Markov-Ketten-Monte-Carlo-Verfahren

Anton Klimovsky

21. Juli 2014

Strichprobenerzeugung aus einer Verteilung (das Samplen). Markov-Ketten-Monte-Carlo-Verfahren. Metropolis-Hastings-Algorithmus. Gibbs-Sampler.

Problem: "Strichprobenerzeugung aus einer Verteilung[1]" Gegeben ist eine Verteilung $\pi \in \mathcal{M}_1(S)$[2] auf einem Zustandsraum S. Wie erzeugt man die (pseudo)Zuffalsvariablen, die die Verteilung π haben?

[1] Das nennt man auch das "Samplen aus einer Verteilung"

[2] **Notation:** $\mathcal{M}_1(S)$ ist die Menge aller *Wahrscheinlichkeitsmaßen* auf S, d.h.

$$\mathcal{M}_1(S) = \{\pi \in \mathbb{R}^S \colon \sum_{s \in S} \pi_s = 1\}. \quad (1)$$

Das $\mathcal{M}$ steht in (1) für das Maß und die 1 steht für die Bedingung, dass "das Gesamtmaß von S" = "die totale Wahrscheinlichkeit" = 1 ist.

Wofür? Simulation eines w-theoretischen Modells, das durch die Verteilung π beschrieben ist. Diese können sehr kompliziert sein: sehr große hoch-dimensionale Zustandsräume S mit einer multivariaten Verteilung π darauf. Dafür braucht man effiziente Algorithmen.

Idee (Markov-Ketten-Monte-Carlo-Verfahren (=MCMC)). Gegeben ist eine *Zielverteilung* $\pi \in \mathcal{M}_1(S)$, die wir simulieren möchten. Konstruiere eine "gute" *Markov-Kette*, die die Verteilung π als Gleichgewicht hat und "schnell" dagegen konvergiert.

Erste Skizze des Algorithmus. Sei $\varepsilon > 0$ der gegebene *Annäherungsfehler*.

1. Sei $\mu_0 \in \mathcal{M}_1(S)$ eine *Anfangsverteilung*. (Diese kann man oft frei wählen.)
2. Sei Q eine Übergangsmatrix. (Diese müssen wir noch genau spezifizieren.) Simuliere $n \in \mathbb{N}$ Schritten der Markov-Kette

$$\pi Q \rightsquigarrow \pi Q^2 \rightsquigarrow \ldots \rightsquigarrow \pi Q^n, \quad (2)$$

so dass

$$d_{\mathrm{TV}}(\pi Q^n, \pi) < \epsilon \quad (3)$$

In Worten bedeutet (3), dass nach n Schritten die Abweichung (in Sinne der totalen Variation) der Verteilung der Q-Markov-Kette von der Zielverteilung π kleiner als ε ist. Damit können wir $\mu_0 Q^n$ als eine *Approximation* für π benutzen

$$\mu_0 Q^n \approx \pi. \quad (4)$$

Interessanterweise stellt sich heraus, dass es in vielen Fällen leichter ist so eine Markov-Kette (2) zu simulieren, als direkt die Verteilung π. Das ist so, weil man die Zielverteilung π oft nicht explizit kennt.

VERTEILUNGEN MODULO NORMIERUNG. Viele Verteilungen $\mu \in \mathcal{M}_1(S)$ sind von der folgenden Bauart[3]

$$\mu(x) = \frac{f(x)}{C}, \quad x \in S \tag{5}$$

wobei $f\colon S \to \mathbb{R}$ ist eine "schöne" Funktion und C eine aufwendig auszurechnende Konstante ist. Konstante C ist oft unbekannt. Notwendigerweise gilt

$$C = \sum_{y \in S} f(y), \tag{6}$$

sonst ist $\mu \notin \mathcal{M}_1(S)$. (Warum?) Sogar für sehr schöne Funktionen f kann die Summe (6) schwer auszurechnen sein[4].

[3] Für stetige Verteilungen gilt dies auch. Z.b. die Gauß'sche Verteilung: $\frac{\mathrm{d}\mu(x)}{\mathrm{d}x} = \frac{1}{C}\exp(-x^2/2)$. Dann ist $C = \int_{\mathbb{R}} \exp(-x^2/2)\mathrm{d}x$. Wie kann man dieses Integral ausrechnen?

[4] Die Dichte der Gauß'schen Verteilung hat auch die Bauart (5): $p(x) = \frac{1}{C}\exp\{-x^2/2\}$. Wie kann man die Normierungskonstante $C = \int_{-\infty}^{\infty} \exp(-x^2/2)\mathrm{d}x$ ausrechnen (ohne dass man aus der Definition der Gauß'schen Verteilung weißt, dass $C = \sqrt{2\pi}$)?

BEISPIEL (FEHLERKORREKTUR). Betrachten wir das Problem der Fehlerkorrektur bei der Datenübertragung: IhrE FreundIn schickt Ihnen eine Nachricht

$$x \in S = \{\text{"Hallo"}, \text{"Tschüss"}, \text{"Wie war Dein Tag so?"}, \ldots\}. \tag{7}$$

Das, was Sie erhalten, ist leider eine etwas verzerrte/verrauschte Version davon: $z \in S$. Aus Erfahrung wissen Sie, dass

(a) die Nachrichten aus der Verteilung $p \in \mathcal{M}_1(S)$ kommen. (Sie wissen, welche Sätze typisch für IhreN FreundIn sind und welche weniger typisch.)

(b) Außerdem haben Sie etwas Erfahrung mit dem Datenübetragungskanal und dem *Rauschen*. (Sie haben früher mehrmals nachgefragt, was Ihr Gegenüber gemeint hatte, weil Sie sie/ihn akustisch nicht verstanden haben). Damit haben sie eine Idee über die Verteilung $p(\cdot \mid x) \in \mathcal{M}_1(S)$. In Worten ist diese die *bedingte Verteilung der empfangenen Nachricht gegeben die gesendete Nachricht* x.

Anhand von den Daten (a) und (b) können Sie jetzt versuchen die ursprünglich gesendete Nachricht wiederherzustellen.[5] Und zwar stochastisch: Man kann die Wahrscheinlichkeit, dass die Nachricht x gesendet wurde gegeben die empfangene Nachricht z ausrechnen! Die Bayes-Regel liefert nämlich Folgendes

[5] Unser Gehirn macht einen recht guten Job dabei.

$$\boxed{p(x \mid z) = \frac{p(z \mid x)p(x)}{\sum_{y \in S} p(z \mid y)p(y)}, \quad x, z \in S.} \tag{8}$$

Somit hat die Bedingte Verteilung (8) die Bauart (5). Die Normierungskonstante $C = \sum_{y \in S} p(z \mid y)p(y)$ kann sehr schwierig auszurechnen sein (sowohl analytisch in geschlossener Form, als auch numerisch, wenn der Zustandsraum S groß ist.).

Allgemeine Formulierung des MCMC-Verfahrens

VORSCHLAGSMATRIX. Sei $\tilde{Q} \in \mathbb{R}^{S\times S}$ eine irreduzieble und symmetrische Übergangsmatrix.

BEMERKUNG. Die Markov-Kette mit der Übergangsmatrix $\tilde{Q}$ soll so einfach wie möglich zu Simulieren sein.[6] Die Vorschlagsmatrix hat normalerweise nichts mit der Zielverteilung π zu tun! Insb. ist π keine Gleichgewichtsverteilung der Markov-Kette $\tilde{Q}$! Deswegen müssen wir die Übergangsmatrix $\tilde{Q}$ etwas korrigieren, so dass π zur Gleichgewichtsverteilung wird.

[6] Z.b. die Übergangsmatrix einer Irrfahrt auf S, falls S ein Graph ist.

AKZEPTANZWAHRSCHEINLICHKEITEN. Für $s_i, s_j \in S$ wählen wir die *Akzeptanzwahrscheinlichkeiten* $\alpha_{i,j} \in (0;1]$ und definieren eine neue Matrix $Q \in \mathbb{R}^{S\times S}$ durch

$$\boxed{q_{i,j} = \begin{cases} \tilde{q}_{i,j}\alpha_{i,j}, & i \neq j \\ 1 - \sum_{i\neq j} q_{i,j}, & i = j. \end{cases}} \tag{9}$$

Lemma 0.1. *Sei $\tilde{Q}$ irreduziebel und symmetrisch und es gelte*

$$\boxed{\mu_i\alpha_{i,j} = \mu_j\alpha_{j,i}}^{7} \tag{10}$$

Dann ist Q irreduziebel und reversibel bzgl. μ. Wenn $\tilde{Q}$ zusätzlich aperiodisch, ist dann es auch Q.

[7] Vergleiche Bedingung (10) mit der Reversibilitätsbedingung.

BEMERKUNG. Lemma 0.1 liefert eine theoretische Begründung der Approximation (3). (Warum?)

BEMERKUNG. Die Akzeptanzwahrscheinlichkeiten hängen von der Zielverteilung ab.

Beweis von Lemma 0.1. □

Jetzt betrachten wir zwei konkrete Ansätze für die Akzeptanzwahrscheinlichkeiten.

Metropolis-Hastings Algorithmus

Durch die Wahl

$$\boxed{\alpha_{i,j} = \min\left\{1, \frac{\mu_j}{\mu_i}\right\}} \tag{11}$$

erhalten wir den *Metropolis-Hastings Algorithmus*.

Im speziellen Fall wurde dieses Verfahren von Nicholas Metropolis gemeinsam mit den zwei verheirateten Paaren mit den Familiennamen Rosenbluth und Teller erfunden [8]. Angeblich hat Nicolas Metropolis diesen Algorithmus im Rahmen des Manhattan Projects entwickelt. Doe allgemeinere Version (11) ist von W.K. Hastings[9] entwickelt worden.

[8] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953

[9] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1): 97–109, 1970

Bemerkung. In (11) ist die Akzeptanzwahrscheinlichkeit immer $= 1$, falls $\mu_j > \mu_i$. Wir akzeptieren also immer die Vorschläge $i \rightsquigarrow j$, die in die *wahrscheinlichere Zustände* führen. Trotzdem hat man eine Chance auch in die *weniger wahrscheinlicheren* Zustände überzugehen. Dies geschieht, falls $\mu_j < \mu_i$ mit der Wahrscheinlichkeit μ_j/μ_i. Deswegen verbringt die Markov-Kette mehr Zeit in den wahrscheinlicheren (bzgl. μ) Zuständen und weniger Zeit in den unwahrscheinlicheren Zuständen. Dies ist eine intuitive (aber nicht rigorose) Begründung, warum man die gewünschte Approximation (3) erwarten kann.

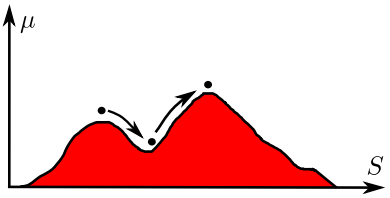

Abbildung 1: Annahme der Vorschlägen im Markov-Ketten-Monte-Carlo-Verfahren: nach unten mit $\mathbb{P} < 1$, nach oben mit $\mathbb{P} = 1$. Symbol • repräsentiert den Zustand der Markov-Kette. Die Fläche unter der Verteilung μ ist in rot dargestellt.

Gibbs-Sampler (= Wärmebad-Algorithmus)

Durch die Wahl

$$\boxed{\alpha_{i,j} = \frac{\mu_i}{\mu_i + \mu_j}} \tag{12}$$

erhalten wir den sogenannten *Gibbs-Sampler Algorithmus*[10].

[10] Genannt nach einem US Amerikanischen Physiker

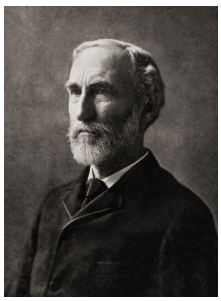

Abbildung 2: Josiah Willard Gibbs (1839–1903)

Bemerkung. In beiden Fällen (11) und (12) brauchen wir lediglich nur den Quotienten $\frac{\mu_j}{\mu_i}$ ausrechnen zu können. Deswegen brauchen wir die Normierungskonstante C in (6) nicht zu wissen, da sie sich in den Quotienten kürzt!

Bemerkung. MCMC ist ein sehr verbreitetes Werkzeug[11,12,13] der numerischen Simulation und findet zahlreiche Anwendungen in der Chemie, Physik, Biologie, Informatik, usw.

Literatur

Pierre Brémaud. *Markov chains*, volume 31 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1999. ISBN 0-387-98509-3. DOI: 10.1007/978-1-4757-3124-8. URL `http://dx.doi.org/10.1007/978-1-4757-3124-8`. Gibbs fields, Monte Carlo simulation, and queues.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[11] Thomas Müller-Gronbach, Erich Novak, and Klaus Ritter. *Monte Carlo-Algorithmen*. Springer, 2012

[12] Gerhard Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer, 2003

[13] Pierre Brémaud. *Markov chains*, volume 31 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1999. ISBN 0-387-98509-3. DOI: 10.1007/978-1-4757-3124-8. URL `http://dx.doi.org/10.1007/978-1-4757-3124-8`. Gibbs fields, Monte Carlo simulation, and queues

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21: 1087–1092, 1953.

Thomas Müller-Gronbach, Erich Novak, and Klaus Ritter. *Monte Carlo-Algorithmen*. Springer, 2012.

Gerhard Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer, 2003.