Albert-Ludwigs-Universität Freiburg Sommersemester 2009

Statistische Methoden der Datenanalyse

Markus Schumacher

Übung XII

Matthew Beckingham und Markus Warsinsky

22.7.2009

Computerübung

Die ROOT-Makros zu dieser Übung liegen in `/home/warsinsk/stat/uebung12/` und werden nach der Übung oder nach Bedarf für alle lesbar gemacht.

Aufgabe 48 *Abdeckungswahrscheinlichkeit von Vertrauensintervallen*

In dieser Aufgabe soll die Abdeckungswahrscheinlichkeit von Vertrauensintervallen untersucht werden. Dabei sollen zwei beliebte Methoden, einerseits der sogenannte „Unified Approach“ nach Feldman und Cousins, und andererseits die von Rolke et al. entwickelte bzw. wiederentdeckte „Profile Likelihood“, betrachtet werden. Beide Methoden sind in ROOT implementiert und können demnach einfach benutzt werden.

In beiden Fällen wollen wir ein einfaches Zählexperiment betrachten. In diesem Zählexperiment erwartet man in seiner Signalregion im Mittel $b = 4{,}7$ Untergrund- und μ Signalereignisse. Wie in der Vorlesung besprochen, kann es durch statistische Fluktuationen dazu kommen, dass weniger Ereignisse, als vom Untergrund erwartet beobachtet werden. Der beste Schätzer auf die Signalrate $\hat{\mu} = n_{obs} - b$ wird also negativ.

(i) Zuerst soll die Methode nach Feldmann und Cousins betrachtet werden. Diese kann in ROOT wie folgt benutzt werden:

```
TFeldmanCousins* mylim=new TFeldmanCousins(0.9);
double b=4.7;
double mu=0.5;
int nobs=gRandom->Poisson(mu+b);
double upperlim=mylim->CalculateUpperLimit(nobs,b);
double lowerlim=mylim->CalculateLowerLimit(nobs,b);
```

wobei `upperlim` bzw. `lowerlim` die obere bzw. untere Grenze auf die wahre Signalrate μ zu einem Vertrauensniveau von 90% sind. `nobs` ist die im Experiment beobachtete Ereigniszahl und `b` die Untergrundvorhersage b. Wir nehmen hierbei an, dass b perfekt vorhergesagt wird.

Im folgenden wollen wir Pseudoexperimente gemäß dieser Parameter durchführen, in jedem Pseudoexperiment das Vertrauensintervall konstruieren und prüfen, ob das gebildete Vertrauensintervall den wahren Wert abdeckt. Dabei sollen verschiedene Werte von μ getestet werden. Gehen Sie dazu wie folgt vor:

- Im folgenden wird es sehr oft notwendig sein, das obere bzw. untere Ende des Vertrauensbereiches für ein gegebenes b und ein zufällig bestimmtes n_{obs} zu ermitteln. Die Benutzung der betreffenden Methoden in ROOT ist allerdings sehr zeitaufwändig. Allerdings können wir uns hierbei einer kleinen Hilfestellung bedienen: Da n_{obs} nur diskrete Werte annehmen kann, muss man, sobald man dasselbe n_{obs} nochmals auswürfelt nicht nochmal das Vertrauensintervall konstruieren. Dazu kann man sich zum Beispiel Felder `double upperlimits[1000];` und `double lowerlimits[1000];` definieren und zu Anfang alle Werte auf eine negative Zahl setzen. Bei einem Pseudoexperiment mit `nobs` beobachteteten Ereignissen, kann man nun prüfen, ob `upperlimits[nobs]` negativ ist. Falls ja, muss man dieses Vertrauensintervall noch konstruieren und `upperlimits[nobs]` und `lowerlimits[nobs]` auf die errechneten

Werte setzen. Ansonsten kann man einfach die bereits errechneten Werte aus dem Feld entnehmen.

- Schreiben Sie eine Schleife über Werte von μ zwischen 0 und 10 im Abstand von 0,5.
- Führen Sie für jeden Wert von μ 100000 Pseudoexperimente durch, in denen Sie die Anzahl an beobachteten Ereignissen poissonverteilt mit einem Mittelwert $\mu + b$ mittels `gRandom->Poisson(b+mu)` bestimmen.
- Ermitteln Sie damit das Vertrauensintervall zu dem jeweiligen Experiment.
- Ermitteln Sie, in wievielen der 100000 Pseudoexperimente der wahre Wert von μ innerhalb des Vertrauensintervalls liegt.
- Tragen Sie diese Abdeckwahrscheinlichkeit gegen μ auf. Sie können dazu einen `TGraph` oder einen `TGraphErrors` verwenden, wobei Sie Punkte des mit dem Befehl `graph->SetPoint(i, x, y)` und Fehler mit dem Befehl `graph->SetPointError(i, ex, ey)` setzen können. Die Fehler auf die Abdeckwahrscheinlichkeit p können Sie beispielsweise als Binomialfehler ansetzen ($\Delta p = \sqrt{p(1-p)/N_{tot}}$, $N_{tot} = 100000$). Wenn der Trick mit der Zwischenspeicherung der Vertrauenintervalle angewendet wird, sollten die Fehler aber so klein sein, dass man auch darauf verzichten könnte. Entspricht die ermittelte Abdeckungswahrscheinlichkeit Ihren Erwartungen?

(ii) Eine zweite in der Vorlesung besprochene Methode ist die sogenannte „Profile Likelihood“. Dabei benutzt man zur Konstruktion von Vertrauensintervallen die Teststatistik

$$\lambda = \frac{L(\vec{x}|H_0)}{L(\vec{x}|H_1)}, \tag{1}$$

die Sie bereits zum Hypothesentest auf Vorhandensein von nur-Untergrund kennengelernt haben. Zur Konstruktion von Vertrauensintervallen wird nun allerdings die Hypothese H_0 definiert als $\mu = \mu_0$ und H_1 als $\mu \neq \mu_0$. Der Parameter μ_0, der in der Teststatistik verbleibt kann dann zur Konstruktion der Vertrauensintervalle benutzt werden. Insbesondere kann sich der Satz von Wilkes zunutze gemacht werden, der besagt, dass $q = -2\ln\lambda$ für hinreichend große Datensätze nach einer χ^2-Verteilung verteilt ist.

Zusätzlich zu der bereits besprochenen Signalregion wollen wir annehmen, dass eine Seitenbandregion existiert, in der um einen Faktor τ mehr Untergrundereignisse und keine Signalereignisse vorhanden sind. Diese Untergrundregion wird dann gemäß einer Poissonverteilung mit Mittelwert τb simuliert. Nehmen Sie im folgenden $\tau = 10$ an.

Dieses Verfahren ist in der ROOT-Klasse `TRolke` implementiert. Leider sind einige Methoden, die eine komfortable Benutzung dieser Klasse erlauben sollte, nicht in der ROOT-Version 5.18 im CIP-Pool vorhanden. Zur Verwendung von `TRolke` kann man dann wie folgt vorgehen:

```
TRolke* myrolke = new TRolke(0.9);
myrolke->CalculateInterval(x, y, 0, 0, 1.0, 1.0, 4, 0., 0., tau, 0., 0);
double upperlim=myrolke->GetUpperLimit();
double lowerlim=myrolke->GetLowerLimit();
```

Dabei sind `x` die Anzahl der Ereignisse in der Signal- und `y` die Anzahl der Ereignisse in der Seitenbandregion, sowie `tau` die relative Anzahl von Untergrundereignissen in der Seitenband- zur Signalregion.

Gehen Sie nun wie folgt vor:

- Im Prinzip könnte man auch für diesen Fall die Vertrauensintervalle zwischenspeichern, um sie nicht jedesmal neu berechnen zu müssen. Dabei müßte man natürlich beachten, dass man zwei ganze Zahlen (Signal- und Untergrundregion) vorhanden hat, die beide in das Vertrauensintervall eingehen. Allerdings ist die Profile Likelihood aufgrund des angewendeten Satzes von Wilkes erheblich schneller als die Konstruktion nach Feldman und Cousins, weshalb man auf dieses Vorgehen verzichten kann.
- Ansonsten können Sie genauso wie im Aufgabenteil (i) vorgehen, denken Sie aber daran, die Ereignisse in der Seitenbandregion gemäss einer Poissonverteilung mit Mittelwert τb zu ermitteln.

Was fällt Ihnen auf? Untersuchen Sie, was passiert, wenn Sie die Signal- und Untergrundrate verzehnfachen.

(iii) Bei der in Unteraufgabe (ii) benutzten „Profile Likelihood“ wurde nicht vorausgesetzt, dass der beste Schätzer für μ innerhalb der physikalisch erlaubten Region liegt. Das Vertrauensintervall wurde in diesem Fall mittels einer komplizierteren Methode konstruiert. Beispielsweise wird im

Falle, dass man weniger Ereignisse als der vorhergesagte Untergrund beobachtet, die Anzahl an Ereignissen in der Signalregion solange künstlich vergrößert, bis man in der physikalisch sinnvollen Region landet und somit eine positive obere Grenze angeben kann. Für Details sei auf die unter `http://lanl.arxiv.org/abs/physics/0403059` zu findende Veröffentlichung verwiesen.

Es existiert noch eine zweite Methode, die auch in der Vorlesung besprochen wurde, indem in der Maximierung der Likelihood eine nicht-negative Signalrate als Randbedingung gefordert wird. Sollte der beste Schätzer im Negativen liegen, wird der beste Schätzer auf 0 gesetzt und von diesem Punkt aus an der Likelihood-Kurve das Vertrauensintervall abgelesen.

Diese Option können Sie mit

```
myrolke->SetSwitch(true);
```

einschalten. Wiederholen Sie das Vorgehen aus Unteraufgabe (ii) mit dieser Einstellung und vergleichen Sie die Ergebnisse.

Aufgabe 49 *95% CL obere Grenzen im Zählexperiment*

Auch in dieser Aufgabe soll wieder ein Zählexperiment betrachtet werden. Diesmal wollen wir einen festen Untergrund mit einem Mittelwert von $b = 3{,}8$ und eine feste Signalrate von $\mu = 0{,}5$ betrachten. Aufgrund statistischer Schwankungen kann es also durchaus möglich sein, dass man im Experiment weniger Ereignisse beobachtet als vom Untergrund erwartet. Im Extremfall kann es auch passieren, dass kein Ereignis beobachtet wird.

Im folgenden wollen wir eine alternative Methode betrachten, die bereits bei LEP benutzt wurde, um Grenzen auf Signalraten anzugeben. Dazu betrachtet man das auffinden einer Ausschlussgrenze als Hypothesentest auf die Hypothese Untergrund plus Signal Wenn man sich ein Vertrauensniveau von $1-\alpha$ vorgibt, wird dann also die obere Grenze auf μ definiert durch den P-Wert dieses Tests:

$$CL_{s+b} = \sum_{n=0}^{n_{obs}} P(n; b+\mu) = \alpha, \tag{2}$$

wobei über die Poissonwahrscheinlichkeiten von 0 bis zum beobachteten n_{obs} summiert wird[1]. Die obere Grenze auf μ kann man dann durch Durchfahren von CL_{s+b} mit verschiedenen μ bestimmen.

Wie wir in dieser Übung sehen werden, stellt sich hierbei allerdings das Problem, dass bei sehr niedrigen Signalraten aufgrund von statistischen Fluktuationen 0 Ereignisse beobachtet werden können. In diesem Fall lässt sich ein beliebig kleines μ zu beliebigem Vertrauensniveau ausschliessen, was offensichtlich nicht akzeptabel ist. Aus diesem Grund kann man mittels

$$CL_b = \sum_{n=0}^{n_{obs}} P(n; b), \tag{3}$$

eine weitere Größe als

$$CL_s = \frac{CL_{s+b}}{CL_b} \tag{4}$$

definieren, die „ad hoc“ einen solchen Ausschluss verhindert, da hier auch die erwartete Untergrundrate eingeht. CL_s wird dann analog zu CL_{s+b} benutzt, um eine obere Grenze anzugeben.

Diese Methodik ist in der ROOT-Klasse `TLimit` definiert. Diese kann man zum Beispiel wie folgt verwenden:

```
TH1D* data=new TH1D("data","data",1,0,1);
TH1D* bkg=new TH1D("bkg","bkg",1,0,1);
TH1D* sig=new TH1D("sig","sig",1,0,1);
double mu=0.5;
double b=3.8;
int nobs=4;// Anzahl beobachteter Ereignisse
sig->SetBinContent(1,mu);
bkg->SetBinContent(1,b);
data->SetBinContent(1,nobs);
TLimitDataSource* mydatasource = new TLimitDataSource(sig,bkg,data);
```

[1] Hinweis: In der Vorlesung wurde CL_{s+b} als $1 - CL_{s+b}$ definiert. In dieser Übung wird auf diese Notation umgestellt, um kompatibel mit den benutzten Methoden in ROOT zu sein.

```
TConfidenceLevel *myconfidence = TLimit::ComputeLimit(mydatasource,5000);
double cls=myconfidence->CLs();
double clsb=myconfidence->CLsb();
```

Das Vorgehen, Signal-, Untergrund- und Datenrate in eindimensionalen Histogrammen zu speichern mag zwar zuerst ein wenig verwunderlich erscheinen, allerdings könnte man auf diese Art und Weise auch problemlos Verteilungen von Observablen (z.B. invariante Massen) betrachten, um eine obere Grenze anzugeben.

Im folgenden sollen die beiden beschriebenen Methoden zur Bestimmung von Ausschlussgrenzen zum 95% Vertrauensniveau anhand des beschriebenen Zählexperimentes verglichen werden. Dies soll anhand von Pseudoexperimenten geschehen, in denen Sie n_{obs} als Zufallszahl nach der Poissonverteilung mit einem Mittelwert $b + \mu$ bestimmen. Gehen Sie dazu wie folgt vor:

(i) Wie oben beschrieben, muss für ein beobachtetes n_{obs} die obere Grenze auf die Signalrate μ_{95} zum 95% Vertrauensniveau gefunden werden, als diejenige Signalrate, bei der CL_{s+b} bzw. CL_s 5% erreicht. Auch in diesem Fall kann n_{obs} aber wieder nur diskrete Werte annehmen. Erstellen Sie sich also vor den eigentlichen Pseudoexperimenten ein Feld, in dem Sie die Ausschlussgrenzen in Abhängigkeit von n_{obs} als Feldindex abspeichern. Beachten Sie dabei die folgenden Hinweise:

- Fahren Sie die Signalrate in 0,1-Schritten durch, bis CL_{s+b} bzw. CL_s 0,05 werden. In diesem Moment können Sie die Schleife mit dem Kommando `break` abbrechen. Genauer als auf 0,1 bestimmen wir die Ausschlussgrenzen also nicht, was Sie bei der Aufteilung der Histogramme beachten sollten.
- Die Ausschlussgrenze für $n_{obs} + 1$ ist immer größer oder gleich der Ausschlussgrenze für n_{obs}.
- Vergessen Sie nicht eventuell innerhalb von Schleifen mit `new` angelegte Zeiger wieder zu löschen.
- Es sollte ausreichend sein, n_{obs} zwischen 0 und 20 durchzufahren.

(ii) Führen Sie danach 10000 Pseudoexperimente durch. Bestimmen Sie jedesmal die Ausschlussgrenze nach der CL_{s+b}- und nach der CL_s-Methode und füllen Sie diese in ein Histogramm.

(iii) Stellen Sie die Histogramme zusammen in verschiedenen Farben dar. Was fällt Ihnen auf?

(iv) Falls noch Zeit ist: Benutzen Sie zusätzlich die in Aufgabe 48 benutzte „Profile Likelihood“, um eine obere Grenze zum 95% Vertrauensniveau zu finden. Benutzen Sie auch wieder ein simuliertes Seitenband, setzten Sie allerdings τ auf einen sehr hohen Wert (z.B. 10000), damit man die damit verbundene statistische Fluktuation vernachlässigen kann. Benutzen Sie `myrolke->SetSwitch(true)`, um den Schätzwert oberhalb von 0 zu haben. Stellen Sie die Ausschlussgrenzen wieder als Histogramm dar. Was fällt Ihnen auf?