

Monte Carlo Integration: Metropolis Algorithmus

-1-

In der letzten Vorlesung haben wir über Monte Carlo-Verfahren zur Integration (höher-dimensionaler) Funktionen gesprochen.

Besonders schnell konvergierende Resultate erhalten wir dabei, wenn wir ein sogenanntes "importance sampling" durchführen. Also anstelle gleichverteilter Zufallswerte $\{x_i\}$ im Integrationsvolumen folgen die Zufallszahlen einer Verteilung $p(x)$, welche möglichst nahe an der zu integrierenden Funktion $f(x)$ liegt, denn es gilt:

$$\int f(x) dx = \int \frac{f(x)}{p(x)} \cdot p(x) dx \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \quad \text{für } p\text{-verteilte Zufallszahlen } \{x_i\}$$

mit statistischem Fehler

$$\Delta = \frac{1}{\sqrt{N}} \cdot \sqrt{\text{Var } \frac{f}{p}}$$

Heute wollen wir der Frage nachgehen, wie man p-verteilte Zufallszahlen für beliebige Verteilungen $p(x)$ erzeugen kann. ($\int dx p(x) = 1$)

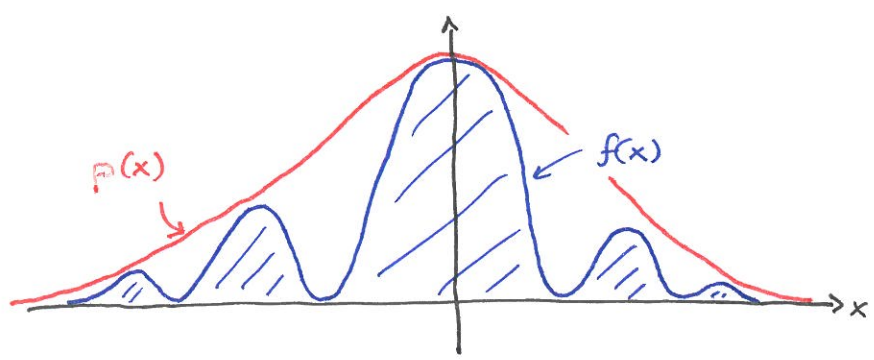
Hierzu werden wir den sogenannten Metropolis-Algorithmus kennenlernen, welcher häufig als der mächtigste Algorithmus überhaupt bezeichnet wird.

Entwickelt wurde der Algorithmus vor knapp 60 Jahren, also im Jahre 1953, von Metropolis, Rosenbluth² und Teller² am Los Alamos National Laboratory, wo seinerzeit auch die ersten

"Super-Computer" in Betrieb genommen wurden, etwa die MANIAC.

Betrachten wir dazu wieder das Beispiel einer zu integrierenden Funktion in einer Dimension:

$$I = \int_{-\infty}^{\infty} f(x) dx$$



Austelle nun zu versuchen, p -verteilte Zufallszahlen direkt zu generieren, wollen wir nach dem folgenden Schema vorgehen:

- $x_i \in \mathbb{R}$ sei gegeben
- Wähle Zufallszahl Δx gleichverteilt im Intervall $[-h, h]$, wobei h eine sogenannte Schrittweite sei.

Damit

$$\tilde{x}_{i+1} = x_i + \Delta x$$

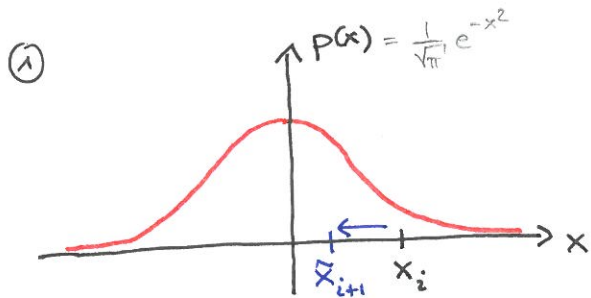
- Berechne $\alpha = \frac{p(\tilde{x}_{i+1})}{p(x_i)}$ wobei $p(x)$ eine gegebene Verteilung sei

- Wähle eine Zufallszahl γ gleichverteilt in $[0, 1)$ und mache folgende Fallunterscheidung

$$\gamma \leq \alpha \quad \rightarrow \quad \tilde{x}_{i+1} \text{ wird akzeptiert, d.h. } x_{i+1} = \tilde{x}_{i+1}$$

$$\gamma > \alpha \quad \rightarrow \quad \tilde{x}_{i+1} \text{ wird verworfen, d.h. } x_{i+1} = x_i$$

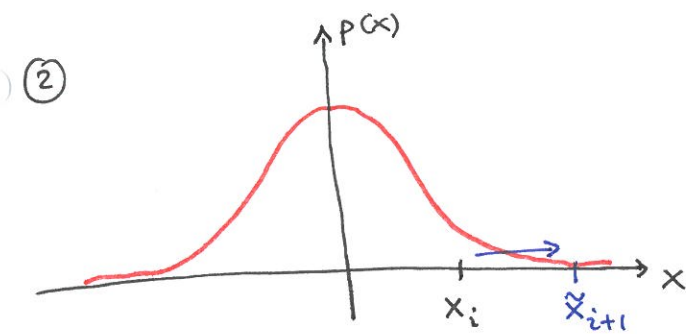
Der Metropolis - Algorithmus erzeugt eine Sequenz von Zahlen $\{x_i\}$, die einem gewichteten Zufallslauf entsprechen. Speziell gilt:



$$\alpha = \frac{p(\tilde{x}_{i+1})}{p(x_i)} > 1$$

d.h. $y < \alpha$ für alle $y \in [0, 1)$

→ Änderung wird auf jeden Fall akzeptiert

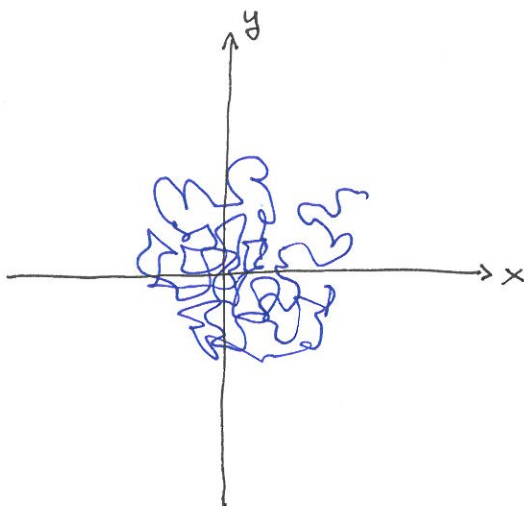


$$0 < \alpha < 1$$

d.h. Änderung wird akzeptiert, falls $y \leq \alpha$, d.h. mit Wahrscheinlichkeit α

Die Sequenz der so erzeugten Zahlen $\{x_i\}$ wird auch als sogenannte Markov - Kette bezeichnet.

Beispiel: Illustration einer Markov - Kette in 2 Dimensionen



$$p(x, y) = \frac{1}{\pi} e^{-(x^2 + y^2)}$$

Markov - Ketten

Im obigen Beispiel haben wir einen Algorithmus kennengelernt, der durch einen geschickt gewichteten Zufallslauf eine Sequenz $\{x_i\}$ von p -verteilten Zufallszahlen in einer sogenannten Markov-Kette erzeugt hat.

Dies wollen wir uns im folgenden noch einmal im Detail anschauen, insbesondere um zu verstehen, welche Bedingungen an den sogenannten Markov-Prozess zur Erzeugung eines solchen Zufallslauf gestellt werden, damit in der Tat eine bestimmte, beliebig vorgegebene Verteilung "gesampled" wird.

Betrachten wir dazu die folgende Sequenz von Konfigurationen

$$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \dots \rightarrow x_i \rightarrow x_{i+1} \rightarrow \dots$$

welche in einem sogenannten Markov-Prozess erzeugt werden sollen. Eine Konfiguration x_i mag dabei eine ein- oder zweidimensionale Koordinate wie im obigen Beispiel der Monte Carlo-Integration sein. Es mag sich dabei aber auch um komplexere Objekte wie etwa einer Spin-Konfiguration handeln, wie wir sie in der kommenden Vorlesung im Zusammenhang mit dem klassischen Ising-Modell besprochen werden.

-5-

Der Übergang von einer gegebenen Konfiguration x zu einer neuen Konfiguration y erfolgt im Markov-Prozess mit einer Wahrscheinlichkeit W_{xy} . Für alle möglichen Konfigurationen x, y definieren diese Wahrscheinlichkeiten eine Übergangsmatrix^W, deren Elemente eben diese W_{xy} sind.

Da die Summe aller Wahrscheinlichkeiten, aus einem Zustand x zu einem beliebigen anderen Zustand zu gehen, genau 1 sein muß, gilt für die Spalten der Übergangsmatrix

$$\sum_y W_{xy} = 1$$

Diese Normierung der Spalten hat wichtige Konsequenzen:

Ein Markov-Prozess erhält die gesamten Wahrscheinlichkeiten.

Und zweitens: Der größte Eigenwert der Übergangsmatrix ist 1, und der zugehörige Eigenvektor mit ausschließlich positiven Einträgen ist eine Gleichgewichtszustellung, welche nach einer großen Zahl von Markov-Schritten erhalten wird.

Die Frage ist nun, wie wir die Übergangsmatrix W wählen müssen, so daß die sich einstellende Gleichgewichtszustellung der gewünschten Verteilung $p(x)$ entspricht.

Ebensolches wird durch die folgenden Bedingungen erfüllt:

-6-

1) Normierung:

$$\sum_y W_{xy} = 1$$

2) Ergodizität: Es muß möglich sein, von einer beliebigen Konfiguration x zu einer beliebigen Konfiguration y in einer endlichen Zahl von Markov-Schritten zu gelangen (unabhängig von der Wahrscheinlichkeit dieses Prozesses), d.h. es gibt für alle Konfigurationen x und y eine positive Zahl $n < \infty$ so daß $(W^n)_{xy} \neq 0$:

$$\forall x, y \exists n: (W^n)_{xy} \neq 0$$

3) Gleichgewicht: Die Wahrscheinlichkeitsverteilung $P_x^{(n)}$ ändert sich bei jedem Markov-Schritt gemäß:

$$\sum_x P_x^{(n)} W_{xy} = P_y^{(n+1)}$$

aber konvergiert zu einer Gleichgewichtsverteilung

$$\lim_{n \rightarrow \infty} P_x^{(n)} = P_x$$

Die Gleichgewichtsverteilung P_x ist ein Eigenvektor der Übergangsmatrix mit Eigenwert 1, so daß die Gleichgewichtsbedingung

$$\sum_x P_x W_{xy} = P_y$$

erfüllt ist.

Wir können sofort sehen, daß dafür die sogenannte "detailed balance" Bedingung hinreichend ist

$$\boxed{\frac{W_{xy}}{W_{yx}} = \frac{P_y}{P_x} \Leftrightarrow P_x W_{xy} = P_y W_{yx}}$$

denn Summation über alle Konfigurationen x liefert:

$$\sum_x P_x W_{xy} = \sum_x P_y W_{yx} = P_y \cdot \underbrace{\sum_x W_{yx}}_{=1 \text{ wegen Normierung}} = P_y$$

Eine Möglichkeit, die "detailed balance" Bedingung zu erfüllen, sind die sogenannten Gibbs-Gewichte:

$$\boxed{W_{xy} = \frac{P_y}{P_x + P_y} \quad W_{yx} = \frac{P_x}{P_x + P_y}}$$

$$\rightarrow \frac{W_{xy}}{W_{yx}} = \frac{P_y}{P_x}$$

Eine zweite Möglichkeit, die "detailed balance" Bedingung zu erfüllen, sind die bereits erwähnten Metropolis-Gewichte:

$$\boxed{W_{xy} = \min\left(1, \frac{P_y}{P_x}\right) \quad W_{yx} = \min\left(1, \frac{P_x}{P_y}\right)}$$

$$\rightarrow \frac{W_{xy}}{W_{yx}} = \frac{P_y}{P_x} \quad \text{unabhängig davon ob } P_y > P_x \text{ oder } P_y < P_x.$$