Debugging Machine Learning Algorithms

William Cohen

Overview of this week

- Debugging tips for ML algorithms
- Graph algorithms
  - A prototypical graph algorithm: PageRank
    - In memory
    - Putting more and more on disk …
  - Sampling from a graph
    - What is a good sample (graph statistics)
    - What methods work (PPR/RWR)
    - HW: PageRank-Nibble method + Gephi
    - Announcement: Gephi recitation on Friday

Debugging for non-ML systems

- “If it compiles, ship it.”

Debugging for ML systems

1. It's definitely *exactly* the algorithm you read about in that paper
2. It also compiles
3. It gets 87% accuracy on the author's dataset
   - but he got 91%
   - so it's not working?
   - or, your eval is wrong?
   - or, *his* eval is wrong?

Debugging for ML systems

1. It's definitely *exactly* the algorithm you read about in that paper
2. It also compiles
3. It gets 97% accuracy on the author's dataset
   - but he got 91%
   - so you have a best paper award!
   - or, maybe a bug…

Debugging for ML systems

- It's always hard to debug software
- It's *especially* hard for ML
  - a wide range of almost-correct modes for a program to be in


EMPIRICAL RESEARCH
It's easy to make assumptions about puppies strapped to missiles, but good science requires testing.
DANGER
EXPLOSIVES

Debugging advice

1. Write tests
2. For subtle problems, write tests
3. If you're still not sure why it's not working, write tests
4. If you get really stuck:
   - take a walk and come back to it in a hour
   - ask a friend
     - If s/he's also in 10-605 s/he can still help as long as no notes are taken (my rules)
   - take a break and write some tests

Debugging ML systems

1. Write tests
   - For a generative learner, write a generator and *generate* training/test data from the *assumed* distribution
     - Eg, for NB: use one small multinomial for pos examples, another one for neg examples, and a weighted coin for the class priors.
   - The learner should (usually) recover the actual parameters of the generator
     - given enough data, modulo convexity, …
   - Test it on the weird cases (eg, uniform class priors, highly skewed multinomials)

Debugging ML systems

1. Write tests
   - For a discriminative learner, similar trick…
   - Also, use what you know: eg, for SGD
     - does taking one gradient step (on a sample task) lower the loss on the training data?
     - does it lower the loss *as expected?*
       - (f(x)-f(x+d))/d should approximate f'(x)
     - does regularization work *as expected*?
       - large mu ➔ smaller param values
     - record training set/test set loss
       - with and without regularization