2. Übungszettel zur LV Computational Health Informatics (WiSe 2020/21)

Ausgabe Mi, 4. November 10 Uhr

Abgabe der Löungen ist bis Di, 10. November 14 Uhr möglich
→ exercise-chi@chi.uni-hannover.de

Aufgabe 2.1: Approximation von Pi (8 Punkte)

Pi kann durch zufällig generierte Punkte (x- und y-Koordinate jeweils zwischen -1 und 1) angenähert werden, indem der Anteil der Punkte bestimmt wird, die innerhalb des Einheitskreises (Radius r = 1) liegen.

Grundlage dafür ist die Formel zur Berechnung des Flächeninhalts eines Kreises:

$$A_{Kreis} = \pi r^2$$

a. **(2 Punkte)** Implementieren Sie die Approximation von Pi in Python mittels NumPy. Generieren Sie dafür mindestens 10 Millionen zufällige Punkte mit `np.random.uniform()`. Geben Sie das Ergebnis der Annäherung auf der Kommandozeile aus.

b. **(4 Punkte)** Implementieren Sie die Visualisierung der generierten Punkte und des Einheitskreises mittels Matplotlib in einer Abbildung mit zwei Subplots. Orientieren Sie sich dabei an der Abbildung auf diesem Übungsblatt.

   - Stellen Sie im linken Subplot 2.500 der generierten Punkte dar.
   - Stellen Sie im rechten Subplot den Einheitskreis im Hintergrund dar und plotten darüber 2.500 der generierten Punkte. Die Punkte außerhalb des

Einheitskreises sollen blau, die Punkte innerhalb rot dargestellt werden.

- Beschriften Sie beide Subplots sowie die Achsen und achten Sie darauf, dass der Kreis nicht zu einer Ellipse verformt dargestellt wird.
- Fügen Sie die generierte Abbildung dem PDF Ihrer Abgabe bei.

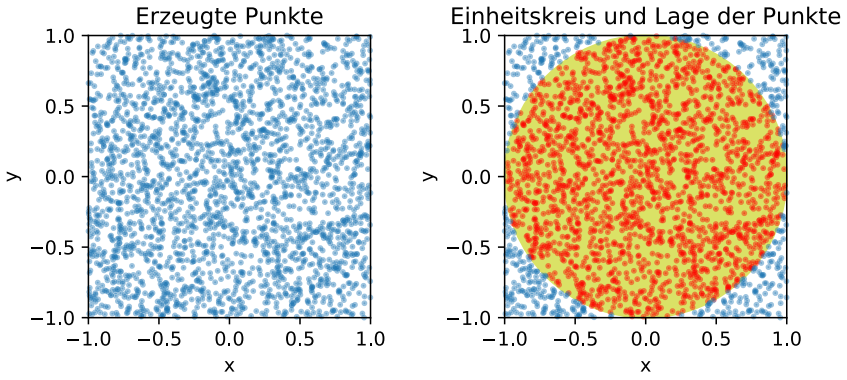

Figure 1. Beispielhafte Visualisierung der Pi-Approximation

NumPy stellt bereits Funktionen zur Distanzberechnung und Filtern eines Arrays basierend auf Bedingungen bereit. Matplotlib stellt Möglichkeiten zum Zeichnen von Kreisen und Punkten zur Verfügung.

Nutzen Sie die Dokumentation von Matplotlib und NumPy für weitere Informationen.

Reduzieren Sie die Anzahl der Punkte, falls der Speicher Ihres Rechners nicht ausreicht und deshalb Fehler auftreten.

c. **(2 Punkte)** Kann Pi mit dem eingesetzten Verfahren genau approximiert werden? Beschreiben Sie das Problem, welches theoretisch durch die Verwendung von `np.random.uniform()` entsteht.

- Ermitteln Sie in einem weiteren Python-Skript für 100 Millionen zufällige Zahlen zwischen -1 und 1 das arithmetische Mittel. Wiederholen Sie dies 10 Mal und geben Sie das arithmetische Mittel und die Standardabeichung der zuvor berechneten Mittelwerte aller Durchgänge aus. Geben Sie das Ergebnis

mit einem Vertrauensbereich von ±s an (vgl. letzte Vorlesung in Stud.IP).

- Was schließen Sie aus dem Ergebnis? Wie aussagekräftig ist dieses?

Verringern sie die Anzahl der Zufallszahlen, falls es zu Speicherproblemen kommt.

Aufgabe 2.2: Fehlerfortpflanzung (2 Punkte)

Berechnen Sie für folgende Funktionen f(x,y) die Standardabweichung s_f gemäß der Formel für die Fehlerfortpflanzung in der Vorlesung:

a. $f(x,y) = x \pm y$
b. $f(x,y) = x \cdot y$
c. $f(x,y) = x / y$
d. $f(x) = \ln(x)$