Algorithmische Mathematik II

Sommersemester 2012
Prof. Dr. Beuchler
Markus Burkow

universitätbonn

Übungsblatt 3. Abgabe am **Mittwoch vor der Vorlesung (bis 10:15 Uhr)**.

Aufgabe 1. (Glücksspiel)

Bei einem Glückspiel setzt man einen Betrag. Dann wird eine faire Münze geworfen ($p = 1/2$). Der eingesetzte Betrag verdoppelt sich, wenn „Zahl“ fällt, bei „Wappen“ ist er verloren. Ein Spieler hat 1 Euro und braucht 3 Euro zusätzlich, also insgesamt 4 Euro. Er setzt den Euro, verliert er ihn ist das Spiel zu Ende. Ansonsten hat er noch 2 Euro, von denen er wieder einen setzt, usw. Bei Gewinn stoppt das Spiel, ebenso bei Bankrott.

- Stellen Sie den zugehörigen Graphen und die Übergangsmatrix auf.
- Berechnen Sie die Übergangswahrscheinlichkeiten μ_i für $i = 1 \dots 4$
- Wie groß ist die Wahrscheinlichkeit, dass der Spieler nach 4 Schritten gewonnen hat.
- Nutzen Sie die Programmieraufgabe von Zettel 1 zur Berechnung von μ_{10}, μ_{25}, μ_{50} und μ_{100}.

(5 Punkte)

Aufgabe 2. (Chinesischer Restsatz)

Für beliebige Zahlen $a, b \in \mathbb{N}$ gibt es $x, y \in \mathbb{N}$ mit

$$ax - by = ggT(a, b)$$

(5 Punkte)

Aufgabe 3. (Irreduzible Matrizen)

Eine Matrix $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ heisst reduzibel, falls eine Permutationsmatrix P existiert mit

$$P^\top AP = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix}$$

mit quadratischen Matrizen $A_{1,1} \in \mathbb{R}^{p \times p}$ und $A_{2,2} \in \mathbb{R}^{(n-p)\times(n-p)}$. Die Matrix A heisst irreduzibel, falls sie nicht reduzibel ist. Unter dem Graphen einer Matrix versteht man den gerichteten Graphen $G = (V, E)$ mit $V = \{1, 2, \dots, n\}$ und $E = \{(i, j) \in V \times V : a_{i,j} \neq 0\}$.

Beweisen Sie:
Eine Matrix $A \in \mathbb{R}^{n \times n}$ ist genau dann irreduzibel, wenn der Graph von A zusammenhängend ist, d.h. $\forall \alpha, \beta \in V \quad \exists \alpha_1, \alpha_2, \dots, \alpha_K \in V$ mit $\alpha_1 = \alpha, \alpha_K = \beta$ und $(\alpha_j, \alpha_{j+1}) \in E, j = 1, \dots, K - 1$.

(5 Punkte)

Aufgabe 4. (Warteschlange)

Eine *Warteschlange* besteht aus einer Person, die gerade bedient wird, und den Personen, die auf die Bedienung warten. Wir betrachten eine Warteschlange zu jeder Sekunde und bezeichnen mit X_n die Anzahl der darin befindlichen Personen nach n Sekunden. Hierbei gilt:

- In der Warteschlange knnen sich zu jedem Zeitpunkt höchstens N Personen befinden (d.h., wenn eine Person eintrifft und sieht, dass bereits N Personen warten bzw. bedient werden, dann verlässt sie die Schlange sofort wieder, ohne sich anzustellen).

- Innerhalb einer Sekunde wird die gerade bediente Person mit Wahrscheinlichkeit r fertig und verlässt die Warteschlange. Gleichzeitig kommt mit Wahrscheinlichkeit p eine neue Person hinzu (alle diese Ereignisse treten unabhängig voneinander ein. Man beachte, dass es möglich ist, dass innerhalb einer Sekunde die bediente Person fertig wird und gleichzeitig eine neue Person kommt, so dass sich die Größe der Warteschlange nicht verändert.).

- Wenn die Warteschlange leer ist, dann sind natürlich nur Neuankömmlinge möglich, wobei wir davon ausgehen, dass kein Neuankömmling noch innerhalb dieser Sekunde bedient werden kann. Wenn die Warteschlange voll ist, also bereits N Personen warten, dann kann nur die gerade bediente Person die Warteschlange verlassen, aber kein Neuankömmling hinzukommen.(Insbesondere lassen wir nicht zu, dass die gerade bediente Person fertig wird, und gleichzeitig eine neue Person hinzukommt. Lediglich ein Neuankömmling pro Sekunde.)

Modellieren Sie diese Situation als Markovkette und stellen Sie die Übergangsmatrix auf. Betrachten Sie ferner die totale Bedienzeit S eines Kunden, also die (zufällige) Zeit, während der ein Kunde bedient wird, sowie die (zufällige) Zeit T zwischen dem Eintreffen zweier Kunden. Bestimmen Sie die Verteilungen von S und T sowie ihre Erwartungswerte.

Programmieraufgabe 1. Simulieren Sie die Markovkette aus 4 (Warteschlange) (mit N=6) mit Hilfe eines Computerprogramms für verschiedene Parameter p und r und verschiedene Anfangsverteilungen $\mu^{(0)}$. Nutzen Sie hierbei, dass bereits bekannt CSR-Format zur Speicherung der Matrix. Zeigen Sie dabei experimentell:

- Die Verteilung von X_n für n sehr groß, hängt (fast) nicht mehr von der Anfangsverteilung ab, d.h., die Konvergenz der Verteilung zu einer Grenzverteilung unabhängig von der Anfangsverteilung lässt sich empirisch beobachten.

- Die Grenzverteilung unterscheidet sich qualitativ je nachdem ob $p < r$, $r < p$ oder $p = r$.

Beachten Sie die folgenden *Hinweise*:

- Wählen Sie n genügend groß(z.B. $n = 1000$), sowie eine große Zahl M (z.B. $M = 1000$) und simulieren Sie M Ausgänge von X_n. Berechnen Sie anschließend die *empirische Verteilung* von X_n. Falls n und M genügend großsind, sollte diese empirische Verteilung, also die relativen Häufigkeiten der Zustände, die Grenzverteilung gut approximieren.

- Zur Simulation von X_n gehen Sie folgendermaßen vor: Simulieren Sie zuerst X_0 anhand der von Ihnen gewählten Anfangsverteilung $\mu^{(0)}$. Danach gehen Sie iterativ vor: gegeben ein simulierter Wert für X_k, simulieren Sie zwei Bernoulli-verteilte Zufallsvariablen mit Parametern p und r, (d.h., simulieren Sie, ob jemand die Warteschlange verlässt oder neu hinzukommt), und bestimmen Sie X_{k+1} entsprechend.

- Die Simulation einer Bernoulli-verteilten Zufallsvariable mit Parameter p kann man beispielsweise so durchführen: simulieren Sie mit einem geeigneten Zufallszahlengenerator eine (Pseudo-)Zufallszahl u nach einer Gleichverteilung auf dem Intervall $[0, 1]$. Falls $u \leq p$, dann setzen Sie die Bernoulli-Zufallszahl auf 1, andernfalls auf 0.(Wie kann man dieses Verfahren verallgemeinern, um Zufallszahlen nach der Anfangsverteilung $\mu^{(0)}$ zu simulieren?)

- Wählen Sie zunächst $p = 0.2$ und $r = 0.7$, dann $p = 0.8$ und $r = 0.3$ und zuletzt $p = r = 0.4$. Simulieren Sie nun in jedem Fall die Grenzverteilung für die folgenden Anfangsverteilungen:

$$\mu^{(0)} = [1\ 0\ 0\ 0\ 0\ 0\ 0],$$
$$\mu^{(0)} = [0\ 0\ 0\ 0\ 0\ 0\ 1],$$
$$\mu^{(0)} = [1/7\ 1/7\ 1/7\ 1/7\ 1/7\ 1/7\ 1/7].$$

  Vergleichen Sie die Ergebnisse je nach der Wahl der Parameter sowie nach der Wahl der Anfangsverteilungen. (Geben Sie Ihre Resultate und die Beobachtungen ab.)

Die Abgabe der Programmieraufgaben erfolgt in den CIP-Pools vom 18.04. bis 24.04.2012. Die Listen für die Anmeldung zu den Abgabe-Terminen hängt in der Woche vom 10.04. bis 17.04.2012 aus.