Sanity Checks

David Duvenaud

Cambridge University
Computational and Biological Learning Lab

April 24, 2013

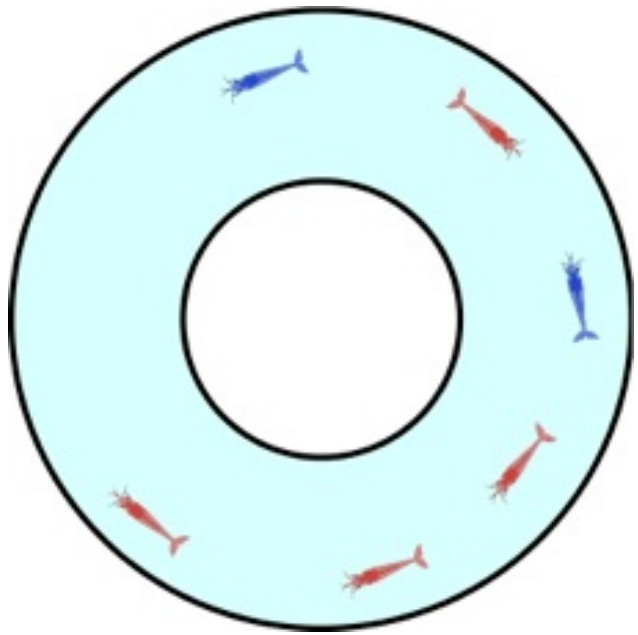
A Simple Example

Comparing Models of Prawn Minds

- Paper: Comparing evidence for different models of prawn behavior.
- Requires inference conditioned on results of experiments.
- Author's highest-impact publication venue so far.

Actual Experiments:

102 reptitions of prawn flocking:



```
function [logP, samples] = logP_mc_ring_memory(theta, direction, N, modelidx, type)

if nargin < 5
    type = 1;

    if nargin <4
        modelidx = 1;
    end
end

%downsample inputs, coreelation length is ~10 frames
for i = 1:numel(theta)
    theta = theta(:, 1:2:end);
    direction = direction(:, 1:2:end);
end

logP = zeros(N, 1, 'double');
samples = zeros(N, 6, 'double');

priormin = [0, 1, -2, -2, 0, -7.5];
priormax = [pi, 5, 2, 2, 1, -7.49];
priorrange = priormax-priormin;

switch modelidx
    case 0
        log_l_pdf = @(x) logP_ring_null(theta, direction, x(1), x(2), x(3:4), x(5), x(6));
    case 1
        log_l_pdf = @(x) logP_ring_mf(theta, direction, x(1), x(2), x(3:4), x(5), x(6));
    case 2
        .
        .
        .
```

Is anything amiss?

```
% downsample inputs,
% coreelation length is ~10 frames
for i = 1:numel(theta)
   theta = theta(:, 1:2:end);
   direction = direction(:, 1:2:end);
end
```

- theta is a cell array, one cell per experiment, each iteration discards half the experiments!
- At the end of the loop, only 1 of 102 experiments left.
- So many pointless experiments!

Is anything amiss?

```
% downsample inputs,
% coreelation length is ~10 frames
for i = 1:numel(theta)
   theta = theta(:, 1:2:end);
   direction = direction(:, 1:2:end);
end
```

- theta is a cell array, one cell per experiment, each iteration discards half the experiments!
- At the end of the loop, only 1 of 102 experiments left.
- So many pointless experiments!

The lesson: never release your code

[update: Was fixed and re-published:
www.ploscompbiol.org/article/
info:doi/10.1371/journal.pcbi.1002961]

Very Common

In Machine Learning

- 2009: Oxford vision group retraction after including test cases in training set.
- A unnamed lab member almost didn't include results in NIPS paper because of sign error in plots.
- Retraction Watch Blog: `retractionwatch.wordpress.com`

In General

- Your code will have bugs!
- My rate: about 1 per line of matlab.

How to trust anything?

When writing code

Standard Advice

- Write Unit Tests
- Physicists have good protocols.
- Compute same thing in different ways.
- Use checkgrad!

Carl's Advice

- Re-write your code until it uses the right data structure.
- Keep your code short and simple, no corner cases.
- Ideally, everything fits on one page.

When writing code

Standard Advice

- Write Unit Tests
- Physicists have good protocols.
- Compute same thing in different ways.
- Use checkgrad!

Carl's Advice

- Re-write your code until it uses the right data structure.
- Keep your code short and simple, no corner cases.
- Ideally, everything fits on one page.

These methods work but slow you down

When running experiments

Things to always compare against:

- A random guesser (finds bugs in evaluation code)
- Always guesses mean/mode (finds too-easy problems)
- 1-nearest neighbour (finds bugs in train/test splitting)

Datasets to include

- A trivial-to-predict dataset (finds major bugs in any method)
- A dataset with no signal (finds bugs in evaluation code)
- A translated, scaled version of dataset (finds bugs in implementation of model)

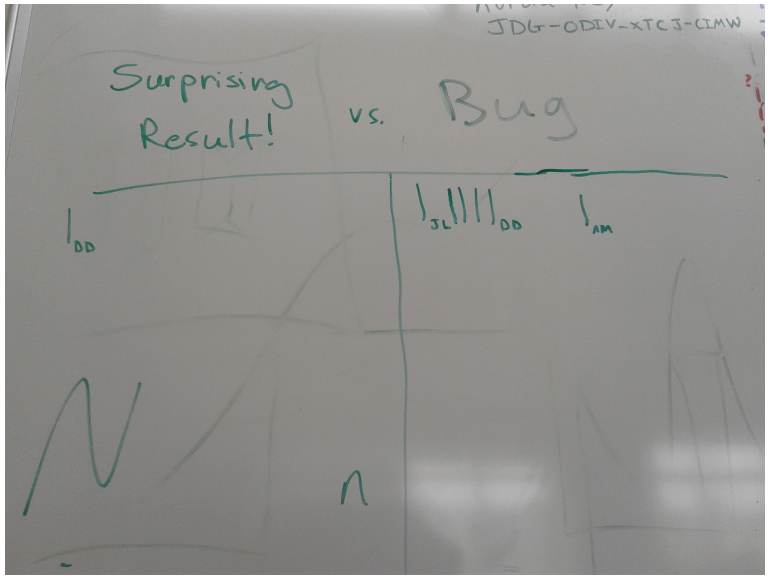
Can detect problems without looking at code

In General

Notice Confusion

- Notice when you're confused
- Notice when you're rationalizing
- Red flag: Looking at only one number and making up a story about why it goes up or down (i.e. cog sci)

Empirical Rates



Look at details until they aren't suprising

Main Takeaways

To keep in mind

- You probably have bugs
- Finding them early saves time
- Finding them before you publish saves retractions

To practice

- Include simple baselines
- Check invariants
- Plot everything
- Keep things simple

A few sanity checks go a long way