Shiny Hastings: An Illustration of the Effect of the Proposal Density on the Metropolis Hastings Algorithm

Zarah Leonie Weiß
zarah-leonie.weiss@student.uni-tuebingen.de

Abstract

TO READ: also cite paper for shiny

1 Introduction

Markov chain Monte Carlo (MCMC) algorithms are used to approximate distributions by sampling from a target distribution. Unlike other sampling methods, MCMC methods do not necessarily draw from the actual target distribution. Instead, a distribution function proportional to the target distribution is sufficient. Therefore, MCMC algorithms are often employed for non-standard distributions from which direct sampling is too difficult or computationally costly, for example if the target distribution has multiple parameters (**???**). The two most commonly known types of MCMC algorithms are *Gibbs sampling* and *Metropolis Hastings*, of which the latter is subject of this article.

****

This characteristic is particularly beneficial for Bayesian inference approaches, when it comes to calculating the posterior distribution $P(\theta|D)$.

Using Bayes' theorem, displayed in Equation 1, the calculation of the actual posterior distribution can be computationally very costly due to the denominator containing an integral.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int_{\theta'} P(D|\theta')P(\theta')d\theta'} \tag{1}$$

In fact, the normalizing constant can be solved easily only, if θ is a single discrete variable with a restricted domain size and if the prior distribution is a conjugate prior for the likelihood function (**???**). However, since MCMC may as well sample from a distribution proportional to the target distribution, the denominator may be omitted, leading to the computationally easier formula in Equation 2, in which the posterior distribution is shown to be proportional to the prior distribution $P(\theta)$ multiplied by the likelihood $P(D|\theta)$.

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{2}$$

****

In order to approximate the target distribution successfully, it is crucial for MCMC methods to explore the sample space as exhaustively as possible, while converging in a reasonable amount of time. The efficiency and effectiveness of Metropolis Hastings (MH) may be tuned by the choice of the *proposal distribution* and the *acceptance rate*. Therefore, a vast amount of literature discusses different possible variants of proposal densities and acceptance rates (e.g. Jackman 2009; Liu 2001). While these discussions mostly remain theoretical and only partially discuss concrete implementations, I implemented a web application using Shiny (**shiny**), a web application framework for R.[1] The *Shiny Hastings* app allows to run a MH implementation using varying proposal distributions, that have been discussed in the literature. It was build to allow users to immediately experience the effect of different proposal distribution implementations on the MH algorithm and is accessible via GitHub[2] under `???`.

[1] `http://shiny.rstudio.com`.

[2] `???`.

In the following section 2, MCMC methods and the MH algorithm are introduced, focussing especially on two families of proposal densities: the *random-walk metropolis* and *independence metropolis*. After introducing the theoretical background and the general algorithm, the implementation of MH in *Shiny Hastings* is discussed in section 2. The report closes with the conclusion in section 4.

2 Theoretical Background

****

- mh / mcmc when to use? too many parameters for grid approximation (Kruschke:144) - sample from mathematically defines distribution instead of experimentally produced samples (Kruschke:145)

While it is not necessary for MCMC methods applied in Bayesian inference statistics, that the integral in Bayes' rule can be evaluated, as the denominator can be dropped, it is crucial, that both, prior and likelihood function can be computed easily for a given parameter θ. Based on this, MCMC returns an approximation of $P(\theta|D)$ in form of a large set of samples of θ values (Kruschke 2015, p. 144).

- history of MCMC reviewed in Robert & Casella 2010 (Robert 2016:1)

– MH algorithm (Jackman 2009:172) i.e. generate ergodic Markov chains wrt. posterior densities (Jackman 2009:172)

- Monte Carlo methods efficient for many dimensions, unlike numerical methods (Hastings 1970:97)

****

As just mentioned in section 1, MH is a sort of MCMC algorithm. Those algorithms may be used to approximate virtually any *target* distribution via sampling, even if direct sampling from that target distribution is computationally not feasible (Liu 2001, p. 105). This is due to the fact that it suffices for the algorithms to sample from distributions that are proportional to the target distribution (Jackman 2009, p. 140).[3] The target density is then characterized based on summary statistics such as mean, central tendency estimates or Highest Density Interval (HDI), calculated with the samples (Kruschke 2015, p. 145). An example for such a target distribution is the posterior distribution $P(\theta|data)$ in Bayesian inference statistics, which may be an arbitrary density not corresponding to any density known by statistics (Jackman 2009, p. 153).

2.1 MCMC

How do MCMC methods work? As their name suggests, they combine two prominent statistical principles: the Monte Carlo principle, a well known technique, that has already been used to solve deterministic and statistical problems in the 1870s (cf. ibid., p. 140), and the Markov chain property, named after Russian mathematician Andrey Markov (1856-1922) (ibid., p. 171):

The Monte Carlo principle states, that it is possible to draw conclusions about a random variable θ based on a large enough number of samples from its density $f(\theta)$ (cf. ibid., p. 133, , p. 1).

The crucial notion ensuring the Monte Carlo principle is *simulation consistency*. Informally speaking, due to the *law of large numbers* the statistical summary of a large enough amount of samples from a density $f(\theta)$ is an estimate of a property of θ, i.e. is *simulation-consistent* (Jackman 2009, p. 134).

****

For a formal definition of simulation consistency please consult Jackman (ibid., p. 138).

********** Due to this principle, it is possible to draw independent random samples from a distribution in order to characterize it, as long as the number of samples is sufficiently large. However, based on the Monte Carlo principle alone it is not possible to characterize a target distribution by sampling from *another* distribution proportional to it. This is, where Markov chains come into play (, p. 1).

The Markov chain property states, that for a sequence of time-indexed random variables, all believes about a given variable are based solely on its immediate past (Jackman 2009, 172f).

[3] Another method of distribution approximation without directly sampling from the actual target distribution is *importance sampling*, where samples instead are drawn from a *trial* distribution resembling the actual target distribution. However, it would be beyond the scope of this report to discuss this alternative approach, please see Hastings (1970), Jackman (2009), and Liu (2001) instead.

********** Put more formally, this means that given a collection of random variables $\theta^{(t)}$, where t is a time index, the following holds: $Pr(\theta^{(t+a)} = y|\theta^{(s)} = x_s, s \leq t) = P(\theta^{(t+a)} = y|\theta^{(t)} = x_t), \forall a > 0$.

********** Every sequence satisfying this property is a Markov chain.[4]

How is this combined with Monte Carlo methods? Markov chains are stochastic processes. If they are constructed appropriately on the parameter space Θ of a target distribution, i.e. set of states θ might take as values, they are *ergodic*. This means, they visit the potential values of θ in the parameter space proportional to the probability assigned to them by the target distribution (Jackman 2009, p. 172). Therefore, the iteration sequence of the Markov chain constitutes a representative sample of the target distribution by the Monte Carlo principle, although these samples are not independent from each other. In fact, *ergodictiy* is a form of the law of large numbers (ibid., p. 171). Yet, using dependently drawn samples is less efficient than using truly independent samples, which is why MCMC methods need more samples to return sufficient approximations than typical Monte Carlo methods.

********** In order to be suitable for MCMC methods, Markov chains need to satisfy certain properties.

- Transition kernel: (Jackman 2009:173) – reformulate markov property as transition probability, that at step t, Markov chain will jump from $\theta^{(t-1)}$ to set A, given Markov chain is at $\theta^{(t-1)}$: $Pr(\theta^{(t)} \in A|\theta^{(t)-1}, \theta^{(t-2)}, \dots) = Pr(\theta^{(t)} \in A|\theta^{(t-1)}), \forall A \subset \Theta$ - in discrete case, transition probabilities representable as square matrix with positive entries, i.e. transition matrix K (Jackman 2009:173), where$K_{ij} = Pr(\theta^{(t)} = j|\theta^{(t-1)} = i)$ – this characterizes the transition Kernel of Markov chain (Jackman 2009:174)

- Stationary distribution (Jackman 2009:177)

- Irreducibility (also called ergodic): – Markov chain needs to be able to access every state theta in Theta at every state Jackman 2009:179 – if it fulfils this, it is irreducible, i.e. if Θ is a communication class Jackman 2009:179) – absorbing state / closed state: cannot reach any other state from that state Jackman 2009:179 – accessible state: state which can be accessed by another state – communicating states are mutually accessible – communication class is formed by communicating states - in irreducible states it does not matter where we start from, at some point we will reach every state

- Recurrence: – state is transient if return time may be infinite – state is recurrent, otherwise – if waiting time for return is finite, it is positive recurrent – an irreducible Markoc chain on discrete state space is either posititive reucrrent for all states or for none – all states are positive recurrent in an irreducible Markov chain on finite state space, i.e. unique stationary distribution exists (Jackman 2009:183)

- Invariant measure (Jackman 2009:184)

- Reversibility (Jackman 2009:185)

****

2.2 MH Algorithm

The MH algorithm was first introduced by **metroplis1953** and generalized by Hastings (1970), henceforth being called the *Metropolis Hastings* algorithm.

****

On Metropolis 1953: - Metropolis algorithm: sample from distribution with a Markov process (Liu 2001:105) - algorithm used in varying scientific fields: biology, chemistry, computer science, economics, enginieering, material scienece, physics, statistics, etc. (Liu 2001:106) - can sample from all distributions regardless of complexity and dimensionality (Liu 2001:106) - problem: – high correlation of samples (Liu 2001:106) – leads to higher variances of samples compared to variances of independent samples (Liu 2001:106)

basic idea: – simulate Markov chain in state space to sample from (Liu 2001:106), such that the stationary distribution of this chain is the target distribution – note: — in Markov chain analysis we usually have a transition rule (Liu 2001:106) and searches for the stationary distribution (Liu 2001:106) — in MCMC simulations we know the stationary distribution, and we want to now an efficient transition rule (Liu 2001:106) - i.e. "The Metropolis algorithm

[4] It should be mentioned, that for the purposes of this paper only discrete time Markov chains were considered, but not the more general continuous time Markov processes. For more information on those, please see Jackman (2009, 172ff).

prescribes a transition rule for a Markov chain" (Liu 2001:111)

important: in Metropolis algorithm proposal function is necessarily symmetric

Hastings 1970 generalizes Metropolis' approach to non-symmetric cases, only restriction on proposal function is $J(x,y) > 0 iff J(y,x) > 0$(Liu 2001:111)

- retrieve information on random variable θ by sampling from its density $P(\theta)$ (Jackman 2009:201) - generate Markov Chains with given target density, i.e. posterior density $P(\theta|D)$, as Markov chain's invariant density (Jackman 2009:201) - i.e. correctly construction transition kernel of the chain (Jackman 2009:201) - sampling from posterior density combining Monte Carlo principle with Markov chain theory is called MCMC (Jackman 2009:201) - MH a core algorithm of MCMC (Jackman 2009:201) - MH defines transition rules to generate Markov chain based on posterior density (Jackman 2009:201)

- originally proposed by Metropolis et al 1953 with acceptance ratio as: $r_M = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$ (Jackman 2009:202) - modified by Hastings 1970 to r from below (Jackman 2009:202) - more acceptance ratios in Liu 2001:111f

Jackman 2009:201f - sample θ values (states in Markov chain) from posterior distribution $P(\theta|D)$ - at iteration t, we are at state θ^{t-1} - make transition to state θ^t based on following algorithm: - acceptance ratio *r*

1. sample θ^* from *proposal distribution*: $J_t(\theta^*, \theta^{t-1})$ 2. $r \leftarrow \frac{P(\theta^*|y)J_t(\theta^*,\theta^{t-1})}{P(\theta^{t-1}|y)J_t(\theta^{t-1},\theta^*)}$ 3. $\alpha \leftarrow min(r,1)$ 4. sample $U\ Ubif(0,1)$ 5. **if** $U \leq \alpha$ **then** 6. $\theta^{(t)} \leftarrow \theta^*$ 7. **else** 8. $\theta^{(t)} \leftarrow \theta(t-1)$ 9. **end if**

****

Assuming a target distribution $P(\theta)$, MH samples values for θ at step t in two stages (see e.g. Liu 2001, 106f):

Proposal Propose $\theta^{'}$, which is a random perturbation of the current value for θ^t. The proposed new state $\theta^{'}$ is generated from the *proposal function* $J(x^t, x^{'})$. Calculate the difference between current and proposed state as $r = x^{(t)} - x^{'}$.

Decision Draw a random number from a uniform distribution $U \sim Unif(0,1)$. TO COMPLETE

****

2. generate a random number; let $x^{(t+1)} = x^{'}, if U \leq \frac{p(x^{'})}{p(x^{(t)})} \equiv exp()$ see (Liu 2001:107) for second step, just type formula in paper accept-reject rule in 2

****

Since the Markov chain is irreducible, the algorithm can start at any point and proceed by iteration proposal and decision until enough samples have been drawn.

****

- main features of Metropolis' sampling method (assume sample from distribution with density p(x)): (Hastings 1970:98) a. computation based on p(x) only through ratios if form p(x')/p(x) with sample points x', x - i.e. no normalizing constant - no factorization of p(x) needed - easy implementation in computer b. acquire sample sequcency by simulating Markov chain - i.e. resulting samples correlated - i.e. estimation of SD and error of an estimate might be more problematic than with independant samples

Metropolis-Hastings Algorithm (Liu 2001:111f): - given current state$x^(t)$ - draw y from proposal distribution$J(x^(t), y)$ - draw$U\ \ uniform(0,1)$ and update: – $x^{(t+1)} = y(if U \leq r(x^(t), y))or x^t otherwise$

Metropolis and Hastings suggested using acceptance rate: $r(x,y)$ as $min(1, \frac{pr(y)*J(y,x)}{pr(x)} * J(x,y))$ (identical to original, if J is symmetric (Liu 2001:112))

Also other definitions for acceptance rate by Barker 1965 and charles stein (liu 112)

NOTE Liu 2001:118: small step-size in proposal transition leads to slow movement in Markov chain; large step-size in proposal transition leads to low acceptance rate in both cases mixing rate of algorithm low

****

2.3 Proposal Densities

1. Implement simple random walk presented by Kruschke

2. Use multivariate random walk from MCMCpack

3. Find Independence metropolis implementation:

https://www.lancaster.ac.uk/pg/jamest/Group/stats4.html

4. use really bad one from HW 3?

- if J generates small values, acceptance rate is low, i.e. inefficient exploration of parameter space (Jackman 2009:202) - if J generates high acceptance values, but only in small neighbourhood around θ^{t-1}, again inefficient exploration of posterior

Random-Walk Metropolis Kruschke2014:149

Independence Metropolis

3 Shiny Hastings App

3.1 Interface

3.2 Settings

3.3 Posterior Densities

Random-walk Metropolis Liu 2001:114f and Jackman 2009:?

Metropolized independent sampler hastings 1970, liu2001:115f

4 Conclusion

References

In:

Hastings, W. K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1, pp. 97–109.

Jackman, Simon (2009). *Bayesian Analysis for the Social Sciences*. Wiley Series in Probability and Statistics.

Kruschke, John K. (2015). *Doing Bayesian Data Analysis. A Tutorial with R, JAGS, and Stan*. 2nd ed. Amsterdam: Elsevier.

Liu, Jun S. (2001). *Monte Carlos Strategies in Scientific Computing*. 1st ed. Springer Series in Statistics. New York: Springer.

Martin, Andrew D., Kevin M. Quinn & Jong Hee Park (2011). "MCMCpack: Markov Chain Monte Carlo in R". In: *Journal of Statistical Software* 42.9, pp. 1–21.

Metropolis, Nicholas et al. (1953). "Equation of State Calculations by Fast Computing Machines". In: *Journal of Chemical Physics* 21.6, pp. 1087–1092.