Professur für Mathematik
Fakultät für Elektrische Energiesysteme
und Informationstechnik

der Bundeswehr Universität München

Praktikum Numerische Simulation in der Technik

Markov Chain Monte Carlo – Verfahren zur globalen Optimierung

Stichworte: Mone Carlo–Verfahren, Markovkette, Metropolis-Algorithmus, Globale Optimierung

Betreuer: Richter

Termine: WT oder FT

Thema: Der Metropolis-Algorithmus ist ein Verfahren, Zufallszahlen nach einer bestimmten, vorgegebenen Wahrscheinlichkeitsverteilung zu erzeugen. Er wird hier benutzt, um Zufallszahlen so zu erzeugen, dass sie mit erhöhter Wahrscheinlichkeit in der Nähe eines globalen Optimums einer multivariaten Funktion liegen. Ähnlich funktioniert die Methode des Simulated Annealing, bei der jedoch die zugrunde liegende Wahrscheinlichkeitsverteilung im Lauf des Verfahrens geändert wird („Cooling Strategy"), damit die erzeugten Zufallszahlen am Ende *sicher* auf eine Optimalstelle fallen. Das funktioniert aber nur, wenn die Änderung sehr langsam vollzogen wird. Beim hier zu untersuchenden Verfahren begnügt man sich damit, in die Nähe einer Optimalstelle zu gelangen und startet von dort ein leistungsfähiges lokales Optimierungsverfahren.

Markovketten. Es sei E eine endliche (aber eventuell *sehr* große) Menge von **Zuständen** (das können Zahlen sein oder Positionen im Raum oder Energiezustände eines physikalischen Systems). Weiter sei $X_0, X_1, X_2, \ldots$ eine Folge von **Zufallsvariablen** mit Werten in E, das heißt jede der Zufallsvariablen X_k, $k \in \mathbb{N}_0$, nimmt zufällig einen Zustand $x \in E$ an. Der Index $k \in \mathbb{N}_0$ wird als (diskrete) Zeit interpretiert, so dass die Folge $(X_k)_{k\in\mathbb{N}_0}$ das zeitliche Verhalten eines Systems modelliert, das in jedem Zeitschritt (zufällig) seinen Zustand ändert. Die Folge heißt **Markovkette**, wenn

$$P(X_{k+1} = x_{k+1} | X_0 = x_0, X_1 = x_1, \ldots, X_k = x_k) = \Pi(x_{k+1}, x_k) \qquad (1)$$

für $x_0, \ldots, x_{k+1} \in E$ und $k \in \mathbb{N}_0$, sofern $P(X_0 = x_0, \ldots, X_k = x_k) > 0$. Hier bezeichnet $P(X_0 = x_0, \ldots, X_k = x_k)$ die Wahrscheinlichkeit, dass die Zufallsvariablen $X_0, \ldots, X_k$ die Werte $x_0, \ldots, x_k$ annehmen und $P(X_{k+1} = x_{k+1} | X_0 = x_0, \ldots, X_k = x_k)$ bezeichnet die bedingte Wahrscheinlichkeit, dass X_{k+1} den Wert x_{k+1} annimmt unter der Bedingung, dass $X_0, \ldots, X_k$ die Werte $x_0, \ldots, x_k$ angenommen haben. Die Aussage (1) bedeutet also, dass $P(X_{k+1} = x_{k+1} | X_0 =$

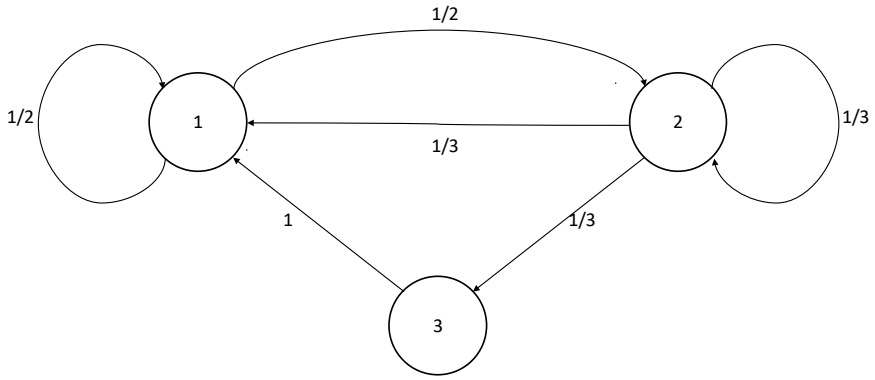
$x_0, \ldots, X_k = x_k)$ nur von den Zuständen x_{k+1} und x_k abhängt, aber *nicht* von den früheren Zuständen der Markov-Kette. In desem Sinn hat die Markovkette nur ein „Kurzzeitgedächtnis“. Man nennt $\Pi(x, y)$ die **Übergangswahrscheinlichkeit** der Markovkette vom Zustand $x \in E$ in den Zustand $y \in E$. Die Übergangswahrscheinlichkeiten bleiben für alle Zeiten gleich. Die Matrix

$$\Pi := (\Pi(x, y))_{(x,y)\in E\times E}$$

nennt man die **Übergangsmatrix** der Markovkette. Alle Elemente der Matrix Π liegen im Intervall $[0, 1]$ und jede ihrer Zeilensummen ist 1:

$$\sum_{y\in E} \Pi(x, y) = 1 \quad \text{für alle} \quad x \in E.$$

Matrizen mit diesen beiden Eigenschaften (Nichtnegativität der Komponenten und Zeilensummen gleich 1) nennt man **stochastische Matrizen**. Markovketten können durch einen **Übergangsgraphen** visualisiert werden. Dies ist ein gerichteter Graph, dessen Knoten die Zustände (meist in einer Nummerierung von 1 bis $N = |E|$) bezeichnen und dessen Kanten die Zustandsübergänge bezeichnen, die in einem Zeitschritt möglich sind. Die Übergangswahrscheinlichkeiten werden an den Kanten angetragen. Nachfolgend ein Beispiel eines Übergangsgraphen zu $E = \{1, 2, 3\}$.



Es wird noch Folgendes festgehalten: Die Wahrscheinlichkeit, in *zwei* Zeitschritten vom Zustand $x \in E$ in den Zustand $y \in E$ zu gelangen, lässt sich über die Formel

$$\sum_{z\in E} \Pi(x, z)\Pi(z, y) = \Pi^2(x, y) \tag{2}$$

als Matrixprodukt berechnen. Begründung: Man kommt von x in zwei Schritten nach y, indem man in einem ersten Schritt von x in einen Zwischenzustand $z \in E$ gelangt und von dort im nächsten Schritt weiter nach y. Die Fälle $z = x$ und $z = y$ sind hierbei eingeschlossen. $\Pi^2(x, y)$ ist das Element der Matrix $\Pi^2 = \Pi \cdot \Pi$ in Zeile

x und Spalte y. Allgemeiner ist $\Pi^n(x,y)$ die Wahrscheinlichkeit eines Übergangs von x nach y in n Zeitschritten.

Metropolis-Algorithmus. Auf der Zustandsmenge E sei eine sogenannte Zähldichte

$$\alpha : E \to]0,1], \quad x \mapsto \alpha(x) \quad \text{mit} \quad \sum_{x\in E} \alpha(x) = 1 \tag{3}$$

gegeben, welche eine Wahrscheinlichkeitsverteilung auf E definiert; $\alpha(x)$ wird als die Wahrscheinlichkeit interpretiert, mit der der Zustand x angenommen wird. (Beachte, dass $\alpha(x) > 0$ für alle $x \in E$.) Der Metropolis-Algorithmus ist ein Verfahren, Zustände $x \in E$ mit Wahrscheinlichkeiten $\alpha(x)$ auszuwählen (genauer gesagt mit einer Wahrscheinlichkeit, die $\alpha(x)$ nahe kommt). Dazu wird eine Markovkette $X_0, X_1, X_2, \ldots$ so konstruiert, dass die Wahrscheinlichkeit, mit der X_k einen beliebigen Wert $x \in E$ annimmt, für $k \to \infty$ gegen $\alpha(x)$ konvergiert. Diese Markovkette wird dann dadurch realisiert, dass in einem beliebigen Zustand $x_0 \in E$ gestartet wird ($X_0 = x_0$) und anschließend Zustände $x_1, x_2, \ldots$ von $X_1, X_2, \ldots$ gemäß den Übergangswahrscheinlichkeiten $\Pi(x_k, x_{k+1})$ erzeugt werden. Die Matrix Π wird so gewählt, dass die Wahrscheinlichkeit des Auftretens von x_k für $k \to \infty$ gegen $\alpha(x_k)$ konvergiert. Das Erzeugen der Zustände x_k nennt man **Sampling** der Markovkette.

Nun zur eigentlichen Konstruktion. Dazu definiert man für jedes $x \in E$ eine Nachbarschaft

$$\{\,\} \neq N_x \subseteq E \setminus \{x\}$$

von Zuständen von x. Für diese soll die Symmetrieeigenschaft

$$y \in N_x \quad \Longleftrightarrow \quad x \in N_y \tag{4}$$

gelten, das heißt wenn x Nachbar von y ist, dann ist umgekehrt y Nachbar von x.

Beispiel. Für das Gitter $E = \{(k,\ell) \in \mathbb{N}^2;\ 1 \le k, \ell \le n\}$ ließe sich

$$N_{(k,\ell)} := \{(i,j) \in E;\ |i-k| \le 1, |j-\ell| \le 1\} \setminus \{(k,\ell)\} \tag{5}$$

definieren als Menge aller zu (k,ℓ) benachbarten Gitterpunkte mit der zusätzlichen Vereinbarung, dass die Umindizierungen

$$0 \longrightarrow n \quad \text{und} \quad n+1 \longrightarrow 1$$

vorgenommen werden (periodische Fortsetzung der Nachbarschaft am gegenüberliegenden Gitterrand).

Es sei $d_x := |N_x|$ die Anzahl der Nachbarn von x. Im obigen Beispiel wäre $d_x = 8$ für alle $x \in E$. Übergangswahrscheinlichkeiten werden nun wie folgt definiert:

$$\Pi(x,y) = \begin{cases} \dfrac{1}{d_x} \min\left\{ \dfrac{\alpha(y)}{\alpha(x)} \dfrac{d_x}{d_y}, 1 \right\}, & \text{falls } y \in N_x \\ 1 - \displaystyle\sum_{z\in N_x} \Pi(x,z), & \text{falls } y = x \\ 0, & \text{sonst} \end{cases} \tag{6}$$

1. Aufgabe. Rechnen Sie nach, dass die Übergangswahrscheinlichkeiten die sogenannte **detailed balance equation**

$$\alpha(x)\Pi(x,y) = \alpha(y)\Pi(y,x) \quad \text{für alle} \quad x,y \in E \tag{7}$$

erfüllen. Dies ist die entscheidende Eigenschaft von Π.

2. Aufgabe. Rechnen Sie nach, dass aus (7) die Identität

$$\alpha(x) = \sum_{y\in E} \alpha(y)\Pi(y,x) \quad \text{für alle} \quad x \in E \tag{8}$$

folgt und dass diese Identität auch in der Form

$$\Pi^\top \alpha = \alpha \quad \text{mit} \quad \alpha = (\alpha(x))_{x\in E} \quad \text{[Spaltenvektor]} \tag{9}$$

ausgedrückt werden kann. Was besagt die Identität (9) mathematisch für den Vektor $\alpha \in \mathbb{R}^N$?

Interpretation von (7) und (9): Nimmt X_k die Zustände $x \in E$ mit den Wahrscheinlichkeiten $\alpha(x)$ an, so folgt

$$P(X_{k+1} = x) = \sum_{y\in E} P(X_k = y)\Pi(y,x) = \sum_{y\in E} \alpha(y)\Pi(y,x) = \alpha(x).$$

Dies bedeutet, dass sich die Wahrscheinlichkeitsverteilung der Zustände der Markovkette mit der Zeit nicht mehr ändert. Die Markovkette beschreibt dann ein System im Gleichgewicht. Man nennt α eine **stationäre Verteilung** der Markovkette.

3. Aufgabe. Zeigen Sie, dass für alle Eigenwerte λ von Π gilt, dass $|\lambda| \leq 1$. Da eine (quadratische) Matrix und ihre Transponierte stets diesselben Eigenwerte haben, gilt diese Aussage dann automatisch auch für $\Pi^\top$.

Anleitung: Ohne Einschränkung können Sie annehmen, dass $E = \{1,\ldots,N\}$ (Durchnummerierung der Zustände). Setzen Sie dann $\Pi = (p_{i,j}) \in \mathbb{R}^{N,N}$. Es sei v ein Eigenvektor von Π zum Eigenwert λ. Nehmen Sie an, dass die 1. Komponente von v die betragsgrößte sei, also: $|v_1| \geq |v_2|,\ldots,|v_N|$ (allgemeiner könnte man annehmen, dass die m-te Komponente die betragsgrößte sei, das würde an der Rechnung nur die Indizes ändern). Schreiben Sie zunächst die 1. Gleichung des Gleichungssystems $\Pi v = \lambda v$ explizit an und folgern Sie unter Benutzung der Dreiecksungleichung und wegen $p_{i,j} \geq 0$, dass

$$|\lambda - p_{1,1}||v_1| \leq (p_{1,2} + \ldots + p_{1,N})|v_1|.$$

Beide Seiten können durch $|v_1|$ geteilt werden (warum?) und dann folgt

$$|\lambda| \leq |\lambda - p_{1,1}| + p_{1,1} \leq \sum_{j=1}^{n} p_{1,j}.$$

Warum ist die Summe rechts gleich 1?

4. Aufgabe. Man setzt $X_0 := x_0 \in E$ (sicherer Start der Markov-Kette im Zustand x_0; der Zustand x_0 könnte seinerseits zufällig ausgewählt werden). Verifizieren Sie anhand von (1) und (2), dass unter der Bedingung des Starts in x_0

$$P(X_k = x) = \Pi^k(x_0, x) \quad \text{für alle} \quad x \in E \tag{10}$$

gilt. Bei einer Durchnummerierung aller $N = |E|$ Zustände sei (ohne Einschränkung) x_0 der erste und es sei $e \in \mathbb{R}^N$ der erste kanonische Einheitsvektor (eine 1 auf Position 1, sonst Nullen). Es sei weiter $p^k \in \mathbb{R}^N$ der Vektor mit Komponenten $(p^k(x))_{x\in E} = (P(X_k = x))_{x\in E}$. Folgern Sie, dass sich (10) kompakt in der Form

$$p^k = (\Pi^\top)^k e \tag{11}$$

schreiben lässt.

Die Interpretation von (11) ist, dass X_k die Zustände $x \in E$ gemäß den Wahrscheinlichkeiten $p^k(x)$ annimmt. Die Verteilung von X_0 ist wegen des sicheren Starts in x_0 durch e gegeben. Wenn gezeigt werden könnte, dass

$$p^k \overset{k\to\infty}{\longrightarrow} \alpha$$

mit der vektoriell geschriebenen Dichte $\alpha = (\alpha(x))_{x\in E}$, dann würde dies die eingangs behauptete Konvergenzeigenschaft des Metropolis-Algorithmus bestätigen.

5. Aufgabe. Zeigen Sie die (exponentielle) Konvergenz $p^k \longrightarrow \alpha$ unter folgenden beiden vereinfachenden Annahmen:

- Es existiere eine Basis $v_1, \ldots, v_N$ des $\mathbb{R}^N$ aus Eigenvektoren von $\Pi^\top$ zu Eigenwerten $\lambda_1, \ldots, \lambda_N$. Nach (9) ist α ein Eigenvektor zum Eigenwert 1. Ohne Einschränkung setze man $v_1 = \alpha$ und $\lambda_1 = 1$.

- $\lambda_1 = 1$ ist der einzige Eigenwert mit Betrag 1. Für die anderen Eigenwerte gelte $|\lambda_2|, \ldots, |\lambda_N| < 1$. (Aus der 3. Aufgabe würde nur $|\lambda_2|, \ldots, |\lambda_N| \leq 1$ folgen.)

Anleitung: Schreiben Sie e unter Benutzung der Eigenvektorbasis in der Form:

$$e = \mu_1 v_1 + \ldots + \mu_N v_N.$$

Leiten Sie hieraus eine explizite Darstellung von p^k in der Eigenvektorbasis ab.

6. Aufgabe. Wie kann die Folge der zufälligen Zustände $x_1, x_2, x_3, \ldots$ erzeugt werden? Es wird folgender Vorschlag gemacht:

1. Setze $k = 0$ und $x := x_0$.

2. Setze $k := k+1$ und wähle ein $y \in N_x$ mit Wahrscheinlichkeit d_x^{-1} aus (Gleichverteilung; realisierbar mit einem Zufallszahlengenerator für ganze Zahlen, in MATLAB etwa `randi`).

3. Erzeuge eine Zufallszahl $u \in [0,1]$ (Gleichverteilung; realisierbar mit einem Generator für auf $[0,1]$ gleichverteilte Zufallszahlen, in MATLAB etwa `rand`).

4. Falls $u < min\{(d_x \cdot \alpha(y))/(d_y \cdot \alpha(x), 1\}$, dann setze $x := y$ (ansonsten wird x nicht geändert)

5. Setze $x_k := x$ und gehe zurück zu Schritt 1.

Verifizieren Sie, dass die Übergänge $x_k \longrightarrow x_{k+1}$ mit diesem Verfahren tatsächlich gemäß der Wahrscheinlichkeiten (6) erfolgen.

Anleitung. Überlegen Sie, mit welcher Wahrscheinlichkeit die Ungleichung $u < min\{(d_x \cdot \alpha(y))/(d_y \cdot \alpha(x), 1\}$ erfüllt ist. Mit welcher Wahrscheinlichkeit wird also der Nachbar y von x ausgewählt (die Auswahl des Kandidaten y in Schritt 2 und die Erzeugung von u geschehen unabhängig voneinander)? Diese Überlegung gilt für jeden Nachbarn von y. Mit welcher Wahrscheinlichkeit wird also kein Nachbar ausgewählt?

Globale Optimierung. Es seien $a, b \in \mathbb{R}^n$ mit $a < b$, das heißt $a_i < b_i$ für $i = 1, \ldots, n$. Gegeben sei ferner eine stetige Funktion $f : [a,b] \longrightarrow \mathbb{R}$. Hierbei versteht man unter $[a,b]$ das n-dimensionale Intervall $[a_1, b_1] \times \ldots \times [a_n, b_n]$. Gesucht ist ein $x^* \in [a,b]$ mit der Eigenschaft $f(x^*) \leq f(x)$, also eine **globale Minimalstelle** der Funktion F.

Man wähle einen Parameter $\lambda > 0$ und definiere die Funktion

$$\alpha : [a,b] \to \mathbb{R}, \quad x \mapsto \frac{e^{-\lambda f(x)}}{\displaystyle\int_{[a,b]} e^{-\lambda f(y)}\, dy} \tag{12}$$

Die Funktion α ist strikt positiv und erfüllt $\int_{[a,b]} \alpha(x)\, dx = 1$, sie ist also eine Wahrscheinlichkeitsdichte auf $[a,b]$. Offenbar nimmt α genau an jenen Stellen x große Werte an, wo f besonders kleine Werte annimmt. Erzeugt man Zufallszahlen, die gemäß der Dichte α verteilt sind, so besteht eine erhöhte Wahrscheinlichkeit, dass sich diese in der Nähe einer Optimalstelle befinden.

7. Aufgabe. Wählen Sie $\lambda > 0$, $M \in \mathbb{N}$ mit $M > 2$. Setzen Sie $h_i := (b_i - a_i)/M$ und definieren Sie das Gitter

$$E := \{(i_1 h_1, \ldots, i_n h_n);\ i_1, \ldots, i_n = 0, \ldots, M\}\,.$$

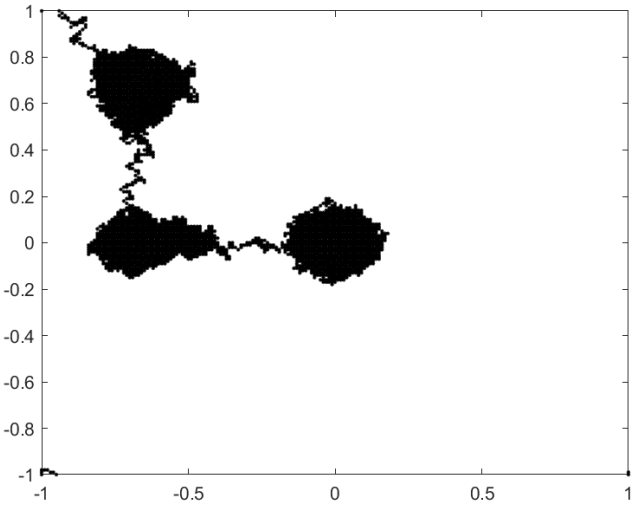
Dieses Gitter besteht aus einer potentiell astronomisch hohen Anzahl von $(M+1)^n$ Gitterpunkten. Auf E wird eine Nachbarschaft analog zu (5) definiert (inklusive der

periodischen Randbehandlung), so dass jedes $x \in E$ eine Anzahl von $d_x = 3^n - 1$ Nachbarn besitzt. Programmieren Sie den Metropolis-Algorithmus, so dass dieser zufällige Gitterpunkte $x \in E$ gemäß der Wahrscheinlichkeitsdichte α erzeugt.

8. Aufgabe. Betrachten Sie als Testfall die Funktion

$$f : [-1, 1]^n \to \mathbb{R}, \quad x \mapsto \sum_{i=1}^{n} \left(4x_i^2 - \cos(8x_i) + 1\right).$$

Diese Funktion besitzt 3^n Minimalstellen, wovon genau eine global ist, nämlich die im Punkt $x = 0$. Die folgende Graphik zeigt 10^6 Punkte, die im Fall $n = 2$, $\lambda = 10$ und $M = 200$ generiert wurde. Startpunkt war der Gitterpunkt $(-1, -1)$, die periodische Fortsetzung des Suchverfahrens über die Gitterränder ist zu erkennen.



Wie schafft es das Verfahren, eine lokale Minimalstelle von f wieder zu verlassen? Welchen Einfluss hat der Parameter λ und warum sollte er nicht zu groß gewählt werden?

Testen Sie, ob die Optimalstelle $x = 0$ bei einem Start in $(-1, -1)$ auch im Fall $n = 30$ gefunden wird. Dazu soll der Metropolis-Algorithmus noch wie folgt erweitert werden: In jenem besuchten Gitterpunkt $x \in E$, in dem der minimale Wert von f festgestellt werden kann, wird eine lokale Minimumssuche gestartet. Dazu kann die in MATLAB bereitgestellte Funktion `fmincon` genutzt werden. Testen Sie in diesem Fall mit $M = 20$ und $\lambda = 1$. Welches Ergebnis erhält man, wenn man die Funktion `fmincon` direkt im Punkt $(-1, -1)$ startet?