Hinweise zu den Programmierprojekten

- Abgabe bis 17.6. im CIP-Pool. Bitte vereinbaren Sie rechtzeitig einen Termin.
- Schreiben Sie Ihre Lösungen bitte direkt in die Notebooks mit den Projektbeschreibungen (download auf der Vorlesungshomepage), oder kopieren Sie die Aufgabenstellungen zu den Lösungen.
- Jedes der Projekte zählt vierfach. Wir empfehlen die Bearbeitung von einem der beiden Projekte - natürlich können Sie auch beide abgeben.

Projekt 1: Simulation des Isingmodells

Einleitung

Wir betrachten ein großes Rechteck

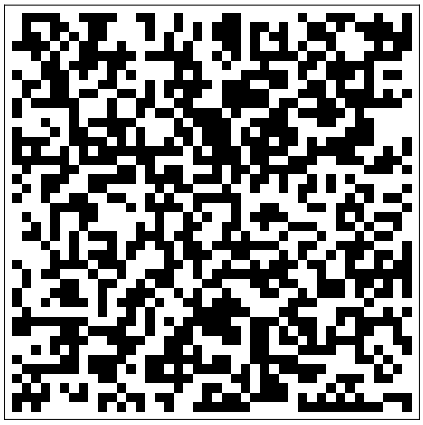
$$\Lambda = \{0,\ 1,\ \ldots,\ 2\,d + 1\} \times \{0,\ 1,\ \ldots,\ 2\,d + 1\}$$

im $\mathbb{Z}^2$, und den Konfigurationsraum

$$S\ =\ \{0,\ 1\}^{\Lambda}\ =\ \left\{x = \left(x_{i,j}\right)_{(i,j)\in\Lambda} :\ x_{i,j} \in \{0,\ 1\} \text{ für } (i,\ j) \in \Lambda\right\}$$

In *Mathematica* können wir ein Element aus S als zweidimensionales Feld von Nullen und Einsen darstellen, und mithilfe von ArrayPlot graphisch darstellen. Eine zufällige (gleichverteilte) Konfiguration aus S erhält man beispielsweise mit:

```
d = 20; x = Table[RandomInteger[], {i, 0, 2 d + 1}, {j, 0, 2 d + 1}]; ArrayPlot[x]
```



Wir wollen nun Wahrscheinlichkeitsverteilungen auf S betrachten, bei denen im Gegensatz zur Gleichverteilung benachbarte Pixel vorzugsweise im selben Zustand sind. Dazu definieren wir die "Energie" einer Konfiguration x als

$$H(x)\ =\ \sum \left|x_{i,j} - x_{k,l}\right|^2\ ,$$

wobei die Summe über alle $i,\ j,\ k,\ l = 0,\ 1,\ \ldots,\ 2\,d + 1$ mit $\|(i,\ j) - (k,\ l)\| = 1$ läuft. Jedes Paar von benachbarten Pixeln wird also **zweimal** berücksichtigt. Man kann $H(x)$ auch als ein Maß für die Größe der Grenze zwischen der schwarzen und der weißen Region in der Konfiguration x interpretieren. Wir definieren nun Wahrscheinlichkeitsverteilungen, unter denen Konfigurationen mit kleiner Energie bevorzugt werden. Dazu beschränken wir uns auf Konfigurationen, für die alle Pixel am Rand des Rechtecks schwarz gefärbt sind. Sei

$$S^+ = \{x \in S\ :\ x_{i,j} = 1 \text{ für alle } (i,\ j) \text{ mit } i = 0 \text{ oder } i = 2\,d + 1 \text{ oder } j = 0 \text{ oder } j = 2\,d + 1\}\,.$$

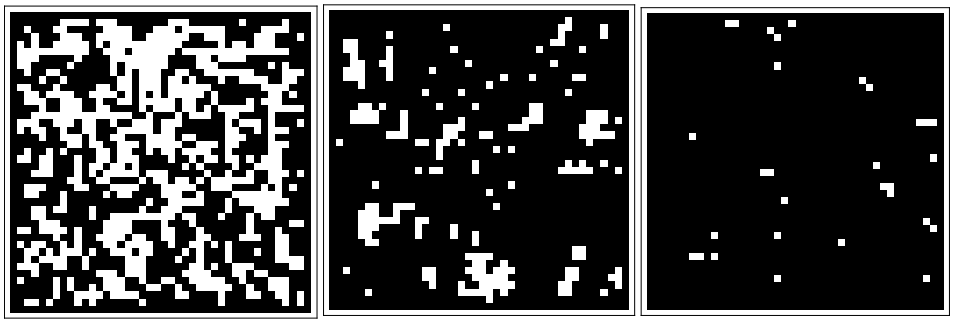
Für $\beta \geq 0$ betrachten wir die Wahrscheinlichkeitsverteilung μ_β auf S^+ mit Massenfunktion

$$\mu_\beta(x)\ =\ \frac{1}{Z_\beta}\, e^{-\beta \cdot H(x)}\quad ,\qquad x \in S^+,$$

wobei

$$Z_\beta = \sum_{x \in S^+} e^{-\beta \cdot H(x)}$$

eine Konstante ist, die μ_β zur Massenfunktion einer Wahrscheinlichkeitsverteilung normiert. Je größer der Wert von β ist, umso stärker werden Konfigurationen mit vielen gleichfarbigen Nachbarpixeln unter der Verteilung μ_β bevorzugt. Insbesondere ist μ_0 die Gleichverteilung auf S^+, während für $\beta \to \infty$ die Wahrscheinlichkeit, daß alle Pixel wie der Rand schwarz sind, gegen 1 konvergiert. Die folgenden Grafiken zeigen Stichproben von μ_β für $d = 20$ und β=0.2, 0.44 bzw. 0.6 :



Hintergrund

Das beschriebene Modell ist eines der fundamentalen Modelle der statistischen Physik, und wurde von Ising 1925 eingeführt. Heute spielt das Ising-Modell auch eine wichtige Rolle in vielen anderen Bereichen, zum Beispiel als A-priori-Verteilung in Bayesschen Modellen der Bildverarbeitung. In der physikalischen Interpretation beschreibt das Modell (auf stark vereinfachte Weise) ein ferromagnetisches Material. An jeder Position des Gitters Λ befindet sich ein Atom, dessen Spin nach oben (+1) oder nach unten (0) zeigen kann. Der Parameter β ist die inverse Temperatur. Bei hoher Temperatur sind die Spins ungeordnet, bei niedriger Temperatur zeigen Sie vorzugsweise in eine Richtung.
Sei $m_\beta(d)$ der Erwartungswert unter μ_β des Wertes $x_{d,d}$ in der Mitte des Rechtecks Λ. Ein tiefes Resultat von Onsager besagt, daß

$$\lim_{d\to\infty} m_\beta(d) = \frac{1}{2} \qquad \text{falls } \sinh(2\beta) \le 1, \text{ während}$$
$$\lim_{d\to\infty} m_\beta(d) = \frac{1}{2} + \frac{1}{2}\left(1 - \sinh(2\beta)^{-4}\right)^{1/8} > \frac{1}{2} \qquad \text{falls } \sinh(2\beta) > 1.$$

Im Fall $\sinh(2\beta)>1$ beeinflussen also die Randwerte den Wert in der Mitte sogar im Limes $d \to \infty$, während für kleinere β der Einfluss der Randwerte auf den Mittelpunkt für $d \to \infty$ verschwindet. Im physikalischen Modell bedeutet dies, daß bei der kritischen inversen Temperatur

$$\beta_{\text{crit}} = \frac{1}{2}\,\text{ArcSinh}(1) \approx 0.44$$

ein Phasenübergang stattfindet.

Simulation des Ising-Modells

Ein einfaches Acceptance-Rejection-Verfahren ist zur Simulation des Isingmodells schon für relativ kleine Werte von d nicht mehr praktikabel. Man verwendet daher meist Markov Chain Monte Carlo Verfahren. Tatsächlich ist die beim Gibbs-Sampler simulierte Markovkette auf dem Raum S^+ sehr ähnlich zu der physikalischen Glauberdynamik des Spinsystems, und wurde ursprünglich auch von dieser motiviert. Um einen Gibbs-Sampler effizient implementieren zu können, unterteilen wir das Gitter Λ (ohne Rand) schachbrettartig in gerade und ungerade Felder:

$$\Lambda^g = \{(i, j) : i, j = 1, 2, \ldots, 2d \text{ mit } i + j \text{ gerade}\},$$
$$\Lambda^u = \{(i, j) : i, j = 1, 2, \ldots, 2d \text{ mit } i + j \text{ ungerade}\},$$

und setzen

$$x^g = \left(x_{i,j} \mid (i,\ j) \in \Lambda^g\right), \qquad x^u = \left(x_{i,j} \mid (i,\ j) \in \Lambda^u\right).$$

Da die Konfigurationen $x \in S^+$ auf dem Rand von Λ überall den Wert 1 annehmen, können wir x identifizieren mit dem Paar (x^g, x^u). Zudem haben die bedingten Verteilungen von x^g gegeben den Wert von x^u und umgekehrt eine einfache Form. Bezeichnet man mit

$$s_{i,j} = x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}$$

die Anzahl der schwarzen Pixel in der Umgebung von $(i,\ j)$, dann gilt

$$\mu_\beta\left(x^g \mid x^u\right) = \prod_{(i,j)\in\Lambda^g} \nu_\beta\left(x_{i,j} \mid s_{i,j}\right), \qquad \text{und} \tag{1}$$

$$\mu_\beta\left(x^u \mid x^g\right) = \prod_{(i,j)\in\Lambda^u} \nu_\beta\left(x_{i,j} \mid s_{i,j}\right), \tag{2}$$

wobei $\nu_\beta(\cdot \mid s)$ die durch

$$\nu_\beta(1 \mid s) = e^{4\beta\cdot(s-2)} \Big/ \left(1 + e^{4\beta\cdot(s-2)}\right),$$
$$\nu_\beta(0 \mid s) = 1 \Big/ \left(1 + e^{4\beta\cdot(s-2)}\right),$$

definierte Bernoulliverteilung auf {0,1} bezeichnet. Unter der bedingten Verteilung gegeben x^u sind die $x_{i,j}$, $(i,\ j) \in \Lambda^g$, also unabhängig mit Verteilung $\nu_\beta\left(\cdot \mid s_{i,j}\right)$, und analog sind unter der bedingten Verteilung gegeben x^g die $x_{i,j}$, $(i,\ j) \in \Lambda^u$, unabhängig mit Verteilung $\nu_\beta\left(\cdot \mid s_{i,j}\right)$. Beide bedingten Verteilungen können wir also leicht simulieren. Damit liegt folgender Algorithmus zur Simulation von μ_β nahe:

- ***Algorithmus (Gibbs-Sampler für das zweidimensionale Isingmodell)***

- Initialisierung:
  - Erzeuge $x_{i,j}^{(0)}$ $(1 \le i,j \le 2d)$ unabhängig, =0,1 jeweils mit W'keit 1/2.
  - Setze $x_{0,j}^{(0)} = x_{2d+1,j}^{(0)} = x_{i,0}^{(0)} = x_{i,2d+1}^{(0)} = 1$ $(0 \le i,j \le 2d+1)$.
- For n:=1,2,... do
  - x:=$x^{(n-1)}$
  - Update $x^g = \left(x_{i,j} :\ i,\ j \in \Lambda^g\right) \sim \mu_\beta(x^g \mid x^u)$
  - Update $x^u = \left(x_{i,j} :\ i,\ j \in \Lambda^u\right) \sim \mu_\beta(x^u \mid x^g)$
  - $x^{(n)}$:=x

Die Wahl der Anfangsbedingung $x^{(0)}$ ist dabei relativ willkürlich - man kann auch andere Anfangsbedingungen verwenden.

Theoretische Resultate

Die in dem Algorithmus simulierte Markovkette $X^{(n)}$ hat die Verteilung μ_β als Gleichgewicht. Aus dem Konvergenzsatz für Markovketten folgt, daß $X^{(n)}$ für große n näherungsweise gemäß μ_β verteilt ist - also ist $x^{(n)}$ eine approximative Stichprobe von μ_β.
Zudem folgt aus dem Ergodensatz (Gesetz der großen Zahlen) für Markovketten, daß für jede Funktion $f : S^+ \to \mathbb{R}$ und für jedes $b \in \mathbb{N}$ die empirischen Mittelwerte

$$\hat{\vartheta}_n := \frac{1}{n} \sum_{k=b}^{b+n-1} f\left(x^{(k)}\right)$$

für $n \to \infty$ mit Wahrscheinlichkeit 1 gegen den Erwartungswert

$$\vartheta \;:=\; \sum f(x)\,\mu_\beta(x)$$

von f unter μ_β konvergieren. Durch hinreichend große Wahl von b kann man die Konvergenz beschleunigen, s.u.

■ Aufgabenstellung

1. Zeigen Sie, daß die bedingte Verteilung

$$\mu_\beta(x^g \mid x^u) \;:=\; \frac{\mu_\beta(x^g,x^u)}{\sum_{x^g}\mu_\beta(x^g,x^u)}$$

durch (1) gegeben ist, und folgern Sie, daß μ_β ein Gleichgewicht für die durch obigen Algorithmus definierte Markovkette $X^{(n)}$ ist. *(Hinweis: Da auf beiden Seiten von Gleichung (1) Wahrscheinlichkeitsverteilungen stehen, genügt es zu zeigen, daß*

$$\mu_\beta(x^g \mid x^u) \;\propto\; \mu_\beta(x^g, x^u) \;\propto\; \prod_{(i,j)\in\Lambda^g} \exp\big(4\,\beta\cdot x_{i,j}\cdot(s_{i,j}-2)\big) \;\propto\; \prod_{(i,j)\in\Lambda^g} \nu_\beta\big(x_{i,j}\,\big|\,s_{i,j}\big)$$

gilt, wobei "$\propto$" bedeutet, daß beide Seiten als Funktionen in x^g proportional sind - sich also nur durch eine von x^u abhängende multiplikative Konstante unterscheiden).

2. Implementieren Sie den oben beschriebenen Gibbs-Sampler in *Mathematica*. Gehen Sie dazu folgendermaßen vor:

2.1. Als Vorübung: Erzeugen Sie ein Schachbrettmuster (d.h. die Konfiguration x mit $x^g \equiv 1$ und $x^u \equiv 0$), und erstellen Sie einen ArrayPlot.

2.2. Erstellen Sie *Mathematica*-Funktionen **updateg[x_]** und **updateu[x_]**, die die Werte $x_{i,j}$ für $(i,\,j)\in\Lambda^g$ bzw. $(i,\,j)\in\Lambda^u$ gemäß den jeweiligen bedingten Verteilungen updaten. Hierbei können β und d als Konstanten verwendet werden, denen vor Aufruf der Funktionen ein fester Wert zugewiesen wird. Wenden Sie jeweils einen Update auf das Schachbrettmuster an, erstellen Sie einen ArrayPlot der erhaltenen Konfigurationen für verschiedene Werte von β und d, und überprüfen Sie, ob Sie die erwarteten Ergebnisse erhalten.

2.3. Erzeugen Sie nun (z.B. mit **NestList[ ]**) eine Liste von ArrayPlots für die ersten 20 Schritte des Algorithmus mit $d = 20$ und β=0.2, 0.44 bzw. 0.6. Führen Sie anschließend 300 Schritte durch, und stellen Sie den Verlauf der Konfigurationen in diesen 300 Schritten mithilfe von **Manipulate[ ]** graphisch dar. Die Berechnung kann nun etwas Zeit in Anspruch nehmen - immerhin müssen 300 mal 1600 Pixel erneuert werden. Was beobachten Sie ? Für welchen Wert von β dauert es am längsten, bis sich ein Gleichgewicht einstellt ?

2.4. Experimentieren Sie auch mit anderen Randbedingungen, z.B.
$x_{0,j} = x_{i,0} = 0,\; x_{2\,d+1,j} = x_{i,2\,d+1} = 1$ für alle $i,\; j$.

3. Verwenden Sie den Gibbs-Sampler, um den Wert von $m_\beta(d)$ zu schätzen. Gehen Sie dabei wie folgt vor:

3.1. Schreiben Sie eine *Mathematica*-Funktion **MCMCSchaetzer[b_,nmax_]**, die die Schätzer

$$\hat{m}_\beta^{(n)} \;:=\; \frac{1}{n}\sum_{k=b}^{b+n-1} x_{d,d}^{(k)} \qquad (n = 1,\,2,\;\ldots,\,\text{nmax})$$

berechnet. Wählen Sie b für die obigen Werte von d und β jeweils so groß, daß Sie den Eindruck haben, daß sich nach b Schritten in guter Näherung ein Gleichgewicht eingestellt hat *("burn-in-time")*. Stellen Sie den Verlauf der Schätzer als Funktion von n graphisch dar, wobei Sie nmax möglichst groß wählen.

3.2. Erzeugen Sie eine Graphik, in der der Verlauf von $m_\beta(d)$ als Funktion von β approximiert wird. Schätzen Sie dazu $m_\beta(d)$ für beispielsweise 40 verschiedene Werte von β zwischen 0.2 und 0.6, und interpolieren Sie die erhaltenen Schätzwerte linear.