# CSE 545, Error Correcting Codes: Combinatorics, Algorithms and Applications

## SPRING 2009

Mon Wed Fri 2:00-2:50pm, Baldy 110

- 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. Instead we will follow the lecture notes from the Fall 2007 offering of the course (see the course webpage for a link to the lecture notes).

The basic material on codes that we will discuss in initial lectures can be found in 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 reminders for deadlines etc.

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).

## Grading Policy

Here is a rough split of grades:

- Scribing Notes/Updating Wikipedia (30-40%)
- Homeworks (40-25%)
- Paper presentations (30-35%)

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

## Scribing Notes

Some of the lectures this semester will cover topics that were not covered in Fall 2007. Each such lecture will be scribed by a student. Each student will most probably have to scribe (at most) one lecture. 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 January 14 lecture are due on Tuesday January 20th. The scribed notes will be graded on the timeliness of completion as well as the quality of the writeup.

## Updating Wikipedia

Every student will pick a coding theory topic that is either absent or not well documented on Wikipedia and write/edit the corresponding entry. (An example of such a page is the one on "list decoding.") You will have the opportunity to work on initial versions of the entry on an in-house wiki page (details on this coming shortly) before the entries are posted on Wikipedia (you'll need to get my OK before you post your entry on Wikipedia).

To ensure a timely completion of this part of the course, you will have to follow the following deadlines:

- *March 23, 2009.* You should inform me about the entry in Wikipedia that you'll work on by this date.
- *March 30, 2009.* You should submit a one page "report" outlining what will appear in your entry. (*Note:* You are allowed to deviate a bit from what you say in this document in your final entry. This document is supposed to me more of a guideline so that I can make sure your entry has the right feel to it.)
- *April 20, 2009.* The deadline to submit your entry on the in-house Wiki.

You will be graded on the quality and depth of your entry.

## 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.

My homework philosophy for this course is that they are meant for you to work on stuff that I did not have time to cover in the lectures. (Don't worry: you will be provided with hints for the tougher problems!) Note that this is **not** the same philosophy for a usual course, where homeworks are used to re-enforce material covered in the lectures.

More details will be available when the homeworks are handed out.

## Paper Presentations

Every student will present a paper on coding theory at the end of the semester. The paper could be on something that could not be covered in the class in detail or could be on a topic that was not covered at all in the class. A suggested list of paper will be made available on the course blog. Students are welcome (and encouraged) to suggest their own papers. However, such papers have to be approved by the instructor.

The paper presentations will be on *April 28* and *April 29, 2009* (the reading days). The time allotted to each student will depend on the total number of registered students.

The *deadline* to choose your paper is *April 13, 2009*, if you are choosing your own paper. If you pick a paper from the suggested list, your deadline is *April 24, 2009.* Papers on the list will be assigned on a first-come-first-serve basis.

You will be graded on the clarity and quality of your presentation.

## Academic Honesty

I will follow the CSE department academic integrity policy. Details can be found at `http://www.cse.buffalo.edu/graduate/policies_acad_integrity.php`

## 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.