

Praktikum 3: Schätztheorie

Theoretische Aufgaben

Aufgabe 1. Seien die Zufallsvariablen X_1 , X_2 und X_3 identisch verteilt mit Erwartungswert μ . Welche der drei Zufallsvariablen

- $T_1 := \frac{1}{3}(X_1 + X_2 + X_3)$,
- $T_2 := \frac{1}{4}(2X_1 + 2X_2)$ und
- $T_3 := \frac{1}{3}(X_1 + 2X_3)$

sind erwartungstreu?

Aufgabe 2. Berechnen Sie den Maximum-Likelihood-Schätzer der Normalverteilung $N(0, \sigma^2)$ für σ .

Aufgabe 3. Berechnen Sie den Maximum-Likelihood-Schätzer der Binomialverteilung mit Parametern $n \in \mathbb{N}$ und $p \in [0, 1]$; dabei soll p geschätzt werden und n als bekannt vorausgesetzt werden. Die Dichte ist gegeben durch

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

Aufgabe 4. Berechnen Sie den Maximum-Likelihood-Schätzer für die Parameter a und b der Gleichverteilung auf dem Intervall $[a, b]$; dabei sind a und b reelle Parameter mit $a < b$.

Aufgabe 5. Begründen Sie, dass eine Verteilung mit Dichte

$$f(x \mid \mu, \sigma^2) = \frac{1}{2} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} + \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right)$$

keinen Maximum-Likelihood-Schätzer hat. (Hinweis: Überlegen Sie, warum die Likelihood-Funktion unbeschränkt ist.)

Programmier- und Verständnis-Aufgaben

Laden Sie sich die Datei `P3vorgabe.zip` herunter und benutzen Sie das darin enthaltene Programm `main3.py`, als Grundlage für die folgenden Programmieraufgaben. Die Signaturen aller zu programmierenden Funktionen sind dort vorgegeben, und enthalten teilweise Hinweise zur Umsetzung. `main3.py` ist so lauffähig, aber liefert natürlich noch falsche Ergebnisse.

Die Programmieraufgaben gliedern sich in zwei Themen:

Maximum-Likelihood-Schätzung

- `L(theta, x, f)`
- `l(theta, x, f)`
- `MLE(x, f, theta0, method=None, log=False, return_status=False)`
- `define_test_data(rng)`

Für die ersten beiden Funktionen sollen die Likelihood- und die Log-Likelihood-Funktionen gemäß Vorlesung definiert werden; dabei ist `theta` der Parameter oder das Parameter-Tupel des statistischen Modells, `x` ein Stichproben-Vektor und `f` die Dichte der zugrundeliegenden Verteilung.

MLE soll für eine Stichprobe `x`, die gemäß Dichte `f` verteilt ist, einen Maximum-Likelihood-Schätzwert berechnen mithilfe von numerischen Optimierungsverfahren. Schauen Sie sich dazu die Dokumentation der Funktion `minimize` in der Bibliothek `scipy.optimize` an. Diese Funktion stellt verschiedene Algorithmen zur Minimierung bereit, deren Name mit `methode` übergeben wird (`None` steht für den Default); außerdem benötigt der Algorithmus einen Startwert `theta0`, welcher auch die Dimension des Parameterraums Θ definiert. Der Parameter `log` gibt an, ob die Likelihood-Funktion `L` (im Fall `log=False`) oder die Log-Likelihood-Funktion `l` (im Fall `log=True`) maximiert wird, und schließlich gibt der Parameter `return_status` an, ob das *Return-Objekt* von `minimize` oder der Schätzwert zurück gegeben wird.

Es soll nun die Güte der Optimierung untersucht werden, indem verschiedene Verteilungen und verschiedene Optimierungsalgorithmen sowie weitere Parameter variiert werden und in Abhängigkeit davon jeweils die Differenz zum analytisch bestimmten MLE gebildet wird.

Hierzu soll in der Funktion `define_model_test_data` ein Dictionary von statistischen Modellen, Hilfsfunktionen und Hilfsvariablen definiert werden. Diese Funktion ist bereits partiell in `main3.py` definiert, und es lässt sich an der Definition für

'normal_std1' die Bedeutung der Einträge ablesen. Für jedes Modell sind folgende 5 Einträge definiert: `theta` für das gewählte Modell zur Stichprobengenerierung, Zufalls-Generator gemäß des Modells, Dichtefunktion der Verteilung zum Parameter `theta` des Modells, exakte Formel für den MLE und schließlich der Startwert für den Optimierungsalgorithmus. Definieren Sie weitere Dictionary-Einträge für folgende Keys: 'normal' (allgemeine Normalverteilung mit Parameter `theta = (mu, sigma)`), 'exp' (Exponentialverteilung mit Parameter `lmbda`), 'uniform' (Gleichverteilung auf dem Intervall $[a, b]$ mit Parameter `theta = (a, b)`), 'binomial10' (Binomialverteilung mit Parameter `p` und festem Umfang $n = 10$).

In der Bibliothek `scipy.stats` finden Sie zu sehr vielen Verteilungen (hier sind `norm`, `expon`, `uniform`, `binom` und unten `t` relevant) zugehörige Dichtefunktionen (`pdf`), Verteilungsfunktionen (`cdf`), deren Umkehrfunktionen also Quantile (`ppf`), usw.

Sind alle Modelle definiert, können Sie `test_MLE_precision(np.random.default_rng())` ausführen. Lesen Sie den Code, um die Ausgaben zu verstehen. Sie sollten nun folgende Fragen beantworten können:

- Welche Optimierungsverfahren funktionieren gut? Welche nicht?
- Wie hängt dies von den Verteilungen ab?
- Wie hängt es von der gewählten Art der Likelihood-Funktion ab? Woher kommen die großen Unterschiede?
- Wie wirkt sich die Größe der Stichprobe aus?
- Warum ist die Gleichverteilung so problematisch?

Intervallschätzung

Im Folgenden ist `alpha` das Signifikanzniveau, und die folgenden beiden Typen von Intervallschätzern für normalverteilte Stichproben sollen gemäß Vorlesung implementiert werden. Rückgabewert ist jeweils ein Paar bestehend aus linker und rechter Intervallgrenze.

- `interval_normal_mu(alpha, sigma, n, mean)`
(Intervallschätzung von `mu` bei bekanntem `sigma`, Input Samplegröße `n` und Mittelwert `mean`)
- `interval_normal_mu_sigma(alpha, n, mean, std)`
(Intervallschätzung von `mu` und `sigma`, Input Samplegröße `n`, Mittelwert `mean` und Standardabweichung `std`)
- `interval_normal_mu_sample(alpha, sigma, x)`
(Intervallschätzung von `mu` bei bekanntem `sigma`, Input Sample `x`)
- `interval_normal_mu_sigma_sample(alpha, x)`
(Intervallschätzung von `mu` und `sigma`, Input Sample `x`)

Abgabe

- Laden Sie das Archiv `P3vorgabe.zip` von [moodle](#) herunter, entpacken Sie es, und testen Sie Ihre Programme, indem Sie `test3.py` im gleichen Verzeichnis mit `python` ausführen. Erhalten Sie `ERROR`, so entspricht Ihr Programm nicht der Spezifikation von oben. Erhalten Sie `FAIL`, so ist Ihr Programm zwar lauffähig, aber die berechneten Werte sind fehlerhaft.
- **Abgaben, bei denen der Test gar nicht durchläuft oder mit `ERROR`, werden nicht akzeptiert**, `FAIL` führt nur zu Punktabzug.
- Sie finden in obigem Archiv auch die Datei `info3.md` mit anzugebenden Informationen zu Ihrem Team und Ihrer Abgabe, **bitte füllen Sie diese nach dortiger Anleitung aus, und vergessen Sie nicht die Quellenangabe**. Abgaben mit unvollständiger Datei `info3.md` können nicht gewertet werden. **Belassen Sie diese Datei in der UTF-8-Kodierung**.
- Komprimieren und bündeln Sie alle oben erzeugten oder geänderten Dateien, indem Sie ein ZIP-Archiv erstellen. Sollten Sie nicht wissen, wie das geht, konsultieren Sie dazu die Dokumentation Ihres Betriebssystems.
- Nennen Sie Ihr ZIP-Archiv `P3.zip`.
- Schreiben Sie eine Email an rosehr@hm.edu mit Betreff **Abgabe Datenanalyse**, Dateianhang **P3.zip** und irgendwelchem sonstigen Inhalt. **Der Automat akzeptiert die Abgabe nur, wenn diese Angaben (Betreff, Dateiname), sowie Ihre persönlichen Angaben (Nach-, Vor- und Teamname) korrekt sind**. Sie erhalten innerhalb der nächsten 24 Stunden (meistens aber deutlich schneller) eine Bestätigungsemail, wenn alles korrekt war.