Perfekte Simulation unter Laufzeitbeschränkung

Fabian Niebling
Markus Schierle

Inhaltsverzeichnis

1 Markov-Ketten

Definition 1.1

- $(\Omega, \mathcal{F}, \mathbb{P})$ *sei Wkt.-Raum und* $E := \{1, ..., l\}$ *Zustandsraum* $(l \in \mathbb{N})$.
- $X_0, X_1, ... : \Omega \to E$ *heißt Markov-Kette* $:\Leftrightarrow$ $\mathbb{P}(X_n = i_n | X_{n-1} = i_{n-1}, ..., X_0 = i_0) = \mathbb{P}(X_n = i_n | X_{n-1} = i_{n-1}) \quad \forall n \in \mathbb{N}, \quad \forall i_k \in E, \quad k = 1, ...n$ *und falls* $\mathbb{P}(X_{n-1} = i_{n-1}, ..., X_0 = i_0) > 0$
- $P \in \mathbb{R}^{l \times l}$ *mit* $p_{ij} = \mathbb{P}(X_n = j | X_{n-1} = i)$ *heißt Übergangsmatrix der Markov-Kette* $(X_n)_{n \in \mathbf{N}}$

Theorem 1.2 (Konstruktion von Markov-Ketten)
Für jede Markov-Kette existiert eine rekursive Darstellung, d.h. die X_n *sind gegeben durch:*

$$X_0 \text{ bel.}$$
$$X_n = \Phi(X_{n-1}, Z_n) \quad \forall n \in \mathbb{N}$$

wobei $Z_1, Z_2, ... : \Omega \to D$ *iid. und* X_0 *unabhängig von* $Z_1, Z_2, ...$*.*

$\Phi : E \times D \to E$ *heißt Updatefunktion.*

Weiter gilt für die Übergangsmatrix $P = (p_{ij})$*:*

$$p_{ij} = \mathbb{P}(\Phi(i, Z_n) = j)$$

Beispiel 1.3 (Random Walk)

$Z_1, Z_2, ... : \Omega \to \{-1, 1\}$ *seien iid ZV. Setze* $X_0 := 0$*,*
$X_n := X_{n-1} + Z_n \quad \forall n \in \mathbb{N}$*. Theorem 1.2* $\Rightarrow$ (X_n) *ist eine MK mit ÜM*

P, wobei

$$p_{ij} = \mathbb{P}(\Phi(i, Z_n) = j) = \mathbb{P}(i + Z_1 = j) = \mathbb{P}(Z_1 = j - i)$$
$$= \begin{cases} 0.5 & j = i - 1 \\ 0.5 & j = i + 1 \\ 0 & \textit{sonst} \end{cases}$$

Definition 1.4
$P^{(n)} = (p_{ij})^{(n)} \in \mathbb{R}^{l \times l}$ *mit* $p_{ij}^{(n)} = \mathbb{P}(X_n = j | X_0 = i)$ *heißt die n-stufige Übergangsmatrix der MK* $(X_n)_{n \in \mathbf{N}_0}$.

Definition 1.5

- *Eine Markov-Kette* $(X_n)_{n \in \mathbf{N}_0}$ *heißt* **ergodisch** :$\Leftrightarrow$
  Die Grenzwerte $\pi_j = \lim\limits_n p_{ij}^{(n)} \quad \forall j \in E$ *existieren, sind positiv, hängen nicht von* $i \in E$ *ab und bilden eine Wahrscheinlichkeitsfunktion.*
- $\pi = (\pi_1, ..., \pi_l)^\top$ *heißt* **Grenzverteilung** *von* $(X_n)_{n \in \mathbf{N}_0}$.

Theorem 1.6
Sei $(X_n)_{n\in\mathbf{N}_0}$ *eine ergodische Markov-Kette.*
$\Rightarrow$ π *ist die Wahrscheinlichkeitsfunktion, die das LGS*

$$\pi^\top = \pi^\top P$$

eindeutig löst.

Später interessieren wir uns für MK $(\tilde{X}_n)_{n\in\mathbf{N}}$, bei denen die "Zeitachse" umgedreht wird, d.h. es gilt:

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_n = i_n) = \mathbb{P}(\tilde{X}_n = i_0, \tilde{X}_{n-1} = i_1, ..., \tilde{X}_0 = i_n)$$

Definition 1.7
Sei $(X_n)_{n\in\mathbf{N}_0}$ *eine ergodische MK mit ÜM* $P = (p_{ij})$ *und stationärer Grenzverteilung* $\pi = (\pi_1, ..., \pi_l)^\top$*. Dann heißt die (ergodische) MK* $(\tilde{X}_n)_{n\in\mathbf{N}}$ *mit ÜM* $\tilde{P} = (\tilde{p}_{ij})$ *gegeben durch*

$$\tilde{p}_{ij} = \frac{\pi_j}{\pi_i} p_{ji}$$

die **zeitinverse MK** *von* X*.*

Definition 1.8
Gilt $P = \tilde{P}$*, dann heißt die MK* $(X_n)_{n\in\mathbf{N}_0} = (\tilde{X}_n)_{n\in\mathbf{N}_0}$ *reversibel.*

Beispiel 1.9

Übergangsmatrix der MK (X_n)*:*

$$P = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \\ 0.5 & 0 & 0.5 \end{pmatrix}$$

$\Rightarrow \pi = (0.4, 0.2, 0.4)^\top$ *und von* $\tilde{X}_n$*:*

$$\tilde{P} = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix}$$

d.h. (X_n) *ist nicht reversibel.*

2 Markov-Chain-Monte-Carlo-Simulation (MCMC)

Motivation

Erzeugung synthetischer Daten als Ersatz für bzw. zur Ergänzung von experimentellen Daten.

Grundlage hierfür sind stochastische Simulationsalgorithmen, die sog. **Pseudozufallszahlen** generieren, welche einer gewünschten Verteilung folgen.

Bei der MCMC-Simulation nutzt man hierfür **Gleichgewichtszustände** der zugrundeliegenden Markov-Kette.

Prinzip

1. Konstruktion einer Realisierung einer Markov-Kette $X_0, X_1, ...$
   (a) mit Zustandsraum E
   (b) einer ergodischen Übergangsmatrix P
   (c) einer Grenzverteilung π.
2. Dann ist für großes n: $X_n \overset{d}{\sim} \pi$ näherungsweise!

Bemerkung 2.1
Man hat die Frage zu untersuchen, wie lange obige Simulation laufen muss, bis die Näherung in einem (noch) zu definierenden mathematischen Sinne gut genug ist.

Einen Ausweg bietet die sogenannte

perfekte Simulation

bei der die Ausgabe der Simulation genau nach π verteilt ist.

Grundlage des bekanntesten Vertreters der auf MCMC-Methoden basierenden Algorithmen ist die **Kopplung** von Markov-Ketten ("Coupling"), weshalb wir diesen Begriff präzisieren.

Theorem 2.2

- *Vor.:*
  - $\forall i \in E$ *sei eine ergodische Markov-Kette* $X^{(i)}$ *gegeben durch:*

$$X_n^{(i)} = \Phi(x_k, U_n^{(i)}), \text{ falls } X_{n-1}^{(i)} = x_k$$

    wobei $U^{(i)} = (U_1^{(i)}, U_2^{(i)}, ...)$ *eine Folge von unabhängigen und* $[0,1]$*-gleichverteilten ZVen ist.*
  - *Die Innovationenfolgen* $(U_k^{(i)})_{k \in N}$ *seien unabhängig* $\forall i \in E$.
- *Beh.:*
  - *Für die sog. Kopplungszeit* $\tau := min\{n \geq 1 : X_n^{(1)} = ... = X_n^{(l)}\}$ *gilt:*

$$P(\tau < \infty) = 1$$

3 Propp-Wilson, Coupling-From-The-Past (CFTP)

3.1 Wiederholung und Spezialfälle

Vor.:

ergodische MK mit Übergangsfunktion Φ, so dass $X_k = \Phi(X_{k-1}, U_k)$.

Motivation:

- Idee: Starte die MK im Zeitpunkt $-\infty$. Dann wäre sie zum Zeitpunkt 0 im Gleichgewichtszustand.
- Problem: "technisch" nicht machbar.
- Lösung: Coupling-From-The-Past

Algorithmus 3.1 (CFTP)

1. *Setze* $t := -T_1$ *und generiere* $U = (U_{-T_1+1}, ..., U_0)$*.*
2. *Starte eine Markov-Kette im Zeitpunkt* t *in allen Zuständen* $i \in E$ *und entwickle bis zum Zeitpunkt 0 unter Verwendung derselben Innovationenfolge U* $\forall i \in E$*:*

   $$X_n^{(i)} = \Phi(x_{n-1}^{(i)}, U_n)$$

   Die Pfade verschmelzen also, sobald sie sich einmal getroffen haben.
3. *Wenn sich alle Pfade zum Zeitpunkt 0 in* $x_0 \in E$ *getroffen haben: STOP. Ausgabe:* x_0*.*
4. *Setze* $t' := -T_2 < -T_1$*, generiere eine Innovationenfolge* $U' = (U_{-t'+1}, ..., U_t)$*, setze* $U := (U', U)$*,* $t := t'$ *und gehe zu 2.*

heuristische Begründung

Bei Terminierung befinden sich alle generierten Ketten des obigen Algorithmus' zum Zeitpunkt 0 im gleichen Zustand $x_0 \in E$. Benutzt eine in $-\infty$ gestartete Kette von $-T_i$ bis 0 die gleichen Innovationen, würde sie zum Zeitpunkt 0 auch x_0 realisieren. Also ist x_0 virtuelles Ergebnis einer in $-\infty$ beginnenden Simulation.

Problem: $|E|$ groß:

- sehr viele Pfade müssen generiert werden
- immenser Rechen- und Speicheraufwand
- Implementierung des allgemeinen Falls nicht möglich

Ausweg: Implementierung nur unter Ausnutzung von Spezialfällen.

wichtigster Spezialfall: Monotonie.

- auf E ist eine Halbordnung $\preceq$ definiert mit Minimalelement $\hat{0}$ und Maximalelement $\hat{1}$.
- Die Updatefunktion $\Phi : E \times [0,1] \to E$ ist monoton bzgl. "$\preceq$", d.h. $\Phi(x,\cdot) \preceq \Phi(y,\cdot)$ für $x \preceq y$
- $\forall i \in E$ wird die gleiche Innovationenfolge benutzt.

Dann reicht es aus, die Kette in den Zuständen $\hat{0}$ und $\hat{1}$ zu starten. Verschmelzen diese beiden Pfade, so auch alle anderen, da diese (wegen Monotonie und Verwendung derselben Innovationenfolge für alle Zustände) dazwischen verlaufen.

3.2 Abhängigkeit des Outputs von der Laufzeit

Betrachte die Übergangsmatrix $P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$ mit stationärer Grenzverteilung $\pi = (\frac{2}{5}, \frac{1}{5}, \frac{2}{5})$.

Wir simulieren mit folgender Updatefunktion:

x=0,2:

$$\Phi(x,U) = \begin{cases} 0 & \text{falls } U \leq \frac{1}{2} \\ min\{x+1,2\} & \text{sonst} \end{cases}$$

x=1:

$$\Phi(1,U) = 2$$

wobei $U \sim \mathcal{U}[0,1]$. Für die Länge der i-ten Iteration gelte $L(i) = 2^{i-1}$.

Beobachtungen

- Verschmelzung kann nicht im ersten Schritt auftreten.
- Danach sind die möglichen Zustände $A = \{0, 2\}$ oder $B = \{1, 2\}$
- In beiden Fällen ist die Verschmelzungswahrscheinlichkeit im nächsten Schritt $\frac{1}{2}$.

$$\begin{aligned}
\mathbb{P}(Z = r|I) &= \frac{\mathbb{P}((\text{Verschmelzung innerhalb I Iterationen}) \wedge (Z = r))}{\mathbb{P}(\text{Verschmelzung innerhalb I Iterationen})} \\
&= \frac{\sum_{k=2}^{N} (1 - \frac{1}{2})^{k-1} \cdot \frac{1}{2}[P^{N-k}(0, r) + P^{N-k}(2, r)]}{1 - (\frac{1}{2})^{N-1}}
\end{aligned}$$

Für $I = 2$ erhalten wir: $\mathbb{P}(Z = 0|I = 2) = \frac{1}{2} \quad \mathbb{P}(Z = 2|I = 2) = \frac{1}{2}$. Für $I \geq 3$ ergibt sich:

$$\begin{aligned}
\mathbb{P}(Z = 0|I) &= \frac{2}{5}\left[\frac{1 - 2^{-N}}{1 - 2^{-N+1}}\right] \\
\mathbb{P}(Z = 1|I) &= \frac{1}{5}\left[\frac{1 - 2^{-N}}{1 - 2^{-N+1}}\right] \\
\mathbb{P}(Z = 2|I) &= \frac{2}{5}\left[\frac{1 - 7 \cdot 2^{-N-1}}{1 - 2^{-N+1}}\right]
\end{aligned}$$

4 Der Fill-Algorithmus

4.1 Einführung

Voraussetzungen

- Auf dem Grundraum E gebe es eine Halbordnung $\preceq$ und eindeutig bestimmte Elemente $\hat{0}$ und $\hat{1}$, so dass $\hat{0} \preceq i \preceq \hat{1} \quad \forall i \in E$.
- $\tilde{P}$ sei stochastisch monoton, d.h.

$$x \preceq y \Rightarrow \tilde{P}(x, \cdot) \preceq_{st} \tilde{P}(y, \cdot)$$

- Es gebe eine monotone Übergangsfunktion $\tilde{\Phi}$ für $\tilde{P}$, d.h.

$$x \preceq y \Rightarrow \tilde{\Phi}(x, \cdot) \preceq \tilde{\Phi}(y, \cdot)$$

Algorithmus 4.1 (Der Algorithmus von Fill für $\tilde{P}$ monoton)

1. *Erzeuge eine Realisierung einer Markov-Kette* $X = (X_0, ..., X_t)$ *mit* $X_0 := \hat{0}$ *gemäß der Vorschrift* $X_k = \Phi(X_{k-1}, U_k)$, U_k *iid.*
2. *Bezeichne* $\tilde{X} := (\tilde{X}_0, ..., \tilde{X}_t) = (X_t, ..., X_0)$ *die zeitinverse Markov-Kette von X mit monotoner Übergangsmatrix* $\tilde{P}$,
   $\tilde{X}_k = \tilde{\Phi}(\tilde{X}_{k-1}, U'_k)$, U'_k *iid.*
   "Gekoppelt" an diesen Pfad $\tilde{X}$ *erzeuge eine neue Markov-Kette* $\tilde{Y} := (\tilde{Y}_0, ..., \tilde{Y}_t)$ *mit* $\tilde{Y}_0 := \hat{1}$ *wie folgt:*

(a) *Generiere eine Folge von bedingt unabhängigen ZVen* $\hat{U}_k$ *gemäß der Verteilung*

$$Q(\tilde{X}_k, \tilde{X}_{k+1})(\cdot) = \mathbb{P}(U'_k \in \cdot | \tilde{\Phi}(\tilde{X}_k, U'_k) = \tilde{X}_{k+1}), \quad k = 0, ..., t-1$$

(b) *Setze*

$$\tilde{Y}_k = \tilde{\Phi}(\tilde{Y}_{k-1}, \hat{U}_k), \quad k = 1, ..., t$$

3. *Ist* $\tilde{Y}_t = \hat{0}$*: STOP. Ausgabe:* $z = X_t$*.*

4. *Sonst: Lösche alle Informationen und starte eine neue Iteration mit t:=2t (z.B.)*

4.2 Begründung

Lemma 4.2 (Verwerfungsmethode)

- *Vor.:*
  - *Es seien π, ν zwei Wkt.-Maße auf demselben Grundraum E.*
  - $\exists\, c > 0 : \pi(i) \leq c \cdot \nu(i) \quad \forall i \in E$
- *Beh.:*
  Folgender Algorithmus terminiert in endlicher Zeit und liefert eine nach π verteilte Stichprobe:
  1. *Generiere eine Stichprobe x von $X \sim \nu$.*
  2. *Sei $x = i$.*
     Akzeptiere diesen Wert mit Wahrscheinlichkeit $\frac{\pi(i)}{c \cdot \nu(i)}$. STOP.
  3. *Falls der Wert verworfen wird, gehe zurück zu 1.*

Beweis

Es gilt: $\mathbb{P}(X = i, X \text{ wird akzeptiert}) = \nu(i) \cdot \frac{\pi(i)}{c \cdot \nu(i)} = \frac{\pi(i)}{c}$

und

$$\begin{aligned}
\mathbb{P}(X \text{ wird akzeptiert}) &\overset{U\sim\mathcal{U}(0,1]}{=} \mathbb{P}(U < \frac{\pi(X)}{c \cdot \nu(X)}) \\
&\overset{bed.Wkt.}{=} \sum_{i:\nu(i)>0} \mathbb{P}(U < \frac{\pi(X)}{c \cdot \nu(X)} | X = i) \cdot \mathbb{P}(X = i) \\
&= \sum_{i:\nu(i)>0} \mathbb{P}(U < \frac{\pi(i)}{c \cdot \nu(i)}) \cdot \nu(i)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i:\nu(i)>0} \frac{\pi(i)}{c \cdot \nu(i)} \cdot \nu(i) \\
&= \frac{1}{c} > 0
\end{aligned}
$$

$\Rightarrow$ Algorithmus terminiert.

weiter gilt:

$$
\mathbb{P}(X = i | X \text{ wird akzeptiert}) = \frac{\mathbb{P}(X = i, X \text{ wird akz.})}{\mathbb{P}(X \text{ wird akzeptiert})} = \frac{\frac{\pi(i)}{c}}{\frac{1}{c}} = \pi(i)
$$

$\Rightarrow$ Behauptung.

- Eine Realisierung von $X_t \overset{d}{\sim} P^t(\hat{0}, \cdot)$ wird benutzt, um eine Realisierung von π zu erhalten. (1. Phase)

- Zur Entscheidung, ob X_t gemäß π verteilt ist, wird die Verwerfungsmethode benutzt. (2. Phase). Wir suchen eine obere Schranke für $\frac{\pi(z)}{P^t(\hat{0},z)} \quad \forall z \in E$.
  Denn: $\frac{\pi(z)}{P^t(\hat{0},z)} \leq c \Leftrightarrow \pi(z) \leq c \cdot P^t(\hat{0}, z)$.
  Aus der Definition der zeitinversen Markov-Kette erhalten wir:

$$\tilde{P}^t(z, \hat{0}) = \frac{\pi(\hat{0})}{\pi(z)} \cdot P^t(\hat{0}, z) \Leftrightarrow \frac{\pi(\hat{0})}{\tilde{P}^t(z, \hat{0})} = \frac{\pi(z)}{P^t(\hat{0}, z)}$$

  und wegen der Monotonie von $\tilde{P}$ können wir $c = \dfrac{\pi(\hat{0})}{\tilde{P}^t(\hat{1}, \hat{0})}$ setzen.

- Gemäß der Verwerfungsmethode akzeptieren wir eine Realisierung von $P^t(\hat{0},\cdot)$ mit der Wahrscheinlichkeit

$$\begin{aligned}\frac{\pi(z)}{c\cdot P^t(\hat{0},z)} &= \frac{\pi(z)}{\frac{\pi(\hat{0})}{\tilde{P}^t(\hat{1},\hat{0})}\cdot P^t(\hat{0},z)}\\ &= \frac{\pi(z)\tilde{P}^t(\hat{1},\hat{0})\pi(\hat{0})}{\pi(\hat{0})\tilde{P}^t(z,\hat{0})\pi(z)}\\ &= \frac{\tilde{P}^t(\hat{1},\hat{0})}{\tilde{P}^t(z,\hat{0})}\end{aligned}$$

Theorem 4.3

$$\mathbb{P}(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{X}_t = \hat{0}, \tilde{Y}_0 = \hat{1}) = \frac{\tilde{P}^t(\hat{1}, \hat{0})}{\tilde{P}^t(z, \hat{0})}$$

Beweis

Durch das Design der MK $\tilde{Y}$ gilt: $\tilde{X}_s \preceq \tilde{Y}_s \quad \forall s \in \{0, ..., t\}$,
insbesondere also $\tilde{Y}_t = \hat{0} \Rightarrow \tilde{X}_t = \hat{0}$.
Denn induktiv ergibt sich:
(IA) $\tilde{X}_0 = X_t = z \preceq \hat{1} = \tilde{Y}_0$
(IS) $\tilde{X}_s = \tilde{\Phi}(\tilde{X}_{s-1}, U'_s) \preceq \tilde{\Phi}(\tilde{Y}_{s-1}, U'_s) \preceq \tilde{\Phi}(\tilde{Y}_{s-1}, \hat{U}_s) = \tilde{Y}_s$

Desweiteren betrachten wir die gemeinsame Verteilung der Ketten $\tilde{Y}$ und $\tilde{X}$:

$$\begin{aligned}
& \mathbb{P}(\tilde{Y} = y, \tilde{X} = x) \\
= \quad & \prod_{s=1}^{k} \tilde{P}(x_{s-1}, x_s)\mathbb{P}(\tilde{Y}_s = y_s | \tilde{Y}_{s-1} = y_{s-1}, \tilde{X}_s = x_s, \tilde{X}_{s-1} = x_{s-1}) \\
= \quad & \prod_{s=1}^{k} \mathbb{P}(\tilde{\Phi}(x_{s-1}, U'_s) = x_s)\mathbb{P}(\tilde{\Phi}(y_{s-1}, \hat{U}_s) = y_s | \tilde{\Phi}(x_{s-1}, \hat{U}_s) = x_s) \\
= \quad & \prod_{s=1}^{k} \mathbb{P}(\tilde{\Phi}(y_{s-1}, \hat{U}_s) = y_s) \\
= \quad & \mathbb{P}(\tilde{Y} = y)
\end{aligned}$$

Mit diesen zwei Teilergebnissen können wir berechnen:

$$\begin{aligned}
\mathbb{P}(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{X}_t = \hat{0}, \tilde{Y}_0 = \hat{1}) &= \frac{\mathbb{P}(\tilde{Y}_t = \hat{0}, \tilde{X}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1})}{\mathbb{P}(\tilde{X}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1})} \\
= \frac{\mathbb{P}(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1})}{\tilde{P}^t(z, \hat{0})} &= \frac{\mathbb{P}(\tilde{Y}_t = \hat{0}, \tilde{X}_0 = z | \tilde{Y}_0 = \hat{1})}{\tilde{P}^t(z, \hat{0}) \cdot \mathbb{P}(\tilde{X}_0 = z, \tilde{Y}_0 = \hat{1})} \\
&= \frac{\tilde{P}^t(\hat{1}, \hat{0})}{\tilde{P}^t(z, \hat{0})}
\end{aligned}$$

woraus die Behauptung folgt.

Bemerkung 4.4

1. *Aus Theorem 4.2 folgt*

$$\mathbb{P}(X_t \textit{ wird akzeptiert}) = \frac{1}{c} = \frac{\tilde{P}^t(\hat{1},\hat{0})}{\pi(\hat{0})} = \frac{P^t(\hat{0},\hat{1})}{\pi(\hat{1})} \longrightarrow 1 \quad (t \to \infty)$$

2. *Die Aussage, dass der Fill-Algorithmus durch einen ungeduldigen Benutzer unbeeindruckt bleibt, folgt aus der Natur der Verwerfungsmethode.*

5 Beispiel: Das Ising-Modell

5.1 Modellbeschreibung

- Betrachten ein Gitter $G = (V, K)$ mit
  - $V = \{v_1, ..., v_{|V|}\}$ Menge der Eckpunkte (endlich)
  - $K \subset V^2$ Menge der Kanten, die jeweils zwei Eckpunkte miteinander verbinden.
  - $K \ni e = (v_i, v_j)$ verbindet die Eckpunkte $v_i, v_j \in V$.
- Jedem Eckpunkt wird der Wert -1 oder 1 zugeordnet.
  $\Rightarrow$ Zustandsraum $E = \{-1, 1\}^{|V|}$ "Menge aller Konfigurationen".
  $x = (x(v), v \in V)$ mit $x(v) \in \{-1, 1\} \quad \forall v \in V$.
- Zur grafischen Veranschaulichung interpretiert man
  - weißes Pixel: $x(v) = -1$
  - schwarzes Pixel: $x(v) = 1$

- Man betrachtet die Wahrscheinlichkeit π_x, dass die Konfiguration $x \in E$ angenommen wird.
  π ist gegeben durch

$$\pi_x := \frac{1}{Z_{G,J}} exp\left(J \sum_{e=(v_i,v_j)\in K} x(v_i)x(v_j)\right)$$

  wobei
  - $J \geq 0$ Parameter
  - $Z_{G,J} := \sum_{x\in E} exp\left(-J \sum_{e=(v_i,v_j)\in K} x(v_i)x(v_j)\right)$
    Normierungskonstante

Algorithmus 5.1 *Gibbs-Sampler*

1. *Wähle eine Komponente* $v \in V$ *aus (nach Gleichverteilung).*
2. *Ändere* $x(v)$ *gemäß der bedingten Verteilung* $\pi_{x(v)|x(-v)}$
   $x(-v) := (x(w), w \in V \setminus \{v\})$.
3. *Übernehme alle anderen Komponenten.*

Man erhält dann die Übergangsmatrix $P = (p_{xx'})$ von X gegeben durch

$$p_{xx'} = \sum_{v \in V} \frac{1}{|V|} \pi_{x'(v)|x(-v)} \mathbb{1}_{(x(-v)=x'(-v))} \quad \forall x, x' \in E$$

mit der bedingten Verteilung

$$\pi_{x'(v)|x(-v)} = \begin{cases} \frac{1}{1+exp(-2J(K_+(x(-v))-K_-(x(-v))))} & \text{falls } x'(v) = 1 \\ \frac{1}{1+exp(-2J(K_-(x(-v))-K_+(x(-v))))} & \text{falls } x'(v) = -1 \end{cases}$$

wobei $K_+(x(-v))$ bzw. $K_-(x(-v))$ die Anzahl derjenigen mit v verbundenen Eckpunkte ist, die auf 1 bzw. -1 gesetzt sind.

Eine Updatefunktion $\Phi : E \times (0,1]^2 \to E$ ist gegeben durch

$$\Phi(x; U_1, U_2) = x', \quad x' = (x'(v), v \in V)$$

wobei

$$x'(v_i) = \begin{cases} 1, & \text{falls } \frac{i-1}{|V|} < U_1 \leq \frac{i}{|V|} \text{ und } U_2 < \pi_{1|x(-v_i)} \\ -1, & \text{falls } \frac{i-1}{|V|} < U_1 \leq \frac{i}{|V|} \text{ und } U_2 \geq \pi_{1|x(-v_i)} \\ x(v_i) & \text{sonst} \end{cases}$$

Für diese Updatefunktion gilt

$$\Phi(x; U_1, U_2) \preceq \Phi(y; U_1, U_2) \quad \forall x \preceq y, \quad \forall U_1, U_2 \in (0,1]$$

d.h. sie ist monoton.

5.2 Implementierung mit CFTP

Algorithmus 5.2 *CFTP für das Ising-Modell*

1. *Wähle T und setze* $X_-T = \hat{0}, Y_-T = \hat{1}$.
2. *Generiere auf* $(0,1]$ *gleichverteilte Pseudozufallszahlen* $U_1^{(1)}, ..., U_1^{(T)}$ *und* $U_2^{(1)}, ..., U_2^{(T)}$.
   Berechne $X_{-T}, ..., X_0$ *und* $Y_{-T}, ..., Y_0$ *mit Hilfe von* Φ.
3. *Falls* $X_0 = Y_0$*: STOP.*
   Falls $X_0 \neq Y_0$*: Erhöhe T und gehe zu Schritt 1.*

5.3 Implementierung mit Fill

- Bezeichne $\tilde{X}$ die zeitinverse MK von X.
- Betrachte den Übergang vom Zustand x in den Zustand x', d.h. $x' = \tilde{\Phi}(x, U_1', U_2')$

Herleitung von Regeln für den Übergang von y nach y':

- Die jeweils für die Wahl des zu ändernden Pixels verantwortliche Innovation U_1 kann für $\tilde{Y}$ übernommen werden: $\hat{U}_1 = U_1$.
- Bei der Art der Änderung sind drei Fälle zu unterscheiden:

1. $x' \neq x$ **und** $x'(v_i) = 1$ **wurde geändert:**
   wegen $\tilde{X}_t \preceq \tilde{Y}_t \quad \forall t$, setzt man $y'(v_i) = 1$.
2. $x' \neq x$ **und** $x'(v_i) = -1$ **wurde geändert:**
   wegen $U_2 \leq \pi_{-1|x(-v)} \Rightarrow \hat{U}_2 \sim \mathcal{U}(0, \pi_{-1|x(-v_i)})$.
   Erzeuge hierzu $U_2^* \sim \mathcal{U}[0,1]$ und setze $\hat{U}_2 = U_2^* \cdot \pi_{-1|x(-v_i)}$.
   Übergangsregel:

$$y'(v_i) = \begin{cases} -1, & \text{falls } U_2^* \leq \frac{\pi_{-1|y(-v_i)}}{\pi_{-1|x(-v_i)}} \\ 1, & \text{sonst} \end{cases}$$

3. $x' = x$
   Setze $\hat{U}_1 = U_1$ und $\hat{U}_2 = U_2$. Denn:

   Betrachte den Fall $U_1 = i, U_2 \leq \pi_{-1|x(-v_i)}$, wir erhalten dann folgende bedingte Verteilung für (U_1, U_2):

$$\begin{aligned} \Rightarrow (U_1, U_2) &\sim F(U_1, U_2 | \tilde{\Phi}(U_1, U_2, x) = x) = F(i, U_2 | \tilde{\Phi}(i, U_2, x) = x) \\ &= F(i, U_2 | U_2 \leq \pi_{-1|x(-v_i)}) = F(i, U_2 \cdot \pi_{-1|x(-v_i)}) \end{aligned}$$

   Andererseits

$$\begin{aligned} (\hat{U}_1, \hat{U}_2) &\sim F(U_1', U_2' | \tilde{\Phi}(U_1', U_2', x) = x) \\ &= F(i, U_2' \cdot \pi_{-1|x(-v_i)}) \end{aligned}$$

   Für $U_2 \stackrel{d}{=} U_2'$ folgt die Behauptung.

Insgesamt erhält man also folgende Update-Regel für $\tilde{Y}$:

$$y'(v_i) = \begin{cases} 1, & \text{falls } x' \neq x \text{ und } x'(v_i) = 1 \text{ geändert} \\ -1, & \text{falls } x' \neq x \text{ und } x'(v_i) = -1 \text{ geändert und } U_2^* \leq \frac{\pi_{-1|y(-v_i)}}{\pi_{-1|x(-v_i)}} \\ 1, & \text{falls } x' \neq x \text{ und } x'(v_i) = -1 \text{ geändert und } U_2^* > \frac{\pi_{-1|y(-v_i)}}{\pi_{-1|x(-v_i)}} \\ y(v_i), & \text{falls } x' \neq x \text{ und } x'(v_i) \text{ nicht geändert} \\ 1, & \text{falls } x' = x \text{ und } \frac{i-1}{|V|} < U_1 \leq \frac{i}{|V|} \text{ und } U_2 \leq \pi_{1|x(-v_i)} \\ -1, & \text{falls } x' = x \text{ und } \frac{i-1}{|V|} < U_1 \leq \frac{i}{|V|} \text{ und } U_2 > \pi_{1|x(-v_i)} \\ y(v_i), & \text{falls } x' = x \text{ und } (U_1 \leq \frac{i-1}{|V|} \text{ oder } U_1 > \frac{i}{|V|}) \end{cases}$$

Algorithmus 5.3 (Fill-Algorithmus für das Ising-Modell)

1. *Wähle T und setze* $X_0 = \hat{0}, \tilde{Y}_0 = \hat{1}$*.*
2. *Generiere auf* $(0,1]$ *gleichverteilte Pseudozufallszahlen* $U_1^{(1)}, ...U_1^{(T)}$ *und* $U_2^{(1)}, ...U_2^{(T)}$*. Berechne* $X_0, ..., X_T$ *mit Hilfe von* Φ *und anschließend die zeitinverse MK* $\tilde{X}$*.*
3. *Berechne* $\tilde{Y}_0, ..., \tilde{Y}_T$ *mit Hilfe eben genannter Regel.*
4. *Falls* $\tilde{Y}_T = \hat{0}$*, akzeptiere* $X_T = z$*. STOP.*
   Falls $\tilde{Y}_T \neq \hat{0}$*, erhöhe T und gehe zu Schritt 2.*

6 Vergleich

	CFTP	**Fill**
Idee	Rückwärtskopplung	Verwerfungsmethode
Anwendbarkeit	erfordert monotone Übergangsfunktion	nicht notwendigerweise monotone Übergangsfunktion, aber aufwendigere Generierung der Ketten
Erweiterbarkeit	stetiger Zustandsraum und inhomogene MK	dto.
Performance	benötigt weniger Speicher	benötigt weniger Iterationen

Literatur

[1] Fill, James A., *An Interruptible Algorithm For Perfect Sampling Via Markov-Chains*, The Johns Hopkins University, Baltimore, 1998

[2] Fismen, Morten, *Exact Simulation using Markov-Chains*, The Norwegian University Of Technology And Science, Trondheim, 1997

[3] Huber, Marc, *Perfect Sampling Without A Lifetime Commitment*, Cornell University, Ithaca

[4] König, Wolfgang, *Vorlesungsskript zu Stochastische Algorithmen*, Universität Köln, 2003

[5] Møller, Jesper und Schladitz, Katja, *Extensions Of Fill's Algorithm For Perfect Simulation*, Aalborg University, 1998

[6] Schmidt, Volker, *Vorlesungsskript zu Markov-Ketten und Monte-Carlo-Simulation*, Universität Ulm, 2003

[7] Thönnes, Elke, *Perfect Simulation Of Some Point Processes For The Impatient User*, University Of Warwick, Coventry, 1998

Vielen Dank für die Aufmerksamkeit!