Sampling Self Avoiding Walks

James Fairbanks and Lianghao Chen

December 3, 2014

Abstract

These notes present the self testing algorithm for sampling self avoiding walks by Randall and Sinclair[3] [4]. They are intended to be used towards the end of a graduate class on Markov Chains. We are solving the problem of sampling Self Avoiding Walks of a fixed length.

The main lessons to learn are.

1. In Baretti-Sokal chain, changing a fixed constant β to a sequence of constants β_i allows one to prove convergence.
2. The mixing rate of the algorithm is polynomial if $c_j c_k \leq c_{j+k} g(j+k)$ for some polynomial g.
3. We can find the necessary β_i, a_i^{-1} by bootstrapping.
4. For any g we can test the necessary conjecture.

1 Introduction

A walk is self avoiding if no vertex is visited more than once. We want to sample self avoiding walks in order to answer physical questions about polymers such as how many possible walk are there of a given length, and how far is the free endpoint from the origin in a typical SAW. Let c_n be the number of self-avoiding walks of size n. We can count lattice walks exactly. At each step there are $2d$ possible neighbors and at each step every neighbor is a valid transition so there are $(2d)^n$ possible walks of length n on the d dimensional lattice. Since there are no iteractions between choices, we can sample exactly from the uniform distribution on lattice walks without a markov chain.

We cannot sample random walks with the process that starts with a walk of length 1 and extends it with equal probability in any direction that does not create a self intersection, because at each step some choices of direction will lead to more possibilities at longer steps.

The pivot algorithm analysed in [2] introduces a Markov Chain that uses very nonlocal moves that have "not too small" acceptance fraction. The authors make a heuristic argument about the autocorrelation time, but do not prove fast mixing. The pivot algorithm takes steps where each vertex of the walk is chosen at random and then a randomly rotatation or reflection is applied to the segment after this point. These steps take points very far in distance but are not proven to have fast mixing.

It is clear that $\frac{c_{n+m}}{c_n c_m} \leq 1$ because it is the probability of picking one walks of length n, m and concatenating them to form a self avoiding walk of length $n+m$.

2 The Markov Chain

Construct Markov Chains $M_1, M_2, ...$, where each M_n has state space $\chi_n = \cup_{i=0}^{n} S_i$ the union of all self avoiding walks of length at most n. The physics community believes that there should be one parameter β that controls the MC but their chain cannot be proven to mix rapidly. Relaxing this to a sequence of parameters β_i is allows us to prove rapid mixing and uniformity for the correct choice of β_i.

For a self-avoiding walk $w \in \chi_n$ of length $i \leq n$, choose one edge randomly from the $2d$ edges that are incident to the free endpoint of w. If this edge coincides with the last step of w, remove this edge. If adding this edge also makes a self-avoiding walk of length at most n, add this edge with probability β_{i+1}.

Define $w \prec w'$ iff $|w| < |w'|$ and the first $|w|$ steps of w' coincides with w [1]. Define $w \prec_1 w'$ iff $w \prec w'$ and $|w'| = |w| + 1$.

$$P_n(w, w') = \begin{cases} \beta_{|w'|}/2d & \text{if } w \prec_1 w' \\ 1/2d & \text{if } w' \prec_1 w \\ r(w) & \text{if } w = w' \\ 0 & \text{otherwise} \end{cases}$$

Stationary distribution is

$$\pi_n(w) = \frac{1}{Z_n} \prod_{i=1}^{|w|} \beta_i$$

for $w \in \chi_n$ where Z_n is the normalizing factor $\sum_{w \in \chi_n} \prod_{i=1}^{|w|} \beta_i$

β_i is chosen as c_{i-1}/c_i so that the stationary distribution is uniform on the levels and uniform within the levels. Physicists believe that there is a limit for c_{i-1}/c_i so Baretti and Sokal chose a single β for all c_{i-1}/c_i. Here β is replaced by a sequence of β_i's which can be determined by bootstrapping. This allows the algorithm to provide correct answers at each level even if there were no constant β that satisfied the conjecture at all of the levels simultaneously.

2.1 Stationary Distribution

Here are the details of the stationary distribution that were skipped in class.

if $w <_1 w'$:

$$\prod_{i=1}^{|w|} \beta_i \beta_{|w'|} \frac{1}{2d} = \prod_{i=1}^{|w'|} \beta_i \frac{1}{2d}$$

if $w' <_1 w$:

$$\prod_{i=1}^{|w'|} \beta_i \beta_{|w|} \frac{1}{2d} = \prod_{i=1}^{|w|} \beta_i \frac{1}{2d}$$

if $w' = w$:

$$\prod_{i=1}^{|w|} \beta_i r(w) = \prod_{i=1}^{|w'|} \beta_i r(w)$$

otherwise both transition probabilities are 0 so the detailed balance equation is satisfied.

Assume that each β_i is equal to c_{i-1}/c_i. Then

$$\pi_n(w) = \frac{1}{Z_n} \prod_{i=1}^{|w|} \frac{c_{i-1}}{c_i} = \frac{1}{Z_n c_{|w|}}$$

2.2 Uniform

Here are the details of the uniformity that were skipped in class. Since the stationary probability is the same for each walk of the same length, conditioning on the length gives the uniform distribution on that length. The distribution of lengths is given by the following where $\hat{\chi_j}$ is the set of paths with length exactly j

$$P(|w| = j) = \frac{1}{Z_n} |\hat{\chi_j}| \prod_{i=1}^{j} \beta_i$$

[1] w' extends w

This tells us that if each of the products $\prod_{i=1}^{j} \beta_i$ is equal to the number of walks with length j, then the distribution is uniform on all walks. This can be achieved by setting $\beta_i = \frac{c_{i-1}}{c_i}$ because the product telescopes appropriately. We might be interested in biasing towards large walks, which would lead to β_i values that are larger. We also cannot a priori state the number of walks of a given length since this is the quantity we are trying to estimate. If β_i were known, then we could run the markov chain and sample efficiently. Since we need to find the β_i, we bootstrap this parameter. The innovation of allowing the β parameter to vary at each level of the tree allows the sampler to work for all levels up to the point where the conjecture is false, if such a point exists. The Baretti and Sokal chain with a fixed β for all levels is not robust in this way.

3 The Mixing Time (exact case)

We define a sequence α_n which bounds the fraction of pairs of self avoiding walks whose concatenation has length n which are self avoiding.

$$\alpha_n = \min_{j+k<n} \frac{c_{j+k}}{c_j c_k}$$

The key idea is to bound the congestion across any edge at length k. We have $Q(e) = \pi_n(w)P_n(w, w') = 1/4dZ_n c_{k+1}$ for the ergodic flow across any edge. and the probability of any subtree S which is the extensions of w' with length $k+1$ is

$$\begin{aligned}
\pi_n(S) &= \sum_{\tilde{w} \succeq w'} \pi_n(\tilde{w}) && (1)\\
&= \frac{1}{Z_n c_{k+1}} \sum_{j=k+1}^{n} \frac{c_{k+1}}{c_j} |\text{extensions of w with } j \text{ steps}| && (2)\\
&\leq \frac{1}{Z_n c_{k+1}} \sum_{j=k+1}^{n} \frac{c_{k+1} c_{j-k-1}}{c_j} && (3)\\
&\leq \frac{n}{Z_n c_{k+1} \alpha_n} && (4)
\end{aligned}$$

Where we use the fact that the number of ways to extend a walk by i steps is less than c_i which is the sub-Cayley property for the tree. So $\rho(\Gamma) \leq 4dn\alpha_n^{-1}$.

From the mixing time bound based on congestion we get a factor of $\log \pi_n(0)^{-1} = \log n$ and a factor of $2n$ from the length of the longest path in a tree. So the final mixing time is $\mathcal{O}\left(dn^2 \alpha_n^{-1} \log n \epsilon^{-1}\right)$

4 Assumed Conjecture

Physicists believe that there is a limiting constant for the expansion probability c_{i-1}/c_i, and that $c_n = \mu^n poly(n)$. It is widely believed that for any dimension d, there exists fixed polynomial g such that, for all n, m,

$$c_n c_m \leq g(n+m) c_{n+m} \tag{5}$$

That is that the probability of successful concatenation is bounded below by $1/g(n)$.

The conjecture implies that $\alpha_n^{-1} \leq g(n)$. The bound on the mixing time of the Markov Chain will be presented in terms of α. So this conjecture will imply fast mixing.

5 Bootstrapping

The mixing time of M_n depends on α_n^{-1} and the transition probabilities depend on β_i. We know neither of these. In fact the β_i are the principle quantity of interest to understanding the set of random walks. So we will assume that we can sample from M_{n-1} and show how to estimate α_n and β_n.

To estimate the α's we generate t_n independent pairs of walks from $S_i \times S_{n-i}$ for each $i \leq n$. Then we count what fraction of pairs of them produce self avoiding concatenations, call this $q_{n,i}$. By taking the minimum of α_{n-1} and $q_{n,i}$ we get an estimator within a factor of 4 of the true answer with sufficient probability. The 0/1 estimator theorem says that $t_n = \mathcal{O}\left(a_n^{-1} \log n/\delta\right)$ is sufficiently large for desired failure probability $1-\delta$. This leads to a $\tilde{O}(n^4\alpha_n^{-2})$ cumulative running time. One factor of α_n^{-1} for the per sample time and another factor from the number of samples.

We can start with the known quantities $\beta_1 = 1/2d$, $\bar{c}_1 = 2d$ $\alpha_1 = 1$. To estimate the $\beta_n = c_{n-1}/c_n$, we generate samples from M_{n-1} and then count what fraction can be extended to length n Then our estimate for the number of walks at length n is $\bar{c}_{n-1}/\beta_n$ and we use this value of β_n to estimate the α_n.

6 Self Testing

Everything said about the chain is true regardless of the conjecture given. The bounds are in terms of α_n^{-1} which might grow superpolynomially. If we want to guarantee that the running time is polynomial then we need to know that α_n^{-1} is polynomially bounded. Since we have not made progress on the conjecture, we adapt the algorithm to self-test. Given any function g, we can test the conjecture $c_m c_n \leq g(m+n)c_{m+n}$. We run the estimate α process above and then check if $\tilde{\alpha_n^{-1}} > 4g(n)$. If it is, then the conjecture has a counterexample with high probability. If our estimates always pass the check, then we can be confident that the answers are reliable with high probabily. The algorithm can output either a reliable answer or a error message with probability $1-\delta$.

7 Later Work

The following are three possible directions that later work can be built based on this paper.

1. Conjecture: For any dimension d, there exists fixed polynomial g,

$$\forall n, m \quad c_n c_m \leq g(n+m)c_{n+m}$$

   Proving this conjecture will make this algorithm the first polynomial time Monte Carlo approximation algorithm for self-avoiding walks.

2. There can be other algorithms that are self-testing like this one. When an algorithm is based on a conjecture, letting the algorithm reject if the conjecture is false will make it a more robust algorithm.

3. There are a few areas in physics and biology where the result in this paper can be applied to.

   Gambin and Wojtowicz[1] applied this algorithm on lattice models of protein folding. Protein folding is a physical process where a protein changes its three-dimensional structure from random coil. While there are several lattice models for protein, Gambin and Wojtowicz used the simplest one called three-dimensional HP model, where each protein is represented by a sequence of "H"s and "P"s ("H" represents hydrophobic amino acid type and "P" represents polar amino acid type). They considered these sequences as self-avoiding walks. Instead of defining c_n as the number of self-avoiding walks of length n, they define it as the sum of weighted self-avoiding walks of length i and introducing weighting factor λ.

$$c_n = \sum_{w \in S_i} \lambda^{h(w)}$$

   They made up two HP sequences and experimented by the EstimateBeta method. They got a table of numerical results and observed a pattern: The quantity α_n^{-1} seems independent with λ and the hydrophobicity pattern of the sequence.

References

[1] Anna Gambin and Damian Wójtowicz. Almost fpras for lattice models of protein folding. In *Experimental and Efficient Algorithms*, pages 534–544. Springer, 2005.

[2] Neal Madras and Alan D Sokal. The pivot algorithm: a highly efficient monte carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50(1-2):109–186, 1988.

[3] Dana Randall and Alistair Sinclair. Testable algorithms for self-avoiding walks. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 593–602, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[4] Dana Randall and Alistair Sinclair. Self-testing algorithms for self-avoiding walks. *Journal of Mathematical Physics*, 41(3):1570–1584, 2000.