RooStats Tutorials

Inversion of Hypothesis Tests
Inversion of Hypothesis Tests

- one-to-one mapping between hypothesis tests and confidence intervals

<table>
<thead>
<tr>
<th>Table 20.1 Relationships between hypothesis testing and interval estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property of test</td>
</tr>
<tr>
<td>Size $= \alpha$</td>
</tr>
<tr>
<td>Power = probability of rejecting a</td>
</tr>
<tr>
<td>false value of $\theta = 1 - \beta$</td>
</tr>
<tr>
<td>Most powerful</td>
</tr>
<tr>
<td>Equal-tails test $\alpha_1 = \alpha_2 = \frac{1}{2}\alpha$</td>
</tr>
</tbody>
</table>

They explained that in statistical theory there is a one-to-one correspondence between a hypothesis test and a confidence interval. (The confidence interval is a hypothesis test for each value in the interval.) The Neyman-Pearson Theorem states that the likelihood ratio gives the most powerful hypothesis test. Therefore, it must be the standard method of constructing a confidence interval.
Performing an hypothesis test at each value of the parameter

Interval can be derived by inverting the p-value curve, function of the parameter of interest (μ)

- value of μ which has p-value α (e.g. 0.05), is the upper limit of 1-α confidence interval (e.g. 95%)
Hypothesis Test Inversion

- use one-sided test for upper limits (e.g. one-side profile likelihood test statistics)
- use two-sided test for a 2-sided interval

Example: 1-σ interval for a Gaussian measurement
HypoTestInverter class

- **Input:**
  - Hypothesis Test calculator (e.g. FrequentistCalculator)
    - possible to customize test statistic, number of toys, etc..
  - N.B: null model is S+B, alternate is B only model

- Interval calculator class
  - scan given interval of $\mu$ and perform hypothesis tests
  - compute upper/lower limit from scan result
    - can use $CL_s = CL_{s+b} / CL_b$ for the p-value
  - store in result (HypoTestInverterResult) also all the hypothesis test results for each scanned $\mu$ value
    - possible to merge later results
  - Can compute expected limits and bands
HypoTestInverter

- **HypoTestInverter** class in RooStats

```cpp
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();
ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);
HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```
Running the HypoTestInverter

Hypothesis test results for each scanned point

Data

Scan result

p-value, CL_{S+B} (or CL_b) is integral of S+B (or B) test statistic distribution from data value

Expected limit and bands are obtained by replacing data test statistic value with quantiles of the B test stat. distribution
Example of Scan

- 95% CL limit on a Gaussian measurement:
  - $\text{Gauss}(x, \mu, 1)$, with $\mu \geq 0$

**Scan of hypothesis tests**

**deficit, observation $x = -1.5$**

**excess, observation $x = 1.5$**

use CL$_s$ as p-value to avoid setting limits which are too good
Limits on bounded measurements

from Bob Cousins:

Downward fluctuations in searches for excesses

Classic example: Upper limit on mean $\mu$ of Gaussian based on measurement $x$ (in units of $\sigma$).

If $\mu \geq 0$ in model, as measured $x$ becomes increasingly negative, standard classical upper limit becomes small and then null.

Issue acute 15-25 years ago in expts to measure $\nu_e$ mass in (tritium $\beta$ decay): several measured $m_\nu^2 < 0$.

Frequentist 1-sided 95% C.L. Upper Limits, based on $\alpha = 1 - \text{C.L.} = 5\%$ (called CL$_{sb}$ at LEP).
For $x < -1.64 \sigma$ the confidence interval is the null set!

Bob Cousins, CMSDAS, 1/2012
Feldman-Cousins intervals

- HypoTestInverter class can compute also a Feldman-Cousins interval
- need to use FrequentistCalculator and CL_{s+b} as p-value
- use the 2-sided profile likelihood test statistic

![Feldman-Cousins Interval](image1)

**Observation x = -1.5**

![Feldman-Cousins Interval](image2)

**Observation x = 1.5**
Feldman-Cousins Interval

from Kyle Cranmer:

Most people think of plot on left when thinking of Feldman-Cousins
- bars are regions “ordered by” $R = P(n|\mu)/P(n|\mu_{\text{best}})$, with $\int_{x_1}^{x_2} P(x|\mu) dx = \alpha$.

But this picture doesn’t generalize well to many measured quantities.
- Instead, just use $R$ as the test statistic... and $R$ is $\lambda(\mu)$

$$t_\mu = -2 \ln \lambda(\mu)$$

$$P_\mu = \int_{t_\mu,\text{obs}}^{\infty} f(t_\mu|\mu) dt_\mu$$
Asymptotic Formulae

- Use the asymptotic formula for the test statistic distributions
- For one-sided profile likelihood test statistic:
  - null model ($\mu = \mu_{\text{TEST}}$)
    - half $\chi^2$ distribution
  - alt model ($\mu \neq \mu_{\text{TEST}}$)
    - non-central $\chi^2$
    - use Asimov data to get the non centrality parameter $\lambda$
  - p-values for null (CL$_{s+b}$) and alt (CL$_b$) can be obtained without generating toys
  - expected limits can be also obtained using the alt distribution

HypoTestInverter

- **AsymptoticCalculator** class in RooStats
- HypoTestCalculator class implementing the asymptotic formulae

```cpp
// create first HypoTest calculator (N.B null is s+b model)
AsymptoticCalculator ac(*data, *bModel, *sbModel);
HypoTestInverter calc(*ac);
// run inverter same as using other calculators
```

![Combined result graph](image-url)
RooStats Exercises (Part 2)
Getting Started

- all RooStats classes are in a namespace
- recommended to add at beginning of macro:
  - using namespace RooStats
- this will also load automatically the RooStats library
- note that RooStats methods start with upper case letter while RooFit start with lower case
- RooStats calculator are quite verbose, useful to suppress many info messages”

  RooMsgService::instance().setGlobalKillBelow(RooFit::WARNING) ;

Roostats reference guide:
  http://root.cern.ch/root/htmldoc/ROOSTATS_Index.html

RooStats tutorial macros:
Exercise 7:

Compute limit on the exponential background model (same as Ex. 6)

run `SPlusBExpoModel.C`
will create a file `SPlusBExpoModel.root`

run the HypoTestInverter using the frequentist calculator
(use macro `HypoTestInverterDemo.C`)

look and try to understand the code of the macro
use the CL$_S$ option (especially for nobls $\leq b$)
use one-sided profile likelihood test statistics and CL$_S$
will take some time
use `npoints = 5;` number of points to scan
`fc->SetToys(200,100);` number of toys

N.B. comment line `toymcs->SetNEventsPerToy(1);`
model is from an extended pdf
Exercise 7b:

- run the asymptotic calculator
  - use same macro, `HypoTestInverterDemo.C` but create the `AsymptoticCalculator` instead of the `FrequentistCalculator`
  - uncomment relevant code in the macro
  - pass to the `AsHypoTestInverter` class to the `HypoTestInverter` constructor
  - look at how the result is plotted (`HypoTestInverterPlot`)
    - `plot->Draw("obs");` plot only observed p-value
    - `plot->Draw();` plot obs+expected limits with bands
    - `plot->Draw("CLb 2CL");` plot $CL_b$, $CL_{s+b}$, $CL_s$
RooStats Exercises (8)

Exercise 8

- Use Poisson model (with background)
  - run macro `PoissonModelWithBackg.C` (generate model)
  - use possibly same values (nobs, b) used for Bayesian limit (exercise 5)
- compute 95% upper limits using the `HypoTestInverter` class

- use macro `HypoTestInverterDemo.C`
  - look and try to understand the code of the macro
  - use the CL$_s$ option (especially for nobs $\lesssim$ b)
    - use one-sided profile likelihood test statistics and CL$_s$
  - look at how the result is plotted (HypoTestInverterPlot)
    - `plot->Draw("obs");` plot only observed p-value
    - `plot->Draw();` plot obs+expected limits with bands
    - `plot->Draw("CLb 2CL");` plot CL$_b$, CL$_s$+b, CL$_s$
RooStats Exercises (8)

Option:

- compute Feldman-Cousins 95% upper limit
- use two-sided profile likelihood test statistics and $CL_{s+b}$ instead of $CL_s$ for p-value to scan

Option:

- use instead of Poisson simple Gauss model (e.g. with $N=100$ or $N=1$)
- generate it with *GaussianModel.C*
- when using FC and GaussianModel with $N=1$ you can check the result with FC paper [http://arxiv.org/pdf/physics/9711021v2.pdf](http://arxiv.org/pdf/physics/9711021v2.pdf) or google Feldman-Cousins
Solution

- use model created with previous exercises
- use macro HypoTestInverterDemo.C passing workspace name
  - comment/uncomment code depending on exercise
- can also use the tutorials/roostats/StandardHypoTestInvDemo.C
How does it work:

- input workspace file, workspace name
- name of S+B model (null) and for B model (alt)
  - if no B model is given, use S+B model with poi = 0
- data set name
- options:
  - calculator type (frequentist, hybrid, or asymptotic)
  - test statistics
  - use CL$_s$ or CL$_{s+b}$ for computing limit
- number of points to scan and min, max of interval

Example:

- load the macro after having create the workspace using given macro (e.g. SPlusBExpoModel.root)
  
  ```
  root[] .L StandardHypoTestInvDemo.C
  ```

- run for CLs (with frequentist calculator (type = 0) and one-side PL test statistics (type = 3) scan 10 points in [0,100]
  
  ```
  root[] StandardHypoTestInvDemo("SPlusBExpoModel.root","w","ModelConfig","","data",0,3, true, 10, 0, 100)
  ```

- run for Asymptotic CLs (scan 20 points in [0,100])
  
  ```
  root[] StandardHypoTestInvDemo(SPlusBExpoModel.root","w","ModelConfig","","data",2,3, true, 20, 0, 100)
  ```

- run for Feldman-Cousins ( scan 10 points in [0,100])
  
  ```
  root[] StandardHypoTestInvDemo(SPlusBExpoModel.root","w","ModelConfig","","data",0,2, false, 10, 0, 15)
  ```
Documentation and user support

- **RooStats TWiki**: [https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome](https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome)
- **RooStats users guide** (under development, to be completed)
- **RooFit's users guide**: [http://root.cern.ch/drupal/content/users-guide](http://root.cern.ch/drupal/content/users-guide)
- **RooStats November tutorials**:
  - Lecture of L. Lista on statistics: [http://indico.cern.ch/conferenceDisplay.py?confId=73545](http://indico.cern.ch/conferenceDisplay.py?confId=73545)
  - Tutorial contents: [http://indico.cern.ch/conferenceDisplay.py?confId=72320](http://indico.cern.ch/conferenceDisplay.py?confId=72320)
- **RooStats user support**:
  - Submit bugs to ROOT Savannah: [https://savannah.cern.ch/bugs/?func=additem&group=savroot](https://savannah.cern.ch/bugs/?func=additem&group=savroot)
- **Contacts for statistical questions**:
  - ATLAS statistics forum: hn-atlas-physics-Statistics@cern.ch (Cowan, Gross et al)
  - CMS statistics committee: (Cousins, Demortier et al)
    - via hypernews: hn-cms-statistics@cern.ch or directly: cms-statistics-committee@cern.ch