Topical Lectures Statistics – Exercises Set 2

Wouter Verkerke (Dec 2018)

General Instructions

Input files

Input files for exercises can be found in three places

1. At nikhef (stbc-i5.nikhef.nl) in directory `~verkerke/stats2018/`
2. At CERN (lxplus7.cern.ch) in directory `~verkerke/public/stats2018`
3. On the web at `http://www.nikhef.nl/~verkerke/stats2018`

Running ROOT

All exercises are based on ROOT. You are recommended to use version 6.14.04, in which all prepared material has been tested. To pick up a pre-installed ROOT version please execute the following setup script

```
source /cvmfs/sft.cern.ch/lcg/app/releases/ROOT/6.14.04/x86_64-centos7-gcc48-opt/root/bin/thisroot.sh
```

This release will work both at Nikhef and at CERN

Where to work

If you have an account at Nikhef, please work on stbc-i5.nikhef.nl or stbc-i6.nikhef.nl only (these run CentOS7 – required for above ROOT version)

If you have an account at CERN, please work on lxplus7.cern.ch, this will select an CentOS7 node (required for above ROOT version)

You can also work directly on your laptop if you have ROOT installed yourself

Exercise 9 – Exploring the Poisson counting model

Copy file `ex09.C`. This macro performs the following steps

- Construct a Poisson probability model P(N|μS+B) with S,B fixed
- Fits model to 25 observed event → returns fitted value of μ
- Alternatively, explicitly constructs the likelihood L(25|μ) and visualizes that
- Also visualized on the same frame is L(μ)/L(μ-hat)

Questions & explorations

- Do you understand the difference between the likelihood and the likelihood ratio curve?
- How does the interval defined by the rise of the likelihood ratio by half a unit compare to the MINOS error?
- Can you construct the 95% interval from the plot?

Copy file `ex09_build_Poisson.C` This macro builds the same Poisson model as `ex09.C`, but doesn't do any analysis on it. Instead it writes the model, along with a RooStats::ModelConfig object to a workspace file for later analysis

- Look at `ex09_build_Poisson.C`, run it, and quit ROOT
- Open a new ROOT session that reads the ROOT file written by `ex09_build_Poisson.C` named `model.root`.
- Retrieve the workspace from the file

  ```
  RooWorkspace* w = gDirectory->Get("w") ;
  ```

  and print its contents, and understand what is in there.

Exercise 10 – Using RooStats calculators

Now that we have a (very) simple statistical model stored in a workspace that is annotated with information (RooStats::ModelConfig) that uniquely defines the statistical problem that we want to be solved, we can try to run some of the RooStats standard interval calculator classes on this problem.

First copy `ex10_roostats_plr_interval.C`, and run it

- Opens the `model.root` file, retrieves the workspace and from that the `ModelConfig` object (unique statistical problem definition) and also the observed data
- Instantiates a `RooStats` **Profile Likelihood Ratio calculator** and lets it calculate the profile likelihood ratio interval on the above problem
- Reports the results on the command line

Questions & explorations for the **Profile Likelihood Calculator**

- How does the profile likelihood calculator result compare to your manual investigation of the likelihood ratio curve in `ex09.C`?
- Calculate the same type of interval at difference confidence levels, e.g. 65% and 95%.

Next, copy `ex10_roostats_bayes_interval.C`, and run it.

- Opens the `model.root` file, retrieves the workspace and from that the `ModelConfig` object (unique statistical problem definition) and the observed data
- Instantiates a RooStats **Bayesian Calculator** and lets it calculate the Bayesian credible interval on the above problem
- Reports the results on the command line

Questions & explorations for the **Bayesian Calculator**

- How does the Bayesian 90% interval compare with flat prior to the Frequentist Profile Likelihood Ratio interval of the same size
- Run the macro for some different interval shapes: e.g. upper limit, or shortest interval
- Explore what happens for various choices of priors, e.g. $1/\sqrt{\mu}$, or a flat prior for $\mu>0$ only?
- *Note that this Bayesian calculator uses a simple numeric integration engine, it may emit warnings about numeric precision if pushed to perform complex integrations.*

Exercise 11 - Build the Poisson on/off problem

The Poisson on/off is a famous 'standard candle' model among statisticians. The reason is that represents a canonical problem, and (exceptionally) that is also possible to calculate results for analytically, so it is a useful vehicle to calibrate numerical calculation methods. Here we will only explore numeric solutions to this model, but we will compare to known analytical result at the end

First copy `ex11_build_PoissonPoisson.C`, and run it

- Constructs the classic statistical model known as 'on/off':
  A Poisson model for the signal region measuring μ*S+B
  A Poisson model for the control region measuring τ*B
- Here tau is a scale factor for the size of the control region, e.g. if τ=3 then a count of 30 in the control region will predict a background rate of 10 in the signal region with a relative error of 10/√30.
- Constructs a RooStats `ModelConfig` and saves everything to a workspace on file.

Questions and explorations

- Do you understand the observed uncertainty on the fitted background rate in the SR? (In terms of given numbers, `N_CR`=200, `tau`=10, `N_SR`=25)?
- Run the RooStats **PLR** and **Bayesian** calculators on this model (from ex10)
- *Can you reproduce the 'standard candle' result* `N_SR`*=178,* `N_CR`*=100,* `tau`*=1 of in the course and confirm that it's significance is exactly 5 sigma?*
  To do so, plot a scan of the profile likelihood ratio of this problem (see `ex09.C` *on how to do that), and look at the value of the PLR for μ=0*

Next, copy `ex11_build_PoissonPoissonGlobs.C`, and run it.

- This macro build exactly the same model as the previous macro, but with a technical difference – *it formulates the observable of the control region as a 'global observable'.*

A regular likelihood of two models is written in terms of data and models as

D(x,y) ←→ F(x,y|param) = F(x|param)*F(y|param)

so that in L(param) the values of x,y for F are taken from D(x,y). Observables of subsidiary measurements (generalized control regions) often have trivial values (in most cases 0), and there are also usually very many of them, so we prefer not to carry those in the dataset, hence the Likelihood construction is modified as follows

D(x) ←→ F(x,y|param) = F(x|param)*F(**y=0**|param)

In this reorganization, the value of the observable y is 'hardcoded' in the model F(y) so that it can be omitted from the dataset, and is called a '**global observable**'

Exercise 12 – The concept of subsidiary measurements

This exercise introduces the concepts of *subsidiary measurements*, which are a generalization of sideband measurement.

A subsidiary measurement can be sideband measurement, but it can also be more abstract: a Gaussian approximation of an external measurement, or even a 'theory measurement', i.e. a likelihood function that encodes the a 'measurement' that represent a theoretical calculation along with its uncertainty.

Along with subsidiary measurement comes the concept of response functions that map the unit Gaussian to the desired response in the physics measurement.
For example for an original measurement written as

$$F(N_{sig},B_{obs}|S,B) = Poisson(N_{sig}|S+B)*Gaussian(B_{obs}|B,\sigma_B)$$

is identically expressed as subsidiary measurement as follows

$$F(N_{sis}|S,B) = Poisson(N_{sig}|\ S + \mathbf{B*(1+\alpha_B*\sigma_B)}\ )*Gaussian(\mathbf{0}|\alpha_B,1)$$

Here the response function **$B*(1+\alpha_B*\sigma_B)$** ensures that the unit Gaussian subsidiary measurement has the same impact on the main measurement as the original Gaussian($B_{obs}|B,\sigma_B$). In the process B_{obs} has also been eliminated from the dataset, as it now replaced by global observable defined inside the dataset

First copy `ex12_build_PoissonGaussGlobs.C` and run it. *This macro implements the above implementation of a Gaussian subsidiary measurement*.

- This macro builds a variant of the model of `ex11_build_PoissonPoissonGlobs.C` – it changes the control region model that measured the background B from a Poisson to a Gaussian.
- It also maps the physics effect (the magnitude of the uncertainty) in a **response function** encoded in the signal region probability model in terms of a nuisance parameter alpha, and reduces the **subsidiary measurement** of alpha to a **unit Gaussian**.
- Writes probability model and RooStats `ModelConfig` to output file

Questions and explorations

- Identify the piece of code that encodes the response function of the systematic uncertainty.
- Modify the response function such that magnitude of the systematic uncertainty is doubled and rerun
- Analyze the model of ex12 with the RooStats **PLR** and **Bayesian** calculators

Exercise 13 – Limit setting procedures with CL_S

A regular 95% limit setting procedure entails finding the value of μ for which the **p-value** of the corresponding test statistic **q_μ is 0.05.** The **PLR test statistic q_μ** is especially designed for upper limit setting, as it will regard any dataset with fluctuation to values greater than expected for the tested hypothesis μ to maximally compatible with μ (so that these are not counted 'against' the hypothesis)

The CL_S technique is one of the procedures used in HEP in limit setting to avoid so-called spurious exclusions. Spurious exclusions happen when the observed event rates is (well) below the background-only expectation, in which case a limit setting procedure may report that *all* signal strengths of a given model are excluded at the stated confidence level (usually 95%).

The CL_S procedure is an 'after burner' on any self-contained limit setting procedure: instead of finding the point where **q_μ=0.05,** the point where **$CL_S=q_\mu/(1-q_0)$=0.05** is found, where the denominator $1-q_0$ gives the p-value for the background-only hypothesis. If **$(1-q_0)$** becomes <<1 that implies the observed data is also unlikely under the background-only hypothesis (e.g. a strong negative fluctuation w.r.t the background), and by dividing by this number the CL_S value is increased and the limit calculation will continue to include this hypothesized μ value in the included interval (i.e. it will not be in the excluded interval of μ values)

First copy `ex13_roostats_cls_limit.C`

- Implements the RooStats **hypothesis test inverter limit calculator**. This is the most general limit calculator.
- In this macro the calculator is configured to use the **profile likelihood ratio test statistic** for the limit calculation, *and to assume its known* **asymptotic distributions**
- In the final limit calculation the CL_S technique is enabled, which is designed to always return non-empty intervals in the range [0,X]

Questions and explorations

- Run the calculator first on several of the models built so far (Single Poisson ex09, Two Poissons ex11, Poisson/Gaussian ex12)

Understand the working of the calculator by identifying its pieces

- An explicit alternative hypothesis is constructed from the workspace (`ModelConfig` for b-only hypothesis). This is needed for the calculation of CL_S and for the calculation of expected limits under the B-only hypothesis
- Set up an (asymptotic) calculator that can calculate the p-value of the data under both hypothesis (B-only, and S+B(for a given value of μ)

- Configure the Inverter, the tool that will vary mu in such a way that

  CL_S = p-value(S+B)/(1-p-value(B))

  corresponds to the desired confidence level. The value of μ for which this is true is then reported as the CL_S upper limit

Optionally, you can try also `ex13_roostats_cls_limit_toys.C`, which is identical to the previous macro except that it does *not assume asymptotic distributions*, but rather samples *these from toy MC runs*. This calculator configuration is also valid at low statistics where the asymptotic formulae are not, **but is very substantially more expensive to evaluate.**