LAK-ComputerMathematik

Projekt **GJ-3**

Box-Muller-Methode

Stand: 29. Oktober 2025, 15:11

Konsultationen zum Projekt: nach Terminvereinbarung (E-mail)

Box-Muller-Methode:

Die Generierung von Zufallszahlen ist eine wichtige Voraussetzung vieler Aufgaben wie numerische Simulationen (Monte-Carlo), das maschinelle Lernen und Kryptografie. In der Regel können Computer aber ohne spezialisiertes Gerät nur Pseudozufallszahlen generieren. Diese sehen zufällig aus, sind es aber nicht. Das ist oft gut genug und auch für uns im Projekt.

In diesem Projekt wird untersucht, wie man aus gleichverteilten Zufallszahlen entsprechende normalverteilte Zufallszahlen X erzeugt. Dazu verwenden wir die Box-Muller-Methode[1] in Teilaufgabe 1.

Des weiteren heißt eine Zufallsvariable X normalverteilt, wenn für alle $a, b \in \mathbb{R}$ $P(x \in [a, b]) = \int_a^b f(t)\,\mathrm{d}t$ gilt wobei die Normalverteilungsdichte f wie folgt definiert ist:

$$f : x \mapsto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \,. \tag{1}$$

Die Parameter μ und σ entsprechen den Erwartungswert und die Standardabweichung. Im Folgendem wird $\mu = 0$ und $\sigma = 1$ angenommen.

1. Schreiben Sie eine Funktion `boxmuller` (ohne Parameter), die eine Zahl $\sqrt{-2\ln(y_1)}\cos(2\pi y_2)$ ausrechnet und zurück gibt, wobei y_1 und y_2 gleichverteilte Standardzufallszahlen (Realisationen von der Gleichverteilung auf $[0, 1]$, siehe `rand`) sind.

Wir wollen jetzt uns anschauen, ob die Zahlen, die `boxmuller` liefert, der Normalverteilungsdichte f entsprechen. Das können wir mit dem folgenden Histogramm machen.

2. Schreiben Sie eine Funktion `vis_compare` mit Parameter `n`, die:
   - `n` Zahlen (eine *(Zufalls)Stichprobe*) mithilfe `boxmuller` generiert und diese in einer Variable `samples` speichert,
   - ein Histogramm davon darstellt, siehe `histogramm` und dessen `nbins` bzw. `Normalization` Optionen (Vorschlag: `histogram(samples, nbins, 'Normalization', 'pdf');` mit `nbins` = 50),
   - in denselben Graph die Funktion f auf dem Intervall $[-3, 3]$ aufzeichnet. Hier können Sie f entweder als symbolische Funktion definieren, oder eine Funktion `normal_density` schreiben und diese für verschiedene Werte auswerten.

3. Rufen Sie `vis_compare` für verschiedene Werte des Parameters `n` (z.B. 10^k mit $k \in \{2, 3, 4, 5\}$) auf und kommentieren Sie diese.

Zusatz [+4 Punkte]: Wir möchten etwas systematischer (= mathematischer) kontrollieren, ob die Stichproben (von `boxmuller` geliefert) normalverteilt sind.

Wir vergleichen dafür grob die Normalverteilungsfunktion $\Phi : x \mapsto \frac{1}{2}(1 + \mathrm{erf}(x/\sqrt{2}))$ und die empirische Verteilungsfunktion F. Für eine Stichprobe $\mathtt{x} = (\mathtt{x}_i)_{1\le i\le n}$ $F : \mathbb{R} \to [0, 1]$ ist wie folgt definiert:

$$F(x) := \frac{\text{Anzahl der } \mathtt{x}_i \le x}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{(-\infty,x]}(\mathtt{x}_i).$$

F ist also eine rechtsstetige, wachsende Treppenfunktion, die an den Stellen $(\mathtt{x}_i)$ nicht stetig ist.

---

[1] `https://de.wikipedia.org/wiki/Box-Muller-Methode`

4. Schreiben Sie eine symbolische Funktion `phi`, die Φ implementiert (nutze Matlab-Funktion `erf`).

5. Schreiben eine Funktion `normal_cdf_error`, die ein Argument `x` (die Stichprobe) nimmt und die den Fehler $\|\Phi(\mathtt{x}) - F(\mathtt{x})\|/n$ berechnet, wobei n die Länge des Vektors `x` entspricht. Siehe `sort` und `norm`.

6. Berechnen Sie die Fehler mithilfe von `normal_cdf_error` für mehrere Stichprobegrößen n z.B. $n = 10^i$, $1 \leq i \leq 5$ (immer noch aus `boxmuller`), und visualisieren Sie diese als Funktion von $1/n$. . Hier ist eine doppelte logarithmische Darstellung von Vorteil, siehe `loglog`.
   Was bemerken Sie?