Communication in the Presence of Noise and Algorithms for Error-Correction MIT2001-04

Progress Report: July 1, 2002—December 31, 2002

Madhu Sudan

Project Overview

The project investigates fundamental problems arising from the theory of communication of information, from the perspective of algorithms, and computational complexity. The project centers on the theme of "list-decoding'', a notion that allows decoding algorithms to output a list of codewords that are close to a word received from a noisy channel. List-decoding allows for more errors to be recovered in a worst-case scenario, and offers a promising route to bridge the "probabilistic analysis", central to Shannon's theory of information, with more adversarial scenarios. The project investigates combinatorial, algorithmic, and complexity-theoretic aspects of list-decoding, including the task of finding very efficient algorithms for correcting errors in very noisy channels. The broader scope of this project includes research in algorithmic problems arising in all aspects of data communication.

Progress Through December 2002

During this period our research investigated some new directions in error-correcting codes and algorithms. Specifically, we investigated the existence of locally testable error-correcting codes and got some promising results. We also investigated the possibility of going decoding beyond the potential barrier to list-decoding. Very recent results (still to be fully written up) are showing surprising promise. Additionally we made some progress in our investigation of digital watermarking.

Some important milestones during this period include:

- Guruswami's thesis has been declared a winner of the Sprowl prize for Ph.D. thesis in the department and will be nominated for an ACM award.
- The PI was invited to give the prestigious Erdos Memorial Lecture Series at the Hebrew University in Jerusalem in March 2002.
- 1 Ph.D. thesis filed by Eric Lehman on grammar-based compression.

Below some of the technical results in detail:

Locally testable codes: When contrasting standard designs of codes, with the nature of error-correction that happens in more natural processes one aspect stands out. Natural processes don't seem to perform error-correction very systematically, while "error-correction by design" typically freezes a source of data in order to correct errors. This motivated Prof. Oded Goldreich (Weizmann Institute) and the PI to consider a new style of error-correction in which the error-correcting process randomly chooses a small (constant) number of bits in a stored word that it knows to be redundant and performs a simple parity check on these locations. If it finds an error, it signals for a systematic error-correction, else it trusts that the data is right. Now if the probability that the parity check fails to detect any errors is low, over the randomness of the error-correcting algorithm, then can we be confident the word is close to a codeword? Not always, it turns out. However, if the code is a carefully designed code then such random tests have a hope of working. Such error-correcting codes, in which random tests can be used to estimate the number of errors, are called *locally testable codes.* In ongoing work with Prof. Goldreich, the PI is investigating the existence, constructiveness, and limits of locally testable codes. Prior to this investigation, it was suspected that such codes could be created provided one is willing to go for a polynomial blowup from the message size to the code size. Of course, for such results to be interesting, the blowup has to be reduced to being linear and it is still not known if that is possible. The investigations from this project however got a promising intermediate result: It shows that the blowup required in locally testable codes can be reduced to being barely superlinear, smaller than N^A for every A>1. A particularly interesting aspect of this notion is that it sheds new light on, and raises new questions about, a classically investigated topic, namely low-density parity check codes. Locally testable codes are codes that have many alternate low-density parity check matrices. We are not aware of any other constructions of low-density codes that exhibit such choice and we intend to investigate this further.

Beyond list-decoding: One of the major advantages of list-decoding is that it allows for a possibility to decode beyond "half the minimum distance" of a code, in a completely adversarial setting. This led to the PI's investigation of this topic and the ensuing algorithms showed this was a viable approach to pushing the limits of error-correction. However, list-decoding has its limitations and in particular one cannot get efficient list-decoding algorithms if the size of the output list is exponentially large. In order to decode efficiently when an adversary can create malicious error patterns, one has to respect the limitations of list-decoding. To correct more errors in "typical" scenarios, one could revert to the notion of random errors and see if one can hope to recover from more errors in such cases. In a recent development, Don Coppersmith (IBM) and the PI have shown that one can design codes where algebraic decoding can correct random errors well beyond known results on list-decoding. At the moment, it is not clear as to whether these actually beat limits on list-decoding because the combinatorial issues related to list-decoding are not fully understood yet. Once again, the topic seems ripe for further exploration.

Research Plan for the Next Six Months

The research plan for the next six months includes some of the specific questions raised by our new successes above. Specifically, we intend to explore:

1. Constructions of locally testable codes.
2. Limitations to locally testable codes.
3. Limitations of list-decoding.
4. Algebraic decoding of random errors and consequences.