Seminarvortrag
Der Propp-Wilson Algorithmus und die Wilson Modifikation

Daniela Duhme

03. Mai 2012

Inhaltsverzeichnis

10 Der Propp-Wilson Algorithmus

10.1 Motivation

Die Grundidee der Markov-Chain-Monte-Carlo Methode ist folgende: Wir erzeugen eine irreduzible und aperiodische Markov-Kette $(X_0, X_1, ...)$, wobei X_n gemäß $\mu^{(n)}$ verteilt ist. Die stationäre Verteilung sei π, sodass $\lim_{n\to\infty} \mu^{(n)} = \pi$. Wie wir aus vorherigen Vorträgen wissen, gibt es hierbei zwei grundlegende Probleme:

1. Wir wissen nur, dass die Verteilung $\mu^{(n)}$ von X_n gegen π konvergiert. Das garantiert aber nicht, dass $\mu^{(n)}$ jemals gleich π sein wird. Oft gilt sogar $\mu^{(n)} \neq \pi$ für alle n. Dann gibt es immer eine Abweichung zwischen der Verteilung $\mu^{(n)}$, die wir durch die MCMC Methode erhalten haben, und der gewünschten Verteilung π.

2. n muss einerseits so groß gewählt werden, dass der Approximationsfehler gering ist, andererseits aber auch so klein, dass der Aufwand für den praktischen Gebrauch vertretbar bleibt.

Diese Probleme werden durch den Propp-Wilson Algorithmus gelöst. Dieser liefert erst dann eine Stichprobenrealisierung, wenn die stationäre Verteilung π erreicht ist. Außerdem enthält er ein automatisches Abbruchkriterium. Im Folgenden wollen wir die Funktionsweise des Algorithmus betrachten und beweisen, dass er unverzerrte Stichproben liefert.

10.2 Funktionsweise des Propp-Wilson Algorithmus

Was wir wollen: Eine Stichprobe gemäß einer Wahrscheinlichkeitsverteilung π aus einem endlichen Zustandsraum $S = \{s_1, ..., s_k\}$

Was wir für den Algorithmus brauchen:

- Die Übergangsmatrix P und eine gültige Updatefunktion $\phi : S \times [0,1] \to S$
- Eine ansteigende Folge natürlicher Zahlen (ohne Null) $(N_1, N_2, ...)$, meist $(1, 2, 4, 8, ...)$
- Eine Folge $(U_0, U_{-1}, U_{-2}, ...)$ von gleichverteilten Zufallszahlen auf [0,1]

So funktioniert der Propp-Wilson Algorithmus:

1. Setze m=1

2. Simuliere für alle s $\in \{s_1, ..., s_k\}$ eine Markov-Kette, die zum Zeitpunkt $-N_m$[1] im Zustand s beginnt und bis zum Zeitpunkt 0 läuft. Verwende hierzu die Updatefunktion ϕ und die Zufallszahlen $U_{-N_m+1}, U_{-N_m+2}, ..., U_{-1}, U_0$

3.
   - Falls alle Ketten im selben Zustand s* zum Zeitpunkt 0 enden: Gebe s* aus und stoppe den Algoithmus.

   - Falls nicht: Erhöhe m um 1 und fahre fort mit Schritt 2.

Wichtig: Einmal generierte Zufallszahlen U_i werden für folgende Schritte immer wiederverwendet! (Sonst erhält man eine verzerrte Stichprobe $\rightarrow$ später).

Das Laufenlassen der Markov-Ketten von einem Zeitpunkt $-N_m$ in der Vergangenheit bis zum Zeitpunkt 0 nennt man auch **Coupling from the Past**.

10.2.1 Beispiel

Die Abbildungen zeigen einen Propp-Wilson Algorithmus mit Zustandsraum $S = \{s_1, s_2, s_3\}$ und $(N_1, N_2, N_3) = (1, 2, 4)$. *(Die Bilder sind nicht Teil des Seminarvortrags. Stattdessen wird an dieser Stelle Folie 1 der „ Animation PWA und WA.pptx"gezeigt.)*

[1] Die Kette beginnt sozusagen in der Vergangenheit zum Zeitpunkt $-N_m$ und läuft bis heute, Zeitpunkt 0.

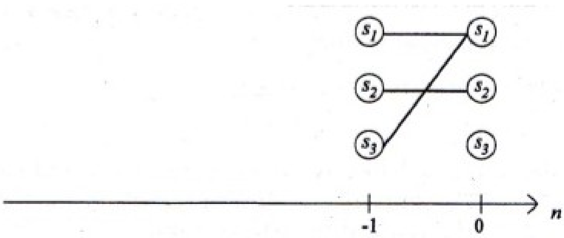


Abbildung 1: Propp-Wilson Algorithmus - Erster Durchlauf

$$\begin{cases} \phi(s_1, U_0) = s_1 \\ \phi(s_2, U_0) = s_2 \\ \phi(s_3, U_0) = s_1 \end{cases}$$

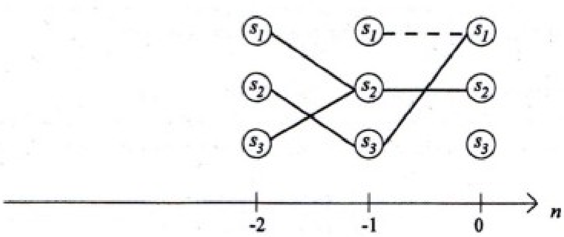


Abbildung 2: Propp-Wilson Algorithmus - Zweiter Durchlauf

$$\begin{cases} \phi(\phi(s_1, \overbrace{U_{-1}}^{\text{neue ZV}}), \overbrace{U_0}^{\text{alte ZV}}) = \phi(s_2, U_0) = s_2 \\ \phi(\phi(s_2, U_{-1}), U_0) = \phi(s_3, U_0) = s_1 \\ \phi(\phi(s_3, U_{-1}), U_0) = \phi(s_2, U_0) = s_2 \end{cases}$$

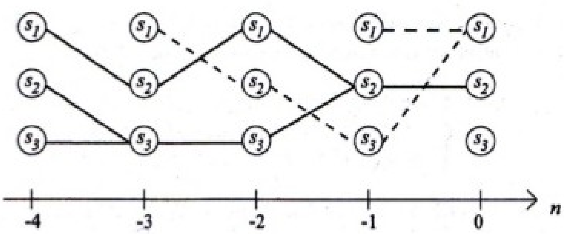


Abbildung 3: Propp-Wilson Algorithmus - Dritter Durchlauf

$$\begin{cases} \phi(\phi(\phi(\phi(s_1, \overbrace{U_{-3}}^{\text{neue ZV}}), \overbrace{U_{-2}}^{\text{neue ZV}}), \overbrace{U_{-1}}^{\text{alte ZV}}), \overbrace{U_0}^{\text{alte ZV}}) = ... = s_2 \\ \phi(\phi(\phi(\phi(s_2, U_{-3}), U_{-2}), U_{-1}), U_0) = ... = s_2 \\ \phi(\phi(\phi(\phi(s_3, U_{-3}), U_{-2}), U_{-1}), U_0) = ... = s_2 \end{cases}$$

Der dritte Neustart des Propp-Wilson Algorithmus war erfolgreich, alle Ketten befinden sich zum Zeitpunkt 0 im Zustand s_2. Somit lautet unsere Stichprobe gemäß π s_2.

Bemerkung: Wenn wir den Algorithmus nach Zusammenlaufen der Ketten weiter laufen lassen, ändert sich an dem Ergebnis nichts mehr, der Output bleibt s* (s*= s_2 im Beispiel 10.2.1).

10.3 Warum der Propp-Wilson Algorithmus funktioniert

Zunächst muss erwähnt werden, dass der Propp-Wilson Algorithmus keinesfalls immer zu einem Ergebnis kommt. Es besteht die Möglichkeit, dass die Markov-Ketten immer weiter in die Vergangenheit zurückgesetzt werden und sie trotzdem nie zusammenlaufen. Somit entsteht also eine unendliche Schleife. Um dieses Problem zu umgehen, werden wir im Folgenden einfach das Abbrechen des Algorithmus voraussetzen.

10.3.1 Theorem

Sei

- P die Übergangsmatrix einer irreduziblen und aperiodischen Markov-Kette
- $S = \{s_1, ..., s_k\}$ der Zustandsraum
- $\pi = (\pi_1, ..., \pi_k)$ die stationäre Verteilung
- ϕ eine gültige Updatefunktion für P
- Y der Output des Algorithmus.

Angenommen, der Propp-Wilson Algorithmus bricht mit Wahrscheinlichkeit 1 ab. Dann gilt:

$$\forall\ i \in \{1, ..., k\}\ :\ \mathbf{P}(Y = s_i) = \pi_i \tag{10.1}$$

Der Algorithmus gibt demnach unverzerrte Stichproben aus.

Beweis

Sei $s_i \in S$ ein fester, beliebiger Zustand. Um (10.1) zu beweisen, genügt es zu zeigen, dass

$$\forall \epsilon > 0\ :\ |\ \mathbf{P}(Y = s_i) - \pi_i\ | \leq \epsilon \tag{10.2}$$

gilt. Sei $\epsilon > 0$ beliebig und fest. Wir haben angenommen, dass der Algorithmus mit Wahrscheinlichkeit 1 abbricht. Somit gilt für genügend großes $M \in \mathbb{N}$:

$$\mathbf{P}(\text{ der Algorithmus benötigt keine frühere Startzeit als } -N_M) \geq 1 - \epsilon \tag{10.3}$$

Wir halten ein genügend großes M fest. Dann simulieren wir eine Markov-Kette, die vom Zeitpunkt $-N_M$ bis zum Zeitpunkt 0 läuft und nennen ihren Output $\tilde{Y}$. Diese Markov-Kette soll dieselbe Updatefunktion ϕ und dieselben Zufallszahlen $U_0, ..., U_{-N_M+1}$ verwenden wie der Algorithmus. Allerdings soll der Anfangszustand dieser Markov-Kette zum Zeitpunkt $-N_M$ gemäß π gewählt werden.

π ist stationäre Verteilung, d.h. nach Definition gilt $\pi \cdot P = \pi$. Der Anfangszustand unserer Markov-Kette wird gemäß π ermittelt, alle folgenden Zustände werden durch die Übergangsmatrix P bestimmt. Somit ist $\tilde{Y}$ gemäß π verteilt.

M wurde so gewählt, dass alle Ketten des Propp-Wilson Algorithmus mit Startzeit $-N_M$ im Zeitpunkt 0 zusammenlaufen, unabhängig von ihrem Anfangszustand. Daher und wegen (10.3) gilt:

$$\mathbf{P}(Y \neq \tilde{Y}) \leq \epsilon \tag{10.4}$$

Außerdem gilt

$$\begin{aligned}
\mathbf{P}(Y = s_i) - \pi_i &\overset{\tilde{Y}\ \pi-verteilt}{=} \mathbf{P}(Y = s_i) - \mathbf{P}(\tilde{Y} = s_i) \\
&= \cancel{\mathbf{P}(Y = s_i, \tilde{Y} = s_i)} + \mathbf{P}(Y = s_i, \tilde{Y} \neq s_i) \\
&\quad -\cancel{\mathbf{P}(\tilde{Y} = s_i, Y = s_i)} - \mathbf{P}(\tilde{Y} = s_i, Y \neq s_i) \\
&\leq \mathbf{P}(Y = s_i, \tilde{Y} \neq s_i) \\
&\leq \mathbf{P}(Y \neq \tilde{Y}) \\
&\overset{(10.4)}{\leq} \epsilon
\end{aligned} \tag{10.5}$$

Analog zeigt man

$$\pi_i - \mathbf{P}(Y = s_i) \leq \epsilon \tag{10.6}$$

Mit (10.5) und (10.6) folgt die Behauptung (10.2).

□

10.4 Zwei (leider scheiternde) Versuche, den Algorithmus zu vereinfachen

10.4.1 Erster Versuch

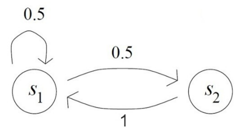
Der Propp-Wilson Algorithmus erscheint auf den ersten Blick ziemlich umständlich. Der aufmerksame Leser/Zuhörer wird sich vermutlich folgende Frage stellen:

Da einmal generierte Zufallszahlen U_i immer wiederverwendet werden müssen, müssen sie auch immer abgespeichert werden. Das scheint eine Verschwendung von Speicherplatz zu sein. Geht das auch einfacher?

Vorschlag: Wir verwenden immer neue Zufallszahlen U_i und führen damit den Algorithmus durch.

Das ist zwar eine gute Idee, aber leider erhalten wir bei diesem modifizierten Algorithmus eine verzerrte Stichprobe. Dies soll anhand des folgenden Beispiels deutlich gemacht werden:

Beispiel:



Gegeben sei der Zustandsraum $S = \{s_1, s_2\}$, der Übergangsgraph und die zugehörige Übergangsmatrix $P = \begin{pmatrix} 0,5 & 0,5 \\ 1 & 0 \end{pmatrix}$. Die stationäre Verteilung der Kette ist $\pi = (\pi_1, \pi_2) = \left(\frac{2}{3}, \frac{1}{3}\right)^2$. Sei $(N_1, N_2, ...) = (1, 2, 4, 8, ...)$

Wir modifizieren den Propp-Wilson Algorithmus, wie im Vorschlag beschrieben. Wir definieren

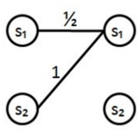
- Y als den Output des modifizierten Algorithmus
- M als die Zufallsvariable, die angibt, wie oft der Algorithmus neu gestartet werden muss bis zum ersten Zusammenlaufen der Ketten.

Dann gilt

$$\begin{aligned}
\mathbf{P}(Y = s_1) &\overset{\text{Randverteilung von Y}}{=} \sum_{m=1}^{\infty} \mathbf{P}(M = m, Y = s_1) \\
&\geq \mathbf{P}(M = 1, Y = s_1) + \mathbf{P}(M = 2, Y = s_1) \\
&\overset{\text{Def. bedingte Wkeit}}{=} \mathbf{P}(M = 1) \cdot \mathbf{P}(Y = s_1 \mid M = 1) + \mathbf{P}(M = 2) \cdot \mathbf{P}(Y = s_1 \mid M = 2) \\
&= \frac{1}{2} \cdot 1 + \frac{3}{8} \cdot \frac{2}{3} \\
&= \frac{3}{4} \\
&> \frac{2}{3}
\end{aligned}$$

Erläuterungen:

$\mathbf{P}(M = 1) = \frac{1}{2}$
Die einzige Möglichkeit, dass Markov-Ketten mit Startzeit -1 (M=1) zusammenlaufen, ist



somit gilt: $\mathbf{P}(M = 1) = \frac{1}{2} \cdot 1 = \frac{1}{2}$

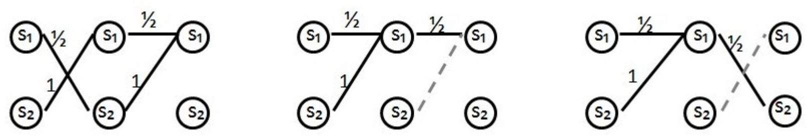
$\mathbf{P}(Y = s_1 \mid M = 1) = 1$
Vorausgesetzt, die Markov-Ketten laufen nach dem ersten Neustart zusammen, so

[2] Die stationäre Verteilung erhält man durch lösen des Gleichungssystems $\pi \cdot P = \pi$.

können sie sich nur im Zustand s_1 treffen.

$\mathbf{P}(M = 2) = \frac{3}{8}$
Es gilt
$\mathbf{P}(M = 2) =$
$\mathbf{P}$(MK mit Startzeit -2 laufen zusammen und MK mit Startzeit -1 laufen nicht zusammen) =
$\mathbf{P}$(MK mit Startzeit -2 laufen zusammen)$\cdot\mathbf{P}$(MK mit Startzeit -1 laufen nicht zusammen) =
...

Möglichkeiten für das Ereignis "Ketten mit Startzeit -2 laufen zusammen":



$... = [(\frac{1}{2})^2 + (\frac{1}{2})^2 + (\frac{1}{2})^2] \cdot \frac{1}{2} = \frac{3}{8}$

$\mathbf{P}(Y = s_1 \mid M = 2) = \frac{2}{3}$
Vorausgesetzt wird M=2, d.h. eine der drei Möglichkeiten im Schaubild tritt ein. Alle sind gleichwahrscheinlich und zwei haben Output s_1.

Fazit: Die Verwendung von neuen Zufallszahlen statt der bereits generierten Zufallszahlen führt zu einem verzerrten Stichprobenergebnis.

10.4.2 Zweiter Versuch

Das Laufenlassen der Markov-Ketten á la Coupling from the Past erscheint ziemlich umständlich.

Vorschlag: Lasse die Markov-Ketten im Zeitpunkt 0 starten und so lange laufen, bis sie zusammenlaufen. Der Zustand, in dem die Makrov-Ketten dann zusammengelaufen sind, soll unsere Stichprobe sein. [Diese Art des Laufenlassens von Markov-Ketten nennen wir **Coupling to the Future**.]

Das ist zwar eine gute Idee, aber leider erhalten wir auch hier eine verzerrte Stichprobe. Dies soll anhand des zuletzt beschriebenen Beispiels verdeutlicht werden:
Wir modifizieren den Propp-Wilson Algorithmus, wie vorgeschlagen, lassen also zwei Markov-Ketten vom Zeitpunkt 0 bis zum Zeitpunkt N des ersten Zusammentreffens laufen.

Eine Kette beginnt in Zustand s_1, die andere in Zustand s_2. Zum Zeitpunkt N-1 befinden sich die Ketten in verschiedenen Zuständen. Durch den Übergangsgraphen wissen wir, dass die Kette, die sich zum Zeitpunkt N-1 in Zustand s_2 befindet, mit Wahrscheinlichkeit 1 zum Zeitpunkt N im Zustand s_1 sein wird. Da die Ketten zum Zeitpunkt N zusammentreffen, ist die Wahrscheinlichkeit für einen Output s_1 1. Dies entspricht ganz offensichtlich nicht der stationären Verteilung. Wir erhalten somit ein verzerrtes Stichprobenergebnis.

11 Propp-Wilson Algorithmus: Die Wilson Modifikation

11.1 Motivation

Wie schon erwähnt ist die Wiederbenutzung der Zufallszahlen U_i im Propp-Wilson Algorithmus ziemlich lästig, da dies sehr viel Speicherplatz benötigt. Dieses Problem wird durch die Wilson Modifikation gelöst:
David Wilson hat den Algorithmus so verfeinert, dass keine Zufallszahlen wiederbenutzt und somit auch nicht gespeichert werden müssen. Im weiteren Verlauf des Vortrags soll die Funktionsweise der Wilson Modifikation erläutert werden.

11.2 Funktionsweise der Wilson Modifikation

Vor der Beschreibung des modifizierten Algorithmus benötigen wir eine Definition:

11.2.1 Definition: Zwillingslauf

Lasse zwei unabhängige Kopien von Markov-Ketten á la Coupling to the Future so lange laufen, bis beide Kopien zusammengelaufen sind. Die Kopie, die zuerst zusammenläuft, heißt **Gewinner**, die andere heißt **Verlierer**. Falls beide gleichzeitig zusammenlaufen, wird eine faire Münze geworfen.

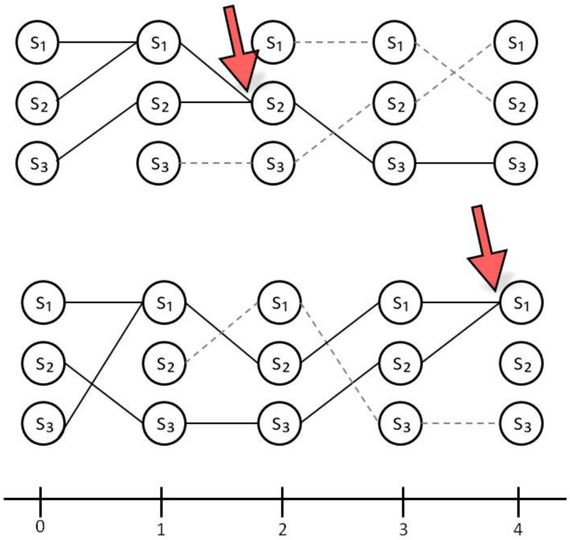


Abbildung 4: Beispiel eines Zwillingslaufes

Der Schlüssel bei der Wilson-Modifikation ist nun, dass wir zuerst Markov-Ketten laufen lassen vom Zeitpunkt $-N_Y$ bis zum Zeitpunkt $-N_{Y-1}$. Anschließend lassen wir Markov-Ketten laufen vom Zeitpunkt $-N_{Y-1}$ bis zum Zeitpunkt $-N_{Y-2}$ usw. bis zum Zeitpunkt 0.

So funktioniert der Propp-Wilson Algorithmus mit Wilson Modifikation

1. Wir erhalten die Entwicklung der Markov-Ketten vom Zeitpunkt $-N_Y$ bis zum Zeitpunkt $-N_{Y-1}$ mithilfe eines Zwillingslaufes. Die Markov-Ketten verhalten sich in diesem Zeitraum so wie der Gewinner des Zwillingslaufes bis zum Zeitpunkt des Zusammenlaufens des Verlierers.

2. Generiere den Wert einer geometrisch verteilten Zufallsvariabe Y [3] zum Parameter $\frac{1}{2}$.

   - Falls Y=1, dann sind die Markov-Ketten zum Zeitpunkt $-N_{Y-1} = 0$ zusammengelaufen. Der Output unseres modifizierten Algorithmus ist der Zustand des Gewinners zum Zeitpunkt des Zusammenlaufens des Verlierers. Wir sind fertig.

[3] Die geometrische Verteilung gibt an, wie viele Durchgänge man bis zum ersten Erfolg benötigt. Y entspricht der Anzahl der benötigten Neustarts der Markov-Ketten bis zum ersten Zusammenlaufen (=Erfolg).

- Falls $Y \geq 2$, müssen wir die Entwicklung der Markov-Ketten noch vom Zeitpunkt $-N_{Y-1}$ bis zum Zeitpunkt 0 simulieren. Dazu weiter mit Schritt 3.

3. Die Entwicklung der Markov-Ketten vom Zeitpunkt $-N_{Y-1}$ bis zum Zeitpunkt $-N_{Y-2}$ erhalten wir folgendermaßen: Wir führen erneut einen Zwillingslauf durch. Die Entwicklung unserer Markov-Ketten im Zeitraum N_{Y-1} bis N_{Y-2} ist nun genau die des Verlierers des Zwillingslaufes bis zum Zeitpunkt des Zusammenlaufens des Gewinners.

4. Schritt 3 wird wiederholt, um den Verlauf der markov-Ketten in den Zeiträumen $-N_{Y-2}$ bis $-N_{Y-3}$, $-N_{Y-3}$ bis $-N_{Y-4}$ usw. zu ermitteln, bis der Zustand der Markov-Ketten zum Zeitpunkt 0 bekannt ist. Dieser Zustand zum Zeitpunkt 0 ist der Output unseres modifizierten Algorithmus.

11.2.2 Beispiele

Zum besseren Verständnis der Wilson Modifikation folgen nun zwei Beispiele. *(Die folgenden Bilder sind nicht Teil des Seminarvortrags. Stattdessen werden an dieser Stelle Folie 2 und 3 der „ Animation PWA und WM.pptx "gezeigt.)*

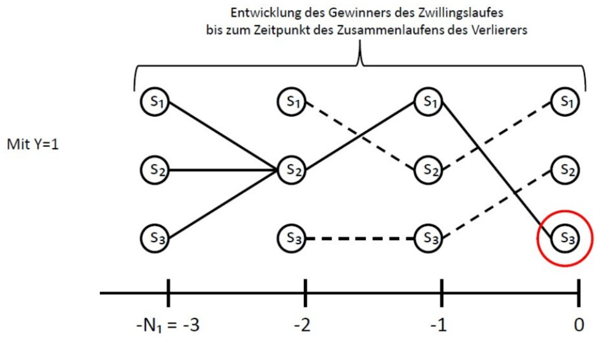


Abbildung 1: Wilson Modifikation- Beispiel 1

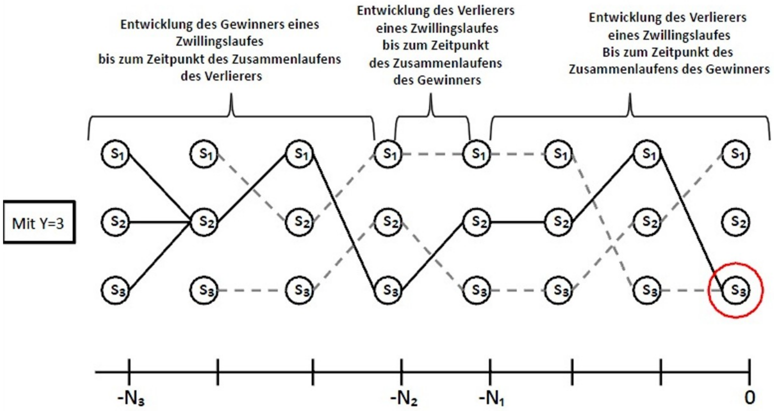


Abbildung 2: Wilson Modifikation- Beispiel 2

Der Output des nach Wilson modifizierten Algorithmus ist gemäß unserer gewünschten Verteilung π verteilt, wir erhalten also eine unverzerrte Stichprobe. Und das Tolle daran ist: Wir mussten überhaupt keine Zufallszahlen speichern!