On Markov chain Monte Carlo methods for sampling self-avoiding walks on the square lattice

Carlos Eduardo Marciano

Universidade Federal do Rio de Janeiro

Trabalho Final MCMC

`cemarciano@poli.ufrj.br`
`https://cemarciano.github.io`

Roadmap

Motivation

- Linear polymers: high density. Present in most plastics, pipes, nylon fabric... The list goes on!
- Comprised of long chains of basic units called monomers.
- How many configurations exist for an n-monomer chain? How far apart are its endpoints, in average [1]?

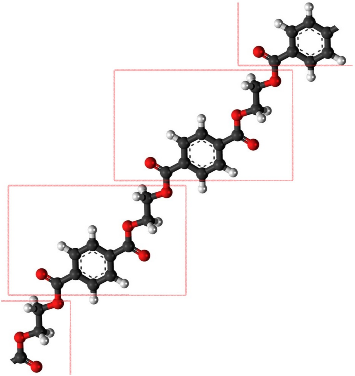


Figure 1: Polyethylene terephthalate
(Creative Commons)

The Self-Avoiding Walk

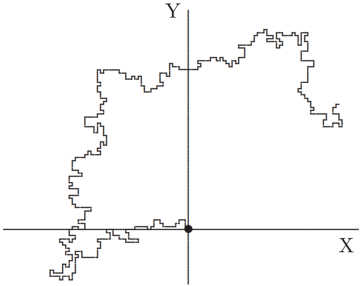


Figure 2: A 500-step *SAW* in the 2*D* lattice starting at the origin [2].

- The Self-Avoiding Walk (SAW): used in the past few decades to model linear polymers.
- Defined as an ordered set of $n+1$ sites in $\mathbb{Z}^d$ where consecutive sites are neighbors and no two sites coincide [1].
- Leads to mathematical problems that are easy enough to state and very hard to solve!

Basic Definitions (from [3])

The set $\mathcal{L}_n$

Set of n-step *SAWs* on the $\mathcal{L}$ d-dimensional lattice starting at the origin and ending anywhere. For simplicity, let's restrict attention to the square lattice $\mathbb{Z}^2$.

Basic Definitions (from [3])

The set $\mathcal{L}_n$

Set of n-step *SAWs* on the $\mathcal{L}$ d-dimensional lattice starting at the origin and ending anywhere. For simplicity, let's restrict attention to the square lattice $\mathbb{Z}^2$.

The value c_n

Cardinality of $\mathcal{L}_n$ (*i.e.:* number of possible n-step *SAWs* on $\mathbb{Z}^2$).

Basic Definitions (from [3])

The set $\mathcal{L}_n$

Set of n-step *SAWs* on the $\mathcal{L}$ d-dimensional lattice starting at the origin and ending anywhere. For simplicity, let's restrict attention to the square lattice $\mathbb{Z}^2$.

The value c_n

Cardinality of $\mathcal{L}_n$ (*i.e.:* number of possible n-step *SAWs* on $\mathbb{Z}^2$).

Mean-square end-to-end distance $< \omega_n^2 >$

$< \omega_n^2 > \equiv \frac{1}{c_n} \sum_x |x|^2 c_n(x)$ (*i.e.* avg. squared *Euclidian* distance).

Horseshoe

- *SAW* 3-element contiguous subsequence whose elements resemble a horseshoe shape.
- Expected value can be of help to some open questions in the theory of percolation.
- Computationally, for each site $i = (x, y)$, just check if $i - 3 \in \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$.

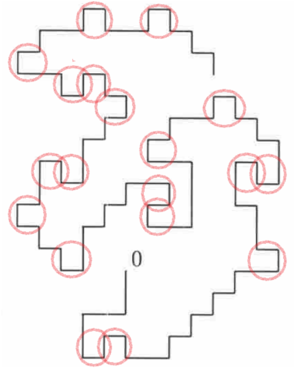


Figure 3: A 105-step *SAW* from [1] with its 19 horseshoes highlighted.

Roadmap

Enumeration

- Simple backtracking algorithm. First node in the backtrack tree has 4 children; all others have 3 or less.

n	c_n	n	c_n	n	c_n
1	4	11	120 292	21	2 408 806 028
2	12	12	324 932	22	6 444 560 484
3	36	13	881 500	23	17 266 613 812
4	100	14	2 374 444	24	46 146 397 316
5	284	15	6 416 596	25	123 481 354 908
6	780	16	17 245 332	26	329 712 786 220
7	2 172	17	46 466 676	27	881 317 491 628
8	5 916	18	124 658 732	28	2 351 378 582 244
9	16 268	19	335 116 620	29	6 279 396 229 332
10	44 100	20	897 697 164	30	16 741 957 935 348

Table 1: Values of c_n on the 2-dimensional square lattice.
Adapted from [1] and [4].

$c_{71} = 4\ 190\ 893\ 020\ 903935\ 054\ 619\ 120\ 005\ 916$

$c_n \sim A\,\mu^n\, n^{\gamma-1}$

Sampling Idea 0: Discard if not self-avoiding

- Problem: Not viable for generating large *SAWs*.

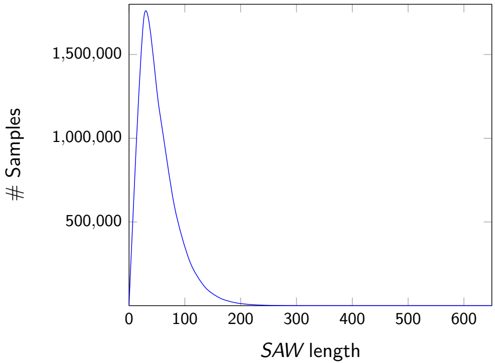


Figure 4: Histogram of *SAW* length after 10^8 trials. Adapted from [5].

Sampling Idea 1: Backtracking

- Iteratively add a site uniformly chosen to a *SAW* over all k $\in \{1, 2, 3, 4\}$ available next steps. Backtrack if trapped.

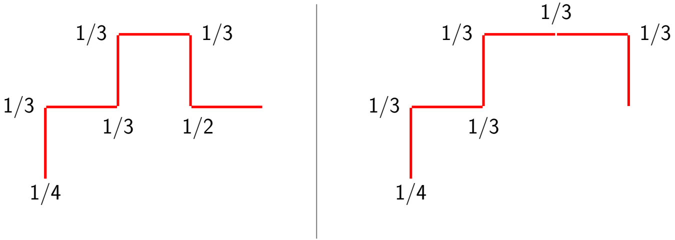


Figure 5: Two 6-step *SAWs* with different probabilities.

- <u>Problem 1:</u> leads to a non-uniform distribution over $\mathcal{L}_n$.

Sampling Idea 1: Backtracking

- Problem 2: Recovering from traps is exponentially difficult as n increases, limiting *SAW* lengths.

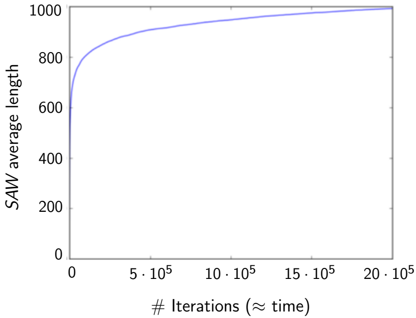


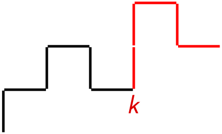
Figure 6: Average length of *SAW* after a number of iterations with 10^5 trials. Adapted from [5].

Promising Idea: Markov chain Monte Carlo

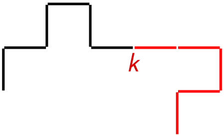
- Create a *Markov* chain with c_n states where each state represents an n-step *SAW*.

- Choose a site k and apply an orthogonal transformation to all sites after k (e.g. 90° rotation; x-axis reflection; etc), leading to another state.

- This approach originates the Pivot Algorithm.



(a) Pivot chosen.



(b) Transformation applied.

Figure 7: Foundations of the Pivot Algorithm.

The Pivot Algorithm (based on [3])

Initialize a self-avoiding walk ω

for *i=1 to num_samples* **do**

  Uniformly choose pivot $k \in [0, N-1]$

  Uniformly choose transformation T

  Apply T to $[k+1, N]$ sites in ω

  if *new walk* ω' *is self-avoiding* **then**

    $\omega \leftarrow \omega'$

  end

  Count ω as a sample

end

- There are a total of $2^d.d!$ orthogonal transf. For $d = 2$:
  1. identity (1)
  2. $\pm 90^\circ$ rotations (2)
  3. 180° rotation (1)
  4. axis reflections (2)
  5. diagonal refl. (2)
- Requires an appropriate data structure in order to achieve efficiency.

Roadmap

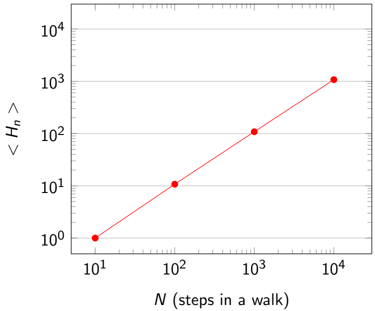
Implementing the Pivot Algorithm

- A hash table and an array are used to redundantly store all (x,y) pairs of the *SAW*, and they serve different purposes:
  1. The hash table provides collision checking in $O(1)$ for each site in the walk;
  2. The array provides random access for each site in the walk.
- As a consequence, each sample is generated in $O(n)$ time.
- Evaluation: mean-square end-to-end distance $<\omega_n^2>$ obtained after 10^7 samples (thermalization of 10^5):

n	$<\omega_n^2>$ (Madras & Sokal [3])	$<\omega_n^2>$ (this work)	Enumeration
15	47.2319 ± 0.0560	$47.2291 \pm ??$	47.2177
20	72.1227 ± 0.0940	$72.0767 \pm ??$	72.0765

Expected Value of Horseshoes in a *SAW*

- Average number of horseshoes $< H_n >$ obtained after 10^6 samples (thermalization of 10^4):



n	$< H_n >$	Enum.
10	1.00068	0.999546
15	1.54204	1.54465
20	2.10965	–
40	4.26707	–
80	8.57384	–
100	10.7461	–
1000	108.314	–
10000	1075.28	–

Figure 8: Expected value of horseshoes in a *SAW*.

Investigating Improvements for the Mixing Time

- Acceptance fraction f of the Pivot Algorithm [3] is estimated to behave as a power law $f \sim n^{-0.19}$
- What would happen if we used 2 pivots?

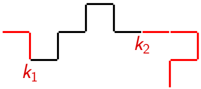


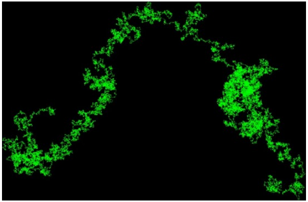
Figure 9: Choosing 2-pivots.

n	1-Pivot Accept. Frac.	2-Pivot Accept. Frac.
10	0.660813	0.475722
25	0.543953	0.322331
100	0.40568	0.17971
250	0.33601	0.12352
500	0.292144	0.092826
1000	0.254208	0.070387
2500	0.212067	0.049579

- Accept. fraction for 2-pivot roughly estimated as $f \sim n^{-0.38}$

Conclusion

- Markov chain Monte Carlo as a powerful tool to reliably estimate many hard to calculate attributes of *SAWs*.
- Future work: effective approaches for diminishing the mixing time of the MC.
- Future work: analytically derive expected value of horseshoes $E[H_n]$.



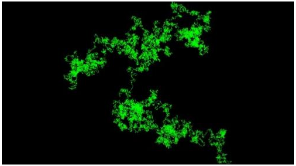


Figure 10: *SAWs* with over 10^7 sites
(by Nathan Clisby, from [6])

Closure

Thank you!

Questions & Answers

This presentation is available in PDF format at:
https://tinyurl.com/inoc2019-32

Bibliography I

[1] SLADE, G., "Self-Avoiding Walks", *The Mathematical Intelligencer*, v. 16, n. 1, pp. 29–35, 1994.

[2] JANSE VAN RENSBURG, E. J., "Monte Carlo methods for the self-avoiding walk", *Journal of Physics A: Mathematical and Theoretical*, v. 42, n. 32, pp. 1–97, July 2009.

[3] MADRAS, N., SOKAL, A. D., "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk", *Journal of Statistical Physics*, v. 50, n. 1/2, pp. 109–186, Jan. 1988.

[4] JENSEN, I., "Enumeration of self-avoiding walks on the square lattice", *Journal of Physics A: General Physics*, v. 37, n. 21, pp. 5503–5524, April 2004.

[5] PRÉA, P., ROUAULT, M., BRUCKER, F., "An optimal algorithm to generate extendable self-avoiding walks in arbitrary dimension", *Electronic Notes in Discrete Mathematics*, v. 59, pp. 37–50, June 2017.

[6] CLISBY, N., "Faster simulation of self-avoiding walks promises to untangle knotty problems in polymer science", 2017. Accessed in 12 June 2019. Available at: `https://clisby.net/projects/saw_feature/`.