A Note an Perfect Slice Sampling

Hörmann, Wolfgang; Leydold, Josef

A Note on Perfect Slice Sampling

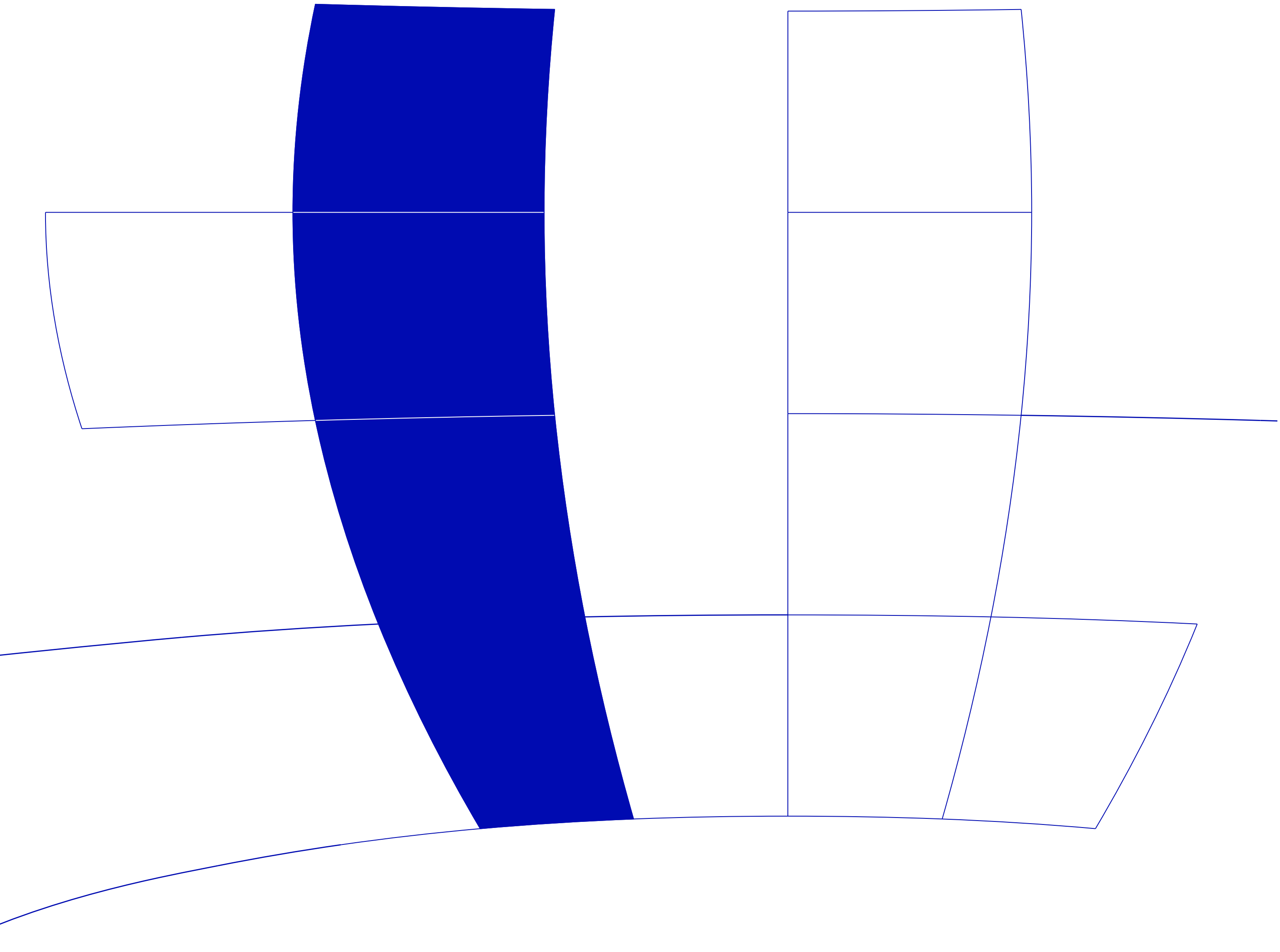
Wolfgang Hörmann, Josef Leydold

Department of Statistics and Mathematics
Wirtschaftsuniversität Wien

Research Report Series

Report 29
March 2006

http://statmath.wu-wien.ac.at/



A Note on Perfect Slice Sampling [*]

Wolfgang Hörmann (hormannw@boun.edu.tr)
Department of Statistics and Mathematics, University of Economics and Business Administration, Augasse 2–6, A-1090 Vienna, Austria, and
Department of Industrial Engineering, Bogazici University Istanbul, 80810 Bebek-Istanbul, Turkey

Josef Leydold (leydold@statistik.wu-wien.ac.at)
Department of Statistics and Mathematics, University of Economics and Business Administration, Augasse 2–6, A-1090 Vienna, Austria

March 8, 2006

Abstract. Perfect slice sampling is a method to turn Markov Chain Monte Carlo (MCMC) samplers into exact generators for independent random variates. We show that the simplest version of the perfect slice sampler suggested in the literature does not always sample from the target distribution.

Keywords: Markov chain Monte Carlo method, perfect slice sampling, coupling from the past

1. Introduction

Markov Chain Monte Carlo (MCMC) samplers are very powerful methods for drawing random samples for quite arbitrary distributions. In particular they are used in the case of simulations that invoke high dimensional integrals. However, as they produce dependent random variables (vectors) they require a convergence assessment. To overcome this problem Propp and Wilson (1996) suggested so called *perfect sampling algorithms* that allow to decide exactly, when convergence is reached. Although first developed for discrete state spaces perfect sampling also can be applied to Markov chains with state space $\mathbb{R}^d$ albeit this is not easy, see Green and Murdoch (2000), Wilson (2000), Murdoch (2000), and Murdoch and Meng (2001).

Coupling from the past (CFTP) suggested by Propp and Wilson (1996) is probably the most popular of these perfect sampling algorithms. The main building block of all CFTP algorithms is the *randomizing operation.* It is a deterministic function ϕ taking as input the state X_t of the chain $\mathbf{X}$ at time t and some intrinsic randomness U_t. The randomizing operation returns the new output state

$$X_{t+1} = \phi(X_t, U_t)\,.$$

[*] This work was supported by the Austrian Science Foundation (FWF), project no. P16767-N12.

Roughly spoken CFTP starts Markov chains from all possible points of the state space at some time $-T$ in the past. Using the randomizing operation with the same intrinsic randomness these chains develop in parallel or coalesce. If at time 0 all these have coalesced in a single state this state is returned. Otherwise the chains are restarted at some earlier time $-T' < -T$; see e.g. Wilson (2000), for a short tutorial.

Mira et al. (2001) suggest a perfect slice sampling algorithm based on CFTP. As that algorithm requires substantial computations in every iteration Casella et al. (2002) and Philippe and Robert (2003) (also included in Robert and Casella, 2004) suggest a simpler and faster variant of the original algorithm. However, it is possible to show that this variant does not always sample from the target distribution. In this short note we give a proof for this fact by deriving the distribution of the random variates generated by that version of the perfect slice sampler for a simple density.

2. A simple perfect slice sampler

Mira et al. (2001) pointed out that the slice sampler is well suited for a combination with the CFTP algorithm as it is stochastically monotone with respect to the natural partial ordering induced by f ie. the ordering $\preceq$ such that

$$f(x) \leq f(y) \quad \text{implies} \quad x \preceq y\,.$$

Hence only two chains (for the minimal and maximal starting point, resp.) have to be run in parallel to keep track of coalescence that is necessary for perfect sampling. To facilitate the notation and to concentrate on the main point we restrict our attention to the simple case of a bounded monotone decreasing density $f(x)$ on $[0, 1]$. Then we can easily formulate the probably simplest possible randomizing operation

$$\phi_0(x, V, U) = U\, f^{-1}(V\, f(x))\,,$$

where both U and V are $U(0, 1)$ uniform variates and we define $f^{-1}(y) = \sup\{x | f(x) \geq y\}$. ϕ_0 is simple and it is not difficult to see that it is stochastically monotone with respect to our ordering as for the same values of U and V, $x \preceq y$ clearly implies $\phi_0(x, V, U) \preceq \phi_0(y, V, U)$, since for decreasing f, $x \preceq y$ implies $x \geq y$. But it is also clear that for this randomizing operation $f(x) \neq f(y)$ implies $\phi_0(x, V, U) \neq \phi_0(y, V, U)$. Thus the chain of the maximal and the minimal state can never coalesce. Mira et al. (2001) have solved this problem by replacing U by a sequence $\mathbf{W} = (W_i)$ of uniform random variates. Thus their algorithm has the obvious disadvantage that it is necessary to generate and store the (possibly long) sequence of $\mathbf{W}_{t-s}$ for every time t necessary in the simulation.

Casella et al. (2002) and Philippe and Robert (2003) have suggested an easier modification that does not require a W-sequence. In the original references it is used to generate from quite general multivariate distributions. As we are here only interested in the details of the CFTP algorithm we present the modification as Algorithm 1 only for a quite simple special case: For non-increasing bounded densities f on $[0, 1]$ with known inverse density f^{-1}.

Algorithm 1 Proposed perfect slice sampler

```
Require: Non-increasing bounded densities $f$ on $[0,1]$, its inverse $f^{-1}$.
Ensure: Random variate $X$.
Set $T \leftarrow 1$.
loop
  Set $X^{(1)}_{-T} \leftarrow 0$ and $X^{(0)}_{-T} \leftarrow 1$.
  for $t = -T$ up to $-1$ do
    if $t < -T/2$ then
      Generate and store $V_t \sim U(0,1)$ and $U_t \sim U(0,1)$.
    else
      Use stored values for $V_t$ and $U_t$.
    Set $X^{(0)}_{t+1} \leftarrow \phi_0(X^{(0)}_t, V_t, U_t) = U_t\, f^{-1}(V_t\, f(X^{(0)}_t))$.
    if $f(X^{(0)}_{t+1}) \geq V_t\, f(X^{(1)}_t)$ then
      Set $X^{(1)}_{t+1} \leftarrow X^{(0)}_{t+1}$.
    else
      Set $\tilde{U}_t \leftarrow U_t$.
      /* Philippe and Robert (2003) */
      /* Or */
      Generate and store $\tilde{U}_t \sim U(0,1)$, (resp., use stored value).
      /* Casella et al. (2002), Fig. 3 */
      Set $X^{(1)}_{t+1} \leftarrow \phi_0(X^{(1)}_t, V_t, \tilde{U}_t) = \tilde{U}_t\, f^{-1}(V_t\, f(X^{(1)}_t))$.
  if $X^{(1)}_0 = X^{(0)}_0$ then /* Coalescence */
    return $X^{(0)}_0$.
  else
    Set $T \leftarrow 2\,T$.
```

It is obvious that coalescence is possible in Algorithm 1 as $X^{(1)}_{t+1} \leftarrow X^{(0)}_{t+1}$ is used for the case that $f(X^{(0)}_{t+1}) \geq V_t\, f(X^{(1)}_t)$. The sequences $\mathbf{X}^{(0)}$ and $\mathbf{X}^{(1)}$ are realizations of the slice sampler. However, when we implemented this algorithm and tested it using a χ^2 goodness-of-fit test and large sample sizes the p-value was always smaller than 0.001. After this observation we checked the reason for this fact and found the following explanation:

The sequence $\mathbf{X}^{(1)}$ depends on the values of $\mathbf{X}^{(0)}$ (see Steps 10 and 11) and is therefore not following the same randomizing operation as $\mathbf{X}^{(0)}$ itself. To be precise it is not a randomizing operation at all as it takes as input also the state of $\mathbf{X}^{(0)}$. This observation alone is no proof that the algorithm is wrong. But we can see for very simple examples that depending whether we start with $T = -1$ or $T = -2$ Algorithm 1 produces different variates which is against the fundamental idea of CFTP that coalescence can only occur if the output is fully determined by the randomness produced so far. Starting farer in the past can never change this output if the CFTP algorithm is correct. The following simple numerical example shows that Algorithm 1 is not implementing a CFTP algorithm.

Example. We consider the density $f(x) = 2 - 2x$ on (0,1). For $T = -1$ we generate and store $V_{-1} = 0.5$ and $U_{-1} = 0.1$. We obtain $X_0^{(0)} \leftarrow \phi_0(X_{-1}^{(0)}, 0.5, 0.1) = 0.1 f^{-1}(0.5\, f(1)) = 0.1$. As $f(0.1) = 1.8 > V f(X_{-1}^{(1)}) = 1$, $X_0^{(1)}$ is set to 0.1 as well. Coalescence is reached at time $t = 0$ and we are finished.

For $T = -2$ we generate and store $V_{-2} = 0.3$ and $U_{-2} = 0.2$. Similar to above we obtain $X_{-1}^{(0)} = X_{-1}^{(1)} = 0.2$. For the last step with $t = -1$ we have to use $V_{-1} = 0.5$ and $U_{-1} = 0.1$ as above. We then get $X_0^{(0)} = X_0^{(1)} = 0.06$. So starting at $T = -2$ leads to a different result than starting with $T = -1$. This fact is not at all influenced by our choice of the variant in Step 13 of Algorithm 1.

For simple densities it is possible to derive the density of the random variates generated by Algorithm 1. ($I_{(a,b)}(x)$ denotes the indicator function over the interval (a, b).)

THEOREM 1. *Consider Algorithm 1 where we use $\tilde{U}_t \leftarrow U_t$ in Step 13 (*Philippe and Robert, 2003*). When we use a target distribution with density*

$$f(x) = \tfrac{3}{2}\, I_{(0,1/2)}(x) + \tfrac{1}{2}\, I_{(1/2,1)}(x)\ ,$$

then this algorithm returns independent random variates with density

$$\tilde{f}(x) = I_{(0,1/4)}(x) + 2\, I_{(1/4,1/2)}(x) + \tfrac{1}{2}\, I_{(1/2,1)}(x)\ .$$

Proof. For this simple density $f^{-1}(y)$ has only two possible values, 1/2 and 1, and thus $\phi_0(X_t, V_t, U_t) = U_t\, f^{-1}(V_t\, f(X_t))$ is either U_t or

$U_t/2$ in Steps 9 and 14, respectively. As a consequence there are at most two different states for X_{t+1} which are computed as U_t and $U_t/2$, respectively.

We can distinguish three cases for the pair (U_t, V_t):

Case A: For $U_t \leq 1/2$ coalescence occurs, $X^{(0)}_{t+1} = X^{(1)}_{t+1}$, the common value $X_{t+1} = U_t$ and $f(X_{t+1}) = 3/2$.

Case B: For $U_t > 1/2$ and $V_t \leq 1/3$ coalescence occurs, $X^{(0)}_{t+1} = X^{(1)}_{t+1}$, the common value $X_{t+1} = U_t$ and $f(X_{t+1}) = 1/2$.

Case C: For $U_t > 1/2$ and $V_t > 1/3$ no coalescence occurs and $f(X_{t+1})$ is always equal to $f(X_t)$.

Thus if the pair (U_{-1}, V_{-1}) belongs to cases A or B coalescence occurs immediately and $X_0 = U_{-1}$ is returned by the algorithm. Otherwise we continue to increase T (in Step 18) and restart the CFTP algorithm until the first time coalescence was found in the chain at some time $t^* < -1$. Then (U_{t^*}, V_{t^*}) either belongs to case A with conditional probability 3/4 and $f(X^{(0)}_{t^*+1}) = f(X^{(1)}_{t^*+1}) = 3/2$, or it belongs to case B with conditional probability 1/4 and $f(X^{(0)}_{t^*+1}) = f(X^{(1)}_{t^*+1}) = 1/2$. The transition probabilities between the two possible states in Step 9 are $\mathrm{P}(f(X_{t+1}) = 1/2 | f(X_t) = 3/2) = 1/6$ and $\mathrm{P}(f(X_{t+1}) = 3/2 | f(X_t) = 1/2) = 1/2$. Thus we have the transition matrix $\begin{pmatrix} 5/6 & 1/2 \\ 1/6 & 1/2 \end{pmatrix}$ which results in a stationary distribution with probabilities $(3/4, 1/4)$, i.e., the same distribution as right after coalescence at time t^*+1. Consequently, if the pair (U_{-1}, V_{-1}) belongs to case C (and thus $U_{-1} > 1/2$) the algorithm returns $U_{-1}/2$ with probability 3/4 and U_{-1} with probability 1/4.

Collecting the above facts we can see that the distribution of the output of the algorithm is a mixture of uniform distributions consisting of

- $\mathcal{U}(0, 1/2)$ with probability 1/2 (case $U_1 \leq 1/2$),
- $\mathcal{U}(1/2, 1)$ with probability 1/6 (case $U_1 > 1/2$ and $V_1 \leq 1/3$),
- $\mathcal{U}(1/4, 1/2)$ with probability 1/4,
- $\mathcal{U}(1/2, 1)$ with probability 1/12.

This uniform mixture distribution clearly has density $\tilde{f}(x)$ which completes the proof. □

It is also not difficult to see that for the simple density f given using the version of Casella et al. (2002) in Step 13 leads to the same output distribution.

REMARK 1. *It is important to note that Theorem 1 only holds when we start with $T = -1$ in Algorithm 1. When we start at $T \leq -2$ the resulting distribution is closer to the target distribution with density f. In particular, if $T \leq -8$ then the deviation from the correct distribution cannot be detected by a χ^2 goodness-of-fit test with sample size 10^6. This is due to the fact that f is the stationary distribution of the randomizing operation in Step 9. Nevertheless, the algorithm is still not sampling exactly from the correct distribution.*

References

Casella, G., K. L. Mengersen, C. P. Robert, and D. M. Titterington: 2002, 'Perfect slice samplers for mixture distributions'. *Journal of the Royal Statistical Society B* **64**, 777–790.

Green, P. J. and D. J. Murdoch: 2000, 'Exact sampling for Bayesian inference: towards general purpose algorithms (with discussion)'. In: J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds.): *Monte Carlo Methods*, Vol. 6 of *Bayesian Statistics*. Oxford, pp. 301–321, Oxford University Press.

Mira, A., J. Møller, and G. O. Roberts: 2001, 'Perfect slice samplers'. *Journal of the Royal Statistical Society B* **63**, 593 – 606.

Murdoch, D. J.: 2000, 'Exact sampling for Bayesian inference: unbounded state space'. In: N. Madras (ed.): *Monte Carlo Methods*, Vol. 26 of *Fields Institute Communications*. pp. 111–121, Americal Mathematical Society.

Murdoch, D. J. and X.-L. Meng: 2001, 'Towards perfect sampling for Bayesian mixture priors'. In: *ISBA 2000, Proceedings*. pp. 381–390, ISBA and Eurostat.

Philippe, A. and C. P. Robert: 2003, 'Perfect simulation of positive Gaussian distributions'. *Statistics and Computing* **13**, 179–186.

Propp, J. G. and D. B. Wilson: 1996, 'Exact sampling with coupled Markov chains and applications to statistical mechanics'. *Random Structures and Algorithms* **9**, 223–252.

Robert, C. P. and G. Casella: 2004, *Monte Carlo Statistical Methods*. Springer Verlag, 2 edition.

Wilson, D. B.: 2000, 'Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP)'. In: N. Madras (ed.): *Monte Carlo Methods*, Vol. 26 of *Fields Institute Communications*. pp. 141–176, American Mathematical Society.