Testing Statistical Tests

Anne M. Archibald (archibald@astron.nl)

ASTRON

2015 April 23

Statistical tests

A statistical test is a procedure that takes data as input and reports whether that data contains a feature of interest.

Examples:

- K-S, Kuiper, or Anderson-Darling test for whether samples have the same distribution
- χ^2 test for whether data is well-fit by a model
- H test for whether circular data is uniform

All yield a number describing the amount of deviation from the null hypothesis.

Testing statistical tests

Understanding the test:

- How significant is the result?
- How sensitive is the test?
- Is the implementation correct?

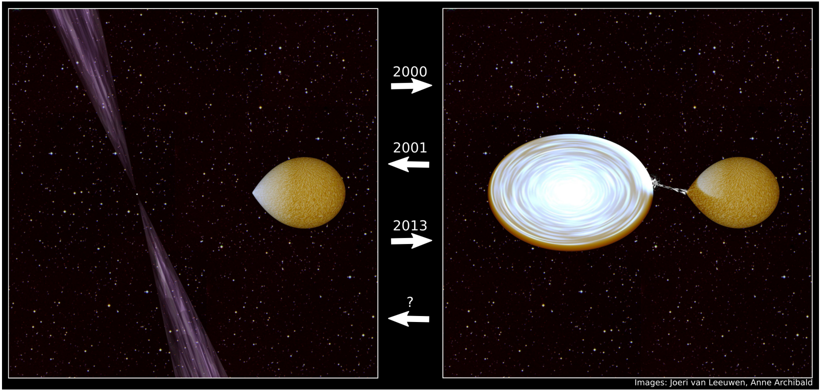
If there is a detection:

- How confident can we be?

If it's a null result:

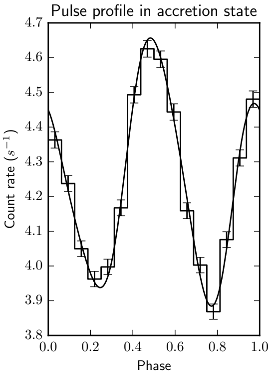
- What upper limits can we place?

The demo problem



PSR J1023+0038 in radio pulsar and X-ray binary states

The demo problem

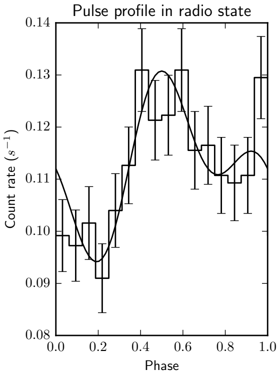


PSR J1023+0038 is an X-ray source in both MSP and LMXB states

- Based on radio timing we can compute pulse phase for each photon
- Period is 1.7 ms, so rotational coverage is uniform
- Much brighter in LMXB than radio state
  - LMXB: 542871 photons
  - Radio: 3746 photons

Are these photons pulsed?

The demo problem



PSR J1023+0038 is an X-ray source in both MSP and LMXB states

- Based on radio timing we can compute pulse phase for each photon
- Period is 1.7 ms, so rotational coverage is uniform
- Much brighter in LMXB than radio state
  - LMXB: 542871 photons
  - Radio: 3746 photons

Are these photons pulsed?

The H test

The H test is based on the "empirical Fourier coefficients":

$$c_m = \sum_{k=1}^{N} e^{2\pi i m \phi_k}$$

It chooses an optimal number n of coefficients to represent the profile and reports the total power in those n coefficients:

$$H = \max_n \sum_{m=1}^{n} 2|c_m|^2/N - 4$$

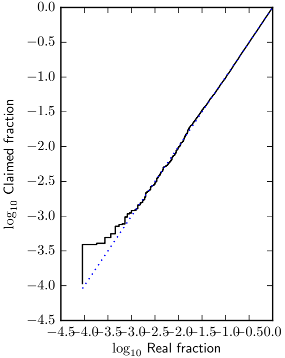
False positive probability

To evaluate the significance of a particular value of H, we need to know the probability of obtaining a value of H this large for photons that are actually uniformly distributed (null hypothesis): the *false positive probability*.

For the plain H test this can be computed analytically:

$$FPP = e^{-0.398405H}$$

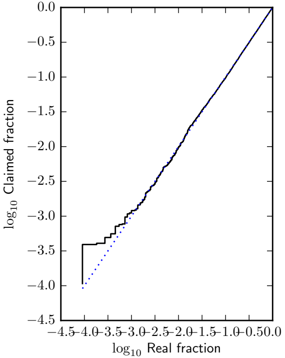
Experimentally testing the FPP



Fortunately such tests are pretty easy to run:

```
def fake_H(n):
    phases = np.random.uniform(size=(n,))
    P = photon_tools.fold_phases(phases)
    return P.H()

fake_Hs_radio = []

for i in range(10000):
    fake_Hs_radio.append(fake_H(len(phases_radio)))

plt.plot(np.log10((np.arange(len(fake_Hs_radio))+1)
                  /len(fake_Hs_radio)),
         fake_Hs_radio/np.log(10),
        color='k', drawstyle='steps')
```

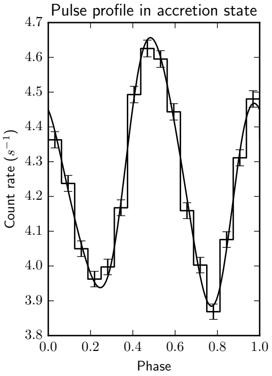
Experimentally testing the FPP



You can also build an automated test:

```
>>> p = scipy.stats.norm.sf(3)
>>> print p
0.00134989803163
>>> n = len([H for H in fake_Hs_radio
                          if H<np.log(p)])
>>> dist = scipy.stats.binom(
        p=p,n=len(fake_Hs_radio))
>>> print (dist.ppf(0.01),
           n,
           dist.ppf(0.99))
7.0 15 24.0
```

Experimentally determining the FPP

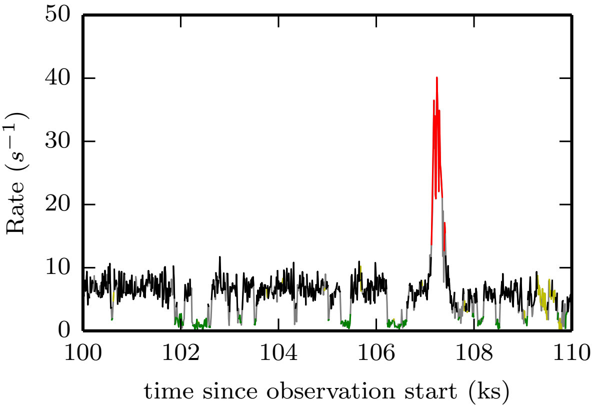


Sometimes you have the “best-fit” H vaue obtained by fitting for a parameter

- Analytic FPP wrong!
- Roughly: multiply by number of independent trials
- Experimentally:
  - Repeat the fitting process on null data
  - Determine how often the null H is more significant than the observed
  - Use the binomial probability distribution to account for the limited number of simulations

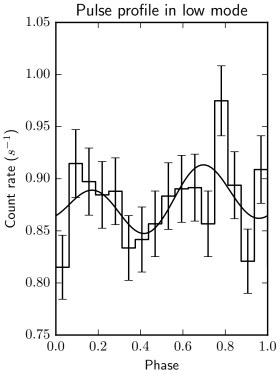
Sensitivity

In the LMXB state, the PSR J1023+0038 light curve shows modes:



Are there pulsations in the low mode?

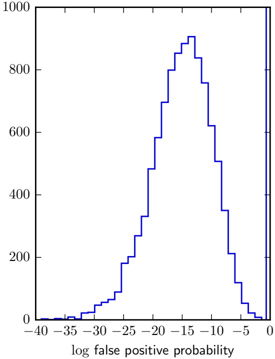
Sensitivity



The H-test false positive probability for the low mode is 0.52

- Definitely a non-detection
- But how strong could the pulsations be?
- In particular, could they be at the same fractional level as in the observation overall?

Sensitivity

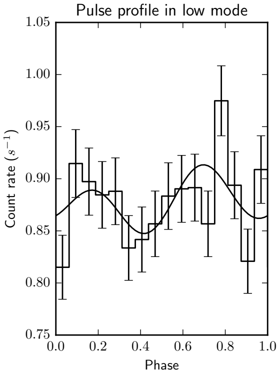


Make fake data by choosing photon phases from the whole observation:

```
def fake_low_H():
    phases = np.random.choice(phases_nov,
                              size=len(phases_low))
    P = photon_tools.fold_phases(phases)
    return P.H()

fake_low_Hs = []

for i in range(10000):
    fake_low_Hs.append(fake_low_H())

astroML.plotting.hist(fake_low_Hs,
                      bins='knuth',
                      histtype='step')
plt.axvline(H_low)
```

So the low mode cannot be as pulsed as the observation as a whole.

Actual upper limits



- Can generate weaker pulsations by mixing in uniformly-distributed phases
- Adjust fraction until 95% of fake data have a higher FPP than observed

Practical concerns

Or, why haven't you written a package to just do this?

- It's slow. Really slow if you have to adjust parameters.
  - Embarrassingly parallel. Ipython parallel notebooks can do it well: `http://lighthouseinthesky.blogspot.ca/2014/10/parallel-ipython-notebooks.html`
- There are often analytical speedups.
  - For the H test you can often work with $\sim$20 Fourier coefficients rather than photons.
- It's very observation-dependent.
  - Your simulations should include all relevant fitting from your data analysis.
- Part of a more general approach to statistics.
  - If you don't understand the behaviour of your analysis procedure (or if you do but want to check), simulate.

What about Bayesian methods?

More complex and less standard:

- Hypothesis testing: null model versus family of signal models
- Need explicit priors
- Computation often (not always) needs an MCMC step
- Result: probability/log-odds of having a signal

Testing is challenging:

- Can draw fake data sets from the prior, then fit
- Obtain a list of (real/fake, claimed probability)
  - Complicated to effectively test whether the probabilities match the truth values