Roman Karawatzki and Josef Leydold

Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions

Working Paper

Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions

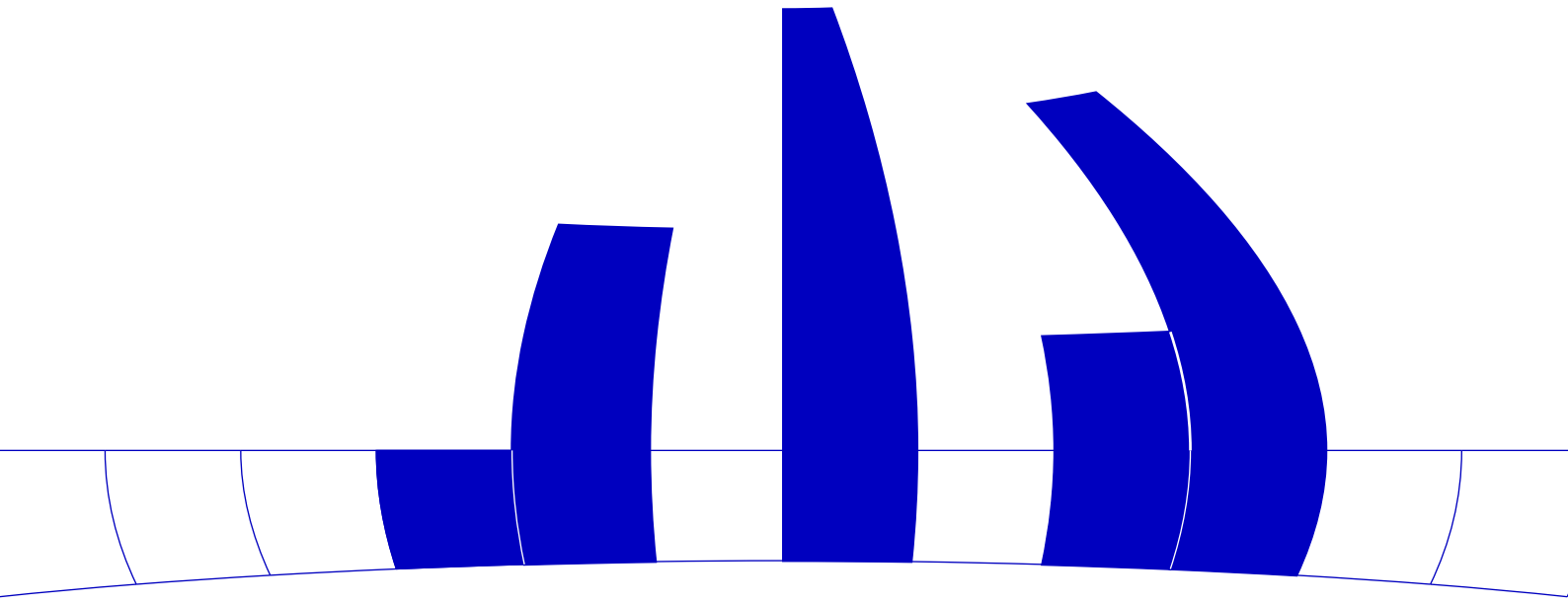
Roman Karawatzki, Josef Leydold

Department of Applied Statistics and Data Processing
Wirtschaftsuniversität Wien

Preprint Series

Preprint 52
January 2005

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



Automatic Markov chain Monte Carlo Procedures for Sampling from Multivariate Distributions

Roman Karawatzki and Josef Leydold *

University of Economics and Business Administration, Department for Applied Statistics and Data Processing, Augasse 2-6, A-1090 Vienna, Austria

Abstract

Generating samples from multivariate distributions efficiently is an important task in Monte Carlo integration and many other stochastic simulation problems. Markov chain Monte Carlo has been shown to be very efficient compared to "conventional methods", especially when many dimensions are involved. In this article we propose a *Hit-and-Run* sampler in combination with the *Ratio-of-Uniforms* method. We show that it is well suited for an algorithm to generate points from quite arbitrary distributions, which include all log-concave distributions. The algorithm works automatically in the sense that only the mode (or an approximation of it) and an oracle is required, i.e., a subroutine that returns the value of the density function at any point x. We show that the number of evaluations of the density increases slowly with dimension.

Subject classification: Simulation: multivariate random variate generation (Markov chain Monte Carlo, hit-and-run sampling, ratio-of-uniforms, log-concave distributions)

1 Introduction

Sampling random vectors is an important part of many stochastic simulation and randomized algorithms. When many dimensions are involved this becomes a very challenging task. Conventional methods that have been developed for the case of univariate random numbers like rejection or composition do not

* Corresponding author. Tel +43 1 313 36–4695. FAX +43 1 313 36–738
Email address: Josef.Leydold@statistik.wu-wien.ac.at (Josef Leydold).

work efficiently for distributions with moderately many (8 or more) dimensions (see e.g. Hörmann, Leydold, and Derflinger, 2004). Markov chain Monte Carlo methods have proven to be more efficient for higher dimensions. The algorithms generate correlated sequences of random points that converge to the target distribution.

In this article we propose the *Hit-and-Run* sampler in combination with the *Ratio-of-Uniforms* method. It is well suited for an algorithm to generate points from quite arbitrary distributions, which include all log-concave distributions. This algorithm works automatically in the sense that only the mode (or an approximation of it) and an oracle is required, i.e., a subroutine that returns the value of the density function at any point $\mathbf{x}$. We use the number of calls to the oracle as a measure of the performance of the algorithm. Some theoretical results and our computational experiences show that this method is fast and its performance decreases only slowly with increasing dimension such that samples from distributions with 100 or more variables can be generated. One should notice, however, that the costs for evaluating the density itself depends on the dimension, e.g. it increases quadratically for a multinormal distribution.

The article is organized as follows: First we shortly describe the Hit-and-Run sampler (Sect. 2) and state the main facts about the Ratio-of-uniforms method (Sect. 3). In Section 4 we propose new algorithms based on these two principles. Our computational experiences are summarized in Sect. 5.

2 The Hit-and-Run Sampler

For the problem of sampling random points uniformly distributed in some fixed but arbitrary bounded open set $S \in \mathbb{R}^n$ Smith (1984) introduced the so called *Hit-and-Run* samplers that are based on the following principle.

0. Choose a starting point $\mathbf{X}_0 \in S$ and set $k = 0$.
1. Generate a random direction $\mathbf{d}_k$ with distribution ν.
2. Generate λ_k uniformly distributed in $\Lambda_k = S \cap \{\mathbf{x} \colon \mathbf{x} = \mathbf{x}_k + \lambda \mathbf{d}_k\}$.
3. Set $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{d}_k$ and $k = k + 1$.
4. Goto 1.

Boneh and Golan (1979) and independently Smith (1980) first considered such an algorithm with ν being the uniform distribution over a hypersphere. Another variant is to use random coordinate directions which can be seen as Gibbs sampling with randomized directions. It has been shown that this Markov chain is mixing fast (Chen and Schmeiser, 1993; Smith, 1984), that is, the distribution of the generated point set converges to the uniform distribution with increasing sample size; in particular when the set S is convex (Lovász,

1999; Lovász and Vempala, 2004). Kaufman and Smith (1998) improved the convergence of the Hit-and-Run sampler by non-uniform direction choice and give a (worst case) optimal distribution ν. We restate the convergence result by Smith (1984).

Theorem 1 (Smith, 1984) *Let $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \ldots$ be the Markov chain generated by the Hit-and-Run Algorithm over an open bounded region $S \in \mathbb{R}^n$. Then for any measureable set $A \subseteq S$,*

$$|P(\mathbf{X}_m \in A | \mathbf{X_0} = \mathbf{x}) - \mu(A)| < \left(1 - (\gamma / n 2^{n-1})\right)^{m-1}$$

where $\mu(\cdot)$ denotes the n-dimensional content of A and γ is the ratio of the n-dimensional content of S to the n-dimensional content of the smallest sphere containing s.

The Hit-and-Run sampler can easily be generalized to sample from non-uniform distributions with density f (Bélisle, Boneh, and Caron, 1998; Belisle, Romeijn, and Smith, 1993; Chen and Schmeiser, 1996; Smith, 1996): Replace the uniform distribution of λ_k in Step 2 of the above algorithm by the conditional distribution of f:

2'. Choose $\lambda_k \in \Lambda_k = S \cap \{\mathbf{x} \colon \mathbf{x} = \mathbf{x}_k + \lambda \mathbf{d}_k\}$ from the distribution with density

$$f_k(\lambda) = \frac{f(\mathbf{x}_k + \lambda \mathbf{d}_k)}{\int_{\Lambda_k} f(\mathbf{x}_k + \theta \mathbf{d}_k)\, d\theta}\,, \qquad \lambda \in \Lambda_k\,. \tag{1}$$

Although it has been shown that this algorithm has fast mixing time as well (Lovász and Vempala, 2003), there are some drawbacks. Besides the existence of distributions where it converges arbitrarily slow (Bélisle, 2000), the necessity of sampling from the conditional density (1) makes it difficult to apply for user-defined distributions. Notice that only one random variate has to be drawn from a particular conditional density. In the last decade automatic algorithms for sampling from large classes of univariate distributions have been developed (Gilks and Wild, 1992; Hörmann et al., 2004). However, these require some setup which can be quite expensive both in time and memory compared to the marginal generation time and thus they are often extremely slow when only one random variate should be generated. Moreover, many of them require the knowledge of some parameters of the (univariate) distributions (e.g. its mode).

3 The Ratio-of-Uniforms Method

The Ratio-of-Uniforms method has been introduced by Kinderman and Monahan (1977) and generalized to the multivariate case by Wakefield, Gelfand, and Smith (1991). It is based on the following theorem.

Theorem 2 (Wakefield, Gelfand, and Smith, 1991) *Let $f(\mathbf{x})$ be a positive integrable function on $\mathbb{R}^n$. Let $r > 0$ and $\mathbf{m} \in \mathbb{R}^n$ be constants. Suppose the point $(\mathbf{U}, V) \in \mathbb{R}^{n+1}$ with $\mathbf{U} = (U_1, \ldots, U_n)$ is uniformly distributed over the region*

$$\mathcal{A}(f) = \mathcal{A}_{r,\mathbf{m}}(f) = \left\{ (\mathbf{u}, v) \colon 0 < v < \sqrt[rn+1]{f(\mathbf{u}/v^r + \mathbf{m})} \right\}, \tag{2}$$

then $\mathbf{X} = \mathbf{U}/V^r + \mathbf{m}$ has probability density function $f(\mathbf{x}) / \int_{\mathbb{R}^n} f(\mathbf{z})\, d\mathbf{z}$.

The proof of this theorem is based on the fact that the map

$$(\mathbf{u}, v) \mapsto (\mathbf{x}, y) = \left(\frac{\mathbf{u}}{v^r} + \mathbf{m}, v^{rn+1} \right) \tag{3}$$

has constant Jacobian (equal to $rn + 1$). We can apply this theorem and get the following algorithm. Notice that f need not be normalized, i.e., it can be any positive multiple of a density function.

1. Generate a point $(\mathbf{U}, V)$ uniformly in $\mathcal{A}_{r,\mathbf{m}}(f)$.
2. Return $\mathbf{X} = \mathbf{U}/V^r + \mathbf{m}$.

An important observation is that the region $\mathcal{A}(f)$ is bounded for many distributions (at least for sufficiently large values of r). Thus the originally proposed algorithm uses rejection from the minimal bounding rectangle $\mathcal{R}_{r,\mathbf{m}}(f)$ which is given by

$$\begin{aligned} v^+ &= \sup_x \left(f(\mathbf{x}) \right)^{1/(rn+1)} , \\ u_i^- &= \inf_{x_i} (x_i - m_i) \left(f(\mathbf{x}) \right)^{r/(rn+1)} , \\ u_i^+ &= \sup_{x_i} (x_i - m_i) \left(f(\mathbf{x}) \right)^{r/(rn+1)} . \end{aligned} \tag{4}$$

Usually $\mathbf{m}$ is set to the mode of the density f as this results in an (almost) optimal rejection constant. However, the acceptance rate decreases exponentially with the dimension in this simple rejection algorithm (see e.g. Hörmann et al., 2004) and hence is impractical for dimensions larger than 10. For example, when $\mathcal{A}(f)$ is a ball then the expected number of points that must be generated within $\mathcal{R}(f)$ to obtain one within $\mathcal{A}(f)$ grows from 1.27 for dimension $n = 1$ to 400 for $n = 10$, 4×10^7 for $n = 20$, and 6×10^{27} for $n = 50$. But we can use the Hit-and-Run sampler to generate a sequence of uniformly distributed points $(\mathbf{u}, v)$ in $\mathcal{A}(f)$ in Step 1 instead of rejection from $\mathcal{R}(f)$ even in high dimensions. As $\mathcal{A}(f)$ is bounded we can expect that the uniform Hit-and-Run sampler is mixing faster that the corresponding Hit-and-Run sampler on the region

$$\mathcal{G}(f) = \{ (\mathbf{x}, y) \colon 0 < y < f(\mathbf{x}) \} . \tag{5}$$

Moreover, this set need not be bounded and thus convergence is not assured.

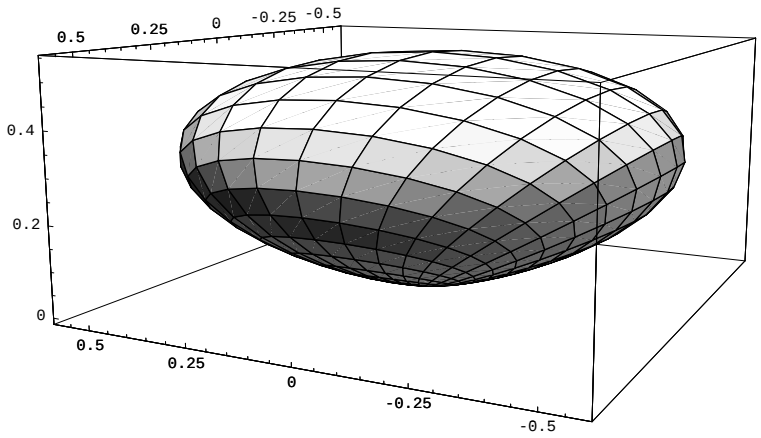


Fig. 1. Region $\mathcal{A}(f)$ for standard bivariate normal distribution ($r = 1$).

The parameter r can be used to control the shape of $\mathcal{A}(f)$. For increasing values of r the family of distributions for which this set is bounded is growing. For the special case $r = 1$ the region $\mathcal{A}(f)$ is convex in many cases and thus the Hit-and-Run sampler converges fast. The following theorem generalizes a result for the univariate case (Leydold, 2000).

Theorem 3 *For a density f and $r = 1$ the region $\mathcal{A}(f) \subset \mathbb{R}^{n+1}$ is convex if and only if the transformed density $T(f(\mathbf{x})) = -(f(\mathbf{x}))^{-1/(n+1)}$ is concave.*

Following Hörmann (1995) we call such a density T_c*-concave* with $c = -1/(n+1)$.

Proof. Since $T(y) = -1/\sqrt[n+1]{y}$ is strictly monotonically increasing, the transformation $(\mathbf{x}, y) \mapsto (\mathbf{X}, T(y))$ maps $\mathcal{G}(f)$ one-to-one onto $\mathcal{T}(f) = \{(\mathbf{x}, y)\colon y < T(f(\mathbf{x}))\}$, i.e. the region below the transformed density. Hence by $T(v^{n+1}) = -1/v$ and transformation (3)

$$\mathbb{R}^n \times (0, \infty) \to \mathbb{R}^n \times (-\infty, 0), \quad (\mathbf{u}, v) \mapsto (\mathbf{x}, y) = (\mathbf{u}/v + \mathbf{m}, -1/v) \qquad (6)$$

maps $\mathcal{A}(f)$ one-to-one onto $\mathcal{T}(f)$. Notice that f is T-concave if and only if $\mathcal{T}(f)$ is convex. Thus it remains to show that $\mathcal{A}(f)$ is convex if and only if $\mathcal{T}(f)$ is convex, and consequently if and only if hyperplanes remain hyperplanes under transformation (6). Now let $\mathbf{a}'\,\mathbf{x} + b\,y = d$ be a hyperplane in $\mathcal{T}(f)$. Then $\mathbf{a}'\,(\mathbf{u}/v + \mathbf{m}) - b/v = d$ or, equivalently, $\mathbf{a}'\,\mathbf{u} - d\,v = b - \mathbf{a}'\,\mathbf{m}$, is a hyperplane in $\mathcal{A}(f)$. Analogously we find for a hyperplane $\mathbf{a}'\,\mathbf{u} + b\,v = d$ in $\mathcal{A}(f)$ the hyperplane $\mathbf{a}'\,\mathbf{x} + d\,y = -b + \mathbf{a}'\,\mathbf{m}$ in $\mathcal{T}(f)$. □

The following result by Hörmann (1995) immediately holds for multivariate distributions.

Theorem 4 (Hörmann, 1995) *If a density f is T_c-concave for some $c \in \mathbb{R}$ then f is T_{c_1}-concave for all $c_1 \le c$.*

The family of transformations T_c contains the special case $T_0(y) = \log(y)$. Thus we have the important corollary; see Fig. 1 for an example.

Corollary 5 *For every log-concave density f and $r = 1$ the region $\mathcal{A}(f)$ is convex.*

We have to note that an analogous (simple) condition for the convexity of $\mathcal{A}_r(f)$ for $r \neq 1$ is not known.

4 The Algorithms

Although we have presented all ingredients of the Hit-and-Run algorithms for sampling from non-uniform multivariate distributions we have to make some considerations about computational details.

4.1 Uniform Sampling

The demanding part of the Hit-and-Run sampler is to generate $\mathbf{X}_{k+1}$ uniformly in the line subset $\Lambda_k = \mathcal{A}(f) \cap \{(\mathbf{u}, v)\colon (\mathbf{u}, v) = (\mathbf{U}_k, V_k) + \lambda \mathbf{d}_k\}$ for a chosen direction $\mathbf{d}_k$. It can be accomplished by the following procedure (Smith, 1996): First the intersection $L_k = \mathcal{R}(f) \cap \{(\mathbf{u}, v)\colon (\mathbf{u}, v) = (\mathbf{U}_k, V_k) + \lambda \mathbf{d}_k\}$ is determined. Then a rejection method is employed by generating one-dimensional uniform points on the line segment L_k until one falls within Λ_k. Notice that L_k can be given by the corresponding values λ_0 and λ_1 of its endpoints. Since this subproblem is one-dimensional the rejection method is typically very efficient for this task.

The expected number of calls to the oracle, i.e., the expected number of iterations is given by the ratio ρ of the length of the line segment L_k, $\mu(L_k)$, and the length of the line segment Λ_k (or the sum of all segments constituting λ_k), $\mu(\Lambda_k)$, see Hörmann et al. (2004, §2.2):

$$\rho = \frac{\mu(L_k)}{\mu(\Lambda_k)}\,. \tag{7}$$

4.2 Adaptive Uniform Sampling

Although the acceptance rate in a univariate rejection step is much better than those for rejection from the bounding rectangle it can still be improved by the following procedure when Λ_k is an open interval. Assume that we have

to sample uniformly from $(0, \rho)$, $\rho \in (0, 1)$. We generate a uniform random number U_1 in $(0, u)$ with $u = 1$. If $U_1 \in (0, u)$ we accept $U = U_1$, otherwise we set $u = U_1$ (and thus shrink the interval $(0, u)$) and generate a new uniform random number U_2, independent from U_1, and try again. We proceed until an U_k is generated with $U_k \in (0, U_{k-1})$. (A similar procedure has been suggested by Neal (2003) for the slice sampler.) The expected number of interations for sampling U is much reduced by this adaptive algorithm. It can be estimated by means of the following equivalent procedure Generate i.i.d. uniform random numbers $U_1, U_2, \ldots \in (0, 1)$ until the first time $U_1 \cdot U_2 \cdot \ldots \cdot U_T < \rho$ and return $U = U_1 \cdot U_2 \cdot \ldots \cdot U_T$.

Lemma 6 *The stopping time $T-1$ of the above sampling procedure is Poisson distributed with parameter $-\log(\rho)$. Thus for the expectation of T we find $\mathbf{E}(T) = 1 - \log(\rho)$.*

Proof. The probability for $T = t$ is given by $\mathrm{P}(T = t) = \mathrm{P}(U_1 \cdot \ldots \cdot U_{t-1} \geq \rho \wedge U_1 \cdot \ldots \cdot U_t < \rho) = \mathrm{P}(Z_{t-1} \leq b \wedge Z_{t-1} + X_t > b)$ where $X_t = -\log(U_t)$ is standard exponential distributed, $Z_{t-1} = -(\log(U_1) + \cdots + \log(U_{t-1}))$ is $\Gamma(t-1, 1)$ distributed and $b = -\log(\rho)$. Hence using the density of the gamma distribution we obtain $\mathrm{P}(T = t) = \int_0^b \mathrm{P}(X_t > b - Z_{t-1} | Z_{t-1} = z)\, z^{t-2} e^{-z} / \Gamma(t-1)\, dz = \int_0^b e^{-(b-z)}\, z^{t-2} e^{-z} / \Gamma(t-1)\, dz = e^b z^{t-1} / (t-1)!$, i.e. $T - 1$ is Poisson distributed with parameter b as proposed. □

This idea can be used to sample from Λ_k more efficiently. For densities where $\mathcal{A}(f)$ is convex, e.g., for log-concave and $T_{-1/(n+1)}$-concave densities when $r = 1$, the set L_k is an open interval. This interval can be described as (λ_0, λ_1). In each iteration step we choose a $\lambda_k \in (\lambda_0, \lambda_1)$ at random. If $\mathbf{X}_{k+1} = (\mathbf{U}_k, V_k) + \lambda_k \mathbf{d}_k \in \Lambda_k$ we accept $\mathbf{X}_{k+1}$. Otherwise we shrink (λ_0, λ_1) and try again. This is done by replacing λ_0 or λ_1 by the generated λ_k depending on the sign of λ_k. Notice that (λ_0, λ_1) always contains 0. By Lemma 6 the expected number of calls to the oracle for generating $\mathbf{X}_{k+1}$ is then given by $1 - \log(\rho)$ where ρ is the ratio of the length of the two line segments L_k and Λ_k.

Computational experience shows that computing the bounding rectangle $\mathcal{R}(f)$ numerically is the most time consuming step in higher dimensions. A possible solution is to replace the bounding rectangle $\mathcal{R}(f)$ by the unbounded "plate" given by $\{(\mathbf{u}, v) \colon 0 < v < \sqrt[n+1]{f(\mathbf{m})}\}$. Notice that the line L_k becomes infinitely long when and only when the v-coordinate of the random direction is 0. Thus for a direction drawn uniformly from the hypersphere (or any other absolutely continuous distribution) L_k has finite length almost surely. However, it can become very long and thus we have to use the adaptive sampling discussed above. Otherwise the number of iterations becomes prohibitively large.

To get a first impression about the expected number of iterations $\mathrm{E}(I)$ when we use the unbounded "plate" we assume that $r = 1$, $f(\mathbf{m}) = 1$, and $\mathcal{A}(f)$ is a

ball of radius 1. The length of a line segment L_K is then given by $\sqrt{1+\tan(\theta)^2}$ where θ is the angle between the direction $\mathbf{d}$ and the v-axis. For a point $\mathbf{X}_k$ and a direction $\mathbf{d}$ with angle θ the expected number of iterations is given by $\log(\sqrt{1+\tan(\theta)^2}/\ell)$ by Lemma 6 where ℓ denotes the length of Λ_k. When the direction $\mathbf{d}$ is uniformly distributed on the hypersphere, θ follows a distribution with density $S_{n-1}/S_n \sin(\theta)^{n-2}$ for $0 \leq \theta \leq \pi$ where $S_n = 2\pi^{n/2}/\Gamma(n/2)$ denotes the area of the surface of the unit n-sphere (i.e. in $\mathbb{R}^n$) and $\Gamma(\cdot)$ denotes the gamma function. Now assume that $\mathbf{X}_k$ is uniformly distributed in the ball $\mathcal{A}(f)$. Then for a given θ the length ℓ of Λ_k depends on the distance δ between $\mathbf{X}_k$ and $\mathbf{d}$, given by $\ell = 2\sqrt{1-\delta^2}$. The density of δ is given by $(n\,S_{n-1})/((n-1)S_n)(1-\delta^2)^{(n-1)/2}$ for $-1 < \delta < 1$. Consequently by a straight forward computation we obtain for the expected number of iterations,

$$
\begin{aligned}
\mathrm{E}(I) &= \int_0^\pi \int_{-1}^1 \log\left(\sqrt{1+\tan(\theta)^2}/2\sqrt{1-\delta^2}\right) \cdot \\
&\qquad \frac{n\,S_{n-1}}{(n-1)S_n}(1-\delta^2)^{\frac{n-1}{2}}\ \frac{S_{n-1}}{S_n}\sin(\theta)^{n-2}\,d\delta\,d\theta \\
&= \frac{n\,S_{n-1}^2}{2(n-1)S_n^2}\int_0^\pi \int_{-1}^1 (\log(1+\tan(\theta)^2) - \log(1-\delta^2)) \cdot \\
&\qquad (1-\delta^2)^{\frac{n-1}{2}}\ \sin(\theta)^{n-2}\,d\delta\,d\theta\ - \log(2) \\
&< \frac{S_{n-1}}{S_n}\left(\int_0^\pi \log(1+\tan(\theta)^2)\sin(\theta)^{n-2}\,d\theta - \int_{-1}^1 \log(1-\delta^2)(1-\delta^2)^{\frac{n-1}{2}}\,d\delta\right) \\
&< \frac{S_{n-1}}{S_n}\left(\int_0^\pi \log(1+\tan(\theta)^2)\,d\theta + \int_{-1}^1 (1-\delta^2)^{\frac{n-3}{2}}\,d\delta\right) \\
&= 1 + \pi\log(4)\,\frac{S_{n-1}}{S_n} = 1 + \sqrt{\pi}\log(4)\,\frac{\Gamma(n/2)}{\Gamma((n-1)/2)}
\end{aligned}
$$

i.e., the expected number of iterations is finite for every dimension n.

4.3 Direction Sampling

Hypersphere sampling, i.e. choosing direction $\mathbf{d}_k$ uniformly distributed on the sphere, seems to be a good choice as it is simple and easy to implement (see e.g. Hörmann et al., 2004, §11.2.1). The improved method by Kaufman and Smith (1998) is another possible algorithm. We made the experience that the Hit-and-Run sampler mixes faster when the bounding rectangle $\mathcal{R}$ is a hypercube. By a linear map we can transform every bounded rectangle $\mathcal{R}$ into the unit hypercube $(0,1)^n$. Thus we can generate points $(\mathbf{U}_k, V_k)$ in $(0,1)^n$ and transform it back into the original scale.

4.4 Floating Point Arithmetic

When stating the theoretical background of our method we have assumed that we have real numbers, $\mathbb{R}$. However, the algorithms are designed to be implemented in real world computers which work with floating point numbers that have only a limited precision; see Overton (2001) for an introduction to floating point arithmetic. Thus expressions like $\sqrt[n+1]{f}$ or v^{n+1} may result in a reduction of significant digits, overflow, or underflow when the number n of dimensions is large. As a consequence the generated distribution deviates too much from the target distribution or the algorithm even may fail too work. Therefore we need two strategies to reduce these problems:

(1) The density f is rescaled such that $f(\mathbf{m}) = 1$ for the mode $\mathbf{m}$.
(2) The oracle should return $\log(f(\mathbf{x}))$.

The latter point is quite convenient as the logarithms of many densities (or of multiples of densities) are often easier to compute.

4.5 HITRO

We have compiled two algorithms [1] (HITRO-I and HITRO-II). The first one uses bounding hyper-rectangle for the rejection steps, the second one uses the unbounded "plate".

There are a few remarks concerning these Algorithms:

- The set $\mathcal{A}(f)$ should be convex. This is, e.g., the case for $r = 1$ when the density f is log-concave. Otherwise the conditions for the convergence theorems are not satisfied, i.e., whenever Λ_k is disconnected and Step 11 in Algorithm HITRO-I is used then
  1. points sampled from Λ_k are not uniformly distributed, and
  2. the Markov chain is not time reversible, i.e., the transition probability function is not symmetric.

  Thus when $\mathcal{A}(f)$ is not convex, then the adaptive Step 11 in Algorithm HITRO-I should be skipped. (Algorithm HITRO-II becomes very slow without Step 11.) Nevertheless, in our computational experiments it seemed that the Markov chain still converged to the target distribution with adaptive uniform sampling even if $\mathcal{A}(f)$ is not convex.
- We can replace the mode $\mathbf{m}$ by any other vector $\mathbf{c}$. This is in particular useful when f is multimodal. Then $\mathbf{c}$ could represent the "center" of the distribution. Notice, however, that f_m should be set to a value close to

[1] *Hitro* is the Slovenian word for *fast*.

Algorithm 1 HITRO-I

Input: Density function f in $\mathbb{R}^n$, mode $\mathbf{m}$; parameter r, sample size N.
Output: Sequence $\mathbf{X}_k$ of random vectors with asymptotic distribution f.

[Setup]
Compute $f_m \leftarrow f(\mathbf{m})$ and bounding rectangle $\mathcal{R}_{r,\mathbf{m}}(f/f_m) = (\mathbf{R}_l, \mathbf{R}_u)$.
Set $(\mathbf{U}_0, V_0) \leftarrow (\mathbf{0}, \frac{1}{2})$ and $k \leftarrow 0$.
[Generate chain]
repeat
  Generate a random direction $\mathbf{d}_k$ uniformly on $(n+1)$-sphere.
  [Generate a point uniformly in Λ_k]
  Compute λ_0 and λ_1. (End points of line segment $L_k = (\mathbf{R}_l, \mathbf{R}_u) \cap \{(\mathbf{u}, v) \colon (\mathbf{u}, v) = (\mathbf{U}_k, V_k) + \lambda \mathbf{d}_k\}$)
  loop
    Generate λ_k uniformly distributed in (λ_0, λ_1).
    Set $(\mathbf{U}_{k+1}, V_{k+1}) \leftarrow (\mathbf{U}_k, V_k) + \lambda_k \mathbf{d}_k$.
    Set $\mathbf{X}_{k+1} \leftarrow \mathbf{U}_{k+1}/(V_{k+1})^r + \mathbf{m}$.
    if $(rn+1)\log(V_{k+1}) \geq \log(f(\mathbf{X}_{k+1})/f_m)$ **then**
      Set $\lambda_0 \leftarrow \lambda_k$ (if $\lambda_k < 0$) or $\lambda_1 \leftarrow \lambda_k$ (otherwise). [Shrink L_k]
    else
      Stop loop.
  [Append to chain]
  Set $k \leftarrow k+1$.
until $k = N$.

Algorithm 2 HITRO-II

Input: Density function f in $\mathbb{R}^n$, mode $\mathbf{m}$; parameter r, sample size N.
Output: Sequence $\mathbf{X}_k$ of random vectors with asymptotic distribution f.

[Setup]
Compute $f_m \leftarrow f(\mathbf{m})$.
Set $(\mathbf{U}_0, V_0) \leftarrow (\mathbf{0}, \frac{1}{2})$ and $k \leftarrow 0$.
[Generate chain]
repeat
  Generate a random direction $\mathbf{d}_k = (\mathbf{d}_u, d_v)$ uniformly on $(n+1)$-sphere.
  Set $\lambda_0 \leftarrow -|v/d_v|$ and $\lambda_1 \leftarrow |(1-v)/d_v|$.
  loop
    Generate λ_k uniformly distributed in (λ_0, λ_1).
    Set $(\mathbf{U}_{k+1}, V_{k+1}) \leftarrow (\mathbf{U}_k, V_k) + \lambda_k \mathbf{d}_k$.
    Set $\mathbf{X}_{k+1} \leftarrow \mathbf{U}_{k+1}/(V_{k+1})^r + \mathbf{m}$.
    if $(rn+1)\log(V_{k+1}) \geq \log(f(\mathbf{X}_{k+1})/f_m)$ **then**
      Set $\lambda_0 \leftarrow \lambda_k$ (if $\lambda_k < 0$) or $\lambda_1 \leftarrow \lambda_k$ (otherwise).
    else
      Stop loop.
  Set $k \leftarrow k+1$.
until $k = N$.

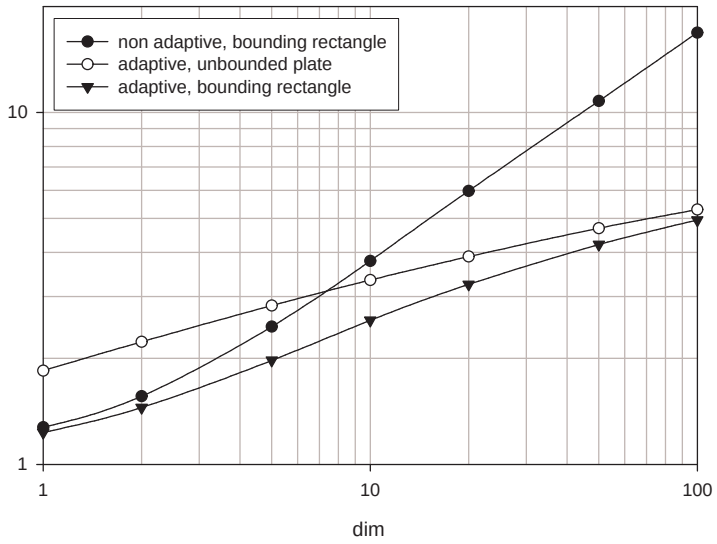


Fig. 2. Average number of oracle-calls (pdf-calls) for standard multivariate normal distribution.

$\max f(\mathbf{x})$ (for computational reasons) in Step 1. Furthermore, the starting point $(\mathbf{U}_0, V_0)$ must be set accordingly.

- $\mathcal{R}(f)$ need not necessarily be the minimal bounding rectangle as defined in (4). It can be larger.

5 Computational Experiences

We ran many experiments with the multinormal distribution and many dimensions and variance-covariance matrices. Figure 2 gives the average number of oracle calls needed for one step of the Hit-and-Run sampler for the standard multinormal distributions are used. The results for correlated components are quite similar. It shows that the performance gain by adaptive uniform sampling is very large in higher dimensions. Moreover, the ratio between the average number of calls needed when using an unbounded plate instead of a bounding box is less than 2 and decreases with increasing dimension. On the other hand using an unbounded plate only requires to compute the mode of the density (or the approximate location of the mode and an upper bound for the density) and thus saves finding all $2\,n + 1$ coordinates for the bounding rectangle. The expected number of oracle-calls stays well below 10 even for dimensions as high as 100. For "conventional" rejection from bounding rectangle the expected number of oracle calls would be 5.06×10^{70}.

The region $\mathcal{A}(f)$ should be convex when adaptive uniform sampling is used.

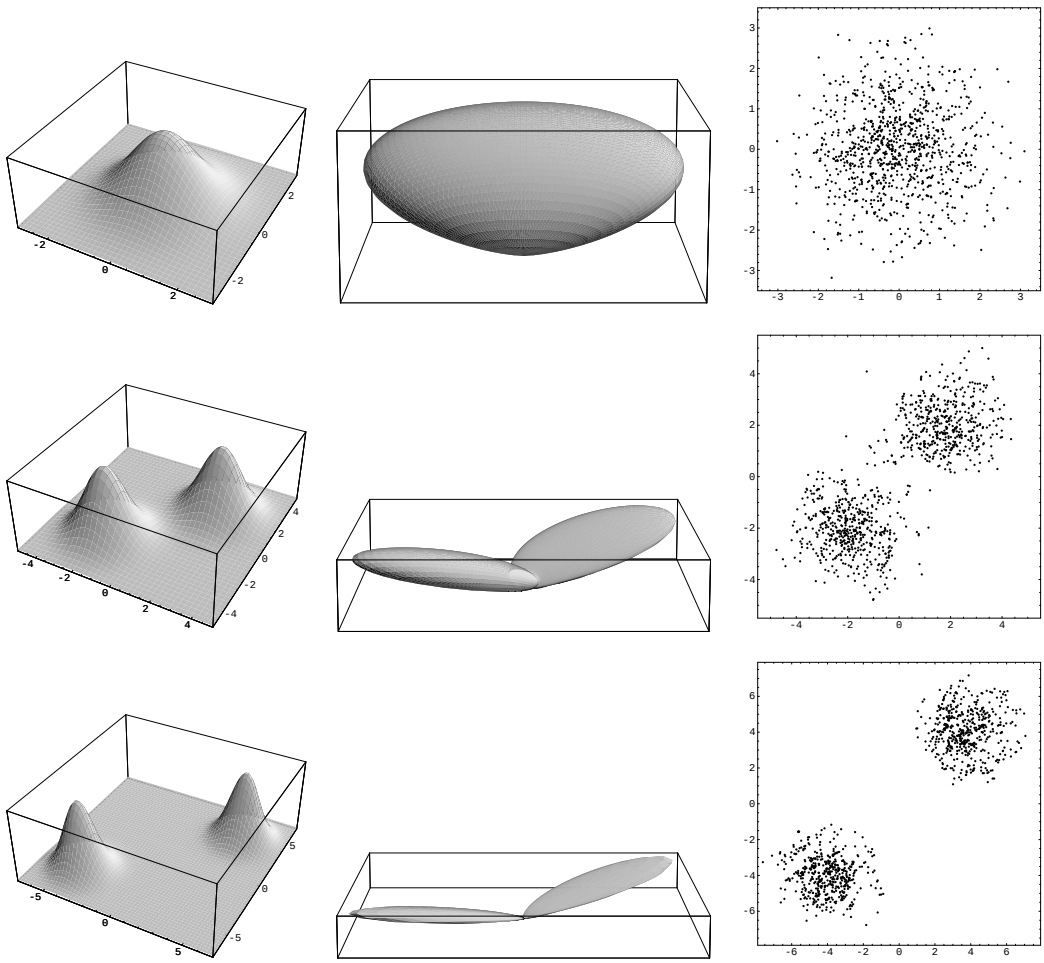


Fig. 3. Graph of mixture density function (8), the surfaces of the RoU-shapes and $N = 1000$ random sample points for $\mu = 0$, $\mu = 2$ and $\mu = 4$ respectively.

Otherwise, the ergodicity of the proposed Hit-and-Run sampler has not been shown, yet. Nevertheless, we ran experiments on normal mixtures. Figure 3 shows graphs, regions $\mathcal{A}(f)$, and chains of length 1000 produced by our algorithm with adaptive uniform sampling for bivariate distributions with densities

$$f(x,y) = \frac{1}{4\pi}(e^{-((x-\mu)^2+(y-\mu)^2)/2} + e^{-((x+\mu)^2+(y+\mu)^2)/2}) \; . \tag{8}$$

Notice that $\mathcal{A}(f)$ is not convex in all cases. Moreover, for large values of μ neither the Gibbs sampler nor a random walk Metropolis sampler would work efficiently. Nevertheless, the Hit-and-Run algorithm seems to mix fast even for this difficult distribution.

6 Conclusion

Adaptive uniform sampling with an unbounded plate (Algorithm HITRO-II) is the best suited practical algorithm among our proposed methods. The computation of the the mode can be seen as equivalent to the burn-in phase of other Markov chain samplers. Indeed, one could use the algorithm to find the mode of the density f by starting with some guess for the upper bound of f, generate a chain and adapt this guess whenever a larger value for f was found (see Zabinsky (1998) for a survey on stochastic optimization).

Compared to the Gibbs sampler and the Random Walk Metropolis sampler it seems to be less sensitive to higher correlation and seems to work also for multimodal distributions like those in Fig. 3. Different to the original Hit-and-Run sampler for multivariate distributions and the Gibbs sampler there is no necessity to sample from non-uniform conditional distributions.

The slice sampler (Neal, 2003) has been proposed to sample points uniformly in the region below the graph of f, $\mathcal{G}(f)$. There points are sampled uniformly in slices $\{\mathbf{x}\colon f(\mathbf{x}) = y\}$. However, this requires a search algorithm to get a cover for such a slice each time. The sampler proposed by Chen and Schmeiser (1998) also generates a Markov chain with uniform stationary distribution in $\mathcal{G}(f)$. However, it requires to fix two parameters which are crucial for the performance of the algorithms.

An important feature of the new algorithm is its simplicity. No proposal distribution has to be adjusted for the target distribution. (The influence of the parameter r is rather small). Of course rescaling can improve the convergence of the sampler.

Remark:
When we finished this paper we became aware of a recent talk by Tierney (2005) who also pointed out the usefulness of the ratio-of-uniforms method for Markov chain Monte Carlo.

Acknowledgments

The authors are grateful for helpful conversations with Wolfgang Hörmann, Klaus Pötzelberger and Günter Tirler. We thank Sylvia Früwirth-Schnatter for calling our attention to the talk of Luke Tierney. This work was supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung*, Proj.No. P16767-N12.

References

C. Bélisle. Slow hit-and-run sampling. *Stat. Probab. Lett.*, 47(1):33–43, 2000.

C. Bélisle, A. Boneh, and R. J. Caron. Convergence properties of hit-and-run samplers. *Commun. Stat., Stochastic Models*, 14(4):767–800, 1998.

C. J. P. Belisle, H. E. Romeijn, and R. L. Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18:255–266, 1993.

A. Boneh and A. Golan. Constraints' redundancy and feasible region boundedness by random feasible point generator (rfpg). In *Third European Congress on Operations Research, EURO III*, Amsterdam, 1979.

M.-H. Chen and B. Schmeiser. Performance of the Gibbs, Hit-and-run, and Metropolis samplers. *Journal of Computational and Graphical Statistics*, 2: 251–272, 1993.

M.-H. Chen and B. Schmeiser. Toward black-box sampling: a random-direction interior-point Markov chain approach. *Journal of Computational and Graphical Statistics*, 7:1–22, 1998.

M.-H. Chen and B. W. Schmeiser. General hit-and-run Monte Carlo sampling for evaluating multidimensional integrals. *Oper. Res. Lett.*, 19(4):161–169, 1996.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41(2):337–348, 1992.

W. Hörmann. A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Software*, 21(2):182–193, 1995.

W. Hörmann, J. Leydold, and G. Derflinger. *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin Heidelberg, 2004.

D. E. Kaufman and R. L. Smith. Direction choice for accelerated convergence in hit-and-run sampling. *Oper. Res.*, 46(1):84–95, 1998.

A. J. Kinderman and J. F. Monahan. Computer generation of random variables using the ratio of uniform deviates. *ACM Trans. Math. Software*, 3 (3):257–260, 1977.

J. Leydold. Automatic sampling with the ratio-of-uniforms method. *ACM Trans. Math. Software*, 26(1):78–98, 2000.

L. Lovász. Hit-and-run mixes fast. *Math. Program.*, 86(3):443–461, 1999.

L. Lovász and S. Vempala. Hit-and-run is fast and fun. Tech. Rep. MSR-TR-2003-05, Microsoft Research, 2003. Available at `http://www-math.mit.edu/~vempala/papers/logcon-hitrun.ps`.

L. Lovász and S. Vempala. Hit-and-run from a corner. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 310–314, Chicago, IL, USA, 2004. ACM Press.

R. M. Neal. Slice sampling. *Ann. Stat.*, 31(3):705–767, 2003. (With discussions and rejoinder).

M. L. Overton. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, Philadelphia, 2001.

R. L. Smith. Monte Carlo procedures for generating random feasible solutions

to mathematical programming. In *ORSA/TIMS Conference*, Whashington, D.C., 1980.

R. L. Smith. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32:1296–1308, 1984.

R. L. Smith. The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions. In J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, editors, *WSC '96: Proceedings of the 28th conference on Winter simulation*, pages 260–264. ACM Press, 1996.

L. Tierney. Markov chain Monte Carlo using the ratio-of-uniforms transformations. URL `http://www.stat.uiowa.edu/~luke/talks/bormio05.pdf`. Talk presented at the 2nd IMS-ISBA Joint Meeting MCMCSki, Bormio, January 12–14, 2005.

J. C. Wakefield, A. E. Gelfand, and A. F. M. Smith. Efficient generation of random variates via the ratio-of-uniforms method. *Statist. Comput.*, 1(2): 129–133, 1991.

Z. B. Zabinsky. Stochastic methods for practical global optimization. *Journal of Global Optimization*, 13:433–444, 1998.