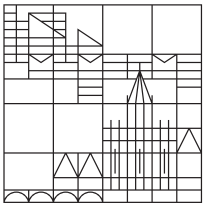
UNIVERSITÄT KONSTANZ
Fachbereich Physik (Theoretische Physik)
Dr. Stefan Gerlach
Raum P 817, Tel. (07531)88-3825
E-mail: stefan.gerlach@uni-konstanz.de

Universität Konstanz



Übungen zu Scientific Computing mit Python
Sommersemester 2024

Übungsblatt 5
Ausgabe 14.6., Übungen KW 25+26, Abgabe bis 1.7.

Wahrscheinlichkeitsverteilungen und Anpassung

1. **Aufgabe: Zentraler Grenzwertsatz und Verteilungsfunktionen**

   (a) Lege eine Feld mit $N = 1000$ **Zufallsvariablen** X_i an, die von der Funktion `random.randint(1, 10)` generiert werden. Visualisiere das Ergebnis in einem **Histogram**. Welche (diskrete) Wahrscheinlichkeitsverteilung erhält man?

   (b) Betrachte mit $Y_k = \frac{1}{M}\sum_{i=1}^{M} X_i$ die **Summe von gleichverteilten Zufallszahlen**. Definiere dazu ein weiteres Feld der Länge N bestehend aus der Summe Y_k von jeweils $M = 100$ neuen Zufallsvariablen und stelle das Feld in einem weiteren Histogram dar.

   (c) Plotte die Histogramme aus (a) und (b) nebeneinander mit plot.subplots und beschrifte die Achsen.

   (d) Definiere eine Funktion die dir eine **Normalverteilung** mit Mittelwert μ und Standardabweichung σ berechnet. Plotte die Funktion für $\{\mu, \sigma\} = \{0, 1\}$ über einen x-Achsenabschnitt von -2 bis 2.

   (e) Extrahiere den **Mittelwert** und die **Standardabweichung** des Histograms aus (b) und nutze diese in der Verteilungsfunktion aus (d) mit einem passenden x-Achsenabschnitt.

   (f) Plotte das Ergebnis aus (e) zusammen mit dem Histogram aus (b) und vergleiche beide.

   (g) Überlege dir, wie du aus der Dichtefunktion aus b) eine sinnvolle **Wahrscheinlichkeitsdichtefunktion** erhältst, sodass die Normalverteilung aus (e) dein Histogram aus (b) annähert.
   Tipp: eine Wahrscheinlichkeitsdichtefunktion ist normiert.

2. **Aufgabe: Poissonverteilung**

   Die Poissonverteilung ist eine Näherung für physikalische Vorgänge, bei denen seltene Ereignisse in sehr vielen Versuchen beobachtet werden. Wir wollen uns hier ein Beispiel anschauen.

(DNA) Punktmutationen, die beim Duplizieren der DNA entstehen, werden durch ausgeklügelte Korrekturmechanismen so gut korrigiert, dass pro Generation nur eine Punktmutation in 50 Millionen Basenpaaren entsteht. Die Zahl der daraus resultierenden Punktmutationen ist Poisson-verteilt.

Das Chromosom Nummer 10 des Menschen hat 135 Millionen Basenpaare. Was sind die Wahrscheinlichkeiten dafür, dass ein Chromosom 10 des Kindes keine, eine, zwei Punktmutationen gegenüber dem elterlichen Chromosom 10 bekommt? Stelle die Ergebnisse graphisch dar.

3. **Aufgabe: Maxwell-Boltzmann-Verteilung**

Die Geschwindigkeitsverteilung von Atomen der Masse m eines idealen Gases der Temperatur T ist gegeben durch

$$f(v) = \left(\frac{m}{2\pi kT}\right)^{3/2} 4\pi v^2 \mathrm{e}^{-mv^2/kT}.$$

Wir wollen die Verteilung am Beispiel von Helium-Atomen ($m = 6{,}65 \cdot 10^{-27}$ kg) bei der Temperatur 300 K untersuchen. Die Boltzmann-Konstante ist $k = 1{,}38 \cdot 10^{-23}$ J/K.

(a) Zeichne die Verteilung und finde dessen Maximum $v_{\max}$.

(b) Berechne den Mittelwert $\langle v \rangle$ und das mittlere Geschwindigkeitsquadrat $v_{\mathrm{RMS}} = \sqrt{\langle v^2 \rangle}$ durch Numerische Integration (z.B. scipy.integrate.quad()).

(c) Zeichne $\langle v \rangle$, v_{RMS} und die Standardabweichung $\sigma = \sqrt{\langle v^2 \rangle - \langle v \rangle^2}$ in die Verteilung ein.

(d) Bestimme den Median v_{m} der Verteilung, definiert durch $\int_0^{v_{\mathrm{m}}} f(v)\,\mathrm{d}v = 0{,}5$. Wo liegt er im Vergleich zu $v_{\max}$ und $\langle v \rangle$?

4. **Aufgabe: (Nicht)lineare Anpassung**

Betrachte die drei Datenpunkte (-1,0), (0,2), (1,1).

(a) Schreibe ein Python-Programm, dass eine **Lineare Regression** des Datensatzes durchführt und plotte das Ergebnis.

(b) Finde mithilfe der **Polynomregression** das **Interpolierende Polynom** (2. Ordnung) und plotte es.

(c) [Optional] Verwende die **Polynomapproximation** mit den Legendre-Polynomen um das interpolierende Polynom linear zu nähern. Vergleiche das Ergebnis mit dem aus a).

(d) Mit Hilfe von Gnuplot, LabPlot, Origin, usw. lassen sich Daten in einer GUI „fitten“. Finde anhand eines Programmes heraus, wie das geht und teste es mit einer linearen Modellfunktion. Vergleiche das Ergebnis mit dem aus a).

(e) Verwende Python (`scipy.optimize.curve_fit`) um die Daten mit einer linearen Funktion zu fitten. Welche Fitparameter liefert Python? Vergleiche das Ergebnis mit dem aus a).