Best Practices for Data Science Projects
CMSC320: Introduction to Data Science

Hector Corrada Bravo
Center for Bioinformatics and Computational Biology

(Slides also by John Dickerson @ UMDCS)

Data Science Lifecycle

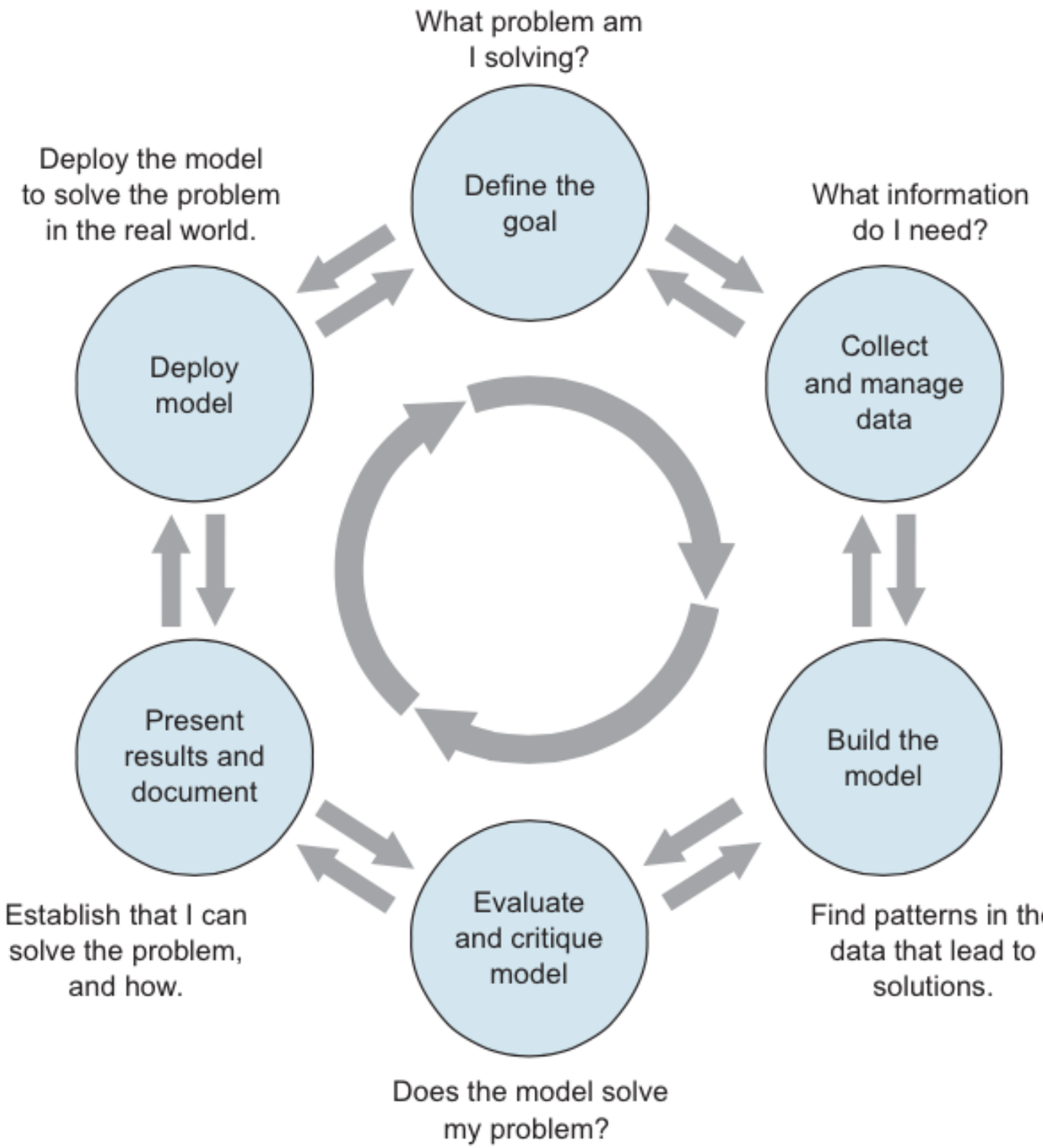

Figure 1.1 The lifecycle of a data science project: loops within loops

Reproducibility

- Extremely important aspect of data analysis
  - ‘Starting from the same raw data, can we reproduce your analysis and obtain the same results?’
- Using libraries helps:
  - Since you don’t reimplement everything, reduce programmer error
  - Large user bases serve as ‘watchdog’ for quality and correctness
- Standard practices help:
  - Version control: git
  - Unit testing: RUnit, testthat
  - Share and publish: github

Reproducibility

Open data:
“Open data is the idea that some data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control”

Open Data movement website

- http://www.opendatafoundation.org/

Practical Tips

- Many tasks can be organized in modular manner:
  - Data acquisition
  - Algorithm/tool development
  - Computational analysis
  - Communication of results

Practical Tips

- Many tasks can be organized in modular manner:
    - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up (wrangling)
    - Algorithm/tool development
    - Computational analysis
    - Communication of results

Practical Tips

- Many tasks can be organized in modular manner:
  - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up
  - Algorithm/tool development: if new analysis tools are required
  - Computational analysis
  - Communication of results

Practical Tips

- Many tasks can be organized in modular manner:
    - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up
    - Algorithm/tool development: if new analysis tools are required
    - Computational analysis: use tools to analyze data
    - Communication of results

Practical Tips

- Many tasks can be organized in modular manner:
  - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up
  - Algorithm/tool development: if new analysis tools are required
  - Computational analysis: use tools to analyze data
  - Communication of results: prepare summaries of experimental results, plots, publication, upload processed data to repositories

Practical Tips

- Many tasks can be organized in modular manner:
  - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up
  - Algorithm/tool development: if new analysis tools are required
  - Computational analysis: use tools to analyze data
  - Communication of results: prepare summaries of experimental results, plots, publication, upload processed data to repositories

Usually a single language or tool does not handle all of these equally well

Practical Tips

- Many tasks can be organized in modular manner:
  - Data acquisition: get data, put it in usable format (many ‘join’ operations), clean it up
  - Algorithm/tool development: if new analysis tools are required
  - Computational analysis: use tools to analyze data
  - Communication of results: prepare summaries of experimental results, plots, publication, upload processed data to repositories

Choose the best tool
for the job!

Practical Tips

- Modularity requires organization and careful thought
- In Data Science we wear two hats
  - Algorithm/tool developer
  - **Experimentalist**: we don’t get trained to think this way enough!
- It helps two consciously separate these two jobs

Think like an experimentalist

- Plan your experiment
- Gather your raw data
- Gather your tools
- Execute experiment
- Analyze
- Communicate

Think like an experimentalist

- Let this guide your organization. I find structuring my projects like this to be useful:

```
project/
| data/
| | processing_scripts
| | raw/
| | proc/
| tools/
| | src/
| | bin/
| exps
| | pipeline_scripts
| | results/
| | analysis_scripts
| | figures/
```

Think like an experimentalist

- Keep a lab notebook!

- Literate programming tools are making this easier for computational projects

  - http://en.wikipedia.org/wiki/Literate_programming

  - http://rmarkdown.rstudio.com/

  - http://jupyter.org/

Think like an experimentalist

- Separate experiment from analysis from communication
    - Store results of computations, write separate scripts to analyze results and make plots/ tables
- **Aim for reproducibility**
    - There are serious consequences for not being careful
        - Publication retraction
        - Worse: http://videolectures.net/cancerbioinformatics2010_baggerly_irrh/
    - Lots of tools available to help, use them! Be proactive: learn about them on your own!

Bias, ethics and responsibility

Examples of Bias

- Genetic testing
  - Genetic tests for heart disorder and race-biased risk (NYTimes)
  - Race-bias in ancestry reports
- Search results / feed optimization
  - Google (How Google could rig the 2016 election)
  - Facebook (Could Facebook swing the election)

Combating bias

Fairness through blindness:

- Don’t let an algorithm look at protected attributes

Examples currently in use ?

- Race
- Gender
- Sexuality
- Disability
- Religion

Problems with this approach ?

Combating Bias

Demographic parity:

- A decision must be independent of the protected attribute

- E.g., a loan application's acceptance rate is independent of an applicant's race (but can be dependent on non-protected features like salary)

Formally: binary decision variable C, protected attribute A

- P{ C = 1 | A = 0 } = P{ C = 1 | A = 1 }

Membership in a protected class should have no correlation with the final decision.

Example from Moritz Hardt's blog

Combating Bias

What if the decision isn't the thing that matters?

"Consider, for example, a luxury hotel chain that renders a promotion to a subset of wealthy whites (who are likely to visit the hotel) and a subset of less affluent blacks (who are unlikely to visit the hotel). The situation is obviously quite icky, but demographic parity is completely fine with it so long as the same fraction of people in each group see the promotion."

Demographic parity allows classifiers that select qualified candidates in the "majority" demographic and unqualified candidate in the "minority" demographic, within a protected attribute, so long as the expected percentages work out.

Example from Moritz Hardt's blog

FATML

This stuff is really tricky (and really important).

- It's also not solved, even remotely, yet!

New community: Fairness, Accountability, and Transparency in Machine Learning (aka FATML)

"… policymakers, regulators, and advocates have expressed fears about the potentially discriminatory impact of machine learning, with many calling for further technical research into the dangers of inadvertently encoding bias into automated decisions."

F is for Fairness

In large data sets, there is always proportionally less data available about minorities.

Statistical patterns that hold for the majority may be invalid for a given minority group.

Fairness can be viewed as a measure of diversity in the combinatorial space of sensitive attributes, as opposed to the geometric space of features.

Thanks to: Faez Ahmed

A is for accountability

Accountability of a mechanism implies an obligation to report, explain, or justify algorithmic decision-making as well as mitigate any negative social impacts or potential harms.

- Current accountability tools were developed to oversee human decision makers
- They often fail when applied to algorithms and mechanisms instead

Example, no established methods exist to judge the intent of a piece of software. Because automated decision systems can return potentially incorrect, unjustified or unfair results, additional approaches are needed to make such systems accountable and governable.

Thanks to: Faez Ahmed

T is for transparency

Automated ML-based algorithms make many important decisions in life.

- Decision-making process is opaque, hard to audit

A transparent mechanism should be:

- understandable;
- more meaningful;
- more accessible; and
- more measurable.

Thanks to: Faez Ahmed

FAT in practice

- Specific questions to ask in three areas:
  - Data collection
  - Modeling
  - Deployment

[Fung, 2016]

Data collection

- What data should (not) be collected
- Who owns the data
- Whose data can (not) be shared
- What technology for collecting, storing, managing data
- Whose data can (not) be traded
- What data can (not) be merged
- What to do with prejudicial data

[Fung, 2016]

Data Modeling

- Data is biased (known/unknown)
  - Invalid assumptions
  - Confirmation bias
- Publication bias
  - WSDM 2017: https://arxiv.org/abs/1702.00502
- Badly handling missing values

[Fung, 2016]

Deployment

- Spurious correlation / over-generalization
- Using “black-box” methods that cannot be explained
- Using heuristics that are not well understood
- Releasing untested code
- Extrapolating
- Not measuring lifecycle performance (concept drift in ML)

We will go over ways to counter this in the ML/stats/hypothesis testing portion of the course

[Fung, 2016]

Guiding principles

- Start with clear user need and public benefit
- Use data and tools which have minimum intrusion necessary
- Create robust data science models
- Be alert to public perceptions
- Be as open and accountable as possible
- Keep data secure

GOV.UK

Thanks to: UK cabinet office

Some references

- Presentation on ethics and data analysis, Kaiser Fung @ Columbia Univ. http://andrewgelman.com/wp-content/uploads/2016/04/fung_ethics_v3.pdf

- O'Neil, Weapons of math destruction. https://www.amazon.com/Weapons-Math-Destruction-Increases-Inequality/dp/0553418815

- UK Cabinet Office, Data Science Ethical Framework. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/524298/Data_science_ethics_framework_v1.0_for_publication__1_.pdf

- Derman, Modelers' Hippocratic Oath. http://www.iijournals.com/doi/pdfplus/10.3905/jod.2012.20.1.035

- Nick D's MIT Tech Review Article. https://www.technologyreview.com/s/602933/how-to-hold-algorithms-accountable/