How to write a good miniproject report, or rather, how to avoid frequently made blunders

Herbert Jaeger, November 1, 2009

Overall structure

A recommendable structuring is the following:

1. Task description (a summary of given task, plus summary of motivation: what specific difficulties will have to be solved and what should be learnt from miniproject)

X. You *do not need to* write a scientific-paper style introduction section with an outline of scientific background, related work or research motivation

2. Summary of what you did: sketch of methods used, results obtained, lesson learnt

3. Methods: description of what you did, in enough detail and formal precision to allow replication!

4. Results: documentation of results (also negative ones, if interesting)

Y. You *do not need to* document your code in the report – that should be done within the code.

5. Discussion: what you learnt, what you would do more/better if you had more time/methods available

6. References list.

Basic hygiene

- All figures need a caption and number, and need to be referenced from within text by their number. All symbols in figure, and axes, must be explained in caption or in text.
- All symbols used in text or equations need an explanation, either in plain English or a formal definition.
- For any non-self-generated graphics, give source
- For any non-self-generated math formula, give source
- Use a uniform and standard and complete format for your references. Check any article from a good scientific journal for a model, and adopt it. Different journals have different citation styles – choose one that you like and stick to it.

Really important

Describe your experiments in enough detail to allow replication. A too superficial treatment is the strongest and most commonly encountered reason for downgrading reports.

Style

- Avoid colloquialism and superlatives. Words like "sort of", "huge", "tremendously" have no place in scientific writing.
- Avoid global judgements of the kind "the most famous...", "the most widely used..." – because you don't actually know this! Statements of this sort (expressed in proper terms) are only admissible for authors that have a long experience and a good standing in a field. In a student's report, they sound presumptuous.

Don't pretend to know what you don't know

- Only cite references that you have actually read. I frequently find cites of papers that are clearly too advanced or too classical (and hard to get) to believe that you read them – and feel duped. Bad for the grade!
- Only write down equations that you understand. I frequently find equations taken out the the hardly understood context of some paper, with not all symbols explained, and not well connected to the rest of the report. Bad for the grade. If you want to use a method that is difficult (like SVNs) but for which ready-made program packages exist which you want to use – you are free to use them, but you should honestly explain the extent to which you understood them. Engineers working "out there" often use methods they don't understand, no-one blames them if the results are good: but if they are not good, it's likely that they didn't use the methods properly. If they (or you) are aware of this, and overtly describe the limitations of your understanding and the details of how you did employ the methods, no reason for blame.

A general machine learning hint

Cross-validation is the key to really most if not all modeling, employ it routinely. Cross-validation is more than splitting the training data into a train and validation set – it has to be done repeatedly, going over all "folds"!