CSE 510C, Error Correcting Codes: Combinatorics, Algorithms and Applications

Fall 2007

Mon Wed Fri 10:00-10:50am, Talbert 106

- Please complete the (anonymous) feedback form. Filling up the form is optional but I encourage you to fill at least the part about your preferred office hours.

Instructor information

Atri Rudra (`http://www.cse.buffalo.edu/∼atri`)

- *Email:* `atri "at" cse`
- *Office:* CSE 123
- *Phone:* 645-3180 x 117
- *Office Hours: TBA*

It is preferable to set up an appointment if you want to talk to me outside of my office hours. However, you can drop by if my office door is open.

Course Description

Error correcting codes are systematic ways of introducing redundancy into data so that the original information can be recovered even when the data is corrupted. Error correcting codes (or just codes) are used ubiquitously in communication systems and data storage. The study of error correcting codes (or coding theory) started with the seminal works of Shannon and Hamming in the late 1940s.

This course will discuss the theoretical aspects of codes and will focus mostly on the worst case noise model pioneered by Hamming. However, we will discuss quite a few results on the stochastic noise model pioneered by Shannon. We will use techniques from combinatorics, probability theory and algebra among other areas.

The course will be roughly divided into three parts. The first part will look at the combinatorial issues in the design of codes. This part will mostly be classical results that talk about limits to what can and cannot be done using codes. The second part of the course will deal with the algorithmic aspects of codes. In particular, we will focus on efficient algorithms that recover the original information from corrupted data (called decoding). In this part we will discuss some exciting recent developments that bridge the "divergent" schools of thoughts of Shannon and Hamming. Finally, we will study some application of codes outside of the "traditional" error correcting applications. In particular, we will see how codes can be used to obtain results in theoretical computer science in general and computational complexity in particular.

Pre-requisites

There is no specific course pre-requisite for this course. However, some "mathematical maturity" will be essential. In particular, comfort with basics of linear algebra (vector spaces, basis, dual spaces); finite fields, field extensions and polynomials over finite fields; elementary probability; analysis of algorithms; and (some exposure to) computational complexity will be useful. Some of these topics (for example finite fields) can be learned on a need to know basis as the course progresses. Email the instructor if you have any questions on the pre-requisites.

References

We will not follow any particular textbook. The closest resource is the excellent set of lecture notes for Madhu Sudan's coding theory course at MIT (see the course webpage for links to the lecture notes).

The basic material on codes that we will discuss in initial lectures can be found in one of many textbooks (some of the standard ones are listed below), but the recent algorithmic developments and applications in computer science are not covered in any of these:

- *Introduction to Coding Theory*, by J. H. van Lint, GTM 86.
- *The Theory of Error Correcting Codes*, by F. J. MacWilliams and N. J. A. Sloane, North-Holland, Amsterdam.
- *Algebraic codes for data transmission*, by Richard E. Blahut.

Class Webpage

`http://www.cse.buffalo.edu/~atri/courses/coding-theory/`

Course Blog

We will be using a blog (`http://codingtheory.wordpress.com`) for the course in lieu of a course newsgroup. All announcements will be made on the blog. If you are attending the course, you **must** check the blog regularly (and consider subscribing to the RSS feed via the link on the blog). These announcements would include the ones that inform if and when classes/office hours are re-scheduled as well as notifications for the availability of online lecture notes.

Usually, I will be the only one who will write the blog entries. There will be an entry for each lecture and homework. You are encouraged to use the comments section to post questions and/or comments. Sometimes, the blog may include side comments or stories that I feel are relevant to the course (but are not directly related to the lectures). Also I plan to write an entry for each project topic that I will propose, so be on the lookout for those.

Grading Policy

Here is a rough split of grades:

- Scribing Notes (30-40%)
- Homeworks (30-15%)
- Survey Report (40-45%)

See the next few sections for more details on each of the above components.

Scribing Notes

Every lecture will be scribed by a student. Depending on the size of the class a student might have to scribe 3-4 lectures. I will typically ask for a volunteer at the beginning of the class.

The notes will have to be scribed using LaTeX. If you do not know how to use it, one can learn how to use it fairly easily (the lecture notes will not require any fancy LaTeX usage). The LaTeX style file is available on the course website.

The scribed notes are due the day before the class the following week. For example, the notes for Wednesday August 29 lecture are due on Tuesday September 4th. The scribed notes will be graded on the timeliness of completion as well as the quality of the writeup.

Homeworks

There will be one or two homeworks: the exact number will depend on how heavy the scribing load turns out to be. Collaboration in groups of size at most three is allowed (and encouraged). However, every student is expected to do their own writeup and clearly state the names of their collaborators. More details will be available when the homeworks are handed out.

Survey Report

Every student is expected to review literature on some aspect of coding theory and submit a report by the end of the semester. The topic can be something that could not be covered in the class in detail or could be a topic that was not covered at all in the class. Working on an open research problem is welcome but not mandatory. A suggested list of topics is available on the website. Short descriptions of the projects along with some helpful starting points will be posted on the course blog. Students are welcome (and encouraged) to suggest their own topics. However, such topics have to be approved by the instructor.

The rough timeline for the project is as follows:

- *September 21, 2007.* The last date by which the instructor will post short description of all the suggested survey topics.
- *October 19, 2007.* Deadline for picking a topic for the report **and** a one page "proposal" that gives a brief overview of the survey along with a list of the major research papers that will be surveyed.
- *December 10, 2007.* Deadline for submitting the final report. Note that this is a **hard** deadline as I will be traveling after that date.

Each survey should be a mix of both classical and current results in the topic. The grade will depend on the clarity/quality of presentation as well as the depth of material covered.

Academic Honesty

I will follow the CSE department academic integrity policy. Details can be found at
`http://www.cse.buffalo.edu/academics-academic_integrity.shtml`

Suggestions or Comments?

I would be happy to get feedback from you. You can either

- Talk/send email to the instructor, or
- Use the comments section of the entries in the course blog, or
- Fill in the feedback forms that will be handed periodically in class.