Übungen zur Vorlesung

Praktische Optimierung, SoSe 2024

Prof. Dr. Günter Rudolph, Dr. Marco Pleines

`http://ls11-www.cs.tu-dortmund.de/people/rudolph/teaching/lectures/POKS/SS2024/lecture.jsp`

Blatt 3, Block A

29.04.2024

Abgabe: 09.05.2024

Aufgabe 3.1: Welch-Test und Wilcoxon-Test (6 Punkte)

In dieser Aufgabe wollen wir uns mit dem Welch-Test und dem Wilcoxon-Test beschäftigen. Dazu betrachten wir erneut die Kompasssuche.

Auf der Moodle-Kursseite wird eine neue Variante der Kompasssuche zur Verfügung gestellt. Kopieren Sie die Datei `kompasssuche_test.py` in Ihr Projektverzeichnis und importieren Sie die Funktion mit:

```
from kompasssuche_test import kompasssuche_test
```

Die Funktion nutzt als Abbruchkriterium nicht mehr die Anzahl an Iterationen sondern eine Abweichung vom wahren Optimum, das fest bei 0.0 erwartet wird. Solange die absolute Abweichung von diesem Optimum größer als die vorgegebene Toleranz 0.001 ist, wird die Kompasssuche fortgesetzt. Nach spätestens 1000 Iterationen wird die Suche abgebrochen, um eine Endlosschleife zu vermeiden. Die importierte Funktion `kompasssuche_test(f, x0, s0, theta)` gibt die Anzahl an Iterationen zurück, die benötigt wird, um nah genug an das Optimum zu kommen. Information darüber, welche Argument die Funktion erwartet, erhalten Sie nach dem Import durch `help(kompasssuche_test)`.

Erzeugen Sie 500 gleichverteilte (Pseudo-)Zufallszahlen aus dem Intervall $[-10, 10] \times [-10, 10]$. Verwenden Sie dazu `seed(1)`. Nutzen Sie diese 500 Zahlen jeweils als Startwert $x^{(0)}$ für die Kompasssuche.

Sie sollen erneut die Funktion $f(x, y) = x^2 + y^2,\ x, y \in [-10, 10]$ minimieren. Verwenden Sie dabei die folgenden Parametereinstellungen für die Kompasssuche:

(1) $s^{(0)} = 1$ und $\theta = 0.5$,

(2) $s^{(0)} = 2$ und $\theta = 0.5$,

(3) $s^{(0)} = 1.5$ und $\theta = 0.8$,

(4) $s^{(0)} = 0.5$ und $\theta = 0.2$.

Wir wollen die tatsächlich benötigte Zeit analysieren, die für die Kompasssuche mit verschiedenen Parametereinstellungen benötigt wird. Diese hängt von der Anzahl der benötigten Iterationen ab. Messen Sie die Laufzeiten für die Kompasssuche mit den vier unterschiedlichen Einstellungen, wobei Sie jede Messung 100-mal wiederholen und die mediane Laufzeit aus den 100 Wiederholungen verwenden sollen, um Messungenauigkeiten zu vermeiden. Sie sollen also $4 \cdot 500 \cdot 100 = 200\,000$ Messungen durchführen und daraus 2000 mediane Laufzeiten berechnen. Zum Messen der durchschnittlichen Laufzeit einer Funktion kann das Modul *timeit* empfohlen werden.

Erstellen Sie in einem ersten Auswertungsschritt Boxplots der gemessenen Laufzeiten für die vier verschiedenen Kombinationen von Parametern. Jeder Boxplot soll dabei aus 500 Werten erstellt werden, die zu je einer Parametereinstellung gehören. Plotten Sie alle vier Boxplots in eine Abbildung und beschriften Sie die Boxplots sinnvoll. Sorgen Sie dafür, dass die y-Achse jeweils die gleichen Wertebereiche abdeckt. Interpretieren Sie die Boxplots im Hinblick auf mögliche Unterschiede bei der Optimierung mit den verschiedenen Parameterkombinationen.

Testen Sie nun die folgenden Hypothesen:

(i) Die Kompasssuche mit $s^{(0)} = 1$ und $\theta = 0.5$ benötigt im Mittel weniger Zeit als die Kompasssuche mit $s^{(0)} = 1.5$ und $\theta = 0.8$.

(ii) Die Kompasssuche mit $s^{(0)} = 1.5$ und $\theta = 0.8$ benötigt im Mittel weniger Zeit als die Kompasssuche mit $s^{(0)} = 0.5$ und $\theta = 0.2$.

(iii) Die Kompasssuche mit $s^{(0)} = 2$ und $\theta = 0.5$ benötigt im Mittel weniger Zeit als die Kompasssuche mit $s^{(0)} = 1.5$ und $\theta = 0.8$.

(iv) Die Kompasssuche mit $s^{(0)} = 1$ und $\theta = 0.5$ benötigt im Mittel weniger Zeit als die Kompasssuche mit $s^{(0)} = 0.5$ und $\theta = 0.2$.

Geben Sie jeweils die Hypothesen H_0 und H_1 an und testen Sie die Hypothesen sowohl mit dem Welch-Test als auch mit dem Wilcoxon-Test:

Welch-Test (unequal variances t-test): Nutzen Sie die Implementierung `ttest_ind()` aus dem Paket *scipy* und setzen Sie den optionalen Parameter `equal_var=False`.

Wilcoxon-Test (Rangsummentest): Nutzen Sie die Implementierung `ranksums()` aus dem Paket *scipy*.

Benutzen Sie als Stichproben die zuvor von Ihnen gemessenen medianen Laufzeiten. Interpretieren Sie die Ergebnisse. Gibt es große Unterschiede bei den Ergebnissen der beiden Tests? Ergeben sich die gleichen Schlussfolgerungen wie bei der Betrachtung der Laufzeiten mithilfe von Boxplots?

Informieren Sie sich nun darüber, welche Annahmen die angewendeten Tests haben und überprüfen Sie die Annahmen für die Tests (es gibt mehrere Annahmen!). Verwenden Sie dabei für die Normalverteilungsannahme die Visualisierung durch Quantil-Quantil-Diagramme (kurz: *Q-Q-Plots*) mit eingezeichneter Referenzlinie (möglich mit der Funktion `qqplot()` dem Paket *statsmodels*) und für die Verschiebung der Verteilungsfunktionen *empirische Verteilungsfunktionen* (Funktion `ECDF()` aus selbigem Paket). Führen Sie die Tests auch dann durch, wenn Sie zu dem Ergebnis kommen, dass die Annahmen verletzt sind.

Aufgabe 3.2: Test auf Gleichheit (2 Punkte)

Sie haben in der Vorlesung gelernt, dass ein (sinnvoller) Test auf Gleichheit, also beispielsweise für das Hypothesenpaar

$$H_0 : \theta \neq \theta_0 \text{ vs. } H_1 : \theta = \theta_0$$

mit $\theta \in \Theta \subseteq \mathbb{R}$ nicht existiert. Begründen Sie, warum ein solcher Test keine höhere Güte unter der Alternativen (Wahrscheinlichkeit eine in Wahrheit nicht zutreffende Nullhypothese zu verwerfen) als das Signifikanzniveau α erreichen kann. Was bedeutet dies inhaltlich?

Hinweis: Es kann gezeigt werden (und darf hier ohne Beweis verwendet werden), dass die Gütefunktion stetig ist. Nutzen Sie dies bei Ihrer Argumentation. Beachten Sie außerdem, dass die Alternative in diesem Fall nur aus einem Punkt besteht.

Aufgabe 3.3: Fehler beim Testen (2 Punkte)

(a) Eine Person formuliert die Hypothese, die sie nachweisen möchte, als H_0. Ist dies ein sinnvolles Vorgehen? Begründen Sie Ihre Antwort.

(b) Es wird die Hypothese

$$H_0\colon \text{ Stichprobe ist normalverteilt vs. } H_1\colon \text{ Stichprobe ist nicht normalverteilt}$$

zum Niveau $\alpha = 0.05$ getestet. Dabei ergibt sich ein p-Wert von 0.08. Die Nullhypothese kann also nicht abgelehnt werden. Die Person, die den Test durchgeführt hat, ist sich nun sicher, dass ihre Stichprobe normalverteilt ist. Ist dies sinnvoll? Wie fällt Ihre Antwort bei einem p-Wert von 0.75 aus? Begründen Sie Ihre Antworten.

(c) Eine Person erhält ein signifikantes Testergebnis. Kann sie sich sicher sein, dass sie damit etwas wissenschaftlich relevantes herausgefunden hat? Begründen Sie Ihre Antwort.