4 Stochastische Simulation und Monte-Carlo-Verfahren

Simulationsverfahren für Stichproben von Wahrscheinlichkeitsverteilungen gehen in der Regel von der Existenz einer Folge von auf dem reellen Intervall $[0, 1]$ gleichverteilten, unabhängigen Zufallszahlen aus, die durch einen Zufallszahlengenerator erzeugt werden. In Wirklichkeit simulieren Zufallszahlengeneratoren natürlich nur auf $\{k\, m^{-1} : k = 0, 1, \ldots, m-1\}$ gleichverteilte Zufallszahlen, wobei m^{-1} die Darstellungsgenauigkeit des Computers ist. Außerdem ist eine Folge von vom Computer erzeugten Pseudozufallszahlen eigentlich gar nicht zufällig, sondern deterministisch. In Abschnitt 4.1 gehen wir kurz auf Verfahren und Probleme bei der Erzeugung von Pseudozufallszahlen mithilfe eines Zufallszahlengenerators ein. Im Abschnitt 4.2 betrachten wir dann verschiedene grundlegenden Verfahren, um Stichproben von allgemeineren Wahrscheinlichkeitsverteilungen aus Stichproben von unabhängigen gleichverteilten Zufallsvariablen zu erzeugen. Schließlich betrachten wir in Abschnitt 4.3 Monte-Carlo-Verfahren, die Gesetze der großen Zahlen verwenden, um Wahrscheinlichkeiten und Erwartungswerte mithilfe von simulierten Stichproben näherungsweise zu berechnen.

Um Simulationsverfahren zu analysieren, benötigen wir noch den Begriff einer auf dem Intervall $[0, 1]$ bzw., äquivalent dazu, auf dem offenen Intervall $(0, 1) \subseteq \mathbb{R}$ gleichverteilten reellwertigen Zufallsvariablen. Die Existenz solcher Zufallsvariablen auf einem geeigneten Wahrscheinlichkeitsraum wird in der Vorlesung Analysis III gezeigt, und hier zunächst vorausgesetzt.

Definition 4.1. Sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum.

(i) Eine Abbildung $U : \Omega \to \mathbb{R}$ heißt **reellwertige Zufallsvariable**, falls die Menge $\{U \le y\} = \{\omega \in \Omega : U(\omega) \le y\}$ für jedes $y \in \mathbb{R}$ in der σ-Algebra $\mathcal{A}$ enthalten ist.

(ii) Eine reellwertige Zufallsvariable U heißt **gleichverteilt auf dem Intervall** $(0, 1)$, falls

$$P[U \le y] = y \qquad \text{für jedes } y \in (0, 1) \text{ gilt.}$$

Im folgenden schreiben wir kurz $U \sim \mathrm{Unif}(0, 1)$ falls U auf $(0, 1)$ gleichverteilt ist.

(iii) Sei I eine beliebige Indexmenge, und seien $U_i : \Omega \to \mathbb{R}$ $(i \in I)$ reellwertige Zufallsvariablen, und $X_i : \Omega \to S$ $(i \in I)$ diskrete Zufallsvariablen mit abzählbarem Zustandsraum S. Dann heißen die Zufallsvariablen U_i und X_i $(i \in I)$ **unabhängig**, falls die Ereignisse $\{U_i \le y_i\}$ und $\{X_i \le a_i\}$, $i \in I$, für alle $y_i \in \mathbb{R}$ und $a_i \in S$ unabhängig sind.

Die Definition ist ein Spezialfall der Definition von Zufallsvariablen mit allgemeinen Zustandsräumen und deren Unabhängigkeit, die in der Vorlesung Einführung in die Wahrscheinlichkeitstheorie gegeben werden.

4.1 Pseudozufallszahlen

Ein *(Pseudo-) Zufallszahlengenerator* ist ein Algorithmus, der eine deterministische Folge von ganzen Zahlen $x_1, x_2, x_3, \ldots$ mit Werten zwischen 0 und einem Maximalwert $m-1$ erzeugt, welche durch eine vorgegebene Klasse statistischer Tests nicht von einer Folge von Stichproben unabhängiger, auf $\{0, 1, 2, \ldots, m-1\}$ gleichverteilter Zufallsgrößen unterscheidbar ist. Ein Zufallszahlengenerator erzeugt also nicht wirklich zufällige Zahlen. Die von „guten“ Zufallszahlengeneratoren erzeugten Zahlen haben aber statistische Eigenschaften, die denen von echten Zufallszahlen in vielerlei (aber nicht in jeder) Hinsicht sehr ähnlich sind.

Zufallszahlengeneratoren

Konkret werden Pseudozufallszahlen üblicherweise über eine deterministische Rekurrenzrelation vom Typ

$$x_{n+1} = f(x_{n-k+1}, x_{n-k+2}, \ldots, x_n), \qquad n = k, k+1, k+2, \ldots,$$

aus *Saatwerten* $x_1, x_2, \ldots, x_k$ erzeugt. In vielen Fällen hängt die Funktion f nur von der letzten erzeugten Zufallszahl x_n ab. Beispiele von Pseudozufallszahlengeneratoren sind lineare Kongruenzgeneratoren und Shift-Register-Generatoren.

Lineare Kongruenzgeneratoren

Bei einem linearen Kongruenzgenerator (LCG) ist die Rekurrenzrelation vom Typ

$$x_{n+1} = (ax_n + c) \mod m, \qquad n = 0, 1, 2, \ldots.$$

Hierbei sind a, c und m geeignet zu wählende positive ganze Zahlen, zum Beispiel:

Generator	m	a	c
ZX81	$2^{16}+1$	75	0
RANDU, IBM 360/370	2^{31}	65539	0
Marsaglia	2^{32}	69069	1
Langlands	2^{48}	142412240584757	11

Ein erstes Problem, dass bei linearen Kongruenzgeneratoren auftreten kann, ist, dass die Folge von Pseudozufallszahlen periodisch mit einer Periode ist, die im Allgemeinen deutlich kleiner als die maximal mögliche Periode m sein kann:

> **Beispiel (LCG mit kleiner Periode).** Wählen wir $m = 63$, $a = 11$ und $c = 0$, dann hat die Folge der vom linearen Kongruenzgenerator erzeugten Pseudozufallszahlen die Periode 6, siehe Abbildung 4.1.

Dieses erste Problem lässt sich leicht mithilfe der folgenden Charakterisierung aller linearen Kongruenzgeneratoren mit der maximal möglichen Periode m umgehen:

Theorem 4.2 (Knuth). Die Periode eines LCG ist gleich m genau dann, wenn

(i) c und m teilerfremd sind,

(ii) jeder Primfaktor von m ein Teiler von $a-1$ ist, und

(iii) falls 4 ein Teiler von m ist, dann auch von $a-1$.

Der Beweis dieses zahlentheoretischen Satzes findet sich in **Knuth:1997:ACP:270146**

Auch wenn ein linearer Kongruenzgenerator die maximal mögliche Periode hat, können weitere Probleme durch versteckte Strukturen und Symmetrien auftreten. Bei einigen einfachen Generatoren werden diese Probleme schon sichtbar, wenn man die Pseudozufallszahlen benutzt, um zwei- oder dreidimensionale Pseudozufallsvektoren zu erzeugen. Dies ist in den Abbildungen 4.2 und 4.3 demonstriert.

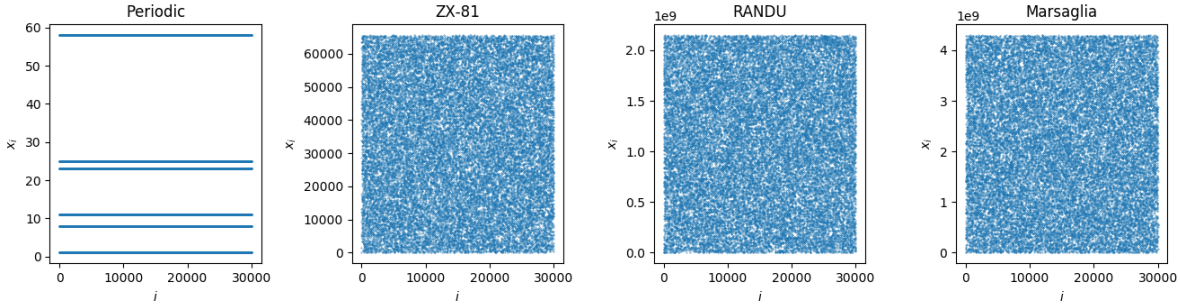


Abbildung 4.1: Plots der Folgen $x_1, \ldots, x_{30000}$ für den LCG mit Periode 6 aus dem Beispiel, sowie für den ZX81-Generator, RANDU, und den Marsaglia-Generator.

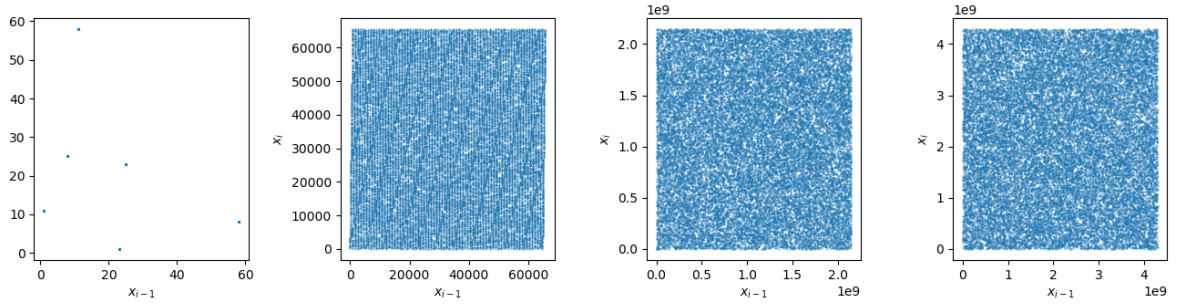


Abbildung 4.2: Fassen wir Paare (x_i, x_{i+1}) von aufeinanderfolgenden Pseudozufallszahlen als Koordinaten eines zweidimensionalen Pseudozufallsvektors auf, und betrachten die empirische Verteilung dieser Vektoren, so ergibt sich beim ZX81-Generator keine besonders gute Approximation einer zweidimensionalen Gleichverteilung.

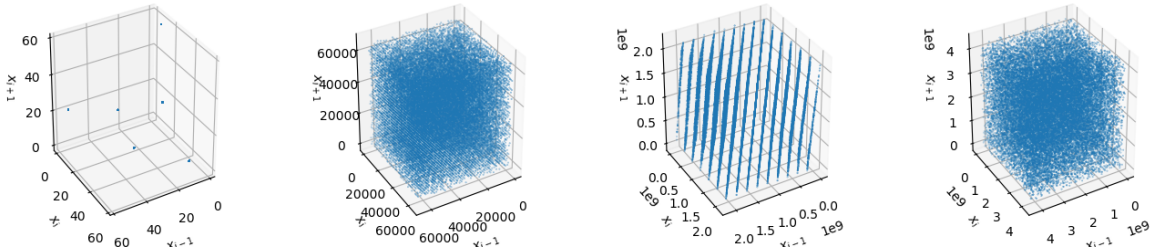


Abbildung 4.3: Fassen wir analog jeweils drei aufeinanderfolgende Pseudozufallszahlen als dreidimensionale Vektoren auf, dann konzentrieren sich diese beim RANDU-LCG auf 15 Hyperebenen.

Der Marsaglia-Generator besteht alle drei Tests; da in Wirklichkeit aber auch dieser deterministische Werte liefert, kann man auch hier einen Test konstruieren, der die Pseudozufallszahlen von echten Zufallszahlen unterscheidet.

Shift-Register-Generatoren

Eine andere Rekurrenzrelation wird zur Erzeugung von Pseudozufallszahlen mit Shift-Register-Generatoren verwendet. Hier interpretiert man eine Zahl $x_n \in \{0,1,\ldots,2^k-1\}$ zunächst als Binärzahl bzw. als Vektor aus $\{0,1\}^k$, und wendet dann eine gegebene Matrix T darauf an, um x_{n+1} zu erhalten:

$$x_{n+1} = Tx_n, \qquad n = 0, 1, 2, \ldots.$$

Kombination von Zufallszahlengeneratoren

Generatoren von Pseudozufallszahlen lassen sich kombinieren, zum Beispiel indem man die von mehreren Zufallszahlengeneratoren erzeugten Folgen von Pseudozufallszahlen aus $\{0,1,\ldots,m-1\}$ modulo m addiert. Auf diese Weise erhält man sehr leistungsfähige Zufallszahlengeneratoren, zum Beispiel den Kiss-Generator von Marsaglia [**marsaglia1993kiss**], der einen LCG und zwei Shift-Register-Generatoren kombiniert, Periode 2^{95} hat, und umfangreiche statistische Tests besteht.

Physikalische Zufallszahlengeneratoren

Alternativ werden Zufallszahlen auch mithilfe von physikalischen und insbesondere quantenmechanischen Vorgängen erzeugt, z.B. durch radioaktive Zerfälle, thermisches Rauschen, Atmosphärenrauschen etc. Ein Nachteil ist, dass auf diese Weise nur eine begrenzte Anzahl unabhängiger Stichproben pro Zeiteinheit erzeugt werden kann. Zudem sind die erhaltenen Ergebnisse nicht reproduzierbar. Auch physikalische Zufallsgeneratoren können mit algorithmischen Pseudozufallszahlengeneratoren kombiniert werden.

Statistische Tests für Zufallszahlengeneratoren

Wie wir schon in den Abbildung 4.1, 4.2 und 4.3 gesehen haben, können Schwachstellen von Zufallszahlengeneratoren mithilfe statistischer Tests aufgezeigt werden. Wir wollen kurz auf die dabei zugrundeliegende Argumentation eingehen. Eine von einem Zufallszahlengenerator erzeugte Folge $x_1, x_2, x_3, \ldots$ soll eine Folge von Stichproben von *unabhängigen* Zufallsvariablen $X_1, X_2, X_3, \ldots$ simulieren, die auf der Menge $\{0, 1, \ldots, m-1\}$ *gleichverteilt* ist. Es stellt sich also die Frage, ob die erzeugte Zahlenfolge zu diesem mathematischen Modell passt. Um dies zu testen, leitet man aus den Modellannahmen Folgerungen her, und überprüft ob die erzeugte Zahlenfolge konsistent mit diesen Folgerungen ist.

> **Beispiel (Blocktest).** Sei d eine natürliche Zahl. Sind die Zufallsvariablen X_i unabhängig und gleichverteilt auf $\{0, 1, \ldots, m-1\}$, dann sind auch die Zufallsvektoren $(X_{(k-1)d+1}, X_{(k-1)d+2}, \ldots, X_{kd})$, $k \in \mathbb{N}$, wieder unabhängig und gleichverteilt auf dem Produktraum $\{0, 1, \ldots, m-1\}^d$. Genau dies haben wir in den Abbildungen 4.2 und 4.3 für $d = 2$ bzw. $d = 3$ graphisch getestet. In höheren Dimensionen versagt zwar der graphische Test, aber wir können weiterhin rechnerisch testen, ob sich zum Beispiel die relativen Häufigkeiten von Werten in einem bestimmten Bereich $A \subseteq \{0, 1, \ldots, m-1\}^d$ der simulierten Zufallsvektoren $(x_{(k-1)d+1}, x_{(k-1)d+2}, \ldots, x_{kd})$, $k = 1, \ldots, n$, für große n der Wahrscheinlichkeit von A unter der Gleichverteilung annähern.

Prinzipiell kann jede Folgerung aus den Modellannahmen zur Konzeption eines statistischen Tests verwendet werden. Beispielsweise haben wir in der Einleitung einen Test für 0-1-Zufallsfolgen betrachtet, der auf der Anzahl der Runs basiert. Da jeder Test nur einen bestimmten Aspekt berücksichtigen kann, ist auch für einen Zufallsgenerator, der viele der üblichen Tests besteht, noch nicht garantiert, dass er für eine konkrete Anwendung wirklich geeignet ist. Es kann daher sinnvoll sein, die Ergebnisse einer stochastischen Simulation mit verschiedenen Generatoren zu reproduzieren.

Simulation von Gleichverteilungen

Aus den von einem Pseudozufallszahlengenerator zunächst erzeugten Pseudozufallszahlen mit Werten in der endlichen Menge $\{0, 1, \ldots, m-1\}$ werden anschließend Pseudo-Stichproben von anderen Gleichverteilungen erzeugt.

Zufallszahlen aus $[0, 1)$

Ein Zufallszahlengenerator kann natürlich nicht wirklich reelle Pseudozufallszahlen erzeugen, die die Gleichverteilung auf dem Intervall $[0, 1)$ simulieren, denn dazu würden unendlich viele „zufällige“ Nachkommastellen benötigt. Stattdessen werden üblicherweise (pseudo-)zufällige Zahlen vom Typ

$$u_n = \frac{x_n}{m}, \qquad x_n \in \{0, 1, \ldots, m-1\},$$

erzeugt, wobei m vorgegeben ist (zum Beispiel Darstellungsgenauigkeit des Computers), und x_n eine Folge ganzzahliger Pseudozufallszahlen aus {0, 1, . . . , m - 1} ist.

Zufallspermutationen

Der folgende Algorithmus erzeugt eine (pseudo-)zufällige Permutation aus $\mathcal{S}_n$:

Algorithmus 1 : Zufällige Permutation

Input : $n \in \mathbb{N}$
Output : Zufällige Permutation der Länge n

```
1 for i ← 1 to n do
2     x_i ← i
3 for i ← 1 to n − 1 do
4     k ← i + ZufälligeGanzzahl({ 0,1,...,n-i});
5     Vertausche(x_i, x_k);
6 return (x_i)_{i=1}^n;
```

Aufgabe. Zeigen Sie, daß Algorithmus 1 tatsächlich eine Stichprobe einer gleichverteilten Zufallspermutation aus $\mathcal{S}_n$ simuliert. *Hinweis: Sei $\tau_{i,j}$ die Transposition von i und j. Zeigen Sie, daß die Abbildung*

$$X(\omega) = \tau_{n-1,\omega_{n-1}} \circ \cdots \circ \tau_{2,\omega_2} \circ \tau_{1,\omega_1}$$

eine Bijektion von $\Omega_n = \{1, 2, \ldots, n\} \times \{2, 3, \ldots, n\} \times \cdots \times \{n-1, n\}$ nach $\mathcal{S}_n$ ist.

4.2 Simulationsverfahren

Wir nehmen nun an, dass wir eine Folge $u_1, u_2, \ldots$ von Stichproben von auf $(0, 1)$ gleichverteilten, unabhängigen Zufallsvariablen $U_1, U_2, \ldots$ gegeben haben. Die in Abschnitt 4.1 beschriebenen Probleme beim Generieren solcher Stichproben werden wir im folgenden ignorieren. Stattdessen wollen wir uns nun überlegen, wie wir aus der Folge (u_n) Stichproben von einer vorgegebenen Wahrscheinlichkeitsverteilung μ auf einer abzählbaren Menge S erzeugen können. Dabei gehen wir in der Regel davon aus, dass wir die Gewichte $\mu(a) = \mu[\{a\}]$ zumindest bis auf eine Normierungskonstante kennen bzw. berechnen können.

Das direkte Simulationsverfahren

Sei $a_1, a_2, \ldots$ eine Abzählung der Elemente von S. Wir betrachten die durch

$$s_k \;:=\; \sum_{i=1}^{k} \mu(a_i) \;=\; \mu[\{a_1, \ldots, a_k\}] \tag{4.1}$$

definierte *kumulative Verteilungsfunktion*. Wir gehen davon aus, dass wir die Werte $\mu(a_i)$ und damit auch s_i für jedes $i \in \mathbb{N}$ berechnen können. Für $n, i \in \mathbb{N}$ setzen wir

$$x_n := a_i \qquad \text{falls } s_{i-1} < u_n \leq s_i.$$

Dann ist x_n eine Stichprobe von der Zufallsvariable

$$X_n := \sum_i a_i \, I_{(s_{i-1}, s_i]}(U_n).$$

Lemma 4.3. *Sind U_n $(n \in \mathbb{N})$ unabhängige Zufallsvariablen mit Verteilung $U_n \sim \mathrm{Unif}(0,1)$, dann sind $X_n (n \in \mathbb{N})$ unabhängige Zufallsvariablen mit Verteilung $X_n \sim \mu$.*

Beweis. Für alle $i \in \mathbb{N}$ gilt

$$P[X_n = a_i] = P[s_{i-1} < U_n \leq s_i] = P[U_n \leq s_i] - P[U_n \leq s_{i-1}] = s_i - s_{i-1} = \mu(a_i).$$

Also hat X_n die Verteilung μ. Der Nachweis der Unabhängigkeit ist eine Übungsaufgabe. ■

Algorithmus 2 : Direkte Simulation einer Stichprobe von einer diskreten Wahrscheinlichkeitsverteilung

Input : Gewichte $(\mu(a_i))_{i \in \mathbb{N}}$
Output : Pseudozufallsstichprobe x von μ

```
1 i ← 1 ;
2 s ← μ(a_1) ;
3 u ← Stichprobe(Unif[0, 1]);
4 while u > s do
5 |   i ← i + 1;
6 |   s ← s + μ(a_i)
7 return x := a_i;
```

Bemerkung (Mittlere Laufzeit). Die mittlere Anzahl von Schritten des Algorithmus ist gleich $\sum i \, \mu(a_i)$.

Nach der Bemerkung ist das direkte Verfahren im Allgemeinen nur dann praktikabel, wenn die Gewichte $\mu(a_i)$ für große i rasch abfallen. In einigen einfachen Spezialfällen kann man jedoch eine explizite Formel zur Berechnung von x_n aus u_n angeben, für deren Auswertung die Schleife in Algorithmus 2 nicht durchlaufen werden muss:

Aufgabe (Simulation von Stichproben einer geometrischen Verteilung). Geben Sie ein direktes Verfahren an, dass in einem Schritt aus einer Stichprobe von der Gleichverteilung auf dem Intervall $(0,1)$ eine Stichprobe von der geometrischen Verteilung mit Parameter $p \in (0,1)$ erzeugt.

Das Acceptance-Rejection-Verfahren

Da das direkte Verfahren oft nicht praktikabel ist, benötigen wir Alternativen. Eine häufig verwendete Methode besteht darin, zunächst unabhängige Stichproben von einer "einfacheren" Wahrscheinlichkeitsverteilung ν auf demselben Zustandsraum S zu generieren, und daraus mit einem Verwerfungsverfahren Stichproben von der Zielverteilung μ zu erzeugen. Dazu nehmen wir an, dass wir die Quotienten $\mu(x)/\nu(x)$ der Gewichte unter μ bzw. ν bis auf eine Proportionalitätskonstante kennen, d.h. für $x \in S$ gilt

$$\mu(x) \propto f(x)\nu(x) \tag{4.2}$$

mit einer explizit bekannten Funktion $f : S \to \mathbb{R}$. Beispielsweise können wir $f(x) = \mu(x)/\nu(x)$ setzen, wenn dieses Verhältnis explizit bekannt ist. Wir setzen zudem voraus, dass wir eine obere Schranke c für die Funktion f kennen, d.h.

$$\text{es gibt ein } c \in [1, \infty), \text{ so dass} \qquad f(x) \le c \qquad \text{für alle } x \in S. \tag{4.3}$$

Angenommen, wir können Folgen von Stichproben x_n, u_n ($n \in \mathbb{N}$) von unabhängigen Zufallsvariablen X_n, U_n mit Verteilung ν bzw. Unif(0, 1) erzeugen. Dann können wir daraus Stichproben von der Zielverteilung μ generieren, indem wir die x_n als Vorschlagswerte betrachten, die mit einer Wahrscheinlichkeit proportional zu $f(x_n)$ akzeptiert, und ansonsten verworfen werden. Aufgrund der Annahme (4.3) können die *Akzeptanzwahrscheinlichkeiten* dabei gleich $f(x)/c$ gewählt werden.

Algorithmus 3 : Acceptance-Rejection-Verfahren (AR)

Input : $f : S \to [0, \infty)$, $c \in [1, \infty)$ mit (4.3)
Output : Stichprobe x von Wahrscheinlichkeitsverteilung μ mit (4.2)

```
repeat
    x ←Stichprobe(ν) ;
    u ←Stichprobe(Unif(0, 1));
until u ≤ f(x)/c;
return x;
```

Wir wollen den Algorithmus nun analysieren. Seien dazu $X_n \sim \nu$ und $U_n \sim \text{Unif}(0, 1)$ ($n \in \mathbb{N}$) unabhängige Zufallsvariablen, die auf einem gemeinsamen Wahrscheinlichkeitsraum definiert sind. Die diskrete Zufallsvariable

$$T(\omega) \;=\; \min\{n \in \mathbb{N} \,:\, U_n(\omega) \le f(X_n(\omega))/c\}$$

beschreibt dann die Anzahl der Durchläufe der Schleife bis erstmals ein Vorschlag X_n akzeptiert wird, und

$$X_T(\omega) \;=\; X_{T(\omega)}(\omega)$$

ist der akzeptierte Wert, der schließlich ausgegeben wird.

Theorem 4.4 (Laufzeit und Output des AR-Verfahrens).

(i) T ist *geometrisch verteilt* mit Parameter $p = \sum_{a \in S} \frac{f(a)\nu(a)}{c}$. Insbesondere ist T fast sicher endlich.

(ii) Die Zufallsvariable X_T hat die Verteilung μ.

Der Satz zeigt, dass der Algorithmus tatsächlich eine Stichprobe von der Verteilung μ liefert. Die mittlere Anzahl von Schritten, bis ein Vorschlag akzeptiert wird, beträgt $E[T] = 1/p$. Ist $f = \mu/\nu$, dann ist $p = 1/c$, also die mittlere Laufzeit gleich c.

Beweis (von Theorem 4.4). (i) Sei $A_n := \{U_n \le f(X_n)/c\}$ das Ereignis, dass der n-te Vorschlag akzeptiert wird. Aus der Unabhängigkeit der Zufallsvariablen $X_1, U_1, X_2, U_2, \ldots$ folgt, daß auch die Ereignisse $A_1, A_2, \ldots$ unabhängig sind. Dies wird in der Vorlesung Einführung in die Wahrscheinlichkeitstheorie allgemein bewiesen, lässt sich im hier betrachteten Spezialfall aber auch direkt überprüfen. Zudem gilt wegen der Unabhängigkeit von X_n und U_n:

$$\begin{aligned} P[A_n] \;&=\; \sum_{a \in S} P\left[\{X_n = a\} \cap A_n\right] \;=\; \sum_{a \in S} P\left[X_n = a, U_n \le f(a)/c\right] \\ &=\; \sum_{a \in S} P\left[X_n = a\right] \cdot P\left[U_n \le f(a)/c\right] \;=\; \sum_{a \in S} \nu(a) f(a)/c = p. \end{aligned}$$

Also ist $T(\omega) = \min\{n \in \mathbb{N} : \omega \in A_n\}$ geometrisch verteilt mit Parameter p.

(ii) Für $a \in S$ gilt

$$\begin{aligned} P[X_T = a] &= \sum_{n=1}^{\infty} P[\{X_T = a\} \cap \{T = n\}] \\ &= \sum_{n=1}^{\infty} P[\{X_n = a\} \cap A_n \cap A_1^C \cap \ldots \cap A_{n-1}^C] \\ &= \sum_{n=1}^{\infty} P[\{X_n = a, U_n \le f(a)/c\} \cap A_1^C \cap \ldots \cap A_{n-1}^C] \\ &= \sum_{n=1}^{\infty} \nu(a) \frac{f(a)}{c} (1-p)^{n-1} \;=\; \frac{f(a)\nu(a)}{pc}. \end{aligned}$$

Hierbei haben wir im letzten Schritt benutzt, dass die Ereignisse $\{X_n = a\}$, $\{U_n \le f(a)/c\}$, sowie $A_1^C, \ldots, A_{n-1}^C$ unabhängig sind. Da μ die einzige Wahrscheinlichkeitsverteilung ist, deren Massenfunktion proportional zu $f(a)\nu(a)$ ist, folgt $X_T \sim \nu$. ■

Aufgabe (Unabhängigkeit). Sei S eine abzählbare Menge, $g : S \to \mathbb{R}$ eine Funktion, und seien $X_1, X_2, \ldots : \Omega \to S$ sowie $U_1, U_2, \ldots : \Omega \to \mathbb{R}$ unabhängige Zufallsvariablen auf $(\Omega, \mathcal{A}, P)$ mit Verteilungen $X_n \sim \mu$, $U_n \sim \mathrm{Unif}(0,1)$. Zeigen Sie, dass die Ereignisse

$$A_n \;:=\; \{U_n \le g(X_n)\}, \qquad n \in \mathbb{N},$$

unabhängig sind. *(Hinweis: Zeigen Sie zunächst die Unabhängigkeit von A_1 und A_2).*

> **Beispiel (Simulation von bedingten Verteilungen).** Das Acceptance-Rejection-Verfahren kann prinzipiell verwendet werden, um Stichproben von einer bedingten Verteilung $\mu[A] = \nu[A|B]$ zu simulieren, wobei $B \subseteq S$ ein Ereignis mit $\nu[B] > 0$ ist. In diesem Fall gilt $\mu(x) = f(x)\nu(x)$ mit
>
> $$f(x) \;=\; I_B(x)/\nu[B] \;\le\; 1/\nu[B] \qquad \text{für alle } x \in S,$$
>
> so dass wir $c = 1/\nu[B]$ wählen können. Das AR-Verfahren erzeugt dann Stichproben von der Verteilung ν und akzeptiert diese mit Wahrscheinlichkeit $I_B(x)$, d.h., Stichproben in B werden stets akzeptiert. Da die mittlere Laufzeit gleich c ist, ist das Verfahren nur dann praktikabel, wenn die Wahrscheinlichkeit von B nicht zu klein ist.

Der Metropolis-Hastings-Algorithmus

Häufig sind direkte oder Acceptance-Rejection-Verfahren zur Simulation von Stichproben einer Wahrscheinlichkeitsverteilung μ nicht praktikabel. Eine Alternative ist die Simulation einer Markovkette (X_n) mit Gleichgewicht μ. Konvergiert die Markovkette ins Gleichgewicht, dann ist die Verteilung von X_n für hinreichend große n ungefähr gleich μ. Eine Stichprobe x_n von X_n ist daher auch eine Näherung einer Stichprobe von μ. Um eine Markovkette mit Gleichgewicht μ zu finden, benutzt man meistens die hinreichende Detailed-Balance-Bedingung (**??**). Die zwei wichtigsten Verfahren, die sich auf diese Weise ergeben, sind der *Metropolis-Hastings-Algorithmus* und der *Gibbs Sampler*.

Wir betrachten zunächst den Metropolis-Hastings-Algorithmus. Sei μ eine beliebige Wahrscheinlichkeitsverteilung auf S mit Gewichten $\mu(x) > 0$ für alle $x \in S$, und sei $q = (q(x,y))_{x,y \in S}$ eine stochastische Matrix, für die

$$q(x,y) = 0 \qquad \Leftrightarrow \qquad q(y,x) = 0 \tag{4.4}$$

für alle $x, y \in S$ gilt. Eine typische Wahl für q ist beispielsweise die Übergangsmatrix eines Random Walks bezüglich einer geeigneten Graphenstruktur. Wie können wir die Matrix q so modifizieren, daß die

Detailed-Balance-Bedingung (**??**) bzgl. μ erfüllt ist ? Die Grundidee des Metropolis-Hastings-Algorithmus ist, Übergänge von x nach y mit den Wahrscheinlichkeiten $q(x, y)$ vorzuschlagen, die Vorschläge aber nur mit einer geeignet gewählten *Akzeptanzwahrscheinlichkeit* $\alpha(x, y)$ zu akzeptieren. Wird ein Vorschlag nicht akzeptiert, dann bleibt die Markovkette an der Stelle x.

Algorithmus 4 : Metropolis-Hastings-Algorithmus (MH)

Input : Stochastische Matrix q, Wahrscheinlichkeitsverteilungen ν, μ

Output : Stichproben $x_0, x_1, \ldots$ von Markovkette mit Startverteilung ν und Gleichgewicht μ

$n \leftarrow 0$; $x_0 \leftarrow$`Stichprobe(`ν`)`;
repeat
  $y_{n+1} \leftarrow$`Stichprobe(`$q(x_n, \bullet)$`)` ;
  $u_{n+1} \leftarrow$`Stichprobe(`Unif$(0, 1)$`)`;
  if $u_{n+1} \le \alpha(x_n, y_{n+1})$ **then** accept: $x_{n+1} \leftarrow y_{n+1}$ **else** reject: $x_{n+1} \leftarrow x_n$;
  $n \leftarrow n + 1$;
until *Abbruchkriterium*;

Die Übergangsmatrix der im Algorithmus simulierten Markovkette ist

$$\pi(x, y) \;:=\; \begin{cases} \alpha(x, y)\, q(x, y) & \text{für } y \neq x, \\ 1 - \sum_{y \neq x} \alpha(x, y)\, q(x, y) & \text{für } y = x. \end{cases} \tag{4.5}$$

Wir müssen noch spezifizieren, wie die Akzeptanzwahrscheinlichkeiten im Algorithmus gewählt werden, damit μ tatsächlich ein Gleichgewicht ist. Die Detailed-Balance-Bedingung lautet in diesem Fall

$$b(x, y) \;=\; b(y, x) \qquad \text{für alle } x, y \in S \text{ mit } x \neq y, \tag{4.6}$$

wobei wir

$$b(x, y) \;:=\; \mu(x)\, \alpha(x, y)\, q(x, y) \tag{4.7}$$

setzen. Um sicherzustellen, dass die Markovkette nicht häufiger als unbedingt nötig an derselben Stelle stehen bleibt, wollen wir diese Bedingung mit möglichst großen Akzeptanzwahrscheinlichkeiten $\alpha(x, y) \in [0, 1]$, also mit möglichst großen Werten für $b(x, y)$ erfüllen. Wegen $\alpha(x, y) \le 1$ muss nach (4.7)

$$b(x, y) \;\le\; \min\left(\mu(x) q(x, y),\, \mu(y) q(y, x)\right) \tag{4.8}$$

gelten, falls die Symmetriebedingung (4.6) erfüllt ist. Damit ergibt sich als maximale Wahl von b mit(4.6) der Wert auf der rechten Seite von (4.8). Entsprechend erhalten wir die MH-Akzeptanzwahrscheinlichkeiten

$$\alpha(x, y) \;=\; \min\left(1,\, \frac{\mu(y) q(y, x)}{\mu(x) q(y, x)}\right) \qquad \text{für alle } x, y \in S \text{ mit } q(x, y) \neq 0. \tag{4.9}$$

Für $x, y \in S$ mit $q(x, y) = 0$ können wir $\alpha(x, y)$ beliebig wählen, da in diesem Fall $\pi(x, y) = 0$ unabhängig von der Wahl von α gilt.

Beispiel (Metropolis-Algorithmus). In der ursprünglich von Metropolis, Rosenbluth, Rosenbluth, Teller und Teller **MRRTT** betrachteten Version des Algorithmus ist die Vorschlagsmatrix symmetrisch, d.h. es gilt $q(x, y) = q(y, x)$ für alle $x, y \in S$. In diesem Fall vereinfacht sich die Formel für die Akzeptanzwahrscheinlichkeiten zu

$$\alpha(x, y) \;=\; \min(1, \mu(y)/\mu(x)). \tag{4.10}$$

Ist beispielsweise der Zustandsraum S ein regulärer Graph, dann liegt es nahe, als Vorschlagsmatrix die symmetrische Übergangsmatrix des Random Walks auf dem Graphen zu wählen.

Definition 4.5. Eine Markovkette (X_n) mit der durch (4.5) und (4.9) definierten Übergangsmatrix heißt **Metropolis-Hastings-Kette** mit Vorschlagsverteilung q und Gleichgewicht μ.

Die Konvergenz ins Gleichgewicht einer Metropolis-Hastings-Kette folgt unter schwachen Voraussetzungen aus dem Konvergenzsatz für Markovketten:

Aufgabe (Konvergenz ins Gleichgewicht für MH). Sei μ eine Wahrscheinlichkeitsverteilung auf einem endlichen Zustandsraum S mit Gewichten $\mu(x) > 0$, und sei $q = (q(x, y))_{x,y \in S}$ eine irreduzible und aperiodische stochastische Matrix auf S, die (4.4) erfüllt. Zeigen Sie, dass μ ein Gleichgewicht ist, und folgern Sie, dass die Verteilung von X_n für eine beliebige Startverteilung ν in Variationsdistanz gegen μ konvergiert.

Der Konvergenzsatz löst aber noch nicht die praktischen Probleme, denn die Konvergenz ins Gleichgewicht kann sehr langsam erfolgen! Wichtig sind daher Abschätzungen der Konvergenzgeschwindigkeit und explizite Fehlerschranken. Diese sind in der Regel stark problemabhängig, und in anwendungsrelevanten Fällen meist nicht leicht herzuleiten.

Aufgabe (Independence Sampler). Sei μ eine Wahrscheinlichkeitsverteilung auf einer endlichen Menge S mit $\mu(x) > 0$ für alle $x \in S$. Der *Independence Sampler* ist ein spezieller MH-Algorithmus, bei dem die Vorschlagsverteilung $q(x, \cdot)$ nicht vom Ausgangspunkt x abhängt, d.h.

$$q(x, y) \;=\; \nu(y)$$

für eine feste Wahrscheinlichkeitsverteilung ν auf S mit $\nu(x) > 0$ für alle $x \in S$.

a) Geben Sie die Übergangsmatrix der entsprechenden Markovkette (X_n) an. Zeigen Sie, dass diese bzgl. des Gleichgewichts μ eine Minorisierungsbedingung mit Konstante $\delta \;=\; \min_{x \in S}(\nu(x)/\mu(x))$ erfüllt.

b) Leiten Sie eine Abschätzung für den Variationsabstand zwischen der Verteilung des Independence Samplers nach n Schritten und dem Gleichgewicht μ her.

c) Alternativ kann man in der obigen Situation eine Stichprobe von μ durch ein Acceptance-Rejection-Verfahren mit Vorschlagsverteilung ν erzeugen. Vergleichen Sie die beiden Verfahren.

Aufgabe (Gibbs-Sampler). Sei μ eine Wahrscheinlichkeitsverteilung auf einem endlichen Produktraum $S = S_1 \times \cdots \times S_d$ mit strikt positiven Gewichten $\mu(x_1, \ldots, x_d)$, und sei

$$\mu_i(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d) \;:=\; \frac{\mu(x_1, \ldots, x_d)}{\sum_{z \in S_i} \mu(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_d)}$$

die Massenfunktion der bedingten Verteilung der i-ten Komponente gegeben die Werte x_k $(k \neq i)$ der übrigen Komponenten. Zeigen Sie, dass durch Algorithmus 5 der Übergangsschritt einer Markovkette mit Gleichgewicht μ realisiert wird. *Hinweis: Schreiben Sie die Übergangsmatrix in der Form $\pi = \pi_d\, \pi_{d-1} \cdots \pi_1$ mit Übergangsmatrizen $\pi_1, \ldots, \pi_d$, die die Detailed Balance Bedingung bzgl. μ erfüllen.*

Algorithmus 5 : Gibbs Sampler, Übergangsschritt

Input : $x = (x_1, \ldots, x_d) \in S$
Output : $y = (y_1, \ldots y_d) \in S$

```
y ← x;
for i = 1 to d do
    y_i ←Stichprobe(μ_i( • | y_1, ... y_{i-1}, y_{i+1}, ..., y_d)) ;
return y;
```

Simulated Annealing

Für viele Optimierungsprobleme, die in der Praxis auftreten, sind keine Lösungen in einer polynomiellen Anzahl von Schritten mit deterministischen Algorithmen bekannt. Zudem bleiben deterministische Optimierungsalgorithmen häufig in lokalen Minima stecken. Daher greift man in diesen Fällen auch auf heuristische stochastische Verfahren zurück. Angenommen, wir wollen das globale Minimum einer Funktion $U : S \to \mathbb{R}$ auf einem endlichen Zustandsraum S bestimmen. In typischen Anwendungen ist S beispielsweise ein hochdimensionaler Produktraum. Um die Gleichverteilung auf den globalen Minima von U anzunähern, betrachtet man die Wahrscheinlichkeitsverteilungen μ_β, $\beta \in [0, \infty)$, mit Gewichten

$$\mu_\beta(x) \;=\; \mathcal{Z}_\beta^{-1} \exp(-\beta U(x)), \qquad x \in S, \tag{4.11}$$

wobei $\mathcal{Z}_\beta$ eine Normierungskonstante ist. In der statistischen Physik ist μ_β die *Boltzmann-Gibbs-Verteilung* im thermodynamischen Gleichgewicht für die Energiefunktion U bei Temperatur $T = 1/\beta$. Für festes β können wir eine Markovkette mit Gleichgewicht μ_β mithilfe des Metropolis-Hastings-Algorithmus simulieren. Ist die Vorschlagsmatrix $q(x, y)$ symmetrisch, dann sind die Akzeptanzwahrscheinlichkeiten nach (4.10) durch

$$\alpha_\beta(x, y) \;=\; \exp\left(-\beta(U(y) - U(x))^+\right) \tag{4.12}$$

gegeben. Wichtig ist, dass die rechte Seite nicht von $\mathcal{Z}_\beta$ abhängt, denn die Normierungskonstante ist meistens nicht explizit bekannt. Sei π_β die entsprechende Übergangsmatrix des MH-Algorithmus mit Gleichgewicht μ_β. Die Idee des Simulated Annealing Verfahrens („simuliertes Abkühlen“) besteht nun darin, eine *zeitlich inhomogene* Markovkette (X_n) mit Übergangskernen $p_n = \pi_{\beta(n)}$ zu simulieren, wobei $\beta(n)$ eine Folge ist, die gegen unendlich konvergiert. Die Gleichgewichtsverteilung der Übergangskerne p_n nähert sich dann für $n \to \infty$ der Gleichverteilung auf der Menge $\mathcal{M}$ der globalen Minima von U an.

Mithilfe ähnlicher Abschätzungen wie im Beweis der Konvergenzsätze für Markovketten kann man zeigen, daß die Verteilung der inhomogenen Markovkette zur Zeit n gegen die Gleichverteilung auf $\mathcal{M}$ konvergiert, falls $\beta(n)$ nur sehr langsam (logarithmisch) gegen $+\infty$ geht. In praktischen Anwendungen wird das Verfahren aber in der Regel mit einem „schnelleren“ *cooling schedule* $\beta(n)$ verwendet. In diesem Fall findet die Markovkette (X_n) im allgemeinen kein globales Minimum von U, sondern kann, ähnlich wie deterministische Optimierungsverfahren, in lokalen Minima „steckenbleiben“. Das Auffinden eines globalen Minimums ist dann also nicht garantiert – trotzdem erhält man ein oft nützliches *heuristisches* Verfahren.

Aufgabe (Konvergenz von Simulated Annealing). a) Zeigen Sie, dass die Boltzmann-Gibbs-Verteilung μ_β in (4.11) für $\beta \to \infty$ in Variationsdistanz gegen die Gleichverteilung auf der Menge $\mathcal{M}$ der globalen Minima von U konvergiert.

b) Sei π_β die Übergangsmatrix des Metropolis-Hastings-Algorithmus mit Gleichgewicht μ_β und Vorschlagsverteilung $q(x, \bullet) = \mathrm{Unif}(S)$. Zeigen Sie, dass π_β für jedes $\beta > 0$ eine Minorisierungsbedingung mit Konstante $\delta_\beta = \exp(-\beta(\max U - \min U))$ bezüglich der Gleichverteilung auf S erfüllt. Folgern Sie, dass es einen cooling schedule $\beta(n)$ gibt, für den die Verteilung von X_n in Variationsdistanz gegen $\mathrm{Unif}(\mathcal{M})$ konvergiert.

4.3 Monte-Carlo-Verfahren

Sei μ eine Wahrscheinlichkeitsverteilung mit Massenfunktion $\mu(x) = \mu[\{x\}]$ auf einer abzählbaren Menge S. Angenommen, wir wollen die Wahrscheinlichkeit

$$p \;:=\; \mu[B] \;=\; \sum_{x \in S} I_B(x)\, \mu(x)$$

eines Ereignisses $B \subseteq S$ beziehungsweise, allgemeiner, den Erwartungswert

$$\theta \;:=\; E_\mu[f] \;=\; \sum_{x \in S} f(x)\,\mu(x)$$

einer reellwertigen Zufallsvariable $f\colon S \to \mathbb{R}$ mit $E_\mu[f^2] < \infty$ (näherungsweise) berechnen, aber die Menge S ist zu groß, um die Summe direkt auszuführen. In einem solchen Fall können wir auf ein Monte-Carlo-Verfahren zurückgreifen. Hierbei simuliert man eine große Anzahl Stichproben $X_1(\omega), \dots, X_n(\omega)$ von unabhängigen Zufallsvariablen mit Verteilung μ (*klassisches Monte-Carlo-Verfahren*), beziehungsweise von einer konvergenten Markovkette mit Gleichgewicht μ (*Markov Chain Monte Carlo*). Nach dem Gesetz der großen Zahlen liefern dann die relativen Häufigkeiten

$$\widehat{p}_n(\omega) \;:=\; \frac{1}{n}\sum_{i=1}^{n} I_B(X_i(\omega)).$$

bzw. die empirischen Mittelwerte

$$\widehat{\theta}_n(\omega) \;:=\; \frac{1}{n}\sum_{i=1}^{n} f(X_i(\omega)).$$

Schätzwerte für p bzw. θ, die sich für $n \to \infty$ den gesuchten Werten annähern. Wir wollen nun verschiedene Abschätzungen für den Approximationsfehler $|\widehat{p}_n - p|$ bzw. $|\widehat{\theta}_n - \theta|$ vergleichen. Dazu nehmen wir an, dass die Zufallsvariablen X_i alle die Verteilung μ haben. Nach dem Transformationssatz (Satz **??**) und der Linearität des Erwartungswerts gilt dann

$$E[\widehat{\theta}_n] \;=\; \frac{1}{n}\sum_{i=1}^{n} E[f(X_i)] \;=\; \frac{1}{n}\sum_{i=1}^{n} E_\mu[f] \;=\; E_\mu[f] \;=\; \theta,$$

d.h. $\widehat{\theta}_n$ ist ein *erwartungstreuer Schätzer*[1] für θ. Der *mittlere quadratische Fehler* („MSE“ = Mean Squared Error) des Schätzers ist daher durch die Varianz der Zufallsvariable $\widehat{\theta}_n$ gegeben:

$$\mathrm{MSE}\left[\widehat{\theta}_n\right] \;:=\; E\left[\left|\widehat{\theta}_n - \theta\right|^2\right] \;=\; \mathrm{Var}\left[\widehat{\theta}_n\right].$$

Explizite Abschätzungen für den Approximationsfehler erhalten wir nun mit denselben Methoden wie beim Beweis von Gesetzen der großen Zahlen in Kapitel **??**. Sind die Zufallsvariablen X_i beispielsweise unabhängig mit Verteilung μ, dann sind die Zufallsvariablen $f(X_i)$ unkorreliert. In diesem Fall ergibt sich ein mittlerer quadratische Fehler

$$\mathrm{MSE}\left[\widehat{\theta}_n\right] \;=\; \mathrm{Var}\left[\widehat{\theta}_n\right] \;=\; \frac{1}{n}\,\mathrm{Var}_\mu[f]$$

von der Ordnung $O(1/n)$. Der mittlere quadratische Fehler fällt also relativ langsam in n ab. Ein großer Vorteil ist jedoch, dass die Abschätzung völlig *problemunabhängig* ist. Aus diesem Grund sind Monte-Carlo-Verfahren sehr universell einsetzbar. In komplizierten Modellen sind sie oft die einzige praktikable Option um Erwartungswerte näherungsweise zu berechnen. Nach der Čebyšev-Ungleichung erhalten wir zudem für $\varepsilon > 0$ und $n \in \mathbb{N}$ die Fehlerabschätzung

$$P\left[\left|\widehat{\theta}_n - \theta\right| \geq \varepsilon\right] \;\leq\; \frac{1}{\varepsilon^2} E\left[\left|\widehat{\theta}_n - \theta\right|^2\right] \;=\; \frac{1}{n\,\varepsilon^2}\,\mathrm{Var}_\mu[f].$$

Insbesondere ist $\widehat{\theta}_n$ eine *konsistente Schätzfolge* für θ, d.h. für jedes $\varepsilon > 0$ gilt

$$P\left[\left|\widehat{\theta}_n - \theta\right| \geq \varepsilon\right] \longrightarrow 0 \quad \text{für } n \to \infty.$$

[1] Als *Schätzer* bezeichnet man in der Statistik eine Funktion der gegebenen Daten (hier Stichproben von $X_1, \dots, X_n$), die zum Schätzen eines unbekannten Parameters verwendet wird.

Alternativ kann man statt der Čebyšev-Ungleichung auch exponentielle Abschätzungen verwenden, um den Schätzfehler zu kontrollieren. Dies demonstrieren wir im folgenden anhand der Monte-Carlo-Schätzung von Wahrscheinlichkeiten.

Bemerkung (Monte-Carlo-Schätzung von hochdimensionalen Integralen). Auch die Werte von mehrdimensionalen Integralen können mit Monte-Carlo-Verfahren näherungsweise berechnet werden. Dies ist besonders in hohen Dimensionen von Interesse, wo klassische numerische Verfahren in der Regel versagen. Soll beispielsweise der Wert des Integrals

$$\theta \coloneqq \int_{[0,1]^d} f(x)\,\mathrm{d}x \coloneqq \int_0^1 \dots \int_0^1 f(x_1,\dots,x_d)\,\mathrm{d}x_1 \dots \mathrm{d}x_d.$$

näherungsweise berechnet werden, dann können wir dazu Stichproben $u_1, u_2, \dots, u_{dn}$ von unabhängigen Zufallsvariablen $U_i \sim \mathrm{Unif}(0,1)$ simulieren. Die d-dimensionalen Zufallsvektoren $X^{(i)} \coloneqq (U_{di+1}, \dots, U_{d(i+1)})$, $i = 1, \dots, n$, sind dann unabhängig und gleichverteilt auf dem Produktraum $(0,1)^d$, siehe Einführung in die Wahrscheinlichkeitstheorie. Daher können wir den Wert θ des Integrals durch den Monte-Carlo-Schätzer

$$\hat{\theta}_n \coloneqq \frac{1}{n}\sum_{i=1}^n f(x^{(i)}) = \frac{1}{n}\sum_{i=1}^n f(u_1, u_2, \dots, u_{dn})$$

approximieren. Ist die Funktion f quadratintegrierbar, dann ergibt sich eine *dimensionsunabhängige* Abschätzung des mittleren quadratischen Fehlers, die nur von der Varianz von f bzgl. der Gleichverteilung auf dem Einheitswürfel $(0,1)^d$ abhängt. Da zum Erzeugen eines Stichprobenvektors $x^{(i)}$ d Zufallszahlen aus $(0,1)$ benötigt werden, beträgt der Aufwand $O(d)$, wenn ein vorgegebener mittlerer quadratischen Fehler für Funktionen mit Varianz kleiner gleich 1 unterschritten werden soll. Klassische numerische Integrationsverfahren haben dagegen in der Regel einen Aufwand, der exponentiell in der Dimension wächst.

Monte Carlo-Schätzung von Wahrscheinlichkeiten

Seien $X_1, X_2, \dots$ auf $(\Omega, \mathcal{A}, P)$ unabhängige Zufallsvariablen mit Verteilung μ. Wir betrachten nun den klassischen Monte-Carlo-Schätzer $\hat{p}_n$ für die Wahrscheinlichkeit $p = \mu[B] = E_\mu[I_B]$ eines Ereignisses $B \subseteq S$. Obere Schranken für den Schätzfehler können sowohl mithilfe der Čebyšev-Ungleichung als auch über die Bernstein-Ungleichung hergeleitet werden. Wir wollen die entsprechenden Schranken nun vergleichen.

Fehlerkontrolle mittels Čebyšev. Mit der Čebyšev-Ungleichung ergibt sich

$$P[|\widehat{p}_n - p| \geq \varepsilon] \leq \frac{1}{\varepsilon^2}\operatorname{Var}(\widehat{p}_n) = \frac{1}{n\varepsilon^2}\operatorname{Var}_\mu(I_B) = \frac{p(1-p)}{n\varepsilon^2} \leq \frac{1}{4n\varepsilon^2}.$$

Gilt beispielsweise $n \geq 5\varepsilon^{-2}$, dann erhalten wir

$$P[p \notin (\widehat{p}_n - \varepsilon, \widehat{p}_n + \varepsilon)] \leq 5\%, \qquad \text{unabhängig von } p,$$

d.h. das zufällige Intervall $(\widehat{p}_n - \varepsilon, \widehat{p}_n + \varepsilon)$ ist ein *95%-Konfidenzintervall* für den gesuchten Wert p.

Fehlerkontrolle mittels Bernstein. Mithilfe der Bernstein-Ungleichung erhalten wir für $\delta > 0$:

$$P\left[p \notin (\widehat{p}_n - \varepsilon, \widehat{p}_n + \varepsilon)\right] = P\left[\left|\frac{1}{n}\sum_{i=1}^n I_B(X_i) - p\right| \geq \varepsilon\right] \leq 2e^{-2n\varepsilon^2} \leq \delta, \quad \text{falls } n \geq \frac{\log(2/\delta)}{2\varepsilon^2}.$$

Für kleine δ ist die erhaltene Bedingung an n wesentlich schwächer als eine entsprechende Bedingung, die man durch Anwenden der Čebyšev-Ungleichung erhält.

Für kleine Werte von p ist in der Regel nicht der absolute, sondern der *relative Schätzfehler* $(\widehat{p}_n - p)/p$ von Interesse. Für diesen ergibt sich die Abschätzung

$$P\left[|\widehat{p}_n - p|/p \geq \varepsilon\right] \;=\; P\left[|\widehat{p}_n - p| \geq \varepsilon p\right] \;\leq\; 2e^{-2n\varepsilon^2 p^2} \;\leq\; \delta \quad \text{für } n \geq \frac{\log(2/\delta)}{2\varepsilon^2 p^2}.$$

Die benötigte Anzahl von Stichproben für eine (ε, δ)-Approximation von p ist also polynomiell in den Parametern ε, $\log(1/\delta)$ und $1/p$. Mit einer etwas modifizierten Abschätzung kann man die Ordnung $\Omega(1/p^2)$ noch auf $\Omega(1/p)$ verbessern, siehe **mitzenmacher2005probability** Trotzdem ist eine direkte Anwendung des einfachen Monte-Carlo-Verfahren für sehr kleine Wahrscheinlichkeiten nicht effektiv.

Varianzreduktion durch Importance Sampling

Häufig ist es sinnvoll, das klassische Monte-Carlo-Verfahren zu modifizieren, indem man zu einer anderen Referenzverteilung übergeht. Beispielsweise wechselt man bei der Monte-Carlo-Berechnung von Wahrscheinlichkeiten seltener Ereignisse zu einer Wahrscheinlichkeitsverteilung, bezüglich der das relevante Ereignis nicht mehr selten ist. Sei also ν eine weitere Wahrscheinlichkeitsverteilung auf S mit Maßenfunktion $\nu(x) = \nu[\{x\}]$. Es gelte $\nu(x) > 0$ für alle $x \in S$. Dann können wir einen unbekannten Erwartungswert $\theta = E_\mu[f]$ auch als Erwartungswert bzgl. ν ausdrücken:

$$\theta \;=\; E_\mu[f] \;=\; \sum_{x \in S} f(x)\,\mu(x) \;=\; \sum_{x \in S} f(x)\,\frac{\mu(x)}{\nu(x)}\nu(x) \;=\; E_\nu[f\,\varrho],$$

wobei

$$\varrho(x) = \frac{\mu(x)}{\nu(x)}$$

der Quotient der beiden Massenfunktionen ist. Ein alternativer Schätzer für θ ist daher durch

$$\widetilde{\theta}_n \;=\; \frac{1}{n}\sum_{i=1}^{n} f(Y_i)\,\varrho(Y_i)$$

gegeben, wobei $Y_1, \ldots, Y_n$ unabhängige Zufallsvariablen mit Verteilung ν sind. Auch $\widetilde{\theta}_n$ ist erwartungstreu, denn

$$E_\nu[\widetilde{\theta}_n] \;=\; E_\nu[f\,\varrho] \;=\; \theta.$$

Für die Varianz erhalten wir aufgrund der Unabhängigkeit

$$\mathrm{Var}_\nu[\widetilde{\theta}_n] \;=\; \frac{1}{n}\,\mathrm{Var}_\nu[f\,\varrho] \;=\; \frac{1}{n}\left(\sum_{x \in S} f(x)^2\varrho(x)^2\nu(x) - \theta^2\right).$$

Bei geeigneter Wahl der Referenzverteilung ν kann die Varianz von $\widetilde{\theta}_n$ deutlich kleiner sein als die des Schätzers $\widehat{\theta}_n$.

Aufgabe (Varianzminimierung bei Importance Sampling). Zeigen Sie, dass für einen endlichen Zustandsraum S die eindeutige Lösung des Variationsproblems

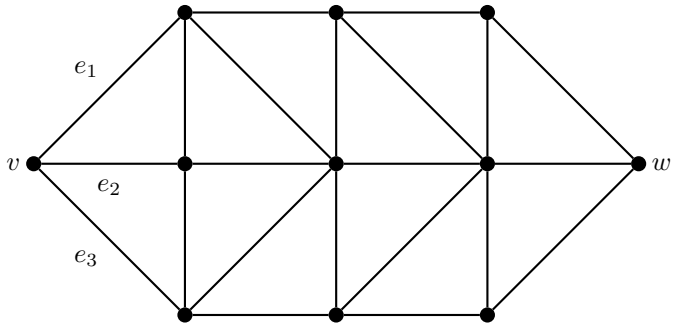
$$\sum_{x \in S} f(x)^2\varrho(x)^2\nu(x) \;\stackrel{!}{=}\; \min \qquad \text{unter der Nebenbedingung} \qquad \sum_{x \in S} \nu(x) = 1$$

auf $\mathbb{R}^S$ durch die Massenfunktion der Wahrscheinlichkeitsverteilung ν mit Gewichten

$$\nu(x) \;\propto\; |f(x)|\,\mu(x) \tag{4.13}$$

gegeben ist. Die Referenzverteilung ν mit minimaler Varianz des Importance-Sampling-Schätzers $\widetilde{\theta}_n$ ist also durch (4.13) bestimmt. In Anwendungen ist es meistens nicht möglich, Stichproben von dieser optimalen Referenzverteilung zu erzeugen. Das obige Ergebnis motiviert aber die Faustregel, dass für eine „gute“ Referenzverteilung die Gewichte $\nu(x)$ groß sein sollten, wenn $|f(x)|$ groß ist - daher auch der Name „Importance Sampling“.

Beispiel (Zuverlässigkeit von Netzwerken). Wir beschreiben ein Netzwerk (z.B. Stromleitungen) durch einen endlichen Graphen (V, E). Dabei stehen die Kanten für Verbindungen, die unabhängig voneinander



mit einer kleinen Wahrscheinlichkeit ε ausfallen. Seien nun $v, w \in E$ vorgegebene Knoten. Wir wollen die Wahrscheinlichkeit

$$p \;=\; P[\text{„}v \text{ nicht verbunden mit } w \text{ durch intakte Kanten“}]$$

approximativ berechnen. Sei dazu

$$S = \{0,1\}^E = \{(x_e)_{e\in E} : x_e \in \{0,1\}\}$$

die Menge der Konfigurationen von intakten ($x_e = 0$) bzw. defekten ($x_e = 1$) Kanten, und sei μ die Wahrscheinlichkeitsverteilung auf S mit Massenfunktion

$$\mu(x) \;=\; \varepsilon^{k(x)}(1-\varepsilon)^{|E|-k(x)},$$

wobei $k(x) = \sum_{e\in E} x_e$ die Anzahl der defekten Kanten ist. Dann ist $p = \mu[B]$ die Wahrscheinlichkeit des Ereignisses

$$B \;=\; \{x \in S : v, w \text{ nicht verbunden durch Kanten } e \text{ mit } x_e = 0\}.$$

Der „klassische“ Monte Carlo-Schätzer

$$\widehat{p}_n \;=\; \frac{1}{n}\sum_{i=1}^{n} I_B(X_i), \qquad X_i \text{ unabhängig mit Verteilung } \mu,$$

hat Varianz $p(1-p)/n$. Wir wollen den relativen Fehler $\sigma(\hat{p}_n)/p$ beschränken, wobei $\sigma(\hat{p}_n)$ die Standardabweichung bezeichnet. Fordern wir zum Beispiel

$$\sigma(\widehat{p}_n) \;=\; \sqrt{\frac{p(1-p)}{n}} \;\overset{!}{\leq}\; \frac{p}{10},$$

dann benötigen wir eine Stichprobenanzahl

$$n \geq \frac{100\,(1-p)}{p},$$

um diese Bedingung zu erfüllen. Für das in der Abbildung dargestellte (relativ kleine) Netzwerk mit Ausfallwahrscheinlichkeit $\varepsilon = 1\%$ können wir die Größenordnung von p folgendermaßen grob abschätzen:

$$10^{-6} = \mu[\text{„}e_1, e_2, e_3 \text{ versagen“}] \leq p \leq \mu(\text{„mindestens 3 Kanten versagen“}) = \binom{22}{3} \cdot 10^{-6} \approx 1,5 \cdot 10^{-3}.$$

Schon hier wird also eine sehr große Stichprobenanzahl benötigt, und für realistischere Netzwerke ist die Verwendung des klassischen Monte-Carlo-Schätzers nicht mehr praktikabel.

Um die benötigte Stichprobenanzahl zu reduzieren, wenden wir nun Importance Sampling an. Dazu wählen wir als Referenzverteilung die Wahrscheinlichkeitsverteilung ν auf S mit Gewichten

$$\nu(x) = t^{-k(x)}(1-t)^{|E|-k(x)}, \qquad k(x) = \sum_{e \in E} x_e,$$

die sich bei Ausfallwahrscheinlichkeit t ergibt. Im Netzwerk aus der Abbildung setzen wir beispielsweise $t :=\coloneqq 3/22$, so dass unter ν im Schnitt 3 Kanten defekt sind. Der Ausfall der Verbindung ist dann bezüglich der Verteilung ν kein seltenes Ereignis mehr. Für den Importance-Sampling-Schätzer

$$\widetilde{p}_n \;=\; \frac{1}{n}\sum_{i=1}^{n} I_B(Y_i)\,\frac{\mu(Y_i)}{\nu(Y_i)}, \qquad Y_i \text{ unabhängig mit Verteilung } \nu,$$

erhalten wir

$$\sigma(\widetilde{p}_n)^2 \;=\; \mathrm{Var}(\widetilde{p}_n) \;=\; \frac{1}{n}\left(\sum_{x \in S} I_B(x)^2 \frac{\mu(x)^2}{\nu(x)^2}\nu(x) - p^2\right).$$

Im Beispiel aus der Abbildung mit $\varepsilon = 0,01$ und $t = 3/22$ ergibt sich

$$\sigma(\widetilde{p}_n)^2 \;\le\; \frac{1}{n}\sum_{k=3}^{22}\binom{22}{k}\left(\frac{\varepsilon^2}{t}\right)^k\left(\frac{(1-\varepsilon)^2}{1-t}\right)^{22-k} \;\le\; 0,0053\,\frac{p}{n}.$$

Diese obere Schranke für die Varianz ist etwa um den Faktor 200 kleiner als die oben berechnete Varianz des einfachen Monte Carlo-Schätzers. Schon mit einem sehr einfachen Ansatz konnten wir also die Varianz, und damit die benötigte Stichprobenanzahl, deutlich reduzieren.

Der im Beispiel verwendete Ansatz, zu einer „kritischen“ Referenzverteilung überzugehen, bezüglich der die relevanten seltenen Ereignisse gerade eine nicht vernachlässigbare Wahrscheinlichkeit haben, ist typisch für den Einsatz von Importance Sampling auf praktische Problemstellungen. Die Hauptschwierigkeit ist dabei die geschickte Wahl der Referenzverteilung, siehe zum Beispiel **AsmussenGlynn**

Markov Chain Monte Carlo

Häufig ist es nicht möglich oder zu aufwändig, unabhängige Stichproben von der Zielverteilung μ oder einer geeigneten Referenzverteilung zu simulieren. In diesem Fall kann man eine Markovkette (X_n) mit Gleichgewicht μ verwenden, um approximative Stichproben zu erhalten. Nach den Resultaten in Abschnitt **??** konvergiert die Verteilung der Verteilung von X_n für $n \to \infty$ unter geeigneten Voraussetzungen in Variationsdistanz gegen μ, sodass wir die Werte $X_n(\omega)$ der Markovkette für $n \ge b$, b hinreichend groß, als approximative Stichproben verwenden können. Diese Stichproben sind jedoch nicht mehr unabhängig, sondern korreliert. Wenn die Kovarianzen schnell abklingen, können wir trotzdem das Gesetz der großen Zahlen anwenden, um Wahrscheinlichkeiten $p = \mu[B]$ und, allgemeiner, Erwartungswerte $\theta = E_\mu[f]$ bezüglich der Gleichgewichtsverteilung durch empirische Mittelwerte der Form

$$\widehat{p}_{n,b} = \frac{1}{n}\sum_{k=b+1}^{b+n} I_B(X_k), \qquad \text{bzw.} \qquad \widehat{\theta}_{n,b} = \frac{1}{n}\sum_{k=b+1}^{b+n} f(X_k)$$

zu approximieren, siehe zum Beispiel Satz **??** und das anschließende Korollar.

Die Analyse des Schätzfehlers ist bei Markov Chain Monte Carlo Verfahren im Allgemeinen diffizil. Aus den Resultaten und Beweisen in den Abschnitten **??** und **??** lassen sich erste Fehlerabschätzungen herleiten. Weitergehende Resultate finden sich zum Beispiel in **LevinPeresWilmer** Für viele Anwendungsprobleme sind jedoch keine brauchbaren Fehlerabschätzungen verfügbar, und man greift auf statistische Methoden zurück, um die Korrelationen abzuschätzen und zu testen, ob die Markovkette sich bereits dem Gleichgewicht angenähert hat. Da die statistischen Tests nicht immer zuverlässig sind, sind die Simulationsergebnisse dann mit entsprechender Vorsicht zu verwenden.

Listings

```
rperm[n_] :=
 Module[{x = Range[n], k, a}, (* Beginn mit Liste {1,2,...,n} *)
  Do[
   k = RandomInteger[{i, n}];
   a = x[[i]]; x[[i]] = x[[k]]; x[[k]] = a;  (* (Vertausche $x[[i]]$ und $x[[k]]$) *)
   , {i, n - 1}]; (* (Schleife, $i$ laeuft von $1$ bis $n-1$) *)
  x    (* (Ausgabe von $x$) *)];
rperm[17] {12, 5, 13, 8, 17, 9, 10, 6, 1, 7, 16, 15, 14, 4, 2, 3, 11}

f[x_] \coloneqq Mod[a x + c, m]
a = 11; c = 0; m = 63; pseudorandomdata = NestList[f, 1, 300]; ListPlot[pseudorandom
```