only one eigenvector, there must be one 1×1 block along the diagonal of $\mathbf{J}$ for the eigenvalue 1. Then $\mathbf{J}^t$ as $t \to \infty$ has only one nonzero element which is a 1 on the diagonal. Eq.(1.7.53) follows, because applying the matrix $\mathbf{P}^t$ always results in the unique column of $\mathbf{S}^{-1}$ that corresponds to the nonzero diagonal element of $\mathbf{J}^t$. By our assumptions,this column must be proportional to $v_1(s)$. This completes our proof and discussion of the Perron-Frobenius theorem.

1.7.4 *Minimization*

At low temperatures, a thermodynamic system in equilibrium will be found in its minimum energy configuration. For this and other reasons,it is often useful to identify the minimum energy configuration of a system without describing the full ensemble. There are also many other problems that can be formulated as minimization or optimization problems.

Minimization problems are often described in a d-dimensional space of continuous variables. When there is only a single valley in the parameter space of the problem,there are a variety of techniques that can be used to obtain this minimum. They may be classified into direct search and gradient-based techniques. In this section we focus on the single-valley problem. In Section 1.7.5 we will discuss what happens when there is more than one valley.

Direct search techniques involve evaluating the energy at various locations and closing in on the minimum energy. In one dimension, search techniques can be very effective. The key to a search is bracketing the minimum energy. Then

each energy evaluation is used to geometrically shrink the possible domain of the minimum.

We start in one dimension by looking at the energy at two positions s_1 and s_2 that are near each other. If the left of the two positions s_1 is higher in energy $E(s_1) > E(s_2)$, then the minimum must be to its right. This follows from our assumption that there is only a single valley—the energy rises monotonically away from the minimum and therefore cannot be lower than $E(s_2)$,anywhere to the left of s_1. Evaluating the energy at a third location s_3 to the right of s_2 further restricts the possible locations of the minimum. If $E(s_3)$ is also greater than the middle energy location $E(s_3) > E(s_2)$, then the minimum must lie between s_1 and s_3. Thus, we have successfully bracketed the minimum. Otherwise, we have that $E(s_3) < E(s_2)$, and the minimum must lie to the right of s_2. In this case we look at the energy at a location s_4 to the right of s_3. This process is continued until the energy minimum is bracketed. To avoid taking many steps to the right,the size of the steps to the right can be taken to be an increasing geometric series, or may be based on an extrapolation of the function using the values that are available.

Once the energy minimum is bracketed, the segment is bisected again and again to locate the energy minimum. This is an iterative process. We describe a simple version of this process that can be easily implemented. An iteration begins with three locations $s_1 < s_2 < s_3$. The values of the energy at these locations satisfy $E(s_1), E(s_3) > E(s_2)$. Thus the minimum is between s_1 and s_3. We choose a new location s_4, which in even steps is $s_4 = (s_1 + s_2) / 2$ and in odd steps is $s_4 = (s_2 + s_3)/2$. Then we eliminate either s_1 or s_3. The one that is eliminated is the one next to s_2 if $E(s_2) > E(s_4)$, or the one next to s_4 if $E(s_2) < E(s_4)$. The remaining three locations are relabled to be s_1, s_2, s_3 for the next step. Iterations stop when the distance between s_1 and s_3 is smaller than an error tolerance which is set in advance. More sophisticated versions of this algorithm use improved methods for selecting s_4 that accelerate the convergence.

In higher-dimension spaces,direct search can be used. However, mapping a multidimensional energy surface is much more difficult. Moreover, the exact logic that enables an energy minimum to be bracketed within a particular domain in one dimension is not possible in higher-dimension spaces. Thus, techniques that make use of a gradient of the function are typically used even if the gradient must be numerically evaluated. The most common gradient-based minimization techniques include steepest descent, second order and conjugate gradient.

Steepest descent techniques involve taking steps in the direction of the most rapid descent direction as determined by the gradient of the energy. This is the same as using a first-order expansion of the energy to determine the direction of motion toward lower energy. Illustrating first in one dimension, we start from a position s_1 and write the expansion as:

$$E(s) = E(s_1) + (s - s_1)\left.\frac{dE(s)}{ds}\right|_{s_1} + O((s - s_1)^2) \tag{1.7.57}$$

We now take a step in the direction of the minimum by setting:

$$s_2 = s_1 - c\left.\frac{dE(s)}{ds}\right|_{s_1} \tag{1.7.58}$$

From the expansion we see that for small enough c, $E(s_2)$ must be smaller than $E(s_1)$. The problem is to carefully select c so that we do not go too far. If we go too far we may reach beyond the energy minimum and increase the energy. We also do not want to make such a small step that many steps will be needed to reach the minimum. We can think of the sequence of configurations we pick as a time sequence, and the process we use to pick the next location as an iterative map. Then the minimum energy configuration is a fixed point of the iterative map given by Eq.(1.7.58). From a point near to the minimum we can have all of the behaviors described in Section 1.1—stable (converging) and unstable (diverging), both of these with or without alternation from side to side of the minimum.Of particular relevance is the discussion in Question 1.1.12 that suggests how c may be chosen to stabilize the iterative map and obtain rapid convergence.

When s is a multidimensional variable, Eq. (1.7.57) and Eq. (1.7.58) both continue to apply as long as the derivative is replaced by the gradient:

$$E(s) = E(s_1) + (s - s_1)\cdot \left.{}_s E(s)\right|_{s_1} + O((s - s_1)^2) \tag{1.7.59}$$

$$s_2 = s_1 - c \left.{}_s E(s)\right|_{s_1} \tag{1.7.60}$$

Since the direction opposite to the gradient is the direction in which the energy decreases most rapidly, this is known as a steepest descent technique. For the multidimensional case it is more difficult to choose a consistent value of c, since the behavior of the function may not be the same in different directions. The value of c may be chosen "on the fly" by making sure that the new energy is smaller than the old. If the current value of c gives a value $E(s_2)$ which is larger than $E(s_1)$ then c is reduced. We can improve upon this by looking along the direction of the gradient and considering the energy to be a function of c:

$$E(s_1 - c \left.{}_s E(s)\right|_{s_1}) \tag{1.7.61}$$

Then c can be chosen by finding the actual minimum in this direction using the search technique that works well in one dimension.

Gradient techniques work well when different directions in the energy have the same behavior in the vicinity of the minimum energy. This means that the second derivative in different directions is approximately the same. If the second derivatives are very different in different directions,then the gradient technique tends to bounce back and forth perpendicular to the direction in which the second derivative is very small, without making much progress toward the minimum (Fig. 1.7.3). Improvements of the gradient technique fall into two classes. One class of techniques makes direct use of the second derivatives, the other does not. If we expand the energy to second order at the present best guess for the minimum energy location s_1 we have

$$E(s) = E(s_1) + (s - s_1) \left.{}_s E(s)\right|_{s_1} + (s - s_1) \overleftarrow{}_s \vec{}_s E(s)\Big|_{s_1} (s - s_1) + O((s - s_1)^3) \tag{1.7.62}$$

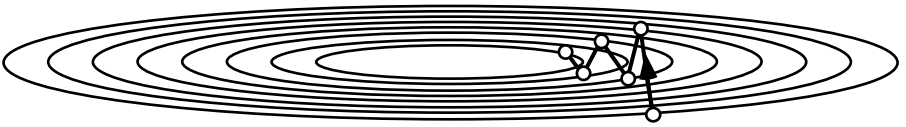

Figure 1.7.3 Illustration of the difficulties in finding a minimum energy by steepest descent when the second derivative is very different in different directions. The steps tend to oscillate and do not make progress toward the minimum along the flat direction. ∎

Setting the gradient of this expression to zero gives the next approximation for the minimum energy location s_2 as:

$$s_2 = s_1 - \frac{1}{2}\left({}_s{}_s E(s)\Big|_{s_1} \right)^{-1} {}_s E(s)\Big|_{s_1} \tag{1.7.63}$$

This, in effect, gives a better description of the value of c for Eq. 1.7.60, which turns out to be a matrix inversely related to the second-order derivatives. Steps are large in directions in which the second derivative is small. If the second derivatives are not easily available, approximate second derivatives are used that may be improved upon as the minimization is being performed. Because of the need to evaluate the matrix of second-order derivatives and invert the matrix,this approach is not often convenient. In addition, the use of second derivatives assumes that the expansion is valid all the way to the minimum energy. For many minimization problems, this is not valid enough to be a useful approximation. Fortunately, there is a second approach called the conjugate gradient technique that often works as well and sometimes better.

Conjugate gradient techniques make use of the gradient but are designed to avoid the difficulties associated with long narrow wells where the steepest descent techniques result in oscillations. This is done by starting from a steepest descent in the first step of the minimization. In the second step, the displacement is taken to be along a direction that does not include the direction taken in the first step. Explicitly, let v_i be the direction taken in the ith step, then the first two directions would be:

$$\begin{aligned} v_1 &= - \; {}_s E(s)\Big|_{s_1} \\ v_2 &= - \; {}_s E(s)\Big|_{s_2} + v_1 \frac{(v_1 \quad {}_s E(s)\Big|_{s_2})}{v_1 \;\; v_1} \end{aligned} \tag{1.7.64}$$

This ensures that v_2 is orthogonal to v_1. Subsequent directions are made orthogonal to some number of previous steps. The use of orthogonal directions avoids much of the problem of bouncing back and forth in the energy well.

Monte Carlo simulation can also be used to find minimum energy configurations if the simulations are done at zero temperature. A zero temperature Monte Carlo means that the steps taken always reduce the energy of the system. This approach works not only for continuous variables, but also for the discrete variables like in the Ising model. For the Ising model,the zero temperature Monte Carlo described above

and the zero temperature Glauber dynamics are the same. Every selected spin is placed in its low energy orientation—aligned with the local effective field.

None of these techniques are suited to finding the minimum energy configuration if there are multiple energy minima, and we do not know if we are located near the correct minimum energy location. One way to address this problem is to start from various initial configurations and to look for the local minimum nearby. By doing this many times it might be possible to identify the global minimum energy. This works when there are only a few different energy minima. There are no techniques that guarantee finding the global minimum energy for an arbitrary energy function $E(s)$. However, by using Monte Carlo simulations that are not at $T = 0$, a systematic approach called simulated annealing has been developed to try to identify the global minimum.

1.7.5 *Simulated annealing*

Simulated annealing was introduced relatively recently as an approach to finding the global minimum when the energy or other optimization function contains many local minima. The approach is based on the physical process of heating a system and cooling it down slowly. The minimum energy for many simple materials is a crystal. If a material is heated to a liquid or vapor phase and cooled rapidly, the material does not crystallize. It solidifies as a glass or amorphous solid. On the other hand, if it is cooled slowly, crystals may form. If the material is formed out of several different kinds of atoms, the cooling may also result in phase separation into particular compounds or atomic solids. The separated compounds are lower in energy than a rapidly cooled mixture.

Simulated annealing works in much the same way. A Monte Carlo simulation is started at a high temperature. Then the temperature is lowered according to a cooling schedule until the temperature is so low that no additional movements are likely. If the procedure is effective, the final energy should be the lowest energy of the simulation. We could also keep track of the energy during the simulation and take the lowest value, and the configuration at which the lowest value was reached.

In general, simulated annealing improves upon methods that find only a local minimum energy, such as steepest descent, discussed in the previous section. For some problems, the improvement is substantial. Even if the minimum energy that is found is not the absolute minimum in energy of the system, it may be close. For example, in problems where there are many configurations that have roughly the same low energy, simulated annealing may find one of the low-energy configurations.

However, simulated annealing does not work well for all problems, and for some problems it fails completely. It is also true that annealing of physical materials does not always result in the lowest energy conformation. Many materials, even when cooled slowly, result in polycrystalline materials, disordered solids and mixtures. When it is important for technological reasons to reach the lowest energy state, special techniques are often used. For example, the best crystal we know how to make is silicon. In order to form a good silicon crystal, it is grown using careful nonuniform cooling. A single crystal can be gradually pulled from a liquid that solidifies only on the surfaces of the existing crystal. Another technique for forming crystals is growth

from the vapor phase, where atoms are deposited on a previously formed crystal that serves as a template for the continuing growth. The difficulties inherent in obtaining materials in their lowest energy state are also apparent in simulations.

In Section 1.4 we considered the cooling of a two-state system as a model of a glass transition. We can think about this simulation to give us clues about why both physical and simulated annealing sometimes fail to find low energy states of the system. We saw that using a constant cooling rate leaves some systems stuck in the higher energy well. When there are many such high energy wells then the system will not be successful in finding a low energy state. The problem becomes more difficult if the height of the energy barrier between the two wells is much larger than the energy difference between the upper and lower wells. In this case, at higher temperatures the system does not care which well it is in. At low temperatures when it would like to be in the lower energy well, it cannot overcome the barrier. How well the annealing works in finding a low energy state depends on whether we care about the energy scale characteristic of the barrier, or characteristic of the energy difference between the two minima.

There is another characteristic of the energy that can help or hurt the effectiveness of simulated annealing. Consider a system where there are many local minimum energy states (Fig. 1.7.4). We can think about the effect of high temperatures as placing the system in one of the many wells of the energy minima. These wells are called basins of attraction.A system in a particular basin of attraction will go into the minimum energy configuration of the basin if we suddenly cool to zero temperature. We

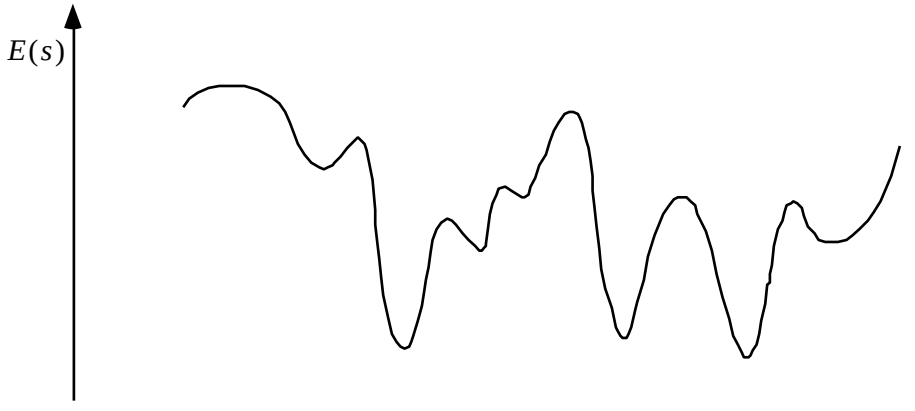

Figure 1.7.4 Schematic plot of a system energy $E(s)$ as a function of a system coordinate s. In simulated annealing, the location of a minimum energy is sought by starting from a high temperature Monte Carlo and cooling the system to a low temperature. At the high temperature the system has a high kinetic energy and explores all of the possible configurations. As the temperature is cooled it descends into one of the wells, called basins of attraction, and cannot escape. Finally, when the temperature is very low it loses all kinetic energy and sits in the bottom of the well. Minima with larger basins of attraction are more likely to capture the system. Simulated annealing works best when the lowest-energy minima have the largest basins of attraction. ∎

also can see that the gradual cooling in simulated annealing will result in low energy states if the size of the basin of attraction increases with the depth of the well. This means that at high temperatures the system is more likely to be in the basin of attraction of a lower energy minimum. Thus,simulated annealing works best when energy varies in the space in such a way that deep energy minima also have large basins of attraction. This is sometimes but not always true both in physical systems and in mathematical optimization problems.

Another way to improve the performance of simulated annealing is to introduce nonlocal Monte Carlo steps. If we understand the characteristics of the energy, we can design steps that take us through energy barriers. The problem with this approach is that if we don't know the energy surface well enough, then moving around in the space by arbitrary nonlocal steps will result in attempts to move to locations where the energy is high. These steps will be rejected by the Monte Carlo and the nonlocal moves will not help. An example where nonlocal Monte Carlo moves can help is treatments of low-energy atomic configurations in solids. Nonlocal steps can allow atoms to move through each other, switching their relative positions, instead of trying to move gradually around each other.

Finally, for the success of simulated annealing, it is often necessary to design carefully the cooling schedule.Generally, the slower the cooling the more likely the simulation will end up in a low energy state. However, given a finite amount of computer and human time,it is impossible to allow an arbitrarily slow cooling. Often there are particular temperatures where the cooling rate is crucial. This happens at phase transitions, such as at the liquid-to-solid phase boundary. If we know of such a transition, then we can cool rapidly down to the transition, cool very slowly in its vicinity and then speed up thereafter. The most difficult problems are those where there are barriers of varying heights leading to a need to cool slowly at all temperatures.

For some problems the cooling rate should be slowed as the temperature becomes lower. One way to achieve this is to use a logarithmic cooling schedule. For example, we set the temperature $T(t)$ at time step t of the Monte Carlo, to be:

$$T(t) = T_0 / \ln(t / t_0 + 1) \tag{1.7.65}$$

where t_0 and T_0 are parameters that must be chosen for the particular problem. In Question 1.7.5 we show that for the two-state system,if $kT_0 > (E_B - E_1)$,then the system will always relax into its ground state.

Question 1.7.5: Show that by using a logarithmic cooling schedule, Eq. (1.7.65), where $kT_0 > (E_B - E_1)$,the two-state system of Section 1.4 always relaxes into the ground state. To simplify the problem, consider an incremental time Δt during which the temperature is fixed.Show that the system will still relax to the equilibrium probability over this incremental time, even at low temperatures.

Solution 1.7.5: We write the solution of the time evolution during the incremental time Δt from Eq. (1.4.45) as:

$$P(1;t + \Delta t) - P(1;\infty) = (P(1;t) - P(1;\infty))e^{-t/\tau(t)} \tag{1.7.66}$$

where $P(1;\infty)$ is the equilibrium value of the probability for the temperature $T(t)$. $\tau(t)$is the relaxation time for the temperature$T(t)$. In order for relaxation to occur we must have that $e^{-t/\tau(t)}<<1$, equivalently:

$$t / \tau(t) >> 1 \tag{1.7.67}$$

We calculate $\tau(t)$ from Eq. (1.4.44):

$$\begin{aligned} 1/\tau(t) &= \nu(e^{-(E_B-E_1)/kT(t)} + e^{-(E_B-E_{-1})/kT(t)}) \\ &> \nu e^{-(E_B-E_1)/kT(t)} = \nu(t/t_0+1)^{-\gamma} \end{aligned} \tag{1.7.68}$$

where we have substituted Eq. (1.7.65) and defined $\gamma = (E_B - E_1)/kT_0$. We make the reasonable assumption that we start our annealing at a high temperature where relaxation is not a problem. Then by the time we get to the low temperatures that are of interest, $t >> t_0$, so:

$$1/\tau(t) > 2\nu(t/t_0)^{-\gamma} \tag{1.7.69}$$

and

$$t/\tau(t) > \nu t_0^{\gamma} t^{1-\gamma} \tag{1.7.70}$$

For $\gamma < 1$ the right-hand side increases with time and thus the relaxation improves with time according to Eq. (1.7.67). If relaxation occurs at higher temperatures, it will continue to occur at all lower temperatures despite the increasing relaxation time. ∎

1.8 Information

Ultimately, our ability to quantify complexity (How complex is it?) requires a quantification of information (How much information does it take to describe it?). In this section, we discuss information. We will also need computation theory described in Section 1.9 to discuss complexity in Chapter 8.A quantitative theory of information was developed by Shannon to describe the problem of communication. Specifically, how much information can be communicated through a transmission channel (e.g., a telephone line) with a specified alphabet of letters and a rate at which letters can be transmitted. The simplest example is a binary alphabet consisting of two characters (digits) with a fixed rate of binary digits (bits) per second. However, the theory is general enough to describe quite arbitrary alphabets,letters of variable duration such as are involved in Morse code, or even continuous sound with a specified band-width. We will not consider many of the additional applications,our objective is to establish the basic concepts.

1.8.1 *The amount of information in a message*

We start by considering the information content of a string of digits $s = (s_1 s_2 ... s_N)$. One might naively expect that information is contained in the state of each digit. However, when we receive a digit, we not only receive information about what the digit is, but

also what the digit is not. Let us assume that a digit in the string of digits we receive is the number 1. How much information does this provide? We can contrast two different scenarios—binary and hexadecimal digits:

1. There were two possibilities for the number, either 0 or 1.

2. There were sixteen possibilities for the number {0, 1, 2,3,4,5, 6, 7,8, 9, A, B, C, D, E, F}.

In which of these did the "1" communicate more information? Since the first case provides us with the information that it is "not 0," while the second provides us with the information that it is "not 0," "not 2," "not 3," etc., the second provides more information. Thus there is more information in a digit that can have sixteen states than a digit that can have only two states. We can quantify this difference if we consider a binary representation of hexadecimal digits {0000,0001,0010,0011,…,1111}. It takes four binary digits to represent one hexadecimal digit. The hexadecimal number 1 is represented as 0001 in binary form and uses four binary digits. Thus a hexadecimal 1 contains four times as much information as a binary 1.

We note that the amount of information does not depend on the particular value that is taken by the digit. For hexadecimal digits, consider the case of a digit that has the value 5. Is there any difference in the amount of information given by the 5 than if it were 1? No, either number contains the same amount of information.

This illustrates that information is actually contained in the distinction between the state of a digit compared to the other possible states the digit may have. In order to quantify the concept of information, we must specify the number of possible states. Counting states is precisely what we did when we defined the entropy of a system in Section 1.3. We will see that it makes sense to define the information content of a string in the same way as the entropy—the logarithm of the number of possible states of the string:

$$I(s) = \log_2(\Omega) \tag{1.8.1}$$

By convention, the information is defined using the logarithm base two. Thus, the information contained in a single binary digit which has two possible states is $\log_2(2) = 1$. More generally, the number of possible states in a string of N bits, with each bit taking one of two values (0 or 1) is 2^N. Thus the information in a string of N bits is (in what follows the function log( ) will be assumed to be base two):

$$I(s) = \log(2^N) = N \tag{1.8.2}$$

Eq.(1.8.2) says that each bit provides one unit of information. This is consistent with the intuition that the amount of information grows linearly with the length of the string. The logarithm is essential, because the number of possible states grows exponentially with the length of the string, while the information grows linearly.

It is important to recognize that the definition of information we have given assumes that each of the possible realizations of the string has equal a priori probability. We use the phrase a priori to emphasize that this refers to the probability prior to receipt of the string—once the string has arrived there is only one possibility.

To think about the role of probability we must discuss further the nature of the message that is being communicated. We construct a scenario involving a sender and a receiver of a message. In order to make sure that the recipient of the message could not have known the message in advance (so there is information to communicate), we assume that the sender of the information is sending the result of a random occurrence, like the flipping of a coin or the throwing of a die. To enable some additional flexibility, we assume that the random occurrence is the drawing of a ball from a bag. This enables us to construct messages that have different probabilities. To be specific, we assume there are ten balls in the bag numbered from 0 to 9. All of them are red except the ball marked 0, which is green. The person communicating the message only reports ifthe ball drawn from the bag is red (using the digit 1) or green (using the digit 0). The recipient of the message is assumed to know about the setup. If the recipient receives the number 0,he then knows exactly which ball was selected,and all that were not selected. However, if he receives a 1, this provides less information, because he only knows that one of nine was selected,not which one. We notice that the digit 1 is nine times as likely to occur as the digit 0.This suggests that a higher probability digit contains less information than a lower probability digit.

We generalize the definition of the information content of a string of digits to allow for the possibility that different strings have different probabilities. We assume that the string is one of an ensemble of possible messages, and we define the information as:

$$I(s) = -\log(P(s)) \tag{1.8.3}$$

where $P(s)$ is the probability of the occurrence of the message s in the ensemble. Note that in the case of equal a priori probability $P(s) = 1/\Omega$, Eq. (1.8.3) reduces to Eq. (1.8.1). The use of probabilities in the definition of information makes sense in one of two cases:(1) The recipient knows the probabilities that represent the conventions of the transmission, or (2) A large number of independent messages are sent, and we are considering the information communicated by one of them. Then we can approximate the probability of a message by its proportion of appearance among the messages sent. We will discuss these points in greater detail later.

Question 1.8.1 Calculate the information, according to Eq.(1.8.3),that is provided by a single digit in the example given in the text of drawing red and green balls from a bag.

Solution 1.8.1 For the case of a 0,the information is the same as that of a decimal digit:

$$I(0) = -\log(1/10) \approx 3.32 \tag{1.8.4}$$

For the case of a 1 the information is

$$I(0) = -\log(9/10) \approx 0.152 \tag{1.8.5}$$ ∎

We can specialize the definition of information in Eq. (1.8.3) to a message $s = (s_1 s_2 ... s_N)$ composed of individual characters (bits, hexadecimal characters, ASCII characters, decimals, etc.) that are completely independent of each other

(for example, each corresponding to the result of a separate coin toss). This means that the total probability of the message is the product of the probability of each character, $P(s) = \prod_i P(s_i)$. Then the information content of the message is given by:

$$I(s) = -\sum_i \log(P(s_i)) \tag{1.8.6}$$

If all of the characters have equal probability and there are k possible characters in the alphabet, then $P(s_i) = 1/k$, and the information content is:

$$I(s) = N \log(k) \tag{1.8.7}$$

For the case of binary digits,this reduces to Eq.(1.8.2). For other cases like the hexadecimal case, $k = 16$,this continues to make sense:the information $I = 4N$ corresponds to the requirement of representing each hexadecimal digit with four bits. Note that the previous assumption of equal a priori probability for the whole string is stronger than the independence of the digits and implies it.

Question 1.8.2 Apply the definition of information content in Eq.(1.8.3) to each of the following cases. Assume messages consist of a total of N bits subject to the following constraints (aside for the constraints assume equal probabilities):

1. Every even bit is 1.
2. Every (odd, even) pair of bits is either 11 or 00.
3. Every eighth bit is a parity bit (the sum modulo 2 of the previous seven bits).

Solution 1.8.2: In each case, we first give an intuitive argument, and then we show that Eq. (1.8.3) or Eq. (1.8.6) give the same result.

1. The only information that is transferred is the state of the odd bits. This means that only half of the bits contain information. The total information is $N / 2$. To apply Eq.(1.8.6), we see that the even bits, which always have the value 1, have a probability $P(1) = 1$ which contributes no information. Note that we never have to consider the case $P(0) = 0$ for these bits, which is good, because by the formula it would give infinite information. The odd bits with equal probabilities, $P(1) = P(0) = 1/2$, give an information of one for either value received.
2. Every pair of bits contains only two possibilities, giving us the equivalent of one bit of information rather than two. This means that total information is $N/2$. To apply Eq.(1.8.6), we have to consider every (odd, even) pair of bits as a single character. These characters can never have the value 01 or 10, and they have the value 11 or 00 with probability $P(11) = P(00) = 1/2$, which gives the expected result. We will see later that there is another way to think about this example by using conditional probabilities.
3. The number of independent pieces of information is $7N / 8$. To see this from Eq. (1.8.6), we group each set of eight bits together and consider

them as a single character (a byte). There are only 2^7 different possibilities for each byte, and each one has equal probability according to our constraints and assumptions. This gives the desired result.

Note: Such representations are used to check for noise in transmission. If there is noise,the redundancy of the eighth bit provides additional information. The noise-dependent amount of additional information can also be quantified; however, we will not discuss it here. ∎

Question 1.8.3 Consider a transmission of English characters using an ASCII representation. ASCII characters are the conventional method for computer representation of English text including small and capital letters, numerals and punctuation. Discuss (do not evaluate for this question) how you would determine the information content of a message. We will evaluate the information content of English in a later question.

Solution 1.8.3 In ASCII, characters are represented using eight bits. Some of the possible combinations of bits are not used at all. Some are used very infrequently. One way to determine the information content of a message is to assume a model where each of the characters is independent. To calculate the information content using this assumption, we must find the probability of occurrence of each character in a sample text. Using these probabilities,the formula Eq.(1.8.6) could be applied. However, this assumes that the likelihood of occurrence of a character is independent of the preceding characters, which is not correct. ∎

Question 1.8.4: Assume that you know in advance that the number of ones in a long binary message is M. The total number of bits is N. What is the information content of the message? Is it similar to the information content of a message of N independent binary characters where the probability that any character is one is $P(1) = M/N$?

Solution 1.8.4: We count the number of possible messages with M ones and take the logarithm to obtain the information as

$$I = \log\left(\begin{pmatrix} N \\ M \end{pmatrix}\right) = \log\left(\frac{N!}{M!(N-M)!}\right) \tag{1.8.8}$$

We can show that this is almost the same as the information of a message of the same length with a particular probability of ones, $P(1) = M/N$, by use of the first two terms of Sterling's approximation Eq. (1.2.36). Assuming $1 << M << N$ (A correction to this would grow logarithmically with N and can be found using the additional terms in (Eq. (1.2.36)):

$$\begin{aligned} I &\approx N(\log(N) - 1) - M(\log(M) - 1) - (N-M)(\log(N-M) - 1) \\ &= -N[P(1)\log(P(1)) + (1 - P(1))\log(1 - P(1))] \end{aligned} \tag{1.8.9}$$

This is the information from a string of independent characters where $P(1) = M / N$. For such a string, the number of ones is approximately $NP(1)$ and the number of zeros $N(1 - P(1))$ (see also Question 1.8.7). ∎

1.8.2 *Characterizing sources of information*

The information content of a particular message is defined in terms of the probability that it,out of all possible messages, will be received. This means that we are characterizing not just a message but the source of the message.A direct characterization of the source is not the information of a particular message, but the average information over the ensemble of possible messages. For a set of possible messages with a given probability distribution $P(s)$ this is:

$$< I > = -\sum_s P(s)\log(P(s)) \tag{1.8.10}$$

If the messages are composed out of characters $s = (s_1 s_2 ... s_N)$,and each character is determined independently with a probability $P(s_i)$, then we can write the information content as:

$$< I > = -\sum_s \left(\prod_i P(s_i)\right)\log\left(\prod_{i'} P(s_{i'})\right) = -\sum_s \left(\prod_i P(s_i)\right)\sum_{i'} \log\left(P(s_{i'})\right) \tag{1.8.11}$$

We can move the factor in parenthesis inside the inner sum and interchange the order of the summations.

$$< I > = -\sum_{i'} \sum_s \left(\prod_i P(s_i)\right)\log\left(P(s_{i'})\right) = -\sum_{i'} \left(\sum_{\{s_i\}_{i\neq i'}} \prod_{i\neq i'} P(s_i)\right)\sum_{s_{i'}} P(s_{i'})\log\left(P(s_{i'})\right) \tag{1.8.12}$$

The latter expression results from recognizing that the sum over all possible states is a sum over all possible values of each of the letters. The sum and product can be interchanged:

$$\sum_{\{s_i\}_{i\neq i'}} \prod_{i\neq i'} P(s_i) = \prod_{i\neq i'} \sum_{s_i} P(s_i) = 1 \tag{1.8.13}$$

giving the result:

$$< I > = -\sum_{i'} \sum_{s_{i'}} P(s_{i'})\log\left(P(s_{i'})\right) \tag{1.8.14}$$

This shows that the average information content of the whole message is the average information content of each character summed over the whole character string. If the characters have the same probability, this is just the average information content of an individual character times the number of characters. If all letters of the alphabet have the same probability, this reduces to Eq. (1.8.7).

The average information content of a binary variable is given by:

$$< I > = -(P(1)\log(P(1)) + P(0)\log(P(0))) \tag{1.8.15}$$

Aside from the use of a logarithm base two, this is the same as the entropy of a spin (Section 1.6) with two possible states $s = \pm 1$ (see Question 1.8.5). The maximum information content occurs when the probabilities are equal, and the information goes to zero when one of the two becomes one,and the other zero (see Fig. 1.8.1). The information reflects the uncertainty in, or the lack of advance knowledge about, the value received.

Question 1.8.5 Show that the expression for the entropy S given in Eq. (1.6.16) of a set of noninteracting binary spins is the same as the information content defined in Eq.(1.8.15) aside from a normalization constant $k\ln(2)$. Consider the binary notation $s_i = 0$ to be the same as $s_i = -1$ for the spins.

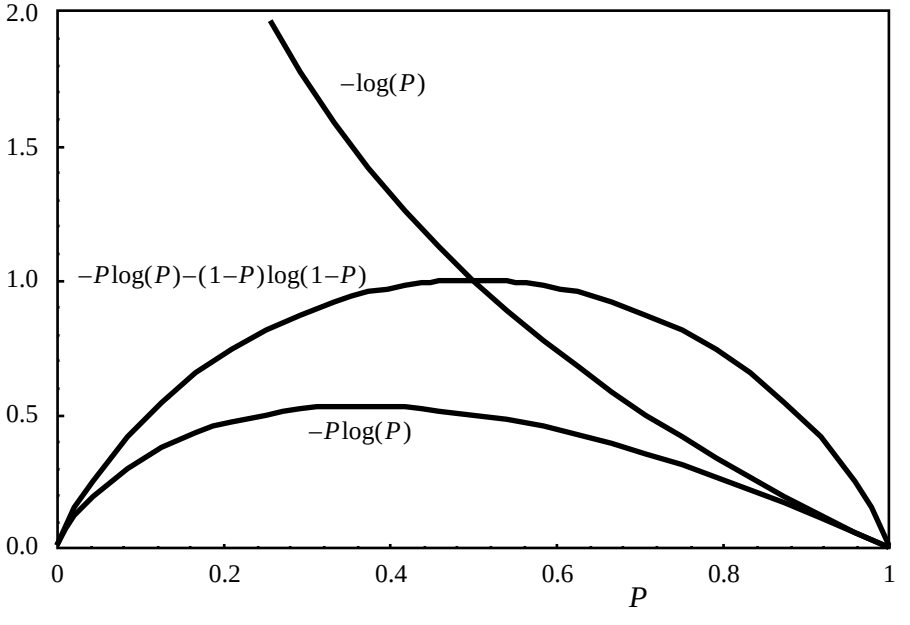

Figure 1.8.1 Plots of functions related to the information content of a message with probability P. $-\log(P)$ is the information content of a single message of probability P. $-P\log(P)$ is the contribution of this message to the average information given by the source. While the information content of a message diverges as P goes to zero, it appears less frequently so its contribution to the average information goes to zero. If there are only two possible messages, or two possible (binary) characters with probability P and $1 - P$ then the average information given by the source per message or per character is given by $-P\log(P) - (1 - P)\log(1 - P)$. ∎

Solution 1.8.5 The local magnetization m_i is the average value of a particular spin variable:

$$m_i = P_{s_i}(1) - P_{s_i}(-1) \tag{1.8.16}$$

Using $P_{s_i}(1) + P_{s_i}(-1) = 1$ we have:

$$\begin{aligned} P_{s_i}(1) &= (1 + m_i)/2 \\ P_{s_i}(-1) &= (1 - m_i)/2 \end{aligned} \tag{1.8.17}$$

Inserting these expressions into Eq. (1.8.15) and summing over a set of binary variables leads to the expression:

$$I = \left[-N \frac{1}{2} \sum_i \left((1+m_i)\log\left(1+m_i\right) + (1-m_i)\log\left(1-m_i\right) \right) \right] = S/k\ln(2) \tag{1.8.18}$$

The result is more general than this derivation suggests and will be discussed further in Chapter 8. ∎

Question 1.8.6 For a given set of possible messages, prove that the ensemble where all messages have equal probability provides the highest average information.

Solution 1.8.6 Since the sum over all probabilities is a fixed number (1), we consider what happens when we transfer some probability from one message to another. We start with the information given by

$$<I> = -P(s')\ln(P(s')) - P(s'')\ln(P(s'')) - \sum_{s \neq s', s''} P(s)\ln(P(s)) \tag{1.8.19}$$

and after shifting a probability of δ from one to the other we have:

$$<I'> = -(P(s') - \delta)\ln(P(s') - \delta) - (P(s'') + \delta)\ln(P(s'') + \delta) - \sum_{s \neq s', s'} P(s)\ln(P(s)) \tag{1.8.20}$$

We need to expand the change in information to first nonzero order in δ. We simplify the task by using the expression:

$$<I'> - <I> = f(P(s') + \delta) - f(P(s')) + f(P(s'') - \delta) - f(P(s'')) \tag{1.8.21}$$

where

$$f(x) = -x\log(x) \tag{1.8.22}$$

Taking a derivative, we have

$$\frac{d}{dx} f(x) = -(\log(x) + 1) \tag{1.8.23}$$

This gives the result:

$$<I'> - <I> = -(\log(P(s')) - \log(P(s'')))\delta \tag{1.8.24}$$

Since $\log(x)$ is a monotonic increasing function, we see that the average information increases ($(<I'> - <I>) > 0$) when probability $\delta > 0$ is transferred from a higher-probability character to a lower-probability character ($P(s'') > P(s')$ $-(\log(P(s')) - \log(P(s'')) > 0$). Thus,any change of the probability toward a more uniform probability distribution increases the average information. ∎

Question 1.8.7 A source produces strings of characters of length N. Each character that appears in the string is independently selected from an alphabet of characters with probabilities $P(s_i)$. Write an expression for the probability $P(s)$ of a typical string of characters. Show that this expression implies that the string gives N times the average information content of an individual character. Does this mean that every string must give this amount of information?

Solution 1.8.7 For a long string, each character will appear $NP(s_i)$ times. The probability of such a string is:

$$P(s) = \prod_{s_i} P(s_i)^{NP(s_i)} \tag{1.8.25}$$

The information content is:

$$I(s) = -\log(P(s)) = -N \sum_{s_i} P(s_i)\log(P(s_i)) \tag{1.8.26}$$

which is N times the average information of a single character. This is the information of a typical string. A particular string might have information significantly different from this. However, as the number of characters in the string increases, by the central limit theorem (Section 1.2), the fraction of times a particular character appears (i.e., the distance traveled in a random walk divided by the total number of steps) becomes more narrowly distributed around the expected probability $P(s_i)$. This means the proportion of messages whose information content differs from the typical value decreases with increasing message length. ∎

1.8.3 *Correlations between characters*

Thus far we have considered characters that are independent of each other. We can also consider characters whose values are correlated. We describe the case of two correlated characters. Because there are two characters,the notation must be more complete. As discussed in Section 1.2, we use the notation $P_{s_1,s_2}(s'_1, s'_2)$ to denote the probability that in the same string the character s_1 takes the value s'_1 and the variable s_2 takes the value s'_2. The average information contained in the two characters is given by:

$$<I_{s_1,s_2}> = -\sum_{s'_1,s'_2} P_{s_1,s_2}(s'_1,s'_2)\log(P_{s_1,s_2}(s'_1,s'_2)) \tag{1.8.27}$$

Note that the notation $I(s_1,s_2)$ is often used for this expression. We use $<I_{s_1,s_2}>$ because it is not a function of the values of the characters—it is the average information carried by the characters labeled by s_1 and s_2. We can compare the information content of the two characters with the information content of each character separately:

$$P_{s_1}(s_1') = \sum_{s_2'} P_{s_1,s_2}(s_1',s_2')$$
$$P_{s_2}(s_2') = \sum_{s_1'} P_{s_1,s_2}(s_1',s_2') \tag{1.8.28}$$

$$<I_{s_1}> = -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2')\log\Big(\sum_{s_2'} P_{s_1,s_2}(s_1',s_2'')\Big)$$
$$<I_{s_2}> = -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2')\log\Big(\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')\Big) \tag{1.8.29}$$

It is possible to show (see Question 1.8.8) the inequalities:

$$<I_{s_2}> + <I_{s_1}> \;\geq\; <I_{s_1,s_2}> \;\geq\; <I_{s_2}>, <I_{s_1}> \tag{1.8.30}$$

The right inequality means that we receive more information from both characters than from either one separately. The left inequality means that information we receive from both characters together cannot exceed the sum of the information from each separately. It can be less if the characters are dependent on each other. In this case, receiving one character reduces the information given by the second.

The relationship between the information from a character s_1 and the information from the same character after we know another character s_2 can be investigated by defining a contingent or conditional probability:

$$P_{s_1,s_2}(s_1'|s_2') = \frac{P_{s_1,s_2}(s_1',s_2')}{\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')} \tag{1.8.31}$$

This is the probability that s_1 takes the value s_1' assuming that s_2 takes the value s_2'. We used this notation in Section 1.2 to describe the transitions from one value to the next in a chain of events (random walk). Here we are using it more generally. We could recover the previous meaning by writing the transition probability as $P_s(s_1'|s_2') = P_{s(t),s(t-1)}(s_1'|s_2')$. In this section we will be concerned with the more general definition, Eq. (1.8.31).

We can find the information content of the character s_1 when s_2 takes the value s_2'

$$<I_{s_1}>\Big|_{s_2=s_2'} = -\sum_{s_1'} P_{s_1,s_2}(s_1'|s_2')\log\big(P_{s_1,s_2}(s_1'|s_2')\big)$$
$$= \frac{-\sum_{s_1'} P_{s_1,s_2}(s_1',s_2')\left(\log(P_{s_1,s_2}(s_1',s_2')) - \log\Big(\sum_{s_1'''} P_{s_1,s_2}(s_1''',s_2')\Big)\right)}{\sum_{s_1'} P_{s_1,s_2}(s_1'',s_2')} \tag{1.8.32}$$

This can be averaged over possible values of s_2, giving us the average information content of the character s_1 when the character s_2 is known.

$$\begin{aligned}
<<I_{s_1|s_2}>> &\equiv <<I_{s_1}>|_{s_2=s_2'}> \\
&= -\sum_{s_2'} P_{s_2}(s_2') \sum_{s_1'} P_{s_1,s_2}(s_1'|s_2') \log\left(P_{s_1,s_2}(s_1'|s_2')\right) \\
&= -\sum_{s_2'} \sum_{s_1'''} P_{s_1,s_2}(s_1''',s_2') \sum_{s_1'} \frac{P_{s_1,s_2}(s_1',s_2')}{\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')} \log\left(P_{s_1,s_2}(s_1'|s_2')\right) \\
&= -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2') \log\left(P_{s_1,s_2}(s_1'|s_2')\right)
\end{aligned} \tag{1.8.33}$$

The average we have taken should be carefully understood. The unconventional double average notation is used to indicate that the two averages are of a different nature. One way to think about it is as treating the information content of a dynamic variable s_1 when s_2 is a quenched (frozen) random variable. We can rewrite this in terms of the information content of the two characters,and the information content of the character s_2 by itself as follows:

$$\begin{aligned}
<<I_{s_1|s_2}>> &= -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2') \left[\log(P_{s_1,s_2}(s_1',s_2')) - \log\left(\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')\right)\right] \\
&= <I_{s_1,s_2}> - <I_{s_2}>
\end{aligned} \tag{1.8.34}$$

Thus we have:

$$<I_{s_1,s_2}> = <I_{s_1}> + <<I_{s_2|s_1}>> = <I_{s_2}> + <<I_{s_1|s_2}>> \tag{1.8.35}$$

This is the intuitive result that the information content given by both characters is the same as the information content gained by sequentially obtaining the information from the characters.Once the first character is known,the second character provides only the information given by the conditional probabilities. There is no reason to restrict the use of Eq.(1.8.27) – Eq.(1.8.35) to the case where s_1 is a single character and s_2 is a single character. It applies equally well if s_1 is one set of characters,and s_2 is another set of characters.

Question 1.8.8 Prove the inequalities in Eq. (1.8.30).

Hints for the left inequality:

1. It is helpful to use Eq. (1.8.35).
2. Use convexity ($f(\ x\ ) >\ f(x)\ $) of the function $f(x) = -x\log(x)$.

Solution 1.8.8 The right inequality in Eq. (1.8.30) follows from the inequality:

$$P_{s_1}(s_1'') = \sum_{s_1'} P_{s_1,s_2}(s_1'',s_2') > P_{s_1,s_2}(s_1',s_2') \tag{1.8.36}$$

The logarithm is a monotonic increasing function, so we can take the logarithm:

$$\log(\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')) > \log(P_{s_1,s_2}(s_1',s_2')) \tag{1.8.37}$$

Changing sign and averaging leads to the desired result:

$$\begin{aligned} <I_{s_2}> &= -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2')\log(\sum_{s_1''} P_{s_1,s_2}(s_1'',s_2')) \\ &< -\sum_{s_1',s_2'} P_{s_1,s_2}(s_1',s_2')\log(P_{s_1,s_2}(s_1',s_2')) = <I_{s_1,s_2}> \end{aligned} \tag{1.8.38}$$

The left inequality in Eq. (1.8.30) may be proven from Eq. (1.8.35) and the intuitive inequality

$$(<I_{s_1}>) > (<<I_{s_1|s_2}>>) \tag{1.8.39}$$

To prove this inequality we make use of the convexity of the function $f(x) = -x\log(x)$. Convexity of a function means that its value always lies above line segments (secants) that begin and end at points along its graph. Algebraically:

$$f((ax + by) / (a + b)) > (af(x) + bf(y)) / (a + b) \tag{1.8.40}$$

More generally, taking a set of values of x and averaging over them gives:

$$f(\,x\,) > \,f(x) \tag{1.8.41}$$

Convexity of $f(x)$ follows from the observation that

$$\frac{d^2 f}{dx^2} = -\frac{1}{x\ln(2)} < 0 \tag{1.8.42}$$

for all $x > 0$, which is where the function $f(x)$ is defined.

We then note the relationship:

$$P_{s_1}(s_1') = \sum_{s_2'} P_{s_2}(s_2')P_{s_1,s_2}(s_1'|s_2') = <P_{s_1,s_2}(s_1'|s_2')>_{s_2} \tag{1.8.43}$$

where, to simplify the following equations, we use a subscript to indicate the average with respect to s_2. The desired result follows from applying convexity as follows:

$$\begin{aligned} <I_{s_1}> &= -\sum_{s_1'} P_{s_1}(s_1')\log(P_{s_1}(s_1')) = \sum_{s_1'} f(P_{s_1}(s_1')) = \sum_{s_1'} f(<P_{s_1,s_2}(s_1'|s_2')>_{s_2}) \\ &> \sum_{s_1'} <f(P_{s_1,s_2}(s_1'|s_2'))>_{s_2} \\ &= -\sum_{s_2'} P_{s_2}(s_2')\sum_{s_1'} P_{s_1,s_2}(s_1'|s_2')\log\big(P_{s_1,s_2}(s_1'|s_2')\big) = <<I_{s_1|s_2}>> \end{aligned} \tag{1.8.44}$$

the final equality following from the definition in Eq. (1.8.33). We can now make use of Eq. (1.8.35) to obtain the desired result. ∎

1.8.4 *Ergodic sources*

We consider a source that provides arbitrarily long messages, or simply continues to give characters at a particular rate. Even though the messages are infinitely long, they are still considered elements of an ensemble. It is then convenient to measure the average information per character. The characterization of such an information source is simplified if each (long) message contains within it a complete sampling of the possibilities. This means that if we wait long enough, the entire ensemble of possible character sequences will be represented in any single message. This is the same kind of property as an ergodic system discussed in Section 1.3. By analogy, such sources are known as ergodic sources. For an ergodic source, not only the characters appear with their ensemble probabilities, but also the pairs of characters, the triples of characters, and so on.

For ergodic sources, the information from an ensemble average over all possible messages is the same as the information for a particular long string. To write this down we need a notation that allows variable length messages. We write $\underline{s}_N = (s_1 s_2 \ldots s_N)$, where N is the length of the string. The average information content per character may be written as:

$$< i_s >= \lim_{N\to\infty} \frac{< I_{\underline{s}_N} >}{N} = -\lim_{N\to\infty} \frac{1}{N} \sum_{\underline{s}_N} P(\underline{s}_N)\log(P(\underline{s}_N)) = -\lim_{N\to\infty} \frac{1}{N} \log(P(\underline{s}_N)) \tag{1.8.45}$$

The rightmost equality is valid for an ergodic source. An example of an ergodic source is a source that provides independent characters—i.e., selects each character from an ensemble. For this case, Eq.(1.8.45) was shown in Question 1.8.7. More generally, for a source to be ergodic, long enough strings must break up into independent substrings, or substrings that are more and more independent as their length increases.

Assuming that N is large enough, we can use the limit in Eq. (1.8.45) and write:

$$P(\underline{s}_N) \approx 2^{-N<i_s>} \tag{1.8.46}$$

Thus, for large enough N, there are a set of strings that are equally likely to be generated by the source. The number of these strings is

$$2^{N<i_s>} \tag{1.8.47}$$

Since any string of characters is possible, in principle, this statement must be formally understood as saying that the total probability of all other strings becomes arbitrarily small.

If the string of characters is a Markov chain (Section 1.2), so that the probability of each character depends only on the previous character, then there are general conditions that can ensure that the source is ergodic. Similar to the discussion of Monte Carlo simulations in Section 1.7, for the source to be ergodic, the transition probabil-

ities between characters must be irreducible and acyclic. Irreducibility guarantees that all characters are accessible from any starting character. The acyclic property guarantees that starting from one substring, all other substrings are accessible. Thus, if we can reach any particular substring, it will appear with the same frequency in all long strings.

We can generalize the usual Markov chain by allowing the probability of a character to depend on several (n) previous characters. A Markov chain may be constructed to represent such a chain by defining new characters, where each new character is formed out of a substring of n characters. Then each new character depends only on the previous one. The essential behavior of a Markov chain that is important here is that correlations measured along the chain of characters disappear exponentially. Thus,the statistical behavior of the chain in one place is independent of what it was in the sufficiently far past. The number of characters over which the correlations disappear is the correlation length. By allowing sufficiently many correlation lengths along the string—segments that are statistically independent—the average properties of one string will be the same as any other such string.

Question 1.8.9 Consider ergodic sources that are Markov chains with two characters $s_i = \pm 1$ with transition probabilities:

a. $P(1|1) = .999, P(-1|1) = .001, P(-1|-1) = 0.5, P(1|-1) = 0.5$

b. $P(1|1) = .999, P(-1|1) = .001, P(-1|-1) = 0.999, P(1|-1) = 0.001$

c. $P(1|1) = .999, P(-1|1) = .001, P(-1|-1) = 0.001, P(1|-1) = 0.999$

d. $P(1|1) = .001, P(-1|1) = .999, P(-1|-1) = 0.5, P(1|-1) = 0.5$

e. $P(1|1) = .001, P(-1|1) = .999, P(-1|-1) = 0.999, P(1|-1) = 0.001$

f. $P(1|1) = .001, P(-1|1) = .999, P(-1|-1) = 0.001, P(1|-1) = 0.999$

Describe the appearance of the strings generated by each source, and (roughly) its correlation length.

Solution 1.8.9 (a) has long regions of 1s of typical length 1000. In between there are short strings of –1s of average length $2 = 1 + 1/2 + 1/4 + \ldots$(there is a probability of $1/2$ that a second character will be –1 and a probability of $1/4$ that both the second and third will be –1, etc.). (b) has long regions of 1s and long regions of –1s, both of typical length 1000. (c) is like (a) except the regions of –1s are of length 1. (d) has no extended regions of 1 or –1 but has slightly longer regions of –1s. (e) inverts (c). (f) has regions of alternating 1 and –1 of length 1000 before switching to the other possibility (odd and even indices are switched). We see that the characteristic correlation length is of order 1000 in (a),(b),(c),(e) and (f) and of order 2 in (d). ∎

We have considered in detail the problem of determining the information content of a message, or the average information generated by a source, when the characteristics of the source are well defined. The source was characterized by the ensemble of possible messages and their probabilities. However, we do not usually have a

well-defined characterization of a source of messages,so a more practical question is to determine the information content from the message itself.The definitions that we have provided do not guide us in determining the information of an arbitrary message. We must have a model for the source. The model must be constructed out of the information we have—the string of characters it produces.One possibility is to model the source as ergodic. An ergodic source can be modeled in two ways, as a source of independent substrings or as a generalized Markov chain where characters depend on a certain number of previous characters. In each case we construct not one, but an infinite sequence of models. The models are designed so that if the source is ergodic then the information estimates given by the models converge to give the correct information content.

There is a natural sequence of independent substring models indexed by the number of characters in the substrings n. The first model is that of a source producing independent characters with a probability specified by their frequency of occurrence in the message. The second model would be a source producing pairs of correlated characters so that every pair of characters is described by the probability given by their occurrence (we allow character pairs to overlap in the message). The third model would be that of a source producing triples of correlated characters,and so on. We use each of these models to estimate the information. The nth model estimate of the information per character given by the source is:

$$< i_s >_{1,n} = \lim_{N\to\infty} \frac{1}{n} \sum_{\underline{s}_n} \tilde{P}_N(\underline{s}_n) \log(\tilde{P}_N(\underline{s}_n)) \tag{1.8.48}$$

where we indicate using the subscript $1,n$ that this is an estimate obtained using the first type of model (independent substring model) using substrings of length n. We also make use of an approximate probability for the substring defined as

$$\tilde{P}_N(\underline{s}_n) = N(\underline{s}_n)/(N-n+1) \tag{1.8.49}$$

where $N(\underline{s}_n)$ is the number of times s_n appears in the string of length N. The information of the source might then be estimated as the limit $n \to \infty$ of Eq. (1.8.48):

$$< i_s > = \lim_{n\to\infty} \lim_{N\to\infty} \frac{1}{n} \sum_{\underline{s}_n} \tilde{P}_N(\underline{s}_n) \log(\tilde{P}_N(\underline{s}_n)) \tag{1.8.50}$$

For an ergodic source, we can see that this converges to the information of the message. The n limit converges monotonically from above. This is because the additional information in $\underline{s}_{n+1}$ given by s_{n+1} is less than the information added by each previous character (see Eq. 1.8.59 below). Thus, the estimate of information per character based on $\underline{s}_n$ is higher than the estimate based on $\underline{s}_{n+1}$. Therefore, for each value of n the estimate $< i_s >_{1,n}$ is an upper bound on the information given by the source.

How large does N have to be? Since we must have a reasonable sample of the occurrence of substrings in order to estimate their probability, we can only estimate probabilities of substrings that are much shorter than the length of the string. The number of possible substrings grows exponentially with n as k^n, where k is the num-

ber of possible characters. If substrings occur with roughly similar probabilities, then to estimate the probability of a substring of length n would require at least a string of length k^n characters. Thus,taking the large N limit should be understood to correspond to N greater than k^n. This is a very severe requirement. This means that to study a model of English character strings of length $n = 10$ (ignoring upper and lower case, numbers and punctuation) would require $26^{10} \sim 10^{14}$ characters. This is roughly the number of characters in all of the books in the Library of Congress (see Question 1.8.15).

The generalized Markov chain model assumes a particular character is dependent only on n previous characters. Since the first n characters do not provide a significant amount of information for a very long chain ($N >> n$), we can obtain the average information per character from the incremental information given by a character. Thus, for the nth generalized Markov chain model we have the estimate:

$$< i_s >_{2,n} = << I_{s_n|\underline{s}_{n-1}} >> = \lim_{N\to\infty} \sum_{\underline{s}_{n-1}} \tilde{P}_N(\underline{s}_{n-1}) \sum_{s_n} \tilde{P}(s_n \mid \underline{s}_{n-1}) \log(\tilde{P}(s_n \mid \underline{s}_{n-1})) \tag{1.8.51}$$

where we define the approximate conditional probability using:

$$\tilde{P}_N(s_n \mid \underline{s}_{n-1}) = N(\underline{s}_{n-1} s_n) / N(\underline{s}_{n-1}) \tag{1.8.52}$$

Taking the limit $n \to \infty$ we have an estimate of the information of the source per character:

$$< i_s > = \lim_{n\to\infty} \lim_{N\to\infty} \sum_{\underline{s}_{n-1}} \tilde{P}_N(\underline{s}_{n-1}) \sum_{s_n} \tilde{P}(s_n \mid \underline{s}_{n-1}) \log(\tilde{P}(s_n \mid \underline{s}_{n-1})) \tag{1.8.53}$$

This also converges from above as a function of n for large enough N. For a given n, a Markov chain model takes into account more correlations than the previous independent substring model and thus gives a better estimate of the information (Question 1.8.10).

Question 1.8.10 Prove that the Markov chain model gives a better estimate of the information for ergodic sources than the independent substring model for a particular n. Assume the limit $N \to \infty$ so that the estimated probabilities become actual and we can substitute $\tilde{P}_N \to P$ in Eq.(1.8.48) and Eq. (1.8.51).

Solution 1.8.10 The information in a substring of length n is given by the sum of the information provided incrementally by each character, where the previous characters are known. We derive this statement algebraically (Eq. (1.8.59)) and use it to prove the desired result. Taking the N limit in Eq. (1.8.48), we define the nth approximation using the independent substring model as:

$$< i_s >_{1,n} = \frac{1}{n} \sum_{\underline{s}_n} P(\underline{s}_n) \log(P(\underline{s}_n)) \tag{1.8.54}$$

and for the nth generalized Markov chain model we take the same limit in Eq. (1.8.51):

$$< i_s >_{2,n} = \sum_{\underline{s}_{n-1}} P(\underline{s}_{n-1}) \sum_{s_n} P(s_n \mid \underline{s}_{n-1}) \log(P(s_n \mid \underline{s}_{n-1})) \tag{1.8.55}$$

To relate these expressions to each other, follow the derivation of Eq.(1.8.34), or use it with the substitutions $s_1 \to \underline{s}_{n-1}$ and $s_2 \to s_n$, to obtain

$$< i_s >_{2,n} = - \sum_{\underline{s}_{n-1}, s_n} P(\underline{s}_{n-1} s_n) \left[\log(P(\underline{s}_{n-1} s_n)) - \log\left(\sum_{s_n} P(\underline{s}_{n-1} s_n) \right) \right] \tag{1.8.56}$$

Using the identities

$$\begin{aligned} P(\underline{s}_{n-1} s_n) &= P(\underline{s}_n) \\ P(\underline{s}_{n-1}) &= \sum_{s_n} P(\underline{s}_{n-1} s_n) \end{aligned} \tag{1.8.57}$$

this can be rewritten as:

$$< i_s >_{2,n} = n < i_s >_{1,n} - (n-1) < i_s >_{1,n-1} \tag{1.8.58}$$

This result can be summed over n from 1 to n (the $n = 1$ case is $<i_s>_{2,1} = <i_s>_{1,1}$) to obtain:

$$\sum_{n'=1}^{n} < i_s >_{2,n'} = n < i_s >_{1,n} \tag{1.8.59}$$

since $< i_s >_{2,n}$ is monotonic decreasing and $< i_s >_{1,n}$ is seen from this expression to be an average over $< i_s >_{2,n}$ with lower values of n, we must have that

$$< i_s >_{2,n} \leq < i_s >_{1,n} \tag{1.8.60}$$

as desired. ∎

Question 1.8.11 We have shown that the two models—the independent substring models and the generalized Markov chain model—are upper bounds to the information in a string. How good is the upper bound? Think up an example that shows that it can be terrible for both, but better for the Markov chain.

Solution 1.8.11 Consider the example of a long string formed out of a repeating substring, for example (000000010000000100000001…). The average information content per character of this string is zero. This is because once the repeat structure has become established,there is no more information. Any model that gives a nonzero estimate of the information content per

character will make a great error in its estimate of the information content of the string, which is N times as much as the information per character.

For the independent substring model,the estimate is never zero. For the Markov chain model it is nonzero until n reaches the repeat distance. A Markov model with n the same size or larger than the repeat length will give the correct answer of zero information per character. This means that even for the Markov chain model, the information estimate does not work very well for n less than the repeat distance. ∎

Question 1.8.12 Write a computer program to estimate the information in English and find the estimate. For simple,easy-to-compute estimates, use single-character probabilities,two-character probabilities,and a Markov chain model for individual characters. These correspond to the above definitions of $< i_s >_{2,1} = < i_s >_{1,1}$, $< i_s >_{1,2}$, and $< i_s >_{2,2}$ respectively.

Solution 1.8.12 A program that evaluates the information content using single-character probabilities applied to the text (excluding equations) of Section 1.8 of this book gives an estimate of information content of 4.4 bits/character. Two-character probabilities gives 3.8 bits/character, and the one-character Markov chain model gives 3.3 bits/character. A chapter of a book by Mark Twain gives similar results. These estimates are decreasing in magnitude, consistent with the discussion in the text. They are also still quite high as estimates of the information in English per character.

The best estimates are based upon human guessing of the next character in a written text. Such experiments with human subjects give estimates of the lower and upper bounds of information content per character of English text. These are 0.6 and 1.2 bits/character. This range is significantly below the estimates we obtained using simple models. Remarkably, these estimates suggest that it is enough to give only one in four to one in eight characters of English in order for text to be decipherable. ∎

Question 1.8.13 Construct an example illustrating how correlations can arise between characters over longer than,say, ten characters. These correlations would not be represented by any reasonable character-based Markov chain model. Is there an example of this type relevant to the English language?

Solution 1.8.13 Example 1: If we have information that is read from a matrix row by row, where the matrix entries have correlations between rows, then there will be correlations that are longer than the length of the matrix rows.

Example 2: We can think about successive English sentences as rows of a matrix. We would expect to find correlations between rows (i.e., between words found in adjacent sentences) rather than just between letters. ∎

Question 1.8.14 Estimate the amount of information in a typical book (order of magnitude is sufficient). Use the best estimate of information content per character of English text of about 1 bit per character.

Solution 1.8.14 A rough estimate can be made using as follows: A 200 page novel with 60 characters per line and 30 lines per page has 4×10^5 characters. Textbooks can have several times this many characters.A dictionary, which is significantly longer than a typical book, might have 2×10^7 characters. Thus we might use an order of magnitude value of 10^6 bits per book. ∎

Question 1.8.15 Obtain an estimate of the number of characters (and thus the number of bits of information) in the Library of Congress. Assume an average of 10^6 characters per book.

Solution 1.8.15 According to information provided by the Library of Congress,there are presently (in 1996) 16 million books classified according to the Library of Congress classification system, 13 million other books at the Library of Congress, and approximately 80 million other items such as newspapers, maps and films. Thus with 10^7–10^8 book equivalents, we estimate the number of characters as 10^{13}–10^{14}. ∎

Inherent in the notion of quantifying information content is the understanding that the same information can be communicated in different ways, as long as the amount of information that can be transmitted is sufficient. Thus we can use binary, decimal, hexadecimal or typed letters to communicate both numbers and letters. Information can be communicated using any set of (two or more) characters. The presumption is that there is a way of translating from one to another. Translation operations are called codes; the act of translation is encoding or decoding. Among possible codes are those that are invertible.Encoding a message cannot add information, it might,however, lose information (Question 1.8.16). Invertible codes must preserve the amount of information.

Once we have determined the information content, we can compare different ways of writing the same information. Assume that one source generates a message of length N characters with information I. Then a different source may transmit the same information using fewer characters. Even if characters are generated at the same rate,the information may be more rapidly transmitted by one source than another. In particular, regardless of the value of N, by definition of information content, we could have communicated the same information using a binary string of length I. It is,however, impossible to use fewer than I bits because the maximum information a binary message can contain is equal to its length. This amount of information occurs for a source with equal a priori probability.

Encoding the information in a shorter form is equivalent to data compression. Thus a completely compressed binary data string would have an amount of information given by its length. The source of such a message would be characterized as a source of messages with equal a priori probability—a random source. We see that ran-

domness and information are related. Without a translation (decoding) function it would be impossible to distinguish the completely compressed information from random numbers. Moreover, a random string could not be compressed.

Question 1.8.16 Prove that an encoding operation that takes a message as input and converts it into another well-defined message (i.e., for a particular input message, the same output message is always given) cannot add information but may reduce it. Describe the necessary conditions for it to keep the same amount of information.

Solution 1.8.16 Our definition of information relies upon the specification of the ensemble of possible messages. Consider this ensemble and assume that each message appears in the ensemble a number of times in proportion to its probability, like the bag with red and green balls. The effect of a coding operation is to label each ball with the new message (code) that will be delivered after the coding operation. The amount of information depends not on the nature of the label, but rather on the number of balls with the same label. The requirement that a particular message is encoded in a well-defined way means that two balls that start with the same message cannot be labeled with different codes. However, it is possible for balls with different original messages to be labeled the same. The average information is not changed if and only if all distinct messages are labeled with distinct codes. If any distinct messages become identified by the same label, the information is reduced.

We can prove this conclusion algebraically using the result of Question 1.8.8, which showed that transferring probability from a less likely to a more likely case reduced the information content. Here we are,in effect, transferring all of the probability from the less likely to the more likely case. The change in information upon labeling two distinct messages with the same code is given by ($f(x) = -x\log(x)$, as in Question 1.8.8):

$$\begin{aligned}\Delta I &= f(P(s_1) + P(s_2)) - (f(P(s_1)) + f(P(s_2))) \\ &= (f(P(s_1) + P(s_2)) + f(0)) - (f(P(s_1)) + f(P(s_2))) < 0\end{aligned} \tag{1.8.61}$$

where the inequality follows because $f(x)$ is convex in the range $0 < x < 1$. ∎

1.8.5 *Human communication*

The theory of information, like other theories, relies upon idealized constructs that are useful in establishing the essential concepts, but do not capture all features of real systems. In particular, the definition and discussion of information relies upon sources that transmit the result of random occurrences, which, by definition, cannot be known by the recipient. The sources are also completely described by specifying the nature of the random process. This model for the nature of the source and the recipient does not adequately capture the attributes of communication between human beings. The theory of information can be applied directly to address questions about

information channels and the characterization of communication in general. It can also be used to develop an understanding of the complexity of systems. In this section, however, we will consider some additional issues that should be kept in mind when applying the theory to the communication between human beings. These issues will arise again in Chapter 8.

The definition of information content relies heavily on the concepts of probability, ensembles, and processes that generate arbitrarily many characters. These concepts are fraught with practical and philosophical difficulties—when there is only one transmitted message,how can we say there were many that were possible? A book may be considered as a single communication.A book has finite length and, for a particular author and a particular reader, is a unique communication. In order to understand both the strengths and the limitations of applying the theory of information,it is necessary to recognize that the information content of a message depends on the information that the recipient of the message already has. In particular, information that the recipient has about the source. In the discussion above,a clear distinction has been made. The only information that characterizes the source is in the ensemble probabilities $P(s)$. The information transmitted by a single message is distinct from the ensemble probabilities and is quantified by $I(s)$. It is assumed that the characterization of the source is completely known to the recipient. The content of the message is completely unknown (and unknowable in advance) to the recipient.

A slightly more difficult example to consider is that of a recipient who does not know the characterization of the source. However, such a characterization in terms of an ensemble $P(s)$ does exist. Under these circumstances, the amount of information transferred by a message would be more than the amount of information given by $I(s)$. However, the maximum amount of information that could be transferred would be the sum of the information in the message,and the information necessary to characterize the source by specifying the probabilities $P(s)$. This upper bound on the information that can be transferred is only useful if the amount of information necessary to characterize the source is small compared to the information in the message.

The difficulty with discussing human communication is that the amount of information necessary to fully characterize the source (one human being) is generally much larger than the information transmitted by a particular message. Similarly, the amount of information possessed by the recipient (another human being) is much larger than the information contained in a particular message. Thus it is reasonable to assume that the recipient does not have a full characterization of the source. It is also reasonable to assume that the model that the recipient has about the source is more sophisticated than a typical Markov chain model, even though it is a simplified model of a human being. The information contained in a message is, in a sense, the additional information not contained in the original model possessed by the recipient. This is consistent with the above discussion, but it also recognizes that specifying the probabilities of the ensemble may require a significant amount of information. It may also be convenient to summarize this information by a different type of model than a Markov chain model.

Once the specific model and information that the recipient has about the source enters into an evaluation of the information transfer, there is a certain and quite reasonable degree of relativity in the amount of information transferred. An extreme example would be if the recipient has already received a long message and knows the same message is being repeated,then no new information is being transmitted.A person who has memorized the Gettysburg Address will receive very little new information upon hearing or reading it again. The prior knowledge is part of the model possessed by the recipient about the source.

Can we incorporate this in our definition of information? In every case where we have measured the information of a message, we have made use of a model of the source of the information. The underlying assumption is that this model is possessed by the recipient. It should now be recognized that there is a certain amount of information necessary to describe this model. As long as the amount of information in the model is small compared to the amount of information in the message, we can say that we have an absolute estimate of the information content of the message. As soon as the information content of the model approaches that of the message itself, then the amount of information transferred is sensitive to exactly what information is known. It might be possible to develop a theory of information that incorporates the information in the model,and thus to arrive at a more absolute measure of information. Alternatively, it might be necessary to develop a theory that considers the recipient and source more completely, since in actual communication between human beings, both are nonergodic systems possessed of a large amount of information. There is significant overlap of the information possessed by the recipient and the source. Moreover, this common information is essential to the communication itself.

One effort to arrive at a universal definition of information content of a message has been made by formally quantifying the information contained in models. The resulting information measure, Kolmogorov complexity, is based on computation theory discussed in the next section. While there is some success with this approach,two difficulties remain. In order for a universal definition of information to be agreed upon,models must still have an information content which is less than the message—knowledge possessed must be smaller than that received. Also, to calculate the information contained in a particular message is essentially impossible, since it requires computational effort that grows exponentially with the length of the message. In any practical case,the amount of information contained in a message must be estimated using a limited set of models of the source. The utilization of a limited set of models means that any estimate of the information in a message is an upper bound.

1.9 Computation

The theory of computation describes the operations that we perform on numbers, including addition, subtraction, multiplication and division. More generally, a computation is a sequence of operations each of which has a definite/unique/well-defined result. The fundamental study of such operations is the theory of logic. Logical