Technische Universität Dortmund
Fakultät Statistik
Lehrstuhl Computergestützte Statistik

Dr. Uwe Ligges
M.Sc. Daniel Horn
M.Sc. Hendrik van der Wurp
Steffen Maletz

Übung zur Vorlesung

# Computergestützte Statistik

Wintersemester 2018/2019

## Übungsblatt Nr. 13

Abgabe ist Montag der 21.01.2019 an CS-abgabe@statistik.tu-dortmund.de oder Briefkasten 138

**Aufgabe 1** **(4 Punkte)**

Implementieren Sie eine Kreuzvalidierung, um die Güte eines linearen Modells zu schätzen. Verwenden Sie dieses Gütemaß zur Variablenselektion. Betrachten Sie dazu erneut den Datensatz `bic_data.RData`.

a) Implementieren Sie eine Funktion, die eine $k$-fache Kreuzvalidierung ausführt und den mittleren quadratischen Fehler (Mean Squared Error, MSE) eines linearen Modells schätzt. In jeder der $k$ Iterationen soll dabei ein lineares Modell auf den Daten `data` mit der Zielvariable `target` geschätzt werden, `data` und `target` sollen dabei Eingabeparameter Ihrer Funktion sein.

b) Betrachten Sie erneut Aufgabe 1 von Blatt 11: Variablenselektion mit einem Evolutionären Algorithmus auf `bic_data.RData`. Führen Sie diese Variablenselektion erneut durch, verwenden Sie nur diesmal den via a) berechneten MSE als Gütemaß.

**Aufgabe 2** **(4 Punkte)**

Wie Ihnen allen vielleicht noch aus der Werbung bekannt ist, gibt es zwei einander gegenüberliegende Fabriken, die linke bzw. rechte Twix-Schokoriegel herstellen. Nach langen Streits, welche der beiden Fabriken denn nun den besseren Riegel herstellt, wurden Sie als Statistiker zur Klärung angestellt. Tagesaktuell werden Sie aus beiden Fabriken Daten zu der sogenannten *Crunchigkeit* der Schokoriegel erhalten, und jeden Tag müssen Sie an Hand eines Tests entscheiden: Sind die Riegel aus den beiden Fabriken gleich crunchig, oder gibt es Unterschiede?

Bevor Sie Ihr Tagesgeschäft beginnen können, wollen Sie zunächst den optimalen Test finden, um die Verteilungen der Crunchigkeiten zu vergleichen. Dazu hat Twix Ihnen die Daten dreier Tage zur Verfügung gestellt (siehe Material chrunchigkeit.RData). Renommierte Testesser haben dabei in aufwendigen Selbstversuchen festgestellt, dass die Crunchigkeit der beiden Riegel sich an diesen Tagen jeweils nicht unterscheidet. Sie müssen diese Daten nun bestmöglich nutzen und denjenigen Test finden, der die beste Power besitzt. Dabei erinnern Sie sich an den lange zurückliegenden Besuch der Computergestützten Statistik und entscheiden sich einen Bootstrap zu verwenden, um so viel wie möglich aus der einen Stichprobe herauszuholen.

a) (1 Punkt) Implementieren Sie eine Funktion bootstrapTest, die einen zweiseitigen Test zum Mittelwertvergleich mittels Bootstrapping implementiert. Rückgabe Ihrer Funktion soll der p-Wert des Tests sein.

b) (2 Punkte) Schreiben Sie eine Funktion bootstrapPower. Diese erhält als Eingabe 2 numerische Vektoren $x$ und $y$ — die Crunchigkeiten der beiden Riegel — sowie eine Funktion, die einen statistischen Test ausführt. Diese Funktion soll 2 numerische Vektoren als Eingabe bekommen und den zugehörigen p-Wert zurückgeben. Erzeugen Sie sich mittel Bootstrapping je 200 Stichproben aus $x$ und $y$, verschieben Sie die Stichprobe $y$ um $\delta$ und zählen Sie, wie oft der Test die Nullhypothese ablehnt. Führen Sie dies für mehrere sinnvoll gewählt Werte für $\delta$ durch.

c) (1 Punkt) Bestimmen Sie mit Ihrer Funktion aus b) jeweils die Power des t-Tests, des Wilcoxon-Tests und des Bootstrap-Tests für die 3 Tagesproduktionen. Welchen Test empfehlen Sie?

**Aufgabe 3** **(4 Punkte)**

In der Datei `langsamerCode.R` finden Sie langsame Implementierung für die vier folgenden Aufgaben:

- Erzeugung einer $1000 \times 1000$ Matrix mit standardnormalverteilten Einträgen
- Bestimmung der Summe jeder Zeile dieser Matrix
- Trunkierung dieser Ergebnis auf den Bereich $[-100, 100]$, d.h. jedes Element kleiner als -100 soll durch -100 und jedes Element größer als 100 durch 100 ersetzt werden
- Bestimmung der 2-Norm des letzten Ergebnisses

Gestalten Sie diese Implementierungen effizienter. Punkte gibt es je nach Effizienz Ihrer Abgabe, bis zu einem Punkt pro Aufgabenteil.