Lessons from coding theory
This talk is about

Error correcting codes
You may be thinking

- What are error correcting codes?
  - Tools for protecting data from errors
- Classical solutions to problems in communication and storage...

*Isn’t that a solved problem?*
Actually...

- Lots of old problems **still open**
  - Fun to think about!

- Lots of **new applications** in communication and storage
  - Distributed storage is a big one

- But even if you are not into communication and storage
  - **Lots of applications everywhere!**
Common set-up

- We have some data
- It has some structure
  - which we don’t control
- We want to take advantage of that structure
  - learn stuff about the data
  - do it efficiently
Taking advantage of structure

Compressed sensing/
sparse recovery
Taking advantage of structure

Yikes...
Taking advantage of structure

High-throughput screening
Taking advantage of structure

De novo sequencing
etc...

- We don’t control the structure
- We do (partially) control what happens to the data
The other side of the coin

- We do control the structure
- We don’t control what happens to the data
Taking advantage of the most advantageous structure
Taking advantage of the most advantageous structure
Error correcting codes

are, by design, the most advantageous structure.

- Used in communication and storage since the 1950’s
- Over half a century of beautiful work!
Two questions

- Haven’t we figured that out by now?
- What about this other stuff?
  - If we control measurements but not structure?
As mentioned before

- Okay, so we haven’t really figured everything out.
  - Coding and information theory for communication and storage is still a vibrant research area.

- But also!
  - Lessons from classical coding theory are relevant in a wide variety of these sorts of applications.
My work

• Some of this:

• Some of that:

• Right now I’m really excited about the interplay between the two.
We don’t have too much time so I’ll just give one example

• Here’s one very classical idea...

This is “Reed-Solomon encoding”
Low-degree polynomials are the most advantageous structure*

- Really nice combinatorial properties
- Admit really fast algorithms
- Used all over the place in communication and storage

*personal opinion
Not just in communication and storage

- High-throughput screening

Pool genetic samples:
Imagine there are way more people than there are pools...

*taking some artistic liberties*
Pool samples according to low-degree polynomials

- Not a new idea [Kautz & Singleton 1964]
- But making it work for modern problems comes with new challenges!

[Thierry-Mieg 2006], ..., [Erlich, Gilbert, Ngo, Rudra, Thierry-Mieg, W., Zielinski, Zuk 2015],...

*taking some artistic liberties
Also, de novo genome sequencing

Sequence this pool altogether, get a bunch of reads...what came from what?

[Lonardi, Duma, Alpert 2013]
[Duma, W., Gilbert, Ngo, Rudra, Alpert, Close, Ciardo, Lonardi 2013]

*Taking even more artistic liberties...
Even if you don’t control the structure...

• If you do control how you process the data, tricks like this work.
• Many examples in compressed sensing, group testing, streaming algorithms,...
  • Cheraghchi, Gilbert, Indyk, Nelson, Ngo, Porat, Rudra, Strauss,...
Summary

• I think error correcting codes are cool.
  • And still an exciting area of research!
  • Talk to me later about distributed storage!

• But even if you don’t care about communication, storage, or information theory,

  error correcting codes are still cool.
Thanks!

Contact me to talk more!  
Mary Wootters  
marykw@stanford.edu  
Gates 468