CS294 Markov Chain Monte Carlo: Foundations & Applications **Fall 2009**

Lecture 1: August 27

Lecturer: Prof. Alistair Sinclair *Scribes: Alistair Sinclair*

1.1 The Markov Chain Monte Carlo Paradigm

Assume that we have a very large but finite set Ω and a positive weight function $w : \Omega \to \Re^+$. Our goal is to sample $x \in \Omega$ with probability $\pi(x) = \frac{w(x)}{Z}$, where the normalizing factor $Z = \sum_{x\in\Omega} w(x)$, often called the "partition function", is usually unknown. (Indeed, in many applications our ultimate goal will be to estimate Z.)

Markov Chain Monte Carlo constructs a Markov Chain (X_t) on Ω that converges to π, ie $\Pr[X_t = y|X_0 = x] \to \pi(y)$ as $t \to \infty$, independent of x. Then we get a sampling algorithm by simulating the Markov chain, starting in an arbitrary state X_0, for sufficiently many steps and outputting the final state X_t. It is usually not hard to set up a Markov chain that converges to the desired stationary distribution; however, the key question is how many steps is "sufficiently many," or equivalently, how many steps are needed for the chain to get "close to" π. This is known as the "mixing time" of the chain. Obviously the mixing time determines the efficiency of the sampling algorithm.

In the remainder of this introductory lecture, we provide motivation for MCMC by sketching a number of application areas in which random sampling from large, complex combinatorial sets arises. We focus on applications in which rigorously justified algorithms can be achieved; for a more practically oriented focus, see the excellent book by Jun Liu [L02].

1.2 Applications

1.2.1 Combinatorics

Applications in combinatorics include:

- Examining typical members of a combinatorial set, which can be used, e.g., for fomulating and testing conjectures.
  For example, by sampling random 3-regular graphs on n vertices, we might formulate the conjecture that they are (almost) all Hamiltonian; this conjecture is actually now a theorem.
- Generating test data for algorithms.
  Often, testing an algorithm on completely random inputs (such as arbitrary random graphs) is uninformative. MCMC can be used to generate inputs from a more complex class (such as sparse connected graphs), which can form the basis of more convincing tests of the algorithm.
- Probabilistic constructions.
  The existence of certain objects (such as networks with desired connectivity properties) can be proven

by the probabilistic method, but in many cases the probabilistic construction required is quite complex and it is not obvious how to realize it algorithmically. For example, recent constructions of efficient Low Density Parity Check codes require a random bipartite graph with specified degrees for each vertex. It is not known how to generate such graphs directly, but they can be generated quite efficiently by MCMC.
A similar example is provided by certain models of the WWW, which are also based on random graphs with specified vertex degrees (sometimes with additional properties).

- Approximate counting.
A much less obvious and more far-reaching combinatorial application is to count the number of elements of the set Ω, which might be (e.g.) the set of cliques in a graph G, or the set of satisfying assignments of a boolean formula ϕ. Almost all such counting problems are *#P-complete* (which is the analog of NP-completeness for decision problems); however, in many cases MCMC provides an efficient *randomized approximation algorithm*.

  The general technique for reducing (approximate) counting to random sampling can be explained in the following folksy scenario for counting the number of people in a large crowd Ω:

  1. Partition the crowd Ω into two parts, B and $\overline{B} = \Omega - B$, according to some property (such as "having black hair").
  2. Estimate the proportion $|B|/|\Omega|$ of people with black hair by taking a small uniform sample from the crowd and letting p be the proportion of the sample that have black hair.
  3. Recursively estimate the size of B by applying the same technique to this smaller crowd but using some other property (such as "being male"). Let $\widehat{N_B}$ be this recursive estimate.
  4. Output the final estimate $\widehat{N} = \widehat{N_B} \cdot \frac{1}{p}$.

  Notice that the choice of properties at each level of the recursion is essentially arbitrary; in particular, we do not require them to be independent. The only thing we require is that the proportion of people in the remaining crowd who have the property is bounded away from 0 and 1: this ensures that (i) the number of levels of recursion (until we get down to a crowd of size 1) is small; and (ii) the sample size needed to get a good estimate p at each level is small.

  To apply this technique in a more mathematical context, let Ω be the set of all cliques of a given graph G. (This is a very large and complex set, whose size is in general exponential in the size of G.) We can partition Ω into those cliques that contain some vertex v, and those that do not. But the first of these sets is equivalent to the set of all cliques in the graph G_v (the subgraph of G consisting of v with all its neigbors); and the second set is equivalent to the set of all cliques in the graph $G - v$ (the subgraph of G with v removed). So both subsets correspond to instances of the same counting problem applied to smaller graphs; hence they can be estimated recursively, as required by the method.

1.2.2 Volume and Integration

Given as input a convex body K in $\Re^n$, the goal is to estimate the volume of K. While in low dimensions exact methods may be used, they become computationally intractable in higher dimensions. (In fact, the problem is #P-complete when the dimension n is treated as part of the input.)

The volume can be estimated by the following reduction to random sampling:

Construct concentric balls $B_0 \subset B_1 \subset B_2 \ldots \subset B_r$, such that $B_0 \subset K \subset B_r$. By a non-trivial geometric result, it can be assumed w.l.o.g. that B_0 is the unit ball, and that B_r has radius $O(n \log n)$, where n is

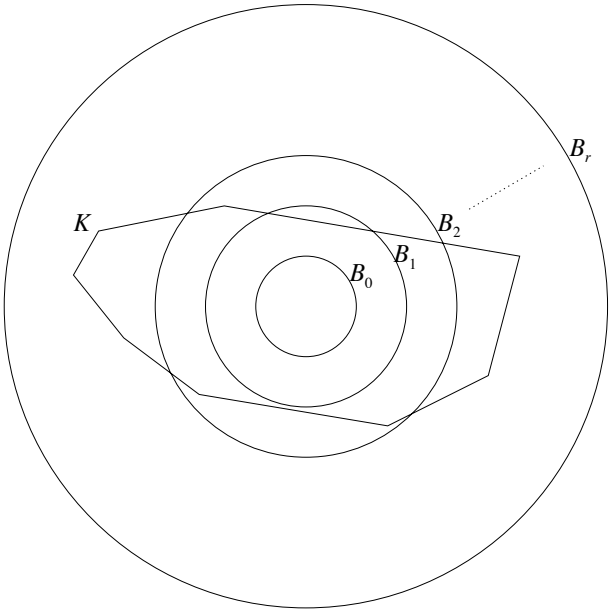

Figure 1.1: Estimating the volume of a convex set by choosing a sequence of increasing balls and computing the ratio $\frac{Vol(K\cap B_i)}{Vol(K\cap B_{i-1})}$

the dimension. (This can be achieved by simple transformations of K.) The construction of the balls is illustrated by Figure 1.1. Then we have

$$Vol(K) = \frac{Vol(K\cap B_r)}{Vol(K\cap B_{r-1})}\times\frac{Vol(K\cap B_{r-1})}{Vol(K\cap B_{r-2})}\times\cdots\times\frac{Vol(K\cap B_1)}{Vol(K\cap B_0)}\times Vol(K\cap B_0).$$

Since $Vol(K\cap B_0) = Vol(B_0)$ is trivial, we can estimate $Vol(K)$ by estimating each of the ratios in the above equation. The ratio $\frac{Vol(K\cap B_i)}{Vol(K\cap B_{i-1})}$ can be estimated by sampling uniformly at random from $Vol(K\cap B_i)$ (which is the intersection of two convex bodies and hence also convex) and counting the proportion of samples falling in B_{i-1}. In this scheme, to ensure that the number of samples needed is small we need to ensure that the ratio $\frac{Vol(K\cap B_i)}{Vol(K\cap B_{i-1})}$ is bounded by a constant. But for this it is enough to make the radius of the balls grow slowly enough, namely $\text{rad}(B_i) = (1+\frac{1}{n})\text{rad}(B_{i-1})$. Notice that this implies that the number of balls is only $r = O(n\log n)$. The random sampling within each $K\cap B_i$ can be done by MCMC.

Observe that the sequence of balls above is necessary because, for a general convex body K in $\Re^n$ that contains the unit ball, the volume of the smallest ball B containing K may be exponentially larger than $Vol(K)$. Hence the "naive Monte Carlo" approach of sampling randomly from B and observing how many samples fall in K is hopeless: we will need exponentially many samples before we even see one that falls in K. The sequence of balls with slowly growing radii ensures that our random sampling always produces an estimator with bounded variance.

The above algorithm was first discovered by Dyer, Frieze and Kannan [DFK89]. The original version had a time complexity of about $O(n^{23})$ (which was a breakthrough because it is polynomial in n). A long sequence of deep and clever improvements has since brought this down to $\tilde{O}(n^4)$ [LV03] (where the $\tilde{O}$ hides logarithmic factors).

It is possible to generalize this approach to the integration of log-concave functions in $\Re^n$.

1.2.3 Combinatorial Optimization

In this setting Ω is the set of feasible solutions to a combinatorial optimization problem (e.g., the set of all cliques in a graph), and there is an *objective function* $f : \Omega \to \Re$ (e.g., the size of the clique). Our goal is to maximize f, i.e., to find an element $x \in \Omega$, such that $f(x) \geq f(y) \quad \forall y \in \Omega$.

Strategy

- Let G be any positive monotone increasing function.
- Sample from the distribution $\pi(x) = \frac{G(f(x))}{Z}$ using MCMC.
  - A popular choice is $G(y) = \lambda^y$, where $\lambda \geq 1$, so that the distribution is $\pi(x) \propto G(f(x))$.
  - We would like to sample using a large λ to get a good approximation to the maximum; however, on the other hand we want to keep λ fairly small (close to 1) because then the distribution π is close to uniform and thus presumably easier to sample from. (When λ gets large, the MCMC algorithm on Ω will tend to get trapped in local maxima.) Thus there is a tension between large and small values of λ.

A simple strategy is to simply choose some intermediate value of λ and hope that it achieves a good compromise between the above two concerns; this is often called the "Metropolis algorithm." A more sophisticated strategy is to start with $\lambda = 1$ and gradually increase λ according to some scheme (often called a "cooling schedule") until λ becomes very large and the algorithm "freezes" into a local maximum (which we hope is good). This latter heuristic is known as "simulated annealing."

1.2.4 Statistical Physics

In statistical physics Ω is the set of configurations of a physical system. Each configuration x has an energy $H(x)$. The probability of finding the system in configuration x is given by the "Gibbs distribution"

$$\pi(x) = \frac{\exp(-\beta H(x))}{Z},$$

where $\beta = \frac{1}{\text{Temp}} > 0$ is the inverse temperature. It can be observed that this Gibbs distribution favors configurations with low energy. Moreover, this effect increases with β: as $\beta \to 0$ (i.e., at high temperature) the Gibbs distribution is close to uniform, while at high β (low temperature) the system is almost certain to be in very low energy states. (Note the similarity with the combinatorial optimization application above, under the correspondence $\lambda = \exp(\beta)$ and $f = -H$. The energy minima can be thought of as local optima.)

The most famous and widely studied model in statistical physics is the *Ising model* of ferromagnetism. Here there is an underlying graph $G = (V, E)$, usually a finite portion of the d-dimensional square lattice, at each vertex of which there is a *spin*, which can be either $+$ or $-$. (The spins correspond to atomic magnets whose magnetization is either "up" or "down".) Thus the set of configurations is $\Omega = \{+, -\}^V$. An example of a configuration is depicted in Figure 1.2.

The energy of an Ising configuration σ is given by

$$H(\sigma) = -\sum_{i \sim j} \sigma_i \sigma_j. \tag{1.1}$$

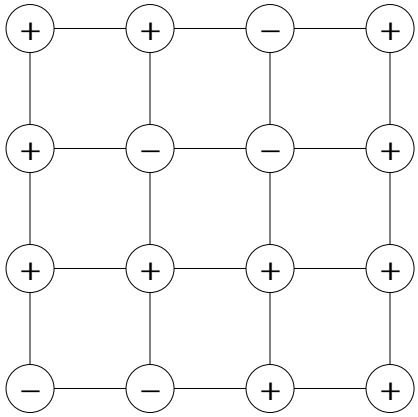

Figure 1.2: Example configuration of a 2-dimensional Ising model.

Thus low energy configurations are those in which many pairs of neighboring spins are aligned with one another. A famous classical fact about the Ising model is that there is a well-defined "critical (inverse) temperature" β_c at which the macroscopic behavior of the system changes dramatically: for $\beta < \beta_c$, the system will be in a "disordered" configuration (consisting of an essentially random sea of $+$ and $-$); and for $\beta > \beta_c$, the system will exhibit long-range order, so that there is likely to be a large region of $+$ (or of $-$) of size comparable to that of the entire graph. This phenomenon is referred to as a "phase transition", and corresponds to the onset of spontaneous magnetization in the ferromagnetic material.

Applications of random sampling from the Gibbs distribution include the following:

1. Examine typical configurations at a given value of β.
2. Compute expectations w.r.t. π (e.g., for the Ising model, the mean magnetization, specific heat, susceptibility).
3. Estimate Z (the partition function). Typically Z carries information about all the thermodynamic properties of the system.

The Glauber Dynamics

A standard approach to MCMC sampling from the Gibbs distribution is known as the Glauber dynamics. We illustrate this for the Ising model, though it is much more general. This is a Markov chain which at each step selects a vertex $v \in V$ at random and sets its spin to $+$ or $-$ with the appropriate probability *conditional on the spins of its neighbors*. Thus the probability of setting the spin to $+$ is $\frac{e^{\beta(n-p)}}{e^{\beta(n-p)}+e^{\beta(p-n)}} = \frac{1}{1+e^{\beta(2n-2p)}}$, where p is the number of neighbors with spin $+$ and n is the number of neighbors with spin $-$. It is not hard to see that this Markov chain converges to the Gibbs distribution on Ω.

The Glauber dynamics is of interest for two reasons:

- It provides an MCMC algorithm for sampling from the Gibbs distribution.
- It is a plausible model for the actual evolution of the physical system (so is interesting as a random process in its own right).

The following remarkable fact about the Glauber dynamics has been proved relatively recently for such a classical model [MO94]:

Theorem 1.1 *The mixing time of the Glauber dynamics for the Ising model on a $\sqrt{n} \times \sqrt{n}$ box in the 2-dimensional square lattice is:*

$$\begin{cases} O(n \log n) & \text{if } \beta < \beta_c; \\ e^{\Omega(\sqrt{n})} & \text{if } \beta > \beta_c, \end{cases}$$

where β_c is the critical point (i.e., phase transition).

This theorem is of algorithmic interest because it says that the physical phase transition has a *computational* manifestation, in the form of a sudden switch in the mixing time from linear to exponential. This gives us an additional motivation for studying MCMC in the context of statistical physics, because of its apparent connection with phase transitions.

1.2.5 Statistical Inference

Consider a statistical model with parameters Θ and a set of observed data X. The aim is to obtain Θ based on the observed data X; one way to formulate this problem is that we should *sample* Θ from the distribution $\Pr(\Theta \mid X)$. Using Bayes' rule, $\Pr(\Theta \mid X)$ translates to

$$\Pr(\Theta \mid X) = \frac{\Pr(X \mid \Theta) \Pr(\Theta)}{\Pr(X)},$$

where $\Pr(\Theta)$ is the *prior* distribution and refers to the information previously known about Θ, $\Pr(X \mid \Theta)$ is the probability that X is obtained with the assumed model, and $\Pr(X)$ is the unconditioned probability that X is observed. $\Pr(\Theta \mid X)$ is commonly called the *posterior* distribution and can be written in the form $\pi(\Theta) = w(\Theta)/Z$, where the weight $w(\Theta) = \Pr(X \mid \Theta) \Pr(\Theta)$ is easy to compute but the normalizing factor $Z = \Pr(X)$ is unknown. MCMC can then be used to sample from $\Pr(\Theta \mid X)$. We can further use the sampling in the following applications:

- Prediction: obtain the probability $\Pr(Y \mid X)$ that some future data Y is observed given X. $\Pr(Y \mid X)$ clearly can be written as $\sum_\Theta \Pr(Y \mid \Theta) \Pr(\Theta \mid X) = \mathrm{E}_\pi \Pr(Y \mid \Theta)$. Therefore we can use sampling to predict $\Pr(Y \mid X)$.
- Model comparison: perform sampling to estimate the normalizing factor $Z = \Pr(X)$ (using methods like those above for approximate counting), and use this to compare different models. Note that $\Pr(X)$ is the probability that the given model generated the observed data, so a model with a large value of $\Pr(X)$ should be preferred to a model with a small value.

References

[DFK89] M. Dyer, A. Frieze and R. Kannan, "A random polynomial time algorithm for estimating volumes of convex bodies," *Proceedings of the 21st ACM STOC*, 1989, pp. 375–381.

[L02] J. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, 2002.

[LV03] L. Lovász and S. Vempala. "Simulated annealing in convex bodies and an $\tilde{O}(n^4)$ volume algorithm." *Proceedings of the 44th IEEE FOCS*, 2003, pp. 650–659.

[MO94] F. Martinelli and E. Olivieri. "Approach to equilibrium of Glauber dynamics in the one phase region I: The attractive case." *Communications in Mathematical Physics* **161** (1994), pp. 447–486.