Albert-Ludwigs-Universität Freiburg Wintersemester 2009

Statistische Methoden der Datenanalyse

Markus Schumacher

Übung VII

Matthew Beckingham und Henrik Nilsen

08.12.2009

Aufgabe 29 *Maximum-Likelihood-Anpassung an exponentiell verteilte Daten*

In Aufgabe 18 wurde eine ROOT-Datei mit simulierten Zerfallszeiten erzeugt.

In dieser Übung sollen Sie versuchen herauszufinden, welchen Wert der Zerfallskonstanten τ ein anderer Übungsteilnehmer benutzt hat, um diese Datei zu erzeugen. Eine Vorlage befindet sich in der Datei `~henrik/public_html/ExpLike_i.C`.

Gehen Sie dazu wie folgt vor:

(i) Lassen Sie sich von einem anderen Übungsteilnehmer die erzeugte Datei geben und öffnen Sie diese in ROOT. Ein Beispiel dafür findet sich im ROOT-Makro `ExpLike_i.C`.

(ii) Berechnen Sie für einen sinnvollen Wertebereich der Zerfallskonstanten τ den Logarithmus der Likelihood für die Exponentialverteilung und die in Aufgabe 18 erzerzeugte Messreihe:

$$\ln \mathcal{L}(\tau_j) = \sum_{i=1}^{N} \left[\ln \left(\frac{1}{\tau_j} \right) - \frac{x_i}{\tau_j} \right] \tag{1}$$

und stellen Sie sie mittels eines `TGraph` der Wert von $\ln \mathcal{L}(\tau_j)$ graphisch dar als Funktion von τ_j. Das TGraph Objekt kan man zeichnen lassen mittels

```
nameVonGraph->Draw("AP");
```

(iii) Ermitteln Sie aus der erzeugten Abbildung eine Abschätzung für die Zerfallskonstante $\hat{\tau}$.

(iv) Erzeugen Sie als nächstes eine eindimensionale Exponentialverteilung (`TF1` Objekt) und benutzen Sie die in ROOT integrierte Anpassungsroutine, um eine Maximum-Likelihood-Anpassung an die Daten mittels

```
TF1* expFunk = new TF1(‘‘expFunk’’,‘‘1/[0]*TMath::Exp(-x/[0])’’, 0.0, 10.0);
expFunk->SetParameter(0, 2.0);
tree->UnbinnedFit("expFunk", "exp", "", "V");
```

Die Option `"exp"` der Funktion "UnbinnedFit" steht für den Namen der Zufallsvariablen im `TTree`. Die Option `"V"` steht für detaillierte Ausgabe. Parameter Nummer 0 wird gleich 2.0 gesetzt vor der Anpassung – bestätigen Sie, dass der gefundene Wert für τ nicht von den ursprunglichen Wert von τ abhängt, indem Sie die "2.0" mit anderen Werten ersetzen.

Aufgabe 30 *Varianz von Maximum-Likelihood Schätzern*

In Aufgabe 29 haben wir den ML-Schätzwert für τ berechnet ($\hat{\tau}$). Als nächstes werden wir die Varianz des ML-Schätzwerts mittels dreier verschiedener Methoden auswerten.

(i) **Analytisch:** In der Vorlesung wurde gezeigt, dass die Varianz des ML-Schätzers $V_{\hat{\tau}} = \frac{\tau^2}{n}$ ist, und dass der ML-Schätzer der Varianz gegeben ist durch $\hat{V}_{\hat{\tau}} = \frac{\hat{\tau}^2}{n}$. Berechnen Sie den ML-Schätzwert der Standardabweichung von $\hat{\tau}$ für die Stichprobe in Aufgabe 29 ($\hat{\sigma}_{\hat{\tau}} = \sqrt{\hat{V}_{\hat{\tau}}}$).

(ii) **Graphisch:** Ermitteln Sie aus dem TGraph von $\ln \mathcal{L}(\tau)$ aus Aufgabe 29 eine Abschätzung für $\sigma_{\hat{\tau}}$. Aus der Vorlesung: $\ln L(\hat{\tau} \pm \hat{\sigma}_{\hat{\tau}}) = \ln L_{\max} - \frac{1}{2}$.

(iii) **MC-Methode:** (Vorlage befindet sich in der Datei `~henrik/public_html/ExpMC_i.C`)

- Nehmen Sie den Schätzwert $\hat{\tau}$ aus Aufgabe 29.
- Generieren Sie eine weitere Stichprobe von 100 exponentiell verteilten Zufallszahlen mit $\tau' = \hat{\tau}$.

- Ermitteln Sie unter Verwendung der ROOT-Anpassungsroutine den Maximum-Likelihood-Schätzwert $\hat{\tau}'$ für diese neue Stichprobe (siehe Anleitung oben). Nachdem die Anpassung durchgeführt ist, steht das Ergebnis zur Verfügung mittels
  ```
  Float_t fittedTauValue = expFunk->GetParameter(0);
  ```
- Wiederholen Sie diese Schritte 1000 mal und füllen Sie die Schätzwerte der Zerfallskonstanten aus jedem Schritt in ein Histogramm.
- Lesen Sie aus diesem Histogramm die Werte für $E[\hat{\tau}]$ und $\hat{\sigma}_{\hat{\tau}}$ ab.

Aufgabe 31 *Schätzer und WDFs: Erwartungstreue, Effizienz und Verzerrung*

In der Datei `~henrik/public_html/Robustheit_i.x` befindet sich ein fast fertiges Root-Makro um Erwartungstreue, Effizienz und Verzerrung von unterschiedlichen Schätzern und WDFs zu studieren. Um das Makro auszuführen, geben Sie folgende Kommandos in Root ein:

```
.L Robustheit_i.x
Robustheit( WDF-Name, # Stichproben, # Werte per Stichprobe)
```

Die möglichen WDF-Namen sind “Gauss” ($\mu = 1$, $\sigma = 1$), “Uniform” ([−1,1]), “Exp” ($\tau = 1$, i.e. e^{-x}) und “BreitWigner” ($\Gamma = 1$, $x_0 = 1$). Um z.B. für eine gaussische WDF und eine einzige Stichprobe von Umfang 10 auszuführen, geben Sie

```
Robustheit( ‘‘Gauss’’, 1, 10)
```

in Root ein.

Aufgaben:

(i) Das Makro enthält halbfertige Funktionen, um den aritmetischen Mittelwert, den Median, (grösstes-kleinstes)/2 und den multiplikativen Mittelwert ($[\prod_{i=1}^{n} x_i]^{1/n}$) zu berechnen. Die Funktionen müssen fertiggeschrieben werden. Wenn man

```
Robustheit( ‘‘Gauss’’, 1, 10)
```

ausführt, sieht man die Werte der Stichprobe und den berechneten Wert, der zu 0 oder 1 initialisiert wird.

(ii) Lassen Sie für jede der vier WDFs das Makro für 1000 Stichproben vom umfang 100 laufen. Versuchen Sie die Unterschiede zwischen den WDFs der 4 Schätzer des Erwartungswertes der WDF zu verstehen (für Breit-Wigner: die geschätzte Grösse ist der Symmetriepunkt – der Erwartungswert ist nicht endlich). Welche Schätzer sind: erwartungstreu? Am effizientesten? Ohne Verzerrung?

(iii) Gibt es einen Fall, in dem ein Schätzer mit Verzerrung zu bevorzugen ist gegenüber einem unverzerrten?

(iv) Falls Sie eine Stichprobe einer unbekannten WDF hätten, welcher Schätzer für den Erwartungswert der Grundgesamtheit wäre die beste Wahl gewesen?