Coding Theory

CSE 445/545

January 28, 2019
Let’s do some introductions

Atri Rudra

319 Davis Hall
atri@buffalo.edu
645-2464
Office hours: Tue, 2-2:45pm
Handouts for today

Syllabus
   Linked from the course webpage

Feedback polls
   Up on piazza
Plug for feedback polls

Completing the form is voluntary & anonymous

Purpose of the form
  For me to get an idea of your technical background
One Stop Shop for the course

CSE 445/545: Coding Theory
Spring 2019

https://cse.buffalo.edu/faculty/atri/courses/coding-theory/webpage/spr19/
CSE 445/545 (Coding Theory)
Syllabus
Spring 2019
Tuesdays and Thursdays, 12:30-1:50pm, Norton 218.

Under Construction
This page is still under construction. In particular, nothing here is final while this sign still remains here.

Please note
It is your responsibility to make sure you read and understand the contents of this syllabus. If you have any questions, please contact the instructor.
CSE 445/545 Spring 19 Schedule

Previous schedule: 2013.

Under Construction
This page is still under construction. In particular, nothing here is final while this sign still remains here.

Future Lectures
The topics for lectures in the future are tentative and subject to change.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue, Jan 29</td>
<td>Introduction</td>
</tr>
<tr>
<td>Th, Jan 31</td>
<td>Definitions</td>
</tr>
<tr>
<td>Tue, Feb 5</td>
<td>Distance of a code</td>
</tr>
</tbody>
</table>
Autolab

Details on Autolab, which will be used for all homework submissions in CSE 4/545.

The main link
We will be using the UB CSE extension to Autolab for submission and grading of CSE 4/545 homeworks. You can access Autolab via https://autograder.cse.buffalo.edu.

Signing up
Follow these steps to setup an account on Autolab (unless you already have one in which case you’ll use your existing account):

1. Go to this page and click on the Sign in with MyUB link. A new account will automatically be created for you.
2. By default, Autolab will use your official UB first and last name. If you have a different preferred name, please let us know ASAP.
3. When you login, the system will ask you to put in your nickname. It seems like to use the system you have to put in a nickname (though it won’t be used for anything in this course).
4. After you have done the above steps, you wait.

What happens next
CSE 445/545: Coding Theory
Spring 2019

Under Construction
This page is still under construction. In particular, nothing here is final while this sign still remains here.
Piazza for discussion

Please use your UB email ID to sign up

Welcome to Piazza for CSE 545!

Students,

Welcome to Piazza! We’ll be conducting all class-related discussion here this term. The quicker you begin asking questions on Piazza (rather than via emails), the quicker you’ll benefit from the collective knowledge of your classmates and instructors. I encourage you to ask questions when you’re struggling to understand a concept—you can even do so anonymously. (You will be anonymous to the students but not to me.)

-Atri Rudra
Feedback polls already up

Background feedback
For me to get a better sense of your background, please fill in these piazza polls:

- Linear Algebra: @8
- Abstract Algebra: @9
- Probability: @10
- Algorithms: @11
- Complexity: @12
- Why are you taking this course?: @13

(I will pin this post so that it is visible.)
#pin
Questions/Comments?

If something doesn’t work (e.g. you cannot post a comment), let me know
References

Draft of a book I’m writing
With Guruswami+Sudan

Standard coding theory texts
MacWilliams and Sloane
van Lint
Blahut

Handbook of coding theory
Pre-requisites

No formal pre-requisites for 545/ CSE 331 for 445
   Probably no one will have all the pre-req’s

Mathematical maturity
   Comfortable with proofs
   Willing to pick up basics of new areas

Will spend one lecture on the pre-req’s
   Linear Algebra
   Finite Fields
   Probability
   Algorithms/ Asymptotic Analysis

Go slower in the first half of the course
Grades and such like

Grading Policy

Here is the split of grades:

<table>
<thead>
<tr>
<th>Course Component</th>
<th>% of grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini project</td>
<td>40%</td>
</tr>
<tr>
<td>Homeworks</td>
<td>30%</td>
</tr>
<tr>
<td>Proof Reading</td>
<td>30%</td>
</tr>
</tbody>
</table>
Mini Project

Groups of size $\leq 3$

Create a Youtube video related to coding theory

Bunch of other details in syllabus
Deadlines

**March 5, 2019.** You should email me the topic and the composition of your group by **11:59pm**.

**April 2, 2019.** You should submit your two-page report by **11:59pm** on **Autolab**.

**April 30, 2019.** You should submit your video by **11:59pm** on **Autolab**.
Proof-reading

Proof-read relevant part of the book
3-4 during the course
   Depends on the class strength
Submit typos, suggestions for improvement
They are due in by noon before next lecture
Notes will be graded on timeliness & quality
Will ask for a volunteer
See syllabus for more details
Questions/Comments?

Check out the syllabus for more details
Homework

3 short ones (545)/ 2 short ones (445)

Collaboration generally allowed
- Work in groups of size at most 3
- Write up your own solutions
- Acknowledge your collaborators
- No source other than book and your notes
- Breaking these rules will be considered as cheating

More details when they are handed out
My homework philosophy for 545

**NOT** to make sure you understand what I teach in the lectures

Homework problems either

- Proofs that were not done in the class; or
- Material that is not covered in the class
  - Closely related to something that is
Questions/Comments?

Check out the syllabus for more details
Some comments

Decide on a Video topic **early**
  - Different topics might need different prep. work
  - Come talk to me

Homeworks might take time
  - Do not wait for the last moment
Academic Dishonesty

All your submissions must be your own work

Penalty:
  Minimum: An grade reduction in course
  Possible: F (or higher penalty) if warranted

YOUR responsibility to know what is cheating, plagiarism etc.
  If not sure, come talk to me

Excuses like “I have a job,” “This was OK earlier/in my country,” “This course is hard,” etc. WON’T WORK

I DO NOT HAVE ANY PATIENCE WITH ANY CHEATING:
YOU WILL GET A GRADE REDUCTION IN THE COURSE FOR YOUR FIRST MISTAKE
If grades are all you care about

You’ll be fine if

   You do your assignments **honestly**
   Make a reasonable attempt at them
Questions/Comments?

Check out the syllabus for more details
Let the fun begin!
Coding theory
Welcome to the class. I hope you will have as much fun as I will have teaching it!
Why did the example work?

English has in built redundancy
Can tolerate “errors”
The setup

- Mapping $C$
  - Error-correcting code or just code
  - Encoding: $x \rightarrow C(x)$
  - Decoding: $y \rightarrow x$
  - $C(x)$ is a codeword

$y = C(x) + \text{error}$
Communication

Internet
  Checksum used in multiple layers of TCP/IP stack

Cell phones

Satellite broadcast
  TV

Deep space telecommunications
  Mars Rover
Samsung and the University of California San Diego recently signed a major license agreement for the telecommunications industry, for a standard-essential error-correction technology developed by engineers from the Jacobs School of Engineering. This new technology plays a key role in making sure 5G networks are reliable and efficient.
“Unusual” applications

Data Storage
- CDs and DVDs
- RAID
- ECC memory

Paper bar codes
- UPS (MaxiCode)

Codes are all around us
Other applications of codes

Outside communication/storage domain

Tons of applications in theory
- Complexity Theory
- Cryptography
- Algorithms

Coding theory is a good tool to have in your arsenal
The birth of coding theory

Claude E. Shannon
“A Mathematical Theory of Communication”
1948
Gave birth to Information theory

Richard W. Hamming
“Error Detecting and Error Correcting Codes”
1950

EE 634
(this semester!)
Structure of the course

Part I: Combinatorics
   What can and cannot be done with codes

Part II: Algorithms
   How to use codes efficiently

Part III: Applications
   Applications in (theoretical) Computer Science
Redundancy vs. Error-correction

**Repetition code**: Repeat every bit say 100 times
  - Good error correcting properties
  - Too much redundancy

**Parity code**: Add a parity bit
  - Minimum amount of redundancy
  - Bad error correcting properties
  - Two errors go completely undetected

Neither of these codes are satisfactory

```
1 1 1 0 0
```

```
1 0 0 0 0
```
Two main challenges in coding theory

Problem with parity example
  Messages mapped to codewords which do not differ in many places
Need to pick a lot of codewords that differ a lot from each other

Efficient decoding
  Naive algorithm: check received word with all codewords
The fundamental tradeoff

Correct as many errors as possible with as little redundancy as possible

Can one achieve the “optimal” tradeoff with efficient encoding and decoding?