Software engineering for cognitive neuroscientists

3rd April 2013
Princeton Psychology Dept

Tuesday, 9 April 2013

I'm going to try and make this interactive

But if that means I don't get to hear as much of my own voice as I'd like, we'll switch tacks

http://github.com/gregdetre/secns

TOOLS

Version control

Subversion (Princeton hosted?)

OR

Git - see GitHub.com and app for Mac

Tuesday, 9 April 2013

Without version control, you're like a Michelin chef trying to cook over a bonfire

WRITE FOR A STRANGER

Imagine the person reading your code is hungry, tired, has a violent history, and knows where you live.

Tuesday, 9 April 2013

The person reading my code is usually ME (in which case, all 4 are true)

In a year's time, you will be a stranger to your present self.

Good comments

High-level goal: what is it trying to achieve?

What kinds of inputs does it expect? Examples

What kinds of outputs does it return? Examples

I tried another way, but ended up doing it this way because...

Explain unusual/complex bits

Comment before you write the code

Examples of bad comments:

Bad comments

```
% I'm so sory about this next bit of code.
...

% Loop over 100 times
For x:1:100
```

Good coding practices

Break functions into bite-sized chunks

Don't repeat yourself

Variable naming

Encapsulation

Etc

http://www.python.org/dev/peps/pep-0020/

https://github.com/thomasdavis/best-practices#programming-best-practices-tidbits

TREAT YOUR DATA LIKE A HOSTILE WITNESS

Tuesday, 9 April 2013

[cf the Cartesian demon]

3 teams:

- Write the analysis
- White-box test the algorithm is working
- The hostile witnesses

LOTS OF BABY STEPS

How do you eat an elephant?

Validate on small data, build up

- Define your metric
- Run it on small data (quick, while prototyping)
- Show that you get better as you add more data

Tuesday, 9 April 2013

how do you eat an elephant? one bite at a time. start small, with a tiny subset of your data. that way, the algorithm runs quickly while you're prototyping

CANARIES IN THE DATACOALMINE

Fake data

Generate data that looks exactly the way you expect

Can be hard to do, but often helps you think things through

Confirm that the output looks as it should

Useful for orienting audience in presentations

Nonsense/scrambled data

Set a trap. Feed your algorithm nonsense data. It had better tell you the results aren't significant!

Easy: shuffle regressors/labels or feed in random numbers as data

This. Will Save. Your. Bacon.

e.g. guard against peeking

TESTING

Unit tests

If I call this function with input X, I expect to get output Y back

See: run_unit_tests.m and unit_*.m (e.g. unit_zscore_runs.m) in the MVPA toolbox

Helps you structure your code – if it's easy to test, it'll be easy to understand and refactor

And the tests serve as a kind of how-to guide

Guard against new bugs in old code

Run your unit tests every time you run your analysis

Otherwise you might break something that used to work, and not realize it

Defensive coding

- Pepper your code with asserts and sanity checks
  - e.g. confirm the dimensions, range of values, type of values
- Fail immediately if things are wrong
  - that way you'll notice early on in time and near to the cause of the problem
  - rather than 2 weeks later and in a downstream part of the analysis

Eyeball it

Run imagemat at large scale. You'll easily spot

- outliers
- stripes (e.g. if the scanner wasn't collecting for a while, or one voxel is all-zeros, baseline differences between runs)
- gradients (baseline drift)

REPRODUCIBILITY

Scripts

Version control everything non-dat

- including config files

Commit often

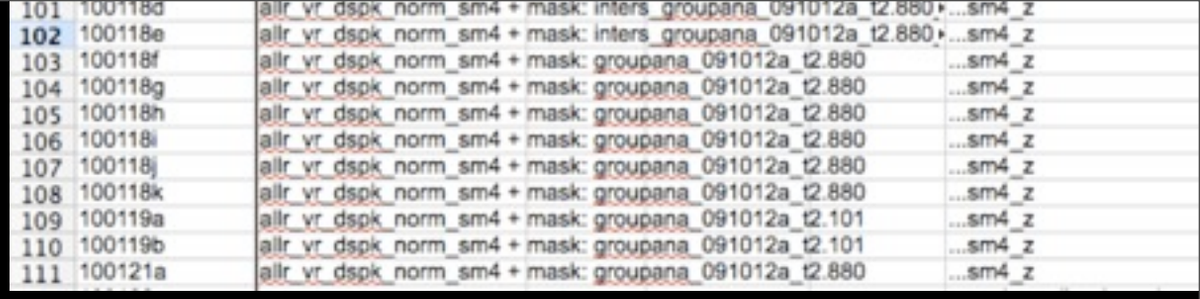
Data

- Keep old versions of your files
  - Structured naming scheme
- Idempotent pipeline scripts
  - so you can effortlessly delete and regenerate intermediate steps

Tuesday, 9 April 2013

On idempotent:
- i.e. they don't mind (give the same results) if you run them multiple times in a row - automatically fill in the blanks as they go, so you can delete intermediate generated data - there are pipeline frameworks that I think are designed handle these kinds of dependencies for you (e.g. based on the old-school 'Make')

Results



101	100118d	allr_vr_dspk_norm_sm4 + mask: inters_groupana_091012a_t2.880	...sm4_z
102	100118e	allr_vr_dspk_norm_sm4 + mask: inters_groupana_091012a_t2.880	...sm4_z
103	100118f	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
104	100118g	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
105	100118h	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
106	100118i	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
107	100118j	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
108	100118k	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z
109	100119a	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.101	...sm4_z
110	100119b	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.101	...sm4_z
111	100121a	allr_vr_dspk_norm_sm4 + mask: groupana_091012a_t2.880	...sm4_z

Structured file names will only get you so far

Spreadsheets are a step up, but hard to manipulate with programs

Use a database!

```
Result.objects \
    .filter(experiment__name='Shiny expt') \
    .filter(classifier__type='ridge',
            classifier__lambda=.2) \
    .filter(mask='PPA') \
    .values('pct_correct', 'running_time')
```

Open sourcing your code

It's good science

Ties you to the mast – standardize data formats, preserve backwards compatibility

Gets you into good habits

Write your code for a reader

Documentation

Package up requirements

Easier to collaborate

Gifts from smart strangers shower down from the sky

Glory!

THE END

INTERACTIVE

Ideas for Movie Lens analysis

Recommendations based on ratings

PCA on genres?

Most popular movies?

Who's the most accurate rater?

Which are the hardest movies to predict?

Which movies are most similar to one another?

What subsets of movies tend to all get rated together?

Creating hostile datasets

try baseline increasing one movie by a big margin

try zeroing out an entire genre

try making all the movies belong to the same genre

try something subtle that won't be obvious visually, e.g. add a little randomness to each of the values (they're supposed to be ints/bools)

REALLY
THE END