Statistische Methoden der Datenanalyse

Markus Schumacher

Übung VIII

Matthew Beckingham und Markus Warsinsky

24.6.2009

Computerübung

Aufgabe 38 χ^2-*Anpassung an simulierte* Z^0-*Produktionsdaten*

In diesem Beispiel soll eine χ^2-Anpassung an einen simulierten Datensatz von Produktionswirkungsquerschnitten für Z^0-Bosonen in e^+e^--Kollisionen durchgeführt werden. Das ROOT-Makro `ZPeakGen.C` generiert Z^0-Boson-Produktionswirkungsquerschnitte als Funktion der Schwerpunktsenergie E_{cm} und verschmiert jeden Datenpunkt gemäß einer Gaussverteilung.

(i) Definieren Sie eine ROOT `TF1`-Funktion gemäß der Breit-Wigner-Verteilung

$$f(x) = A\frac{\frac{\Gamma}{2}}{(x-x_0)^2+\frac{\Gamma^2}{4}},$$

wobei Γ die Breite der Verteilung, x_0 die Position des Maximums (also die Masse des betreffenden Teilchens) und A eine Normierungskonstante ist.

(ii) Benutzen Sie diese Funktion für eine Anpassung an den `TGraphErrors`, der die simulierten Datenpunkte enthält, indem Sie den Befehl

```
graph->Fit("fitfunkname");
```

benutzen.

(iii) Benutzen Sie danach die Methoden von `TF1`

```
TF1::GetChisquare()
TF1::GetNDF()
TF1::GetProb(),
```

um Informationen über die durchgeführte Anpassung zu erhalten.

(iv) Benutzen Sie die Monte-Carlo-Methode, um den P-Wert zu berechnen, und mit demjenigen aus der ROOT-Anpassung zu vergleichen. Gehen Sie dazu wie folgt vor:

a) Lesen Sie den Wert y_1 des ersten Datenpunktes aus dem `TGraphErrors` ein. Die Syntax dafür kann aus dem ROOT-Makro entnommen werden.

b) Benutzen Sie den Wert von y_1 als Mittelwert einer Poissonverteilung, gemäß der ein neuer Datenpunkt y_1' ermittelt wird. Benutzen Sie dazu den ROOT-Zufallszahlengenerator `gRandom->Poisson(mean)`.

c) Wiederholen Sie dieses Verfahren für alle Datenpunkte, um einen neuen Datensatz y_i' zu erhalten.

d) Füllen Sie die y_i' in ein Array $\vec{y}'$ und benutzen Sie dieses, um einen neuen `TGraphErrors` mit den Monte-Carlo-Daten zu erstellen. Benutzen Sie als Fehler auf die Monte-Carlo-Werte die Poisson-Fehler.

e) Führen Sie eine χ^2-Anpassung an diese neuen Monte-Carlo-Daten durch.

f) Füllen Sie den χ^2-Wert nach der Anpassung in ein Histogramm.

g) Wiederholen Sie diese Schritte 1000mal.

(v) Bestimmem Sie aus dem Histogramm der χ^2-Werte den P-Wert dafür, ein gleich großes oder noch grösseres χ^2 wie in der ursprünglichen Anpassung zu erhalten. Benutzen Sie dazu die Methode von `TH1`

```
TH1::Integral(int binMin, int binMax),
```

mittels der man das Integral zwischen den Bins `binMin` und `binMax` erhält. Sie können die Methode von `TH1`

```
TH1::FindBin(float x)
```

benutzen, um die Binnummer für die Position `x` auf der x-Achse zu erhalten.

Aufgabe 39 *Anpassungsgüte bei einer Maximum Likelihood Anpassung*

In dieser Übung sollen Sie die Ergebnisse der 'Binned' Maximum Likelihood Anpassung an die $e^+e^- \to \mu^+\mu^-$-Ereignisse aus Übung VI benutzen. Betrachtet werden sollen die Verhältnisse der Likelihoodfunktionen

$$\lambda = \frac{\mathcal{L}(\vec{n}|\vec{\nu})}{\mathcal{L}(\vec{n}|\vec{n})}.$$

Damit soll gezeigt werden, dass

$$\chi_p^2 = -2\ln\lambda_p$$

gemäß einer χ^2-Verteilung mit $N-m$ Freiheitsgraden verteilt ist. Dabei ist N die Anzahl der Datenbins und m die Anzahl der angepassten Parameter.

Das Makro `MLChi2.C` gibt ein Beispiel, wie man die generierten $e^+e^- \to \mu^+\mu^-$-Ereignisse (Anzahl N_{tot}) einliest und ihre $\cos\theta$-Verteilung berechnet.

(i) Definieren Sie eine `TF1`-Funktion gemäß

$$f(x;\alpha,\beta) = A\frac{1+\alpha x+\beta x^2}{2+\frac{2\beta}{3}},$$

und führen Sie eine Anpassung dieser Funktion an das $\cos\theta$-Histogramm mit dem Befehl

```
hist->Fit("FunkName","IL");
```

durch. Die Option `"IL"` steht für eine Maximum Likelihood Anpassung unter Benutzung der Integrale über die Bins des Histogramms `hist`.

(ii) Definieren Sie als nächstes eine weitere `TF1`-Funktion gemäß $f(x;\alpha,\beta)$ unter Benutzung der angepassten Werte $\hat{\alpha}$ und $\hat{\beta}$. Die Werte von $\hat{\alpha}$ und $\hat{\beta}$ können Sie mit

```
TF1::GetParameter(int i)
```

ermitteln, wobei der Wert des `i`-ten Parameters zurückgegeben wird. Benutzen Sie dann

```
TF1::GetRandom(),
```

um eine gemäß der so definierten Funktion verteilte Zufallszahl zu erhalten.

(iii) Erzeugen Sie N_{tot} neue Werte für $\cos\theta$ und füllen Sie diese in ein Histogramm. Führen Sie eine Anpassung der Funktion $f(x;\alpha,\beta)$ an das neues $cos(\theta)$-Histogramm durch.

(iv) Ermitteln Sie als nächstes den χ^2-Wert für das erzeugte Histogramm, indem Sie die Formel

$$\chi_p^2 = 2\sum_{i=1}^{N}\left(n_i \ln\frac{n_i}{\hat{\nu}_i} + \hat{\nu}_i - n_i\right)$$

benutzen. Dabei sind N die Anzahl der Histogrammbins und $\hat{\nu}_i$ die Erwartungswerte der angepassten Funktion $f(x;\hat{\alpha},\hat{\beta})$, die gegeben sind durch

$$\hat{\nu}_i = N_{\text{tot}}\int_{x_i^{\min}}^{x_i^{\max}} f(x;\hat{\alpha},\hat{\beta})\mathrm{d}x.$$

Zum Integrieren der Funktion $f(x;\hat{\alpha},\hat{\beta})$ sollten Sie die Methode

```
TF1::Integral(int binMin,int binMax)
```

verwenden. Die untere Bingrenze des `i`-ten Bins erhalten Sie mittels

```
TH1::GetBinLowEdge(int i).
```

Schreiben Sie eine Schleife über alle Histogrammbins, in der Sie die Einzelbeiträge zu χ_m^2 aufsummieren.

(v) Wiederholen Sie dieses Vorgehen 1000mal und füllen Sie jedesmal den Wert von χ_m^2 in ein Histogramm ein.

(vi) Stellen Sie das Histogramm am Bildschirm dar und überzeugen Sie sich davon, dass es einer χ^2-Verteilung folgt. Führen Sie, falls Sie noch Zeit haben, eine Anpassung einer χ^2-Funktion an das Histogramm durch.