technische universität dortmund

Fakultät für Informatik
Lehrstuhl 4

Prof. Dr. Peter Buchholz, Clara Scherbaum M.Sc.

Sommersemester 2019

Modellgestützte Analyse und Optimierung
Übungsblatt 6

Ausgabe: 10.05.2019, **Abgabe:** 20.05.2019

Aufgabe 6.1: Modellierung von Eingabedaten (8 Punkte)

Folgende Zwischenankunftszeiten von Bauteilen an einer Maschine wurden gemessen:

10.478	7.268	2.775	0.381	0.979	1.226	1.392	4.706
0.003	5.004	2.835	2.441	1.838	10.780	0.010	4.732
3.042	8.817	0.670	1.693	11.576	2.469	9.905	2.265
1.184	0.093	5.187	1.828	7.140	0.136	1.408	6.349
3.717	7.170	0.609	3.084	0.122	3.154	3.574	0.420

Sie finden diese Daten zusätzlich in einer CSV-Datei auf der Übungs-Webseite.

1. Zeichnen Sie ein Histogramm für diese Stichprobe. Unterteilen Sie dazu das Intervall $[0, 12]$ in 6 Teilbereiche.
2. Die Form des Histogramms lässt vermuten, dass die beobachteten Zwischenankunftszeiten gut durch eine Exponentialverteilung beschrieben werden können. Bestimmen Sie den Parameter λ der Exponentialverteilung, so dass diese bestmöglich an die Stichprobe angepasst wird. Nutzen Sie dazu eine Methode zur Parameterschätzung aus der Vorlesung. Runden Sie Ihr Ergebnis auf vier Nachkommastellen.
3. Bestimmen Sie zum Test der Anpassungsgüte die beiden Abstandsmaße D und D'.
4. Zeichnen Sie den Q-Q-Plot und den P-P-Plot.
5. Führen Sie den Chi-Quadrat Test wie im Skript beschrieben durch. Unterteilen Sie dazu die angepasste theoretische Verteilungsfunktion in $k = 4$ Intervalle, so dass Werte aus den Intervallen mit gleicher Wahrscheinlichkeit $p_i = 1/k = 0,25, i = 1, ..., k$, angenommen werden. Berechnen Sie die Teststatistik und vergleichen Sie das Ergebnis mit den kritischen Werten der χ^2-Verteilung (siehe Tabelle 1). Wie müssen die Freiheitsgrade gewählt werden? Akzeptieren Sie die Exponentialverteilung als gute Approximation der Stichprobe oder lehnen Sie sie ab?
6. Führen Sie den Kolmogorov-Smirnov-Test wie in der Vorlesung beschrieben durch. Berechnen Sie zuerst die Teststatistik D_n. Vergleichen Sie dann die angepasste Teststatistik (siehe Tabelle 2) mit den kritischen Werten aus Tabelle 2. Akzeptieren Sie die Exponentialverteilung als gute Approximation der Stichprobe oder lehnen Sie sie ab?

ν	$\chi^2_{0.005}$	$\chi^2_{0.01}$	$\chi^2_{0.025}$	$\chi^2_{0.05}$	$\chi^2_{0.1}$
1	7.88	6.63	5.02	3.84	2.71
2	10.60	9.21	7.38	5.99	4.61
3	12.84	11.34	9.35	7.81	6.25
4	14.96	13.28	11.14	9.49	7.78

Tabelle 1: Kritische Werte χ^2_α der χ^2-Verteilung mit ν Freiheitsgraden

ν	$1-\alpha$				
	0.850	0.900	0.950	0.975	0.990
1	1.138	1.224	1.358	1.480	1.628
2	0.926	0.990	1.094	1.190	1.308

Tabelle 2: Kritische Werte für angepasste Kolmogorov-Smirnov Teststatistiken

Fall	**Angepasste Teststatistik**
Alle Verteilungsparameter bekannt	$\left(\sqrt{n} + 0.12 + \frac{0.11}{\sqrt{n}}\right) \cdot D_n$
Exponentialverteilung (λ aus Stichprobe geschätzt)	$\left(D_n - \frac{0.2}{n}\right) \cdot \left(\sqrt{n} + 0.26 + \frac{0.5}{\sqrt{n}}\right)$

Aufgabe 6.2: Agentenbasierte Modellierung mit AnyLogic - Conway's Game of Life (4 Punkte)

Realisieren Sie Conway's Game of Life (http://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens) als einfache agentenbasierte Simulation mit AnyLogic. Das quadratische Spielfeld ist dabei in Zellen aufgeteilt, die entweder lebendig oder tot sind. In jedem Schritt ändern die Zellen ihren Zustand anhand der folgenden Regeln:

- Eine tote Zelle mit genau drei lebendigen Nachbarn wird im nächsten Schritt lebendig.
- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben.
- Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt am Leben.
- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben.

Hinweise zur Modellierung:

- Die Zellen sollen durch Agenten realisiert werden.
- Um eine Agentenpopulation zu erzeugen, wählen Sie die *Agent* Bibliothek und ziehen Sie ein *Agent* Objekt in das Modellierungsfenster. In dem nun erscheinenden Agenten-Wizard wählen Sie *Population of Agents*. Wählen Sie als Name für den Typ *Cell*. Die Population erhält automatisch den Namen *cells*. In den folgenden Dialogen verzichten wir zunächst auf eine Animation und geben auch keine Parameter an. Die Population soll aus 10000 Agenten bestehen. Als Parameter für die Umgebung wählen Sie *discrete*, *arranged* und *Moore*. Dies hat zur Folge das die Region in diskrete Zellen aufgeteilt wird und die Agenten in Reihen und Spalten angeordnet werden. Durch die Option *Moore* hat jede Zelle 8 Nachbarn.
- In dem Modellierungsfenster des Agenten *Cell* werden zwei Variablen benötigt: Die Variable *alive* ist vom Typ *boolean* und hat als Startwert *randomTrue(0.2)*. Zu Beginn ist eine Zelle also mit einer Wahrscheinlichkeit von 0.2 lebendig. Die zweite Variable *counter* ist vom Typ *int* und wird später genutzt, um die lebendigen Nachbarn zu zählen.
- Zeichnen Sie außerdem ein Rechteck in das Fenster. Setzen Sie die Position auf X=0, Y=0 und die Breit und Höhe jeweils auf 5. Tragen Sie als Füllfarbe den Ausdruck *alive?green:red* ein. Abhängig vom Wert der Variable *alive* ist das Rechteck entweder grün oder rot.
- Wählen Sie im Modellierungsfenster des Hauptmodells die *cells* Population. Mit Hilfe von *Show presentation* können Sie das gerade erstellte Rechteck hier anzeigen lassen.
- Da die Simulation schrittweise erfolgen soll, aktivieren Sie in den Eigenschaften des Main-Agenten die Option *Enable steps*.

- In den Eigenschaften der Agenten können nun für *On before step* und *On step* Aktionen durchgeführt werden. Diese Aktionen nutzen wir, um das Verhalten unmittelbar vor einem Schritt und zum Zeitpunkt eines Schritts festzulegen. Vor dem Schritt muss die Anzahl der lebendigen Nachbarn ermittelt werden. Zum Zeitpunkt des Schritts muss dann der Status der Zelle geändert werden. Zur Ermittlung der Nachbarn kann der folgende Code genutzt werden:

```
counter = 0;
for (Agent a:getNeighbors())
  if (((Cell)a).alive)
    counter++;
```

  Die Änderung des Status einer Zelle erfolgt mit folgender Abfrage:

```
if (alive && counter <2)
  alive = false;
else if (!alive && counter==3)
  alive = true;
else if (alive && counter > 3)
  alive = false;
```