Computational Cognitive Science

Lecture 10: Models in psychology

Chris Lucas

School of Informatics

University of Edinburgh

October 17, 2019

Reading

- F&L Chapter 12

Recommended:

- Between the Devil and the Deep Blue Sea: Tensions Between Scientific Judgement and Statistical Model Selection by Navarro

When / why to build a model?

Typically:

- To formalize and test theories

But also:

- To understand existing theories/ideas more deeply
  - Models can be compared to inuitions; not just data
  - Models can provide new predictions that go beyond our intuitions
  - Example: Developmental differences in causal learning

Models and data

Models are applied in several contexts, notably:

1. To (better) explain a familiar pattern
2. To explain a recently-discovered or unexplained pattern
3. To predict new data

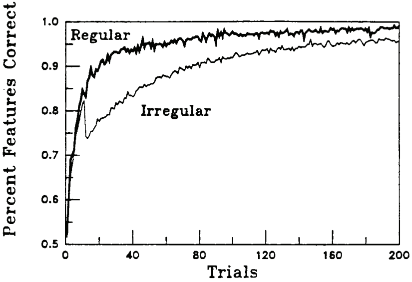
Models and data

1. Model that explains a familiar pattern

Examples:

- proof of concept; common for new kinds of models

Rumelhart & McClelland's model of past tense[1]:



[1] Rumelhart, D. E., & McClelland, J. L. (1985). On learning the past tenses of English verbs.

Models and data

1. Model that explains a familiar pattern

Examples:

- "my model is better than your model"; typically requires decisively better results or new predictions.

Models and data

2. Model that explains a pattern that nothing else does
   - e.g., unifying phenomena that were previously treated separately, e.g., prototype and exemplar models of categorization[2]

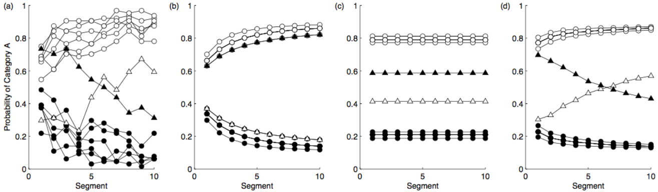


Figure 2 of Griffiths et al., (2007), building on Rosseel. (a) Data; (b) Exemplar; (c) Prototype; (d) Unified model[3]

---

[2] Rosseel, Y. (2002). Mixture models of categorization. Journal of Mathematical Psychology, 46(2), 178-210.

[3] Griffiths, T., Canini, K., Sanborn, A., & Navarro, D. (2007). Unifying rational models of categorization via the hierarchical Dirichlet process.

Models and data

3. Model versus new data – predictions!

- Even models that are proposed to explain old data tend to offer new predictions
- Predictions are sometimes less “pure” than one might think: pilot experiments can inform design

Assessing models: Sufficiency vs necessity

Sufficiency:

If we need to explain/predict data, is the model sufficient?

- are the data consistent with the model's predictions?
- in probabilistic terms: Is $p(\mathbf{y}|\mathcal{M})$ reasonably high?

Assessing models: Sufficiency vs necessity

Necessity:

If we need to explain/predict data, is it necessary to use the model?

- Does the model provide our only good explanation for the data?
- Are the data obvious or trivial to predict?
- i.e., is $\sum_{\mathcal{M}' \neq \mathcal{M}} P(\mathcal{M}') p(\mathbf{y}|\mathcal{M}')$ reasonably low?

Assessing models: Sufficiency vs necessity

Necessity:

- To demonstrate necessity, it helps to compare to alternative models.
- Absent existing models, one can demonstrate "local necessity" by "lesioning" a model:
  - Relax or change key assumptions; look at nested models
- Practical tip: Much seems obvious in hindsight; it can be useful to show how predictions are counter-intuitive

Assessing models: Sufficiency vs necessity

Example: A Bayesian model of preference:

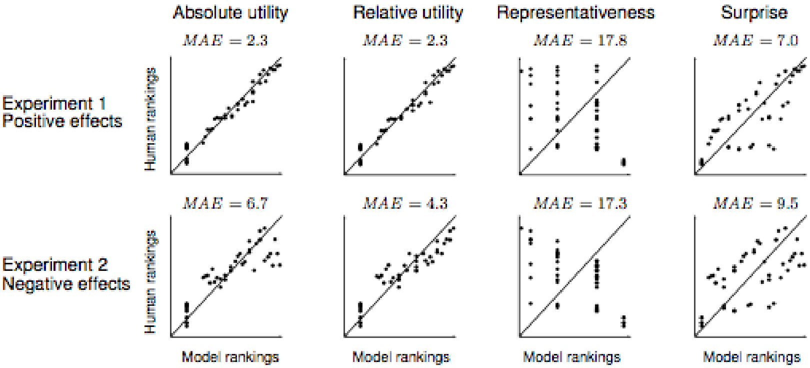


Figure 2 of Kemp et al., (2011). (a-b) Variations on main model; (c) Un-normalized likelihood; (d) Just the normalizing constant[4]

---

[4] Jern, A., Lucas, C. G., & Kemp, C. (2011). Evaluating the inverse decision-making approach to preference learning. In Advances in neural information processing systems (pp. 2276-2284).

Explaining models

Models require us to make lots of decisions. Some of these are essential, others less so. The distinction isn't always obvious.

- What assumptions are central?
- What assumptions are incidental?
- Under what conditions or priors does the model *not* fit the data?

(Perfors; F&L 12.6)

Examples:

- Are independence assumptions fundamental, or there for simplicity?
- Are specific distributions or hyperparameter choices essential or incidental?

Scientific versus statistical thinking

Quantitative model evaluation is only one piece of the puzzle.

- Models express theories completely; auxiliary assumptions are both a help and a hindrance
- Use common sense in addition to mathematical tools
- Don't treat quantitative model selection as the be-all, end-all
  - a model that captures qualitative patterns is still useful
- Take generalization seriously – not just test data in a study, but across studies
  - Example: Rescorla-Wagner (recommended Navarro reading).

Practical recommendations

Practical recommendations

- Organizing data and code
- Testing
- Reproducible analyses

Organizing data and code

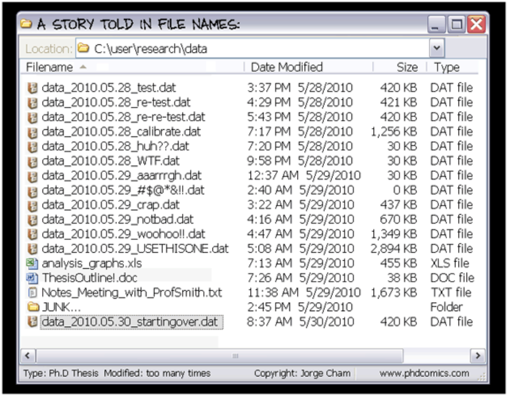
1. Think of prospective collaborators; this includes your future self

- Avoid (or complement) proprietary file formats where possible (.mat, .sav, .xlsx)
- Prefer human *and* machine-readable formats
  - e.g., CSV, json, arguably xml
- Retain your raw data. Try not to keep multiple copies
  - Data-cleaning should be documented and reproducible
- Avoid messy data [link to some advice]
- Document data; a little readme goes a long way

Organizing data and code

1. Think of prospective collaborators; this includes your future self

   - Use informative names for files (prefer absolute references, e.g., dates, over relative ones)



(PhD comics: http://phdcomics.com/comics.php?f=1323)

Organizing data and code

- Use version control (like git)
  - History and backup in one package
  - Peace of mind when refactoring
  - Not always feasible to keep everying in a repository (big data sets, sensitive data)

Organizing data and code

2. Be aware of ethical/legal (e.g., GDPR) concerns. Get trained

Some general rules:

- Don't collect data that allow you to identify people, unless you must (get trained)
- Don't collect sensitive data, unless you must (get trained)

Testing and code hygiene

- Inspect your data visually and with code
  - Small mistakes can have major implications
- Test your code early and often
  - Unit tests are boring, but often save time in the long run
- If something looks peculiar, investigate
- Identifiability simulations can double as tests of code and intuitions
- Harder in some ways than traditional software engineering
  - In some settings, looking like it works means it works; not here
  - "Does my model fit my data" is not sufficient!

Best practices: Reproducible analyses and reusable code

Documented and interleaved analyses are nice, e.g.,

- RMarkdown
- Jupyter notebooks

but don't rely too heavily on notebooks:

- creates incentives to re-write / copy-paste code
  - less testing, more bugs
  - more effort in longer/larger projects
- can impair good use of version control
  - e.g., serialized images in Jupyter notebooks

Summary

- We previously covered general issues in assessing, comparing, and using models
- Today we considered contexts in which models are used, as well as practical and social considerations
- Sufficiency versus necessity
- Recommended practices:
    - model explanation
    - data and code management
    - testing

The remainder of the course will go into specific psychological phenomena and models in more detail, starting with categorization and language.