WIKIPEDIA

Markov chain Monte Carlo

In statistics, **Markov chain Monte Carlo** (**MCMC**) methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain. The more steps that are included, the more closely the distribution of the sample matches the actual desired distribution. Various algorithms exist for constructing the Markov chain including the Metropolis–Hastings algorithm.

Contents

Application domains

Markov chain Monte Carlo methods are primarily used for calculating numerical approximations of multi-dimensional integrals, for example in Bayesian statistics, computational physics[1], computational biology,[2] and computational linguistics.[3][4]

In Bayesian statistics, the recent development of Markov chain Monte Carlo methods has been a key step in making it possible to compute large hierarchical models that require integrations over hundreds or even thousands of unknown parameters.[5]

In rare event sampling, they are also used for generating samples that gradually populate the rare failure region.

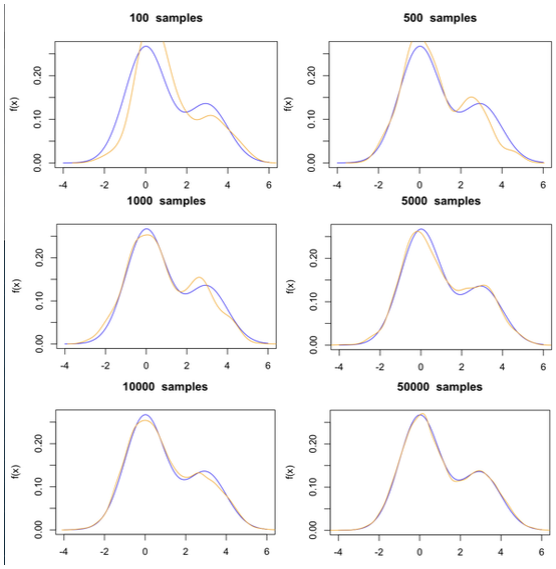
General explanation

Markov chain Monte Carlo methods create samples from a possibly multi-dimensional continuous random variable, with probability density proportional to a known function. These samples can be used to evaluate an integral over that variable, as its expected value or variance.

Practically, an ensemble of chains is generally developed, starting from a set of points arbitrarily chosen and sufficiently distant from each other. These chains are stochastic processes of "walkers" which move around randomly according to an algorithm which looks for places with a reasonably high contribution to the integral to move into next, assigning them higher probabilities.

Random walk Monte Carlo methods are a kind of random simulation or Monte Carlo method. However, whereas the random samples of the integrand used in a conventional Monte Carlo integration are statistically independent, those used in Markov chain Monte Carlo methods are autocorrelated. Correlations of samples introduces the need to use the Markov chain central limit theorem when estimating the error of mean values.

These algorithms create Markov chains such that they have an equilibrium distribution which is proportional to the function given.



Convergence of the Metropolis–Hastings algorithm. Markov chain Monte Carlo attempts to approximate the blue distribution with the orange distribution.

Reducing correlation

While MCMC methods were created to address multi-dimensional problems better than simple Monte Carlo algorithms, when the number of dimensions rises they too tend to suffer the curse of dimensionality: the regions of higher probability tend to stretch and get lost in an increasing volume of space that gives little contribution to the desired integral. One way to address this problem could be shortening the steps of the walker, so that it doesn't continuously try to exit the highest probability region, though this way the process would be highly autocorrelated and quite ineffective (i.e. many steps would be required for an accurate result). More sophisticated methods use various ways of reducing the autocorrelation, while managing to keep the process in the regions that give a higher contribution to the integral. These algorithms usually rely on a more complicated theory, and may be harder to implement, but they usually exhibit faster convergence (fewer steps required).

Examples

Examples of random walk Monte Carlo methods include the following:

- Metropolis–Hastings algorithm: This method generates a Markov chain using a proposal density for new steps and a method for rejecting some of the proposed moves. It is actually a general framework which includes as special cases the very first and simpler MCMC (Metropolis algorithm) and many more recent alternatives listed below.
  - Gibbs sampling: This method requires all the conditional distributions of the target distribution to be sampled exactly. When drawing from the full-conditional distributions is not straightforward other samplers-within-Gibbs are used (e.g., see [6][7][8]). Gibbs sampling is popular partly because it does not require any 'tuning'.
  - Metropolis-adjusted Langevin algorithm and other methods that rely on the gradient (and possibly second derivative) of the log target density to propose steps that are more likely to be in the direction of higher probability density.[9]
  - Pseudo-marginal Metropolis–Hastings: This method replaces the evaluation of the density of the target distribution with an unbiased estimate and is useful when the target density is not available analytically, e.g. latent variable models.
- Slice sampling: This method depends on the principle that one can sample from a distribution by sampling uniformly from the region under the plot of its density function. It alternates uniform sampling in the vertical direction with uniform sampling from the horizontal 'slice' defined by the current vertical position.
- Multiple-try Metropolis: This method is a variation of the Metropolis–Hastings algorithm that allows multiple trials at each point. By making it possible to take larger steps at each iteration, it helps address the curse of dimensionality.[10][11]
- Reversible-jump: This method is a variant of the Metropolis–Hastings algorithm that allows proposals that change the dimensionality of the space.[12] Markov chain Monte Carlo methods that change dimensionality have long been used in statistical physics applications, where for some problems a distribution that is a grand canonical ensemble is used (e.g., when the number of molecules in a box is variable). But the reversible-jump variant is useful when doing Markov chain Monte Carlo or Gibbs sampling over nonparametric Bayesian models such as

those involving the Dirichlet process or Chinese restaurant process, where the number of mixing components/clusters/etc. is automatically inferred from the data.

- Hamiltonian (or Hybrid) Monte Carlo (HMC): Tries to avoid random walk behaviour by introducing an auxiliary momentum vector and implementing Hamiltonian dynamics, so the potential energy function is the target density. The momentum samples are discarded after sampling. The end result of Hybrid Monte Carlo is that proposals move across the sample space in larger steps; they are therefore less correlated and converge to the target distribution more rapidly.

Training-based Markov chain Monte Carlo

Unlike most of the current Markov chain Monte Carlo methods that ignore the previous trials, using a new algorithm the Markov chain Monte Carlo algorithm is able to use the previous steps and generate the next candidate. This training-based algorithm is able to speed-up the Markov chain Monte Carlo algorithm by an order of magnitude.[13]

Interacting Markov chain Monte Carlo methodologies are a class of mean field particle methods for obtaining random samples from a sequence of probability distributions with an increasing level of sampling complexity.[14] These probabilistic models include path space state models with increasing time horizon, posterior distributions w.r.t. sequence of partial observations, increasing constraint level sets for conditional distributions, decreasing temperature schedules associated with some Boltzmann-Gibbs distributions, and many others. In principle, any Markov chain Monte Carlo sampler can be turned into an interacting Markov chain Monte Carlo sampler. These interacting Markov chain Monte Carlo samplers can be interpreted as a way to run in parallel a sequence of Markov chain Monte Carlo samplers. For instance, interacting simulated annealing algorithms are based on independent Metropolis-Hastings moves interacting sequentially with a selection-resampling type mechanism. In contrast to traditional Markov chain Monte Carlo methods, the precision parameter of this class of interacting Markov chain Monte Carlo samplers is *only* related to the number of interacting Markov chain Monte Carlo samplers. These advanced particle methodologies belong to the class of Feynman-Kac particle models,[15][16] also called Sequential Monte Carlo or particle filter methods in Bayesian inference and signal processing communities.[17] Interacting Markov chain Monte Carlo methods can also be interpreted as a mutation-selection genetic particle algorithm with Markov chain Monte Carlo mutations.

Markov Chain quasi-Monte Carlo (MCQMC).[18][19]

The advantage of low-discrepancy sequences in lieu of random numbers for simple independent Monte Carlo sampling is well known.[20] This procedure, known as Quasi-Monte Carlo method (QMC),[21] yields an integration error that decays at a superior rate to that obtained by IID sampling, by the Koksma-Hlawka inequality. Empirically it allows the reduction of both estimation error and convergence time by an order of magnitude. The Array-RQMC method combines randomized quasi-Monte Carlo and Markov chain simulation by simulating n chains simultaneously in a way that the empirical distribution of the n states at any given step is a better approximation of the true distribution of the chain than with ordinary MCMC.[22] In empirical experiments, the variance of the average of a function of the state sometimes converges at rate $O(n^{-2})$ or even faster, instead of the $O(n^{-1})$ Monte Carlo rate.[23]

Convergence

Usually it is not hard to construct a Markov chain with the desired properties. The more difficult problem is to determine how many steps are needed to converge to the stationary distribution within an acceptable error.[24] A good chain will have rapid mixing: the stationary distribution is reached quickly starting from an arbitrary position. A standard empirical method to assess convergence is to run several independent simulated Markov chains and check that the ratio of inter-chain to intra-chain variances for all the parameters sampled is close to 1.[24][25]

Typically, Markov chain Monte Carlo sampling can only approximate the target distribution, as there is always some residual effect of the starting position. More sophisticated Markov chain Monte Carlo-based algorithms such as coupling from the past can produce exact samples, at the cost of additional computation and an unbounded (though finite in expectation) running time.

Many random walk Monte Carlo methods move around the equilibrium distribution in relatively small steps, with no tendency for the steps to proceed in the same direction. These methods are easy to implement and analyze, but unfortunately it can take a long time for the walker to explore all of the space. The walker will often double back and cover ground already covered.

Software

Several software programs provide MCMC sampling capabilities, for example:

- Packages that use dialects of the BUGS model language:
  - WinBUGS / OpenBUGS/ MultiBUGS (https://www.multibugs.org/)
  - JAGS
  - NIMBLE (https://r-nimble.org/)
- greta (https://greta-dev.github.io/greta/), a Bayesian statistical modeling language / R package which uses TensorFlow behind the scenes,[26] similar to PyMC3's use of Theano as the computational back-end
- MCSim
- PyMC3
- pymcmcstat (https://github.com/prmiles/pymcmcstat/wiki)
- R (programming language) with the packages adaptMCMC, atmcmc, BRugs, mcmc, MCMCpack, ramcmc, rjags, rstan, etc.
- Stan
- TensorFlow Probability (https://www.tensorflow.org/probability/) (probabilistic programming library built on TensorFlow)
- MCL (http://micans.org/mcl/) (a cluster algorithm for graphs)[27] and HipMCL (https://bitbucket.org/azadcse/hipmcl/wiki/Home) (a parallelized version)[28]
- emcee (http://dfm.io/emcee/current/) (MIT licensed pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo Ensemble sampler)
- MacMCMC (https://www.causascientia.org/software/MacMCMC/MacMCMC.html) (Standalone, full-featured MCMC application for Mac OS)

See also

- Coupling from the past
- Metropolis-adjusted Langevin algorithm
- Markov chain central limit theorem

References

Citations

1. Kasim, M.F.; Bott, A.F.A.; Tzeferacos, P.; Lamb, D.Q.; Gregori, G.; Vinko, S.M. (September 2019). "Retrieving fields from proton radiography without source profiles". *Physical Review E*. **100**. doi:10.1103/PhysRevE.100.033208 (https://doi.org/10.1103%2FPhysRevE.100.033208).
2. Gupta, Ankur; Rawlings, James B. (April 2014). "Comparison of Parameter Estimation Methods in Stochastic Chemical Kinetic Models: Examples in Systems Biology" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4946376). *AIChE Journal*. **60** (4): 1253–1268. doi:10.1002/aic.14409 (https://doi.org/10.1002%2Faic.14409). PMC 4946376 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4946376). PMID 27429455 (https://www.ncbi.nlm.nih.gov/pubmed/27429455).
3. See Gill 2008.
4. See Robert & Casella 2004.
5. Banerjee, Sudipto; Carlin, Bradley P.; Gelfand, Alan P. (2014-09-12). *Hierarchical Modeling and Analysis for Spatial Data* (Second ed.). CRC Press. p. xix. ISBN 978-1-4398-1917-3.

6. Gilks, W. R.; Wild, P. (1992-01-01). "Adaptive Rejection Sampling for Gibbs Sampling". *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. **41** (2): 337–348. doi:10.2307/2347565 (https://doi.org/10.2307%2F2347565). JSTOR 2347565 (https://www.jstor.org/stable/2347565).
7. Gilks, W. R.; Best, N. G.; Tan, K. K. C. (1995-01-01). "Adaptive Rejection Metropolis Sampling within Gibbs Sampling". *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. **44** (4): 455–472. doi:10.2307/2986138 (https://doi.org/10.2307%2F2986138). JSTOR 2986138 (https://www.jstor.org/stable/2986138).
8. Martino, L.; Read, J.; Luengo, D. (2015-06-01). "Independent Doubly Adaptive Rejection Metropolis Sampling Within Gibbs Sampling". *IEEE Transactions on Signal Processing*. **63** (12): 3123–3138. arXiv:1205.5494 (https://arxiv.org/abs/1205.5494). Bibcode:2015ITSP...63.3123M (https://ui.adsabs.harvard.edu/abs/2015ITSP...63.3123M). doi:10.1109/TSP.2015.2420537 (https://doi.org/10.1109%2FTSP.2015.2420537). ISSN 1053-587X (https://www.worldcat.org/issn/1053-587X).
9. See Stramer 1999.
10. Liu, Jun S.; Liang, Faming; Wong, Wing Hung (2000-03-01). "The Multiple-Try Method and Local Optimization in Metropolis Sampling". *Journal of the American Statistical Association*. **95** (449): 121–134. doi:10.1080/01621459.2000.10473908 (https://doi.org/10.1080%2F01621459.2000.10473908). ISSN 0162-1459 (https://www.worldcat.org/issn/0162-1459).
11. Martino, Luca; Read, Jesse (2013-07-11). "On the flexibility of the design of multiple try Metropolis schemes". *Computational Statistics*. **28** (6): 2797–2823. arXiv:1201.0646 (https://arxiv.org/abs/1201.0646). doi:10.1007/s00180-013-0429-2 (https://doi.org/10.1007%2Fs00180-013-0429-2). ISSN 0943-4062 (https://www.worldcat.org/issn/0943-4062).
12. See Green 1995.
13. Tahmasebi, Pejman; Javadpour, Farzam; Sahimi, Muhammad (August 2016). "Stochastic shale permeability matching: Three-dimensional characterization and modeling" (https://www.researchgate.net/publication/307626119). *International Journal of Coal Geology*. **165**: 231–242. doi:10.1016/j.coal.2016.08.024 (https://doi.org/10.1016%2Fj.coal.2016.08.024).
14. Del Moral, Pierre (2013). *Mean field simulation for Monte Carlo integration* (http://www.crcpress.com/product/isbn/9781466504059). Chapman & Hall/CRC Press. p. 626. "Monographs on Statistics & Applied Probability"
15. Del Moral, Pierre (2004). *Feynman-Kac formulae. Genealogical and interacting particle approximations* (https://www.springer.com/mathematics/probability/book/978-0-387-20268-6). Springer. p. 575. "Series: Probability and Applications"
16. Del Moral, Pierre; Miclo, Laurent (2000). *Branching and Interacting Particle Systems Approximations of Feynman-Kac Formulae with Applications to Non-Linear Filtering* (http://archive.numdam.org/ARCHIVE/SPS/SPS_2000__34_/SPS_2000__34__1_0/SPS_2000__34__1_0.pdf) (PDF). *Lecture Notes in Mathematics*. **1729**. pp. 1–145. doi:10.1007/bfb0103798 (https://doi.org/10.1007%2Fbfb0103798). ISBN 978-3-540-67314-9.
17. Del Moral, Pierre (2006). "Sequential Monte Carlo samplers - P. Del Moral - A. Doucet - A. Jasra - 2006 - Journal of the Royal Statistical Society: Series B (Statistical Methodology) - Wiley Online Library". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. **68** (3): 411–436. arXiv:cond-mat/0212648 (https://arxiv.org/abs/cond-mat/0212648). doi:10.1111/j.1467-9868.2006.00553.x (https://doi.org/10.1111%2Fj.1467-9868.2006.00553.x).
18. Chen, S.; Dick, Josef; Owen, Art B. (2011). "Consistency of Markov chain quasi-Monte Carlo on continuous state spaces". *Annals of Statistics*. **39** (2): 673–701. doi:10.1214/10-AOS831 (https://doi.org/10.1214%2F10-AOS831).
19. Tribble, Seth D. (2007). *Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences* (Diss.). Stanford University. ProQuest 304808879 (https://search.proquest.com/docview/304808879).
20. Papageorgiou, Anargyros; Traub, J. F. (1996). "Beating Monte Carlo". *Risk*. **9** (6): 63–65.
21. Sobol, Ilya M (1998). "On quasi-monte carlo integrations". *Mathematics and Computers in Simulation*. **47** (2): 103–112. doi:10.1016/s0378-4754(98)00096-2 (https://doi.org/10.1016%2Fs0378-4754%2898%2900096-2).
22. L'Ecuyer, P.; Lécot, C.; Tuffin, B. (2008). "A Randomized Quasi-Monte Carlo Simulation Method for Markov Chains" (https://hal.inria.fr/inria-00070462/file/RR-5545.pdf) (PDF). *Operations Research*. **56** (4): 958–975. doi:10.1287/opre.1080.0556 (https://doi.org/10.1287%2Fopre.1080.0556).
23. L'Ecuyer, P.; Munger, D.; Lécot, C.; Tuffin, B. (2018). "Sorting Methods and Convergence Rates for Array-RQMC: Some Empirical Comparisons". *Mathematics and Computers in Simulation*. **143**: 191–201. doi:10.1016/j.matcom.2016.07.010 (https://doi.org/10.1016%2Fj.matcom.2016.07.010).

24. Gelman, A.; Rubin, D.B. (1992). "Inference from iterative simulation using multiple sequences (with discussion)" (http://www.stat.duke.edu/~scs/Courses/Stat376/Papers/ConvergeDiagnostics/GelmanRubinStatSci1992.pdf) (PDF). *Statistical Science*. **7** (4): 457–511. Bibcode:1992StaSc...7..457G (https://ui.adsabs.harvard.edu/abs/1992StaSc...7..457G). doi:10.1214/ss/1177011136 (https://doi.org/10.1214%2Fss%2F1177011136).
25. Cowles, M.K.; Carlin, B.P. (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review". *Journal of the American Statistical Association*. **91** (434): 883–904. CiteSeerX 10.1.1.53.3445 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.3445). doi:10.1080/01621459.1996.10476956 (https://doi.org/10.1080%2F01621459.1996.10476956).
26. "greta's software dependencies and inspirations" (https://greta-dev.github.io/greta/software.html). *greta-dev.github.io*. Retrieved 2018-10-02.
27. Enright, AJ; Van Dongen, S; Ouzounis, CA (1 April 2002). "An efficient algorithm for large-scale detection of protein families" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC101833). *Nucleic Acids Research*. **30** (7): 1575–84. doi:10.1093/nar/30.7.1575 (https://doi.org/10.1093%2Fnar%2F30.7.1575). PMC 101833 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC101833). PMID 11917018 (https://www.ncbi.nlm.nih.gov/pubmed/11917018).
28. Azad, A; Pavlopoulos, GA; Ouzounis, CA; Kyrpides, NC; Buluç, A (6 April 2018). "HipMCL: a high-performance parallel implementation of the Markov clustering algorithm for large-scale networks" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5888241). *Nucleic Acids Research*. **46** (6): e33. doi:10.1093/nar/gkx1313 (https://doi.org/10.1093%2Fnar%2Fgkx1313). PMC 5888241 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5888241). PMID 29315405 (https://www.ncbi.nlm.nih.gov/pubmed/29315405).

Sources

- Christophe Andrieu, Nando De Freitas, Arnaud Doucet and Michael I. Jordan *An Introduction to MCMC for Machine Learning* (http://www.cs.princeton.edu/courses/archive/spr06/cos598C/papers/AndrieuFreitasDoucetJordan2003.pdf), 2003
- Asmussen, Søren; Glynn, Peter W. (2007). *Stochastic Simulation: Algorithms and Analysis*. Stochastic Modelling and Applied Probability. **57**. Springer.
- Atzberger, P. "An Introduction to Monte-Carlo Methods" (http://www.math.ucsb.edu/~atzberg/spring2006/monteCarloMethod.pdf) (PDF).
- Berg, Bernd A. (2004). *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*. World Scientific.
- Bolstad, William M. (2010). *Understanding Computational Bayesian Statistics*. Wiley. ISBN 978-0-470-04609-8.
- Casella, George; George, Edward I. (1992). "Explaining the Gibbs sampler". *The American Statistician*. **46** (3): 167–174. CiteSeerX 10.1.1.554.3993 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.554.3993). doi:10.2307/2685208 (https://doi.org/10.2307%2F2685208). JSTOR 2685208 (https://www.jstor.org/stable/2685208).
- Gelfand, A.E.; Smith, A.F.M. (1990). "Sampling-Based Approaches to Calculating Marginal Densities". *Journal of the American Statistical Association*. **85** (410): 398–409. CiteSeerX 10.1.1.512.2330 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.512.2330). doi:10.1080/01621459.1990.10476213 (https://doi.org/10.1080%2F01621459.1990.10476213).
- Gelman, Andrew; Carlin, John B.; Stern, Hal S.; Rubin, Donald B. (1995). *Bayesian Data Analysis* (1st ed.). Chapman and Hall. *(See Chapter 11.)*
- Geman, S.; Geman, D. (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **6** (6): 721–741. doi:10.1109/TPAMI.1984.4767596 (https://doi.org/10.1109%2FTPAMI.1984.4767596).
- Gilks, W.R.; Richardson, S.; Spiegelhalter, D.J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC.
- Gill, Jeff (2008). *Bayesian methods: a social and behavioral sciences approach* (2nd ed.). Chapman and Hall/CRC. ISBN 978-1-58488-562-7.
- Green, P.J. (1995). "Reversible-jump Markov chain Monte Carlo computation and Bayesian model determination". *Biometrika*. **82** (4): 711–732. CiteSeerX 10.1.1.407.8942 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.407.8942). doi:10.1093/biomet/82.4.711 (https://doi.org/10.1093%2Fbiomet%2F82.4.711).
- Neal, Radford M. (2003). "Slice Sampling". *Annals of Statistics*. **31** (3): 705–767. doi:10.1214/aos/1056562461 (https://doi.org/10.1214%2Faos%2F1056562461). JSTOR 3448413 (https://www.jstor.org/stable/3448413).

- Neal, Radford M. (1993). "*Probabilistic Inference Using Markov Chain Monte Carlo Methods*" (http://www.cs.utoronto.ca/~radford/review.abstract.html).
- Robert, Christian P.; Casella, G. (2004). *Monte Carlo Statistical Methods* (2nd ed.). Springer. ISBN 978-0-387-21239-5.
- Rubinstein, R.Y.; Kroese, D.P. (2007). *Simulation and the Monte Carlo Method* (2nd ed.). Wiley. ISBN 978-0-470-17794-5.
- Smith, R.L. (1984). "Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions". *Operations Research*. **32** (6): 1296–1308. doi:10.1287/opre.32.6.1296 (https://doi.org/10.1287%2Fopre.32.6.1296).
- Spall, J.C. (April 2003). "Estimation via Markov Chain Monte Carlo". *IEEE Control Systems Magazine*. **23** (2): 34–45. doi:10.1109/mcs.2003.1188770 (https://doi.org/10.1109%2Fmcs.2003.1188770).
- Stramer, O.; Tweedie, R. (1999). "Langevin-Type Models II: Self-Targeting Candidates for MCMC Algorithms". *Methodology and Computing in Applied Probability*. **1** (3): 307–328. doi:10.1023/A:1010090512027 (https://doi.org/10.1023%2FA%3A1010090512027).

Further reading

- Diaconis, Persi (April 2009). "The Markov chain Monte Carlo revolution" (http://www.ams.org/bull/2009-46-02/S0273-0979-08-01238-X/S0273-0979-08-01238-X.pdf) (PDF). *Bull. Amer. Math. Soc.* **46** (2): 179–205. doi:10.1090/s0273-0979-08-01238-x (https://doi.org/10.1090%2Fs0273-0979-08-01238-x). S 0273-0979(08)01238-X.
- Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. (2007), "Section 15.8. Markov Chain Monte Carlo" (http://apps.nrbook.com/empanel/index.html#pg=824), *Numerical Recipes: The Art of Scientific Computing* (3rd ed.), Cambridge University Press, ISBN 978-0-521-88068-8
- Richey, Matthew (May 2010). "The Evolution of Markov Chain Monte Carlo Methods" (http://stat.wharton.upenn.edu/~stjensen/stat542/lecture14.mcmchistory.pdf) (PDF). *The American Mathematical Monthly*. **117** (5): 383–413. CiteSeerX 10.1.1.295.4478 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.295.4478). doi:10.4169/000298910x485923 (https://doi.org/10.4169%2F000298910x485923).

External links

- MCMC sampling and other methods in a basic overview (https://web.archive.org/web/20110531150413/http://www.bioss.ac.uk/students/alexm/MCMCintroPresentation.pdf), by Alexander Mantzaris (original link - now broken (http://www.bioss.ac.uk/students/alexm/MCMCintroPresentation.pdf))
- PyMC (https://pymc-devs.github.io/pymc/) - Python module implementing Bayesian statistical models and fitting algorithms, including Markov chain Monte Carlo.
- IA2RMS (http://a2rms.sourceforge.net) is a Matlab code of the "Independent Doubly Adaptive Rejection Metropolis Sampling" method, Martino, Read & Luengo (2015), for drawing from the full-conditional densities within a Gibbs sampler.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Markov_chain_Monte_Carlo&oldid=917727035"

This page was last edited on 25 September 2019, at 04:55 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.