

Markov chain Monte Carlo (MCMC)

Beate Weidenhammer und Lena Kalleske

14. Januar 2011

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

Eine Anwendung

Beispiele

Hard-Core Model

Zufällige q -Färbungen

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

Eine Anwendung

Beispiele

Hard-Core Model

Zufällige q -Färbungen

Gegeben : Markovkette mit Zustandsraum $S = \{s_1, s_2, \dots, s_k\}$,
Anfangsverteilung $\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_k^{(0)})$ und
Übergangsmatrix P

↔ Wie simuliert man eine solche Kette?

Annahme: Wir sind in der Lage, $n + 1$ unabhängige $U([0, 1])$ -verteilte Zufallsvariablen zu simulieren.

↔ Simulation von U_0, U_1, \dots, U_n

Definition der Initiativfunktion $\psi : [0, 1] \mapsto S$:

$$\psi(u_0) = \begin{cases} s_1 & , \text{ für } u_0 \in [0, \mu_1^{(0)}) \\ s_2 & , \text{ für } u_0 \in [\mu_1^{(0)}, \mu_1^{(0)} + \mu_2^{(0)}) \\ \vdots & \\ s_i & , \text{ für } u_0 \in [\sum_{j=1}^{i-1} \mu_j^{(0)}, \sum_{j=1}^i \mu_j^{(0)}) \\ \vdots & \\ s_k & , \text{ für } u_0 \in [\sum_{j=1}^{k-1} \mu_j^{(0)}, 1] \end{cases}$$

Die Initiativfunktion ist:

- ▶ stückweise konstant und
- ▶ Länge des Intervalls, auf dem gilt $\psi(u_0) = s_i$, ist $\mu_i^{(0)}$

Definition der Update-Funktion $\phi : S \times [0, 1] \mapsto S$:

$$\phi(s_i, u_k) = \begin{cases} s_1 & , \text{ für } u_k \in [0, P_{i,1}) \\ s_2 & , \text{ für } u_k \in [P_{i,1}, P_{i,1} + P_{i,2}) \\ \vdots & \\ s_j & , \text{ für } u_k \in [\sum_{l=1}^{j-1} P_{i,l}, \sum_{l=1}^j P_{i,l}) \\ \vdots & \\ s_k & , \text{ für } u_k \in [\sum_{l=1}^{k-1} P_{i,l}, 1] \end{cases}$$

Die Update-Funktion ist für fixiertes $s_j \in S$ ebenfalls:

- ▶ stückweise konstant und
- ▶ Länge des Intervalls, auf dem gilt $\phi(s_i, u_k) = s_j$, ist $P_{i,j}$

Insgesamt erhält man also mit:

$$\begin{aligned}X_0 &= \psi(U_0) \\X_1 &= \phi(X_0, U_1) \\X_2 &= \phi(X_1, U_2) \\&\vdots \\X_n &= \phi(X_{n-1}, U_n)\end{aligned}$$

eine simulierte Markovkette.

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

Eine Anwendung

Beispiele

Hard-Core Model

Zufällige q -Färbungen

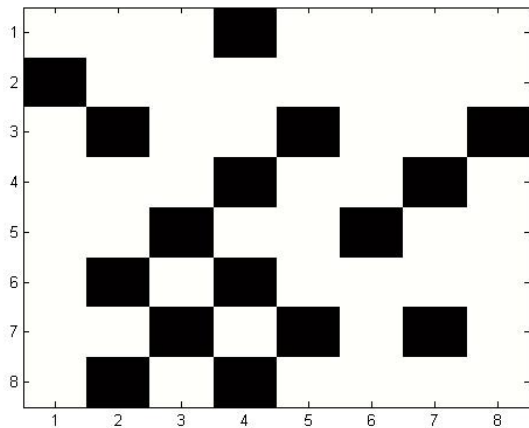
- ▶ Gegeben: Wahrscheinlichkeitsverteilung π auf $S = \{s_1, \dots, s_k\}$
- ▶ Problem: Simulation einer Zufallsvariablen X mit Werten in S und Verteilung π

Sei U eine auf $[0, 1]$ uniform-verteilte, simulierte Zufallsvariable.

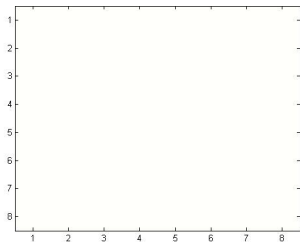
$$X = \begin{cases} s_1 & , \text{ für } U \in [0, \pi(s_1)) \\ s_2 & , \text{ für } U \in [\pi(s_1), \pi(s_1) + \pi(s_2)) \\ \vdots & \\ s_i & , \text{ für } U \in [\sum_{j=1}^{i-1} \pi(s_j), \sum_{j=1}^i \pi(s_j)) \\ \vdots & \\ s_k & , \text{ für } U \in [\sum_{j=1}^{k-1} \pi(s_j), 1] \end{cases}$$

Für **großes** k unmöglich!

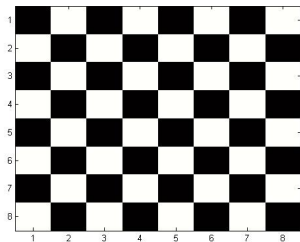
Beispiel: Hard-Core Model ($n = 8$)



Beispiel: Hard-Core Model ($n = 8$)



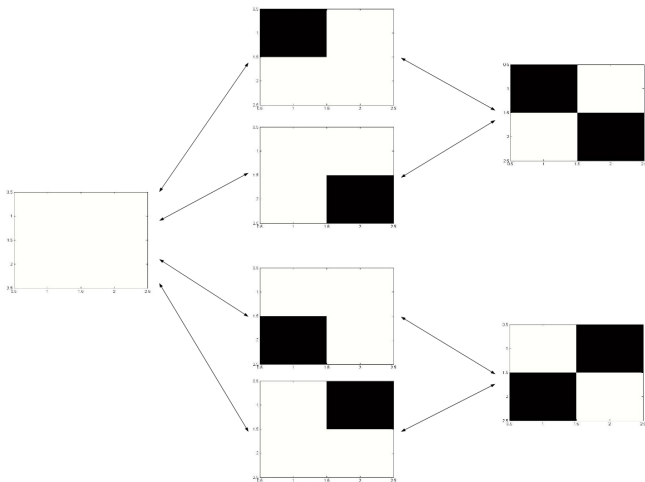
$$\leq 2^{n^2}$$



$$\geq 2^{n^2/2}$$

$2^{\Theta(n^2)}$ viele zulässige Konfigurationen.

Beispiel: Hard-Core Model ($n = 2$) als Markovkette



Gesucht: Übergangsmatrix P , sodass
 P irreduzibel, aperiodisch mit invarianter Verteilung π

$\Rightarrow X$ kann z.B. als X_{1000} gewählt werden bei simulierter
Markovkette $(X_n)_{n \geq 0}$ Markov(λ, P), wegen Konvergenz

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

Eine Anwendung

Beispiele

Hard-Core Model

Zufällige q -Färbungen

Gegeben: Zufallsvariable X auf Zustandsraum $S = \{s_1, s_2, \dots, s_k\}$
und Verteilung $\pi = (\pi_1, \pi_2, \dots, \pi_k)$

Gesucht: Übergangsmatrix P mit Grenzverteilung π

- ▶ Wähle eine Übergangsmatrix R auf S
- ▶ Setze

$$P_{i,j} = \begin{cases} R_{i,j} \wedge \left(\frac{\pi_j}{\pi_i} \cdot R_{j,i}\right) & , \text{ für } i \neq j \\ 1 - \sum_{j \neq i} P_{i,j} & , \text{ für } i = j \end{cases}$$

Für die so definierte Übergangsmatrix P gilt Invarianz von π :

$$\pi_i \cdot P_{i,j} = \pi_j \cdot P_{j,i}$$

Damit P irreduzibel ist, muss gelten:

$$\forall i \neq j \exists n \geq 1, \{t_0, t_1, \dots, t_n\} \subset S : \\ t_0 = s_i \wedge t_n = s_j \wedge R_{t_{k-1}, t_k} > 0 \wedge R_{t_k, t_{k-1}} > 0 \quad \forall 1 \leq k \leq n$$

In der Praxis wählt man R wie folgt:

- ▶ Wähle einen nicht-orientierten Graphen G auf S , so dass
 $\forall i \neq j \exists n \geq 1, \{t_1, t_2, \dots, t_{n+1}\} \subset S : t_1 = s_i \wedge t_{n+1} = s_j$
 $\wedge \forall 1 \leq k \leq n : (t_k, t_{k+1}) \in G$
- ▶ Wähle R so, dass : $R_{i,j} > 0 \Leftrightarrow (s_i, s_j) \in G$
 $\Rightarrow P$ ist irreduzibel

Es gibt zwei klassische Methoden R nach der Wahl von G zu wählen:

- ▶ den Gibbs-Sampler und
- ▶ den Metropolis-Algorithmus

Der Gibbs-Sampler ist wie folgt definiert:

$$R_{i,j} = \begin{cases} (\sum_{l:(s_i, s_l) \in G} \pi_l)^{-1} \cdot \pi_j & , \text{ für } (s_i, s_j) \in G \\ 0 & , \text{ für } (s_i, s_j) \notin G \end{cases}$$

Für den Metropolis-Algorithmus gilt:

$$R_{i,j} = \begin{cases} (n_i)^{-1} & , \text{ für } (s_i, s_j) \in G \\ 0 & , \text{ für } (s_i, s_j) \notin G \end{cases}$$

$$\text{mit } n_i = |\{l : (s_i, s_l) \in G\}|$$

Dabei ist die Idee, G so zu wählen, dass n_i viel kleiner als k (Kardinalität von S) ist, damit MCMC effizienter als das direkte Simulieren ist.

Nach der Wahl von R muss nun noch die Simulation einer Markovkette mit Übergangsmatrix R mit dem **Hastings-Algorithmus** kombiniert werden. So erhält man eine Markovkette mit der Übergangsmatrix P :

- ▶ Für festes $X_n = s_i$ simuliere man Y_n nach $R_{i,\bullet}$
- ▶ Mit simuliertem $Y_n = s_j$ wird in einer neuen unabhängigen Simulation X_{n+1} nach

$$X_{n+1} = \begin{cases} Y_n & , \text{ mit Wahrscheinlichkeit } \frac{\pi_j \cdot R_{j,i}}{\pi_i \cdot R_{i,j}} \wedge 1 \\ X_n & , \text{ mit Wahrscheinlichkeit } 1 - \frac{\pi_j \cdot R_{j,i}}{\pi_i \cdot R_{i,j}} \wedge 1 \end{cases}$$

erzeugt. Also mit $U \sim U([0, 1])$:

$$X_{n+1} = \mathbf{1} \left\{ U \leq \frac{\pi_j \cdot R_{j,i}}{\pi_i \cdot R_{i,j}} \right\} \cdot Y_n + \mathbf{1} \left\{ U > \frac{\pi_j \cdot R_{j,i}}{\pi_i \cdot R_{i,j}} \right\} \cdot X_n$$

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

Eine Anwendung

Beispiele

Hard-Core Model

Zufällige q -Färbungen

- ▶ Zustandsraum S der Form $S = E^\Lambda$ mit $s_j \in S$ als Zuteilung:
 $m \in \Lambda \mapsto s_j(m) \in E$
- ▶ Λ ist der Raum der Punkte oder Pixel eines diskretisierten Bildes
- ▶ E ist der Raum der Farbabstufungen
- ▶ in der Regel ist Λ sehr groß und E relativ klein, also ist S insgesamt sehr groß

- ▶ für $m \in \Lambda$ ist also $X_n(m)$ eine E-wertige Zufallsvariable
- ▶ zwischen den Zeitpunkten n und $n+1$ wird nur eine Komponente von X geändert, d.h.
 $\exists m \in \Lambda : X_{n+1} \stackrel{m}{\sim} X_n$
- ▶ alle anderen Komponenten bleiben unverändert:
 $\forall m' \neq m : X_{n+1}(m') = X_n(m')$
- ▶ man wählt den Graphen G so, dass
 $(s_i, s_j) \in G \Leftrightarrow s_i$ und s_j unterscheiden sich an genau einer Stelle
- ▶ Simulation der Markov-Kette wird dabei beinhalten, in jedem Schritt ein $m \in \Lambda$ zu wählen, an dem der Wert von X_n modifiziert wird

Wie soll sich $X_n(m)$ zu $X_{n+1}(m)$ verändern?

- ▶ dies ist beschrieben durch eine Übergangsmatrix $P^{(m)}$ mit der Eigenschaft $P_{i,j}^{(m)} = 0$, wenn nicht $s_i \stackrel{m}{\sim} s_j$
- ▶ also alle Komponenten von s_i und s_j bleiben unverändert bis auf die Komponente m
- ▶ wir fordern Invarianz von π unter $P^{(m)}$:
$$\pi_i \cdot P_{i,j}^{(m)} = \pi_j \cdot P_{j,i}^{(m)} \forall s_i, s_j \in S, m \in \Lambda$$
- ▶ gegebene Übergangsmatrix $R^{(m)}$ mit $R_{i,j}^{(m)} \neq 0 \Leftrightarrow s_i \stackrel{m}{\sim} s_j$
- ▶ dann gilt: $\pi_i \cdot P_{i,j}^{(m)} = (\pi_i \cdot R_{i,j}^{(m)}) \wedge (\pi_j \cdot R_{j,i}^{(m)})$ und
$$P_{i,i}^{(m)} = 1 - \sum_{j \neq i} P_{i,j}^{(m)} \geq 0$$

Der Hastings-Algorithmus ist also:

- ▶ für gegebenes $X_n = s_i$ simulieren wir Y_n nach $R_{i,\bullet}^{(m)}$
- ▶ mit so simuliertem $Y_n = s_j$ wähle man X_{n+1} nach:

$$X_{n+1} = \begin{cases} Y_n & , \text{ mit Wahrscheinlichkeit } \frac{\pi_j \cdot R_{j,i}^{(m)}}{\pi_i \cdot R_{i,j}^{(m)}} \wedge 1 \\ X_n & , \text{ mit Wahrscheinlichkeit } 1 - \frac{\pi_j \cdot R_{j,i}^{(m)}}{\pi_i \cdot R_{i,j}^{(m)}} \wedge 1 \end{cases}$$

Für den Gibbs-Sampler gilt:

- ▶ $P_{i,j}^{(m)} = R_{i,j}^{(m)} = (\sum_{s_l \sim s_i} \pi_l)^{-1} \cdot \pi_j$, falls $s_i \stackrel{m}{\sim} s_j$
- ▶ $X_{n+1}(m)$ wird nach π unter den gegebenen anderen Komponenten gewählt

Für den Metropolis-Algorithmus dagegen:

- ▶ $R_{i,j}^{(m)} = (|E| - 1)^{-1}$, falls $s_i \stackrel{m}{\sim} s_j$
- ▶ $P_{i,j}^{(m)} = (|E| - 1)^{-1} \cdot (\frac{\pi_j}{\pi_i} \wedge 1)$, falls $s_i \stackrel{m}{\sim} s_j$ und $s_i \neq s_j$
- ▶ man wählt zunächst gleichverteilt auf $E \setminus \{X_n(m)\}$ einen Wert $s_j^{(m)}$ am Punkt m
- ▶ dann ist $X_{n+1}(m) = s_j^{(m)}$ mit Wahrscheinlichkeit $\frac{\pi_j}{\pi_i} \wedge 1$

Wie kommt man zu den Punkten $m \in \Lambda$?

- ▶ eine Möglichkeit wäre es Λ in einer gewissen Reihenfolge abzuarbeiten und dies immer zu wiederholen
- ▶ oder man wählt zu jedem Zeitpunkt n zufällig ein $m \in \Lambda$ nach einer Gleichverteilung auf Λ aus
 - ▶ dann ist $\{X_n; n \geq 1\}$ eine homogene Markovkette mit Übergangsmatrix $P = |\Lambda|^{-1} \sum_{m \in \Lambda} P^{(m)}$
 - ▶ dann gilt auch $\pi_i \cdot P_{i,j} = \pi_j \cdot P_{j,i}$, P ist irreduzibel und somit π ist die eindeutige Grenzverteilung der Markov-Kette $\{X_n; n \geq 1\}$

Simulation einer Markovkette

Motivation

Markov Chain Monte Carlo

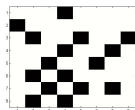
Eine Anwendung

Beispiele

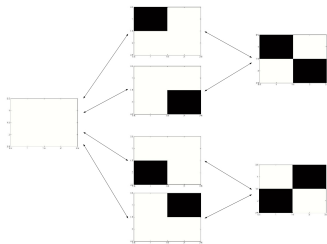
Hard-Core Model

Zufällige q-Färbungen

Beispiel: Allgemeines Hard-Core Model



- ▶ $E = \{0, 1\}$
- ▶ $\Lambda = \{1, \dots, n\}^2$
- ▶ $s \in S \subset E^\Lambda$ ist eine zulässige Konfiguration.



Definiere Graphen G mit Knotenmenge S und Kanten $(s_i, s_j) \in S^2$
gdw. $s_i \stackrel{m}{\sim} s_j$, d.h

$$\exists m \in \Lambda : (s_i(m) \neq s_j(m) \wedge \forall m' \neq m : s_i(m') = s_j(m'))$$

Wahrscheinlichkeitsmaß π :

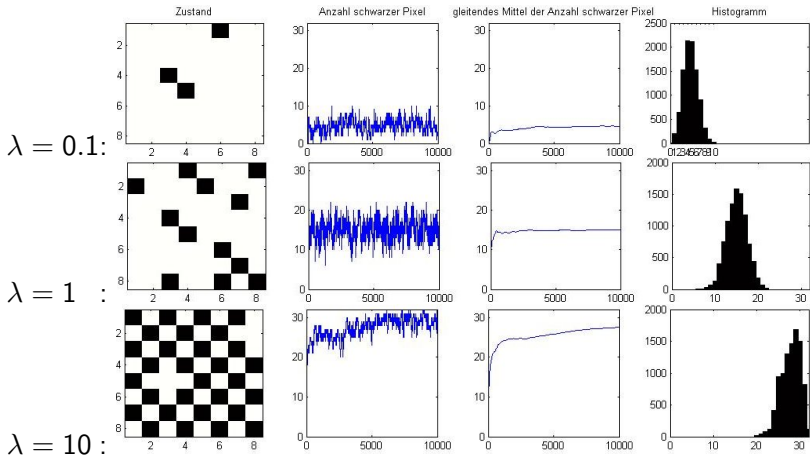
$$\pi_s = \frac{\lambda^{n(s)}}{Z_\lambda} \quad \text{für alle } s \in S$$

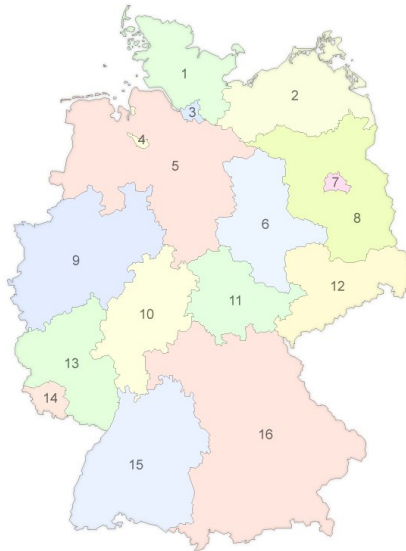
mit

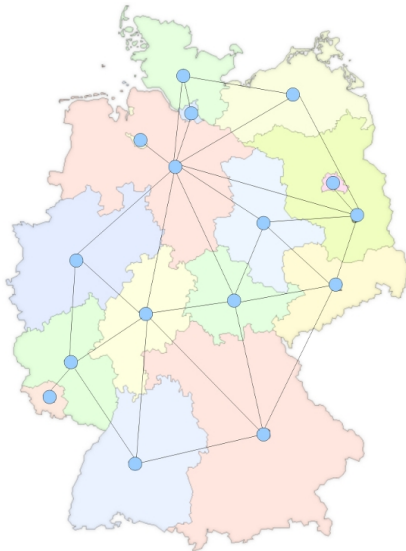
- ▶ Parameter $\lambda > 0$
- ▶ Anzahl von Einsen $n(s) = \sum_{m \in \Lambda} s(m)$
- ▶ Normalisierungsfaktor $Z_\lambda = \sum_{s \in S} \lambda^{n(s)}$

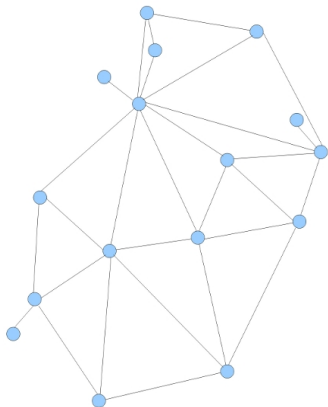
MCMC anwenden:

- ▶ Gibbs Sampler
- ▶ Metropolis-Algorithmus
- ▶ Hastings-Algorithmus
- ▶ irreduzibel ?
- ▶ aperiodisch ?









Beispiel: Zufällige q-Färbungen

- ▶ Graph $H = (V, F)$, V Knotenmenge, $F \subset V^2$ Kanten
- ▶ $E = [q] = \{1, \dots, q\}$
- ▶ $s \in E^V$ Färbung
- ▶ s ist eine gültige Färbung, falls zwei benachbarte Knoten unterschiedlich gefärbt sind, d.h.

$$\forall f = (v, w) \in F : s(v) \neq s(w)$$

- ▶ Sei S die Menge aller gültigen Färbungen

Definiere Graphen G mit Knotenmenge S und Kanten $(s_i, s_j) \in S^2$
gdw. $s_i \stackrel{m}{\sim} s_j$, d.h

$$\exists m \in \Lambda : (s_i(m) \neq s_j(m) \wedge \forall m' \neq m : s_i(m') = s_j(m'))$$

Wahrscheinlichkeitsmaß π :

$$\pi_s = \frac{n(s)^{-1}}{Z_q} \quad \text{für alle } s \in S$$

mit

- ▶ Parameter $q \in \mathbb{N}$
- ▶ Anzahl verschiedener Farben $n(s) = |s(V)|$
- ▶ Normalisierungsfaktor $Z_q = \sum_{s \in S} n(s)^{-1}$

MCMC anwenden:

- ▶ Gibbs Sampler
- ▶ Metropolis-Algorithmus
- ▶ Hastings-Algorithmus
- ▶ irreduzibel ?
- ▶ aperiodisch ?

- ▶ Falls $q \geq \Delta + 1$, so existiert eine gültige Färbung.
- ▶ Falls $q \geq \Delta + 2$, so ist der Graph von zulässigen Färbungen verbunden und damit ist die Markovkette irreduzibel.
- ▶ Falls $q \geq \Delta + 3$, so ist die Markovkette aperiodisch.

Dabei ist Δ der Maximalgrad im Graphen H .

Thank you!