Automating Feedback
CS398
Chris is out of town :(
Hi I’m Mike!

- Grew up in San Diego
- Research on probabilistic inference
- I have two cats! (Monster and Jax)
Things to get out of this lecture!

- Why predicting feedback is hard (unsolved?)
- Understand Zipf distributions
- Generative models for Zipfs (HW 3!!!)
Peak at HW3

- Students learning nested loops
- 50k students with **1.5 million submissions** to a curriculum of **8 exercises**.
- **800 human labels** across 2 of the exercises.

**NO MORE SIMULATIONS**

**100% REAL DATA**
Curriculum: a series of 8 challenges.

Dataset of 50,190 students. How many unique submissions?
**Curriculum**: a series of 8 challenges.

Dataset of **50,190** students. How many unique submissions? **48,606**
Curriculum: a series of 8 challenges.

Dataset of 50,190 submissions. How many unique solutions? 48,606
Curriculum: a series of 8 challenges.

Q: what are the different ways you could see a student tackling this problem? Think of common mistakes or strategies!

- For (n) to (m) by (k) loop block
- Move(forward/backward)(x) block
- Turn(left/right)(x) block
- Repeat(n) block
Curriculum: a series of 8 challenges.

HW 3: Formalize these intuitions in a “probabilistic grammar”.
Feedback in Online Education
Cost Disease of Education

MOOCs are $\frac{1}{2}$ of Story

**Code Foundations**
Interested in learning how to code, but unsure where to start? This is the path for you!
- Computer Science History
- Career Exploration
- Applications
- In-progress

**Computer Science**
Looking for an introduction to the theory behind programming? Master Python while learning data structures, algorithms, and more!
- Python
- Data Structures
- Command Line
- Git

**Data Science**
Learn SQL and Python and build the skills you need to query, analyze, and visualize data.
- Python
- SQL
- Data Visualization
- Machine Learning
Feedback is Labor Intensive

Online classes have not solved the feedback problem [1].

Online Feedback isn’t Great

Try me!
Maybe this is a resource problem?
Datasets: Visual Genome

Who is wearing glasses?
- man
- woman

Where is the child sitting?
- fridge
- arms

Is the umbrella upside down?
- yes
- no

How many children are in the bed?
- 2
- 1

Remember Code.org?

- One of the largest online coding platforms
- They thought about feedback too...
In 2014, Code.org asked 100,000s of instructors to give feedback to student solutions. After 1000s of hours, barely covered the tip of the iceberg. The initiative was cancelled and the effort has not been reproduced since.
Crowdsourcing Teacher Feedback

- In 2014, Code.org asked 100,000s of instructors to give feedback to student solutions.
- After 1000s of hours, barely covered the tip of the iceberg.
- The initiative was cancelled and the effort has not been reproduced since.

Is this a viable strategy?

- What about classrooms of 30?
- What about changing curriculums? Or different topics?
- No “warm up” cost… Help the first student…
Chris and 7 TAs labelled 800 unique solutions to question 8...
Chris and 7 TAs labelled 800 unique solutions to question 8...
Personal Experience

Chris and 7 TAs labelled 800 unique solutions to question 8...
Deep Learning to the rescue?
We’ve seen **DIRT**. This is a similar setup:

- Given a student solution (code), train a neural network to infer feedback (assume pool of feedback).
- This is a classification task!?
- Use modern RNNs / Transformers to handle natural text (code).

So what we have < 1000 labelled examples? Deep learning will definitely work?

Recall

\[
\text{Loss}(\theta) = \sum_{i=0}^{N} y^{(i)} \log(\sigma(\theta^T x^{(i)})) + (1 - y^{(i)}) \log(1 - \sigma(\theta^T x^{(i)}))
\]
Goals for this Exercise:

- Get used to real world (text) student data
- Get familiar with how to use RNNs
- Gauge how well deep learning works for predicting feedback

Things to try:

- Explore the data!
- Plot frequency vs rank of programs
- Plot log-frequency by log-rank
- Try training the RNN! What is the test performance?
- Try playing with the pretrained RNN.

Take 15-20 Minutes
Generalization

- The most complex model is not always good
- Risk of memorizing
- Poor performance on unseen data.

**True function** (usually unknown)  
**Observed data** (sparse + noisy)
Jupyter Notebook
Neural Networks don’t just work

Takeaway from this section:

● Deep neural networks are powerful but also
  ○ Uninterpretable
  ○ Sometimes Inaccurate
  ○ Data hungry

● Not sufficient for the education domain
  ○ Even with 1000 examples (1000 students), not near human level.
Intermission
Feedback is a really hard problem.
A Funny Pattern

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (120))))))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (60))))))))

Example: (Program (WhenRun)(Move (Forward)(Value (Number (50)))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Right)(Value (Number (120))))))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (90))))))))

Example: (Program (WhenRun)(Move (Forward)(Value (Number (100)))))

Example: (Program (WhenRun))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (45)))))))

Example: (Program (WhenRun)(Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (120)))(Move (Forward)(Value (Number (50))))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Right)(Value (Number (60))))))))

Example: (Program (WhenRun)(Move (Forward)(Value (Number (50)))(Turn (Left)(Value (Number (90)))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Right)(Value (Number (90))))))))

Example: (Program (WhenRun)(Repeat (Value (Number (3)))(Body (Move (Forward)(Value (Number (50)))(Turn (Right)(Value (Number (50))))))
This weird behavior of linearity in log-log space is not uncommon...

(a) Code.org  (b) Liftoff  (c) Pyramid  (d) Power
Zipf Distributions

- Informal definition: observations are linear on a log-frequency by log-rank.
- A few values in the head dominate the frequency
- Most of the mass lives in the tail (infinitely dense)
- 2nd most common appears ½ as much, 3rd appears ⅓ as much, etc.
Zipf Distributions

- These distributions make generalization really hard. (why?)
- Neural networks assume test distribution is the same as the train distribution.

- **Not true for Zipf:** Unseen student solutions can (or will) be completely different.
If we assume student solutions are Zipf, then maybe we do not need lots of data:

Model the *generative process* for a Zipf distribution (student work). Then we could generate **infinite** data and have to worry less about generalization.

- Similar to how we built a simulator for IRT.
  - We could simulated data to train NN
- Requires **no** data (“zero-shot”)

But what is the generative process for a Zipf?
Three Hypotheses
Take conversation between two people:

- Speakers want to use a few simple words to convey meaning
- Listeners want to hear more specific (complex) words
- Balancing these two produces a Zipf Principle of Least Resistance

How was your day?
Principle of Least Resistance

Take conversation between two people:

- Speakers want to use a few simple words to convey meaning
- Listeners want to hear more specific (complex) words
- Balancing these two produces a Zipf
Take conversation between two people:

- Speakers want to use a few simple words to convey meaning
- Listeners want to hear more specific (complex) words
- Balancing these two produces a Zipf

Vsauce, “The Zipf Mystery”
Idea: “Rich get richer”. Words that people use more will get used more often.

Step 1
Idea: “Rich get richer”. Words that people use more will get used more often.

Step 1

Step 2
Idea: “Rich get richer”. Words that people use more will get used more often.

Step 1

Step 2

Step 3
Monkey Typewriter

- Randomly type characters. “Space” terminates a word.

- Exponentially more different long words than short words

- Independent probability of pressing “space” = longer words are less likely.

These are fun to play with.
Are Students Monkeys?

Is this the right model?
Now we are venturing into research territory. **But something doesn’t feel right...**

- Are student writing programs by randomly choosing tokens / characters?
- *(I hope not?)* 😢 😢 😢
A Different Perspective: Grammars

Grammar $G = (M, T, R, S, P)$

- $M$ is the set of non-terminal symbols
- $T$ is the set of terminal symbols
- $R$ is the set of production rules
- $S$ is the start symbol
- $P$ is the set of probabilities on production rules
A Different Perspective: Grammars

Grammar $G = (M,T,R,S,P)$

- $M$ is the set of non-terminal symbols
- $T$ is the set of terminal symbols
- $R$ is the set of production rules
- $S$ is the start symbol
- $P$ is the set of probabilities on production rules

Sample from $G$: start at root and roll a weighted $K$-sided die to pick next child. Continue until terminate.

Example Sample

```
Example: { Program { WhenRun } { Repeat { Value { Number { 3 } } } { Body { Move { Forward } { Value { Number { 50 } } } } { Turn { Left } { Value { Number { 120 } } } } } } } }
```
A Different Perspective: Grammars

Grammar $G = (M,T,R,S,P)$

- $M$ is the set of non-terminal symbols
- $T$ is the set of terminal symbols
- $R$ is the set of production rules
- $S$ is the start symbol
- $P$ is the set of probabilities on production rules

The distribution from grammars is exponential (lighter tails than Zipf). So it’s not quite right but we get to model the student thinking process in a more interpretable way...
Chris will be back next class! He will go into more detail on these grammars.
That's all Folks!