A fast MCMC algorithm for the uniform sampling of binary matrices with fixed margins

Guanyang Wang

Stanford Math/Rutgers Stats

May 27, 2021

Outline

MCMC: History

- One sentence definition: A class of algorithms for sampling from a probability distribution.
- Invented by Metropolis, Ulam and co. at Los Alamos National Laboratory in 1950's
- Generalized by Hastings in 1970's
- 'Top 10 algorithms in the 20-th century'

Outline

Motivating Example: Darwin's Finches

Motivating Example: Darwin's Finches

- In 1835, Charles Darwin joined a geological expedition to the Galápagos archipelago, where he collected samples at each of the 17 islands
- Darwin noticed the species of finches (see below) varied from island to island
- This observation is often regarded as one of the main sparks that led to his theory of evolution.

	Islands																	
Species	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
B	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	14
C	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	14
D	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	13
E	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	12
F	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	11
G	1	1	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	10
H	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	10
I	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	10
J	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	6
K	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2
L	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
M	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Total	11	10	10	10	10	9	9	9	8	8	7	4	4	4	3	3	3	122

Motivating Example: Darwin's Finches

- **Ecologist's question**: Is the observed occurrence table the result of pure chance, or does it significantly differ from a random table?

Motivating Example: Darwin's Finches

- **Ecologist's question**: Is the observed occurrence table the result of pure chance, or does it significantly differ from a random table?
- **Statistician's answer**: Test by uniformly sampling binary matrices with fixed number of species per island (column sums) and fixed number of islands on which a species is found (row sums).

Motivating Example: Darwin's Finches

- **Ecologist's question**: Is the observed occurrence table the result of pure chance, or does it significantly differ from a random table?
- **Statistician's answer**: Test by uniformly sampling binary matrices with fixed number of species per island (column sums) and fixed number of islands on which a species is found (row sums).
- **Difficulty**: Sampling binary matrices with fixed margins is computationally very challenging.
- **Who cares?**: Ecologist, Sociologist, Biologist, Mathematician, Computer Scientist ...

Swap Algorithm

Given matrix size $m \times n$, row sums $\vec{r} = (r_1, \cdots, r_m)$, column sums $\vec{c} = (c_1, \cdots, c_n)$, we define $\Sigma_{\vec{r},\vec{c}}$ be all the binary matrices with row sums $\vec{r}$ and column sums $\vec{c}$

- **Target**: Uniformly sample from $\Sigma_{\vec{r},\vec{c}}$.

Swap Algorithm

- **Target**: Uniformly sample from $\Sigma_{\vec{r},\vec{c}}$.
- **Swap Algorithm**:
  Start with a matrix $M \in \Sigma_{\vec{r},\vec{c}}$.
  At each iteration
  1. Pick two rows and two columns at random
  2. If the intersection are one of the following two types

  $$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

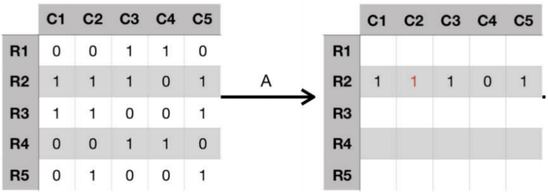
     then switch to the other type
  3. Otherwise, do nothing.

Remarks on Swap Algorithm

- Can be viewed as a Metropolis-Hastings algorithm with stationary distribution Unif$(\Sigma_{\vec{r},\vec{c}})$
- Easy to implement and widely used in practice in the last few decades
- Difficult to analyze theoretically. A famous conjecture proposed by Kannan, Tateli and Vempala [6] is still open for more than 20 years
- Very inefficient when the matrix is too sparse or dense

Rectangle Loop Algorithm: Main idea

- Choose **one** row and **one** column uniformly at random (R2 and C2 here)



Rectangle Loop Algorithm: Main idea

- Choose one row and one column uniformly at random (R2 and C2 here)
- Choose a **column** uniformly at random among all the 0 entries in R2 (C4 here is our only choice)

	C1	C2	C3	C4	C5
R1	0	0	1	1	0
R2	1	1	1	0	1
R3	1	1	0	0	1
R4	0	0	1	1	0
R5	0	1	0	0	1

$\xrightarrow{A}$

	C1	C2	C3	C4	C5
R1					
R2	1	1	1	0	1
R3					
R4					
R5					

$\xrightarrow{B}$

	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4				1	
R5				0	

Rectangle Loop Algorithm: Main idea

- Choose one row and one column uniformly at random
- Choose a **column** uniformly at random among all the **0s** in R2 (C4)
- Choose a **row** uniformly at random among all the **1s** in C4 (R4)

	C1	C2	C3	C4	C5
R1	0	0	1	1	0
R2	1	1	1	0	1
R3	1	1	0	0	1
R4	0	0	1	1	0
R5	0	1	0	0	1

A

	C1	C2	C3	C4	C5
R1					
R2	1	1	1	0	1
R3					
R4					
R5					

B

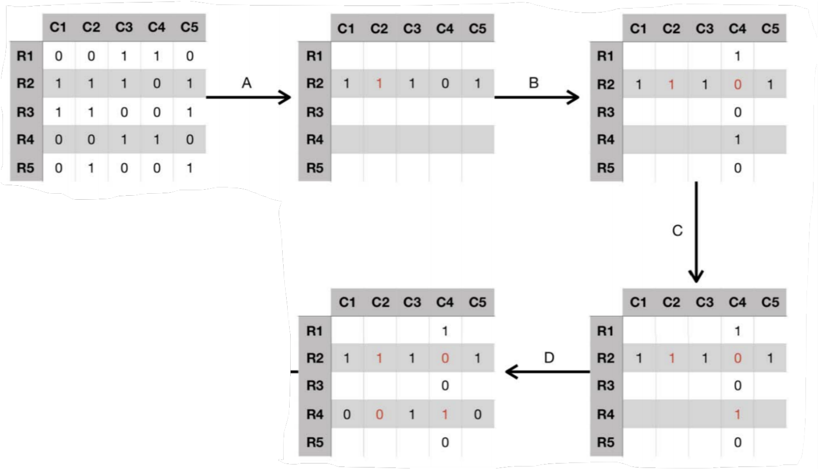
	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4				1	
R5				0	

C

	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4				1	
R5				0	

Rectangle Loop Algorithm: Main idea

- Choose a **column** uniformly at random among all the **0s** in R2 (C4)
- Choose a **row** uniformly at random among all the **1s** in C4 (R4)
- The three entries altogether give us the fourth entry (R4, C2)



Rectangle Loop Algorithm: Main idea

- Choose a **row** uniformly at random among all the **1s** in C4 (R4)
- The three entries altogether give us the fourth entry (R4, C2)
- If the 2×2 submatrix is 'swappable', swap it!

	C1	C2	C3	C4	C5
R1	0	0	1	1	0
R2	1	1	1	0	1
R3	1	1	0	0	1
R4	0	0	1	1	0
R5	0	1	0	0	1

A

	C1	C2	C3	C4	C5
R1					
R2	1	1	1	0	1
R3					
R4					
R5					

B

	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4				1	
R5				0	

C

	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4				1	
R5				0	

D

	C1	C2	C3	C4	C5
R1				1	
R2	1	1	1	0	1
R3				0	
R4	0	0	1	1	0
R5				0	

E

	C1	C2	C3	C4	C5
R1	0	0	1	1	0
R2	1	0	1	1	1
R3	1	1	0	0	1
R4	0	1	1	0	0
R5	0	1	0	0	1

Rectangle Loop Algorithm

Algorithm 3 Rectangle Loop Algorithm

Input: initial binary matrix A_0, number of iterations T

for $t = 1, \cdots T$ **do**
  Choose one row and one column (r_1, c_1) uniformly at random
  if $A_{t-1}(r_1, c_1) = 1$ **then**
    Choose one column c_2 at random among all the 0 entries in r_1
    Choose one row r_2 at random among all the 1 entries in c_2
  else $A_{t-1}(r_1, c_1) = 0$
    Choose one row r_2 at random among all the 1 entries in c_1
    Choose one column c_2 at random among all the 0 entries in r_2
  end if
  if the submatrix extracted from r_1, r_2, c_1, c_2 is a 'checkerboard unit' **then**
    Swap the submatrix
  else $A_t \leftarrow A_{t-1}$
  end if
end for

Rectangle Loop Algorithm: Theoretical Justification

- Given $\vec{r}, \vec{c}$ and an initial matrix $A_0 \in \Sigma_{\vec{r},\vec{c}}$, the Rectangale Loop algorithm defines an aperiodic, irreducible Markov chain with stationary distribution Unif$(\Sigma_{\vec{r},\vec{c}})$
- The Rectangle Loop algorithm dominates the Swap Algorithm in Peskun's ordering

Rectangle Loop Algorithm: Empirical Results

We ran both algorithms on 100×100 matrices with different filled portions, each for $10,000$ iterations.

Method	Filled portion	Number of swaps	Time per swap (/s)
Rectangle Loop	1%	586	1.18×10^{-5}
Swap		8	3.67×10^{-4}
Rectangle Loop	5%	977	5.30×10^{-6}
Swap		42	3.52×10^{-5}
Rectangle Loop	10%	1838	3.23×10^{-6}
Swap		156	1.25×10^{-5}
Rectangle Loop	20%	3271	2.64×10^{-6}
Swap		509	5.68×10^{-6}
Rectangle Loop	30%	4222	2.10×10^{-6}
Swap		803	5.06×10^{-6}
Rectangle Loop	40%	4794	1.27×10^{-6}
Swap		1160	4.98×10^{-6}
Rectangle Loop	50%	5080	1.37×10^{-6}
Swap		1271	5.36×10^{-6}

Rectangle Loop Algorithm: Empirical Results

- (Statistical Efficiency) Rectangle Loop Algorithm produces $4 - 73$ times more successful swaps **for a fixed number of iterations**.
- (Computational Efficiency) Rectangle Loop Algorithm produces $4 - 31$ times more successful swaps **for a fixed amount of time**.

Future Directions

- **Theory:** How to bound the mixing time of the Rectangle Loop algorithm?
- **Methodology:** How to sample from higher-dimensional contingency tables?
- **Computation:** How to design scalable Metropolis-Hastings algorithms?
- **Application:** Differential Privacy, Ecology, Social Network Study $\cdots$

Thank you all the audience!

Jose H Blanchet et al.
Efficient importance sampling for binary contingency tables.
The Annals of Applied Probability, 19(3):949–982, 2009.

Yuguo Chen, Persi Diaconis, Susan P Holmes, and Jun S Liu.
Sequential monte carlo methods for statistical analysis of tables.
Journal of the American Statistical Association, 100(469):109–120, 2005.

Alan E Gelfand and Adrian FM Smith.
Sampling-based approaches to calculating marginal densities.
Journal of the American statistical association, 85(410):398–409, 1990.

Stuart Geman and Donald Geman.
Stochastic relaxation, gibbs distributions, and the bayesian restoration of images.
IEEE Transactions on pattern analysis and machine intelligence, (6):721–741, 1984.

W Keith Hastings.
Monte carlo sampling methods using markov chains and their applications.
1970.

Ravi Kannan, Prasad Tetali, and Santosh Vempala.
Simple markov-chain algorithms for generating bipartite graphs and tournaments.
Random Structures & Algorithms, 14(4):293–308, 1999.

Kerrie L Mengersen, Richard L Tweedie, et al.
Rates of convergence of the hastings and metropolis algorithms.
The Annals of Statistics, 24(1):101–121, 1996.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller.
Equation of state calculations by fast computing machines.
The journal of chemical physics, 21(6):1087–1092, 1953.

Sean P Meyn and Richard L Tweedie.
Markov chains and stochastic stability.
Springer Science & Business Media, 2012.

Iain Murray, Zoubin Ghahramani, and David MacKay.
Mcmc for doubly-intractable distributions.
arXiv preprint arXiv:1206.6848, 2012.

Peter H Peskun.
Optimum monte-carlo sampling using markov chains.
Biometrika, 60(3):607–612, 1973.

Gareth Roberts and Jeffrey Rosenthal.
Geometric ergodicity and hybrid markov chains.
Electronic Communications in Probability, 2:13–25, 1997.

Martin A Tanner and Wing Hung Wong.
The calculation of posterior distributions by data augmentation.
Journal of the American statistical Association, 82(398):528–540, 1987.

Luke Tierney.
A note on metropolis-hastings kernels for general state spaces.
The Annals of Applied Probability, 8(1):1–9, 1998.