DOI:10.1145/1467247.1467268

Technical Perspective

The Beauty of Error-Correcting Codes

By Daniel A. Spielman

ERROR-CORRECTING CODES ARE the means by which we compensate for interference in communication, and are essential for the accurate transmission and storage of digital data. All communication mechanisms and storage devices are subject to interference, typically called "noise," which corrupts communicated messages and stored data. Thus, for a communication system to faithfully transmit data, it must build redundancy into its transmissions in such a way that even if a transmission is partially corrupted, the intended message may be reconstructed. Error-correcting codes provide the mapping from messages to redundant transmissions.

For example, a message is usually a string of zeros and ones. A redundant encoding of a message may be obtained by appending a few parity bits to the original message, to form a *codeword*. The *rate* of a code is the ratio of the length of a message to the length of a codeword, and equals the reciprocal of the redundancy. A communication medium, called a channel, might transmit bits, and noise could flip bits from zero to one or one to zero. For example, the Binary Symmetric Channel with crossover probability p transmits bits, and flips each bit with probability p, independently. An error-correcting code is designed with an abstract model of the target communication channel in mind.

Given a model of a channel, one should design a code that maximizes the rate while minimizing some tradeoff of error-probability, delay, and the computational complexity of encoding and decoding. While the goal of achieving low probability of error in a communication system is fundamentally probabilistic, major advances in the field have been made through a worst-case, deterministic, approach. The paper here by Guruswami and Rudra surveys developments in the worst-case approach to the coding problem, and explains their own recent contributions. They build on the classical Reed-Solomon codes.

Reed-Solomon codes employ a signaling alphabet containing more elements than just zero and one: each symbol is an element of a finite field, such as the integers modulo a prime. In a Reed-Solomon code of rate R, classic decoding algorithms can efficiently reconstruct a message so long as at most a $(1–R)/2$ fraction of the symbols in the transmitted codeword are corrupted. This is exactly the fraction of errors up to which the problem is guaranteed to have a unique solution: there exist rare patterns containing just one more error for which two codewords are equally close to the corrupted transmission.

A major advance in the decoding of Reed-Solomon codes was Sudan's[4] algorithm for list decoding Reed-Solomon codes. A list-decoding decoder returns the list of all codewords within some distance of a corrupted transmission. While the closest codeword is usually unique, the algorithmic task is simplified by the option of returning a list. Guruswami and Sudan's[1] improvement of Sudan's list decoder efficiently returns the list of all codewords that differ from a corrupted transmission in at most a $1 – \sqrt{R}$ fraction of symbols, and the list is guaranteed to be short.

This was a big improvement over previous decoding algorithms, but made little difference at the desirable high rates (near 1), where $1 – \sqrt{R}$ is approximately the same as $(1–R)/2$. Guruswami and Rudra's advance exploits an idea of Parvaresh and Vardy[3] for bundling Reed-Solomon alphabet symbols together. This makes the signaling alphabet slightly larger, but greatly increases the fraction of errors under which efficient list decoding is possible. They obtain codes of rate R from which one can efficiently produce the list of all codewords that differ from a corrupted transmission in a fraction of symbols approaching $1 – R$. For high-rate codes, this is almost twice as many errors as previous schemes could correct. Moreover, we know that one cannot hope to do better.

While a tremendous theoretical advance, more work is required before these codes can be used in practical communication systems. The decoding algorithms run in polynomial time, but need to be faster before they can be applied in practice. They also need to be extended to incorporate information from lower levels of the communication system. Few communication media naturally transmit finite field elements, or even zeros and ones. These symbols are usually converted into analog waveforms. Receivers of partially corrupted waveforms can do more than just report which valid waveform is closest: they can return the likelihood of each valid waveform. A *soft-decision decoder* incorporates this information into the decoding process.

Koetter and Vardy[2] figured out how to incorporate such information in the Guruswami-Sudan algorithm, and an analogous discovery may be required before we communicate using Guruswami-Rudra codes.

References

1. Guruswami, V. and Sudan, M. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Trans. on Info. Theory, 45* (1999), 1757–1767.
2. Koetter, R. and Vardy, A. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. on Info. Theory 49*, 112 (2003), 2809–2825.
3. Parvaresh, F. and Vardy, A. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science* (2005), 285–294.
4. Sudan, M. Decoding the Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity 13*, 1 (1997), 180–193.

Daniel A. Spielman (Spielman@cs.yale.edu) is a professor of applied mathematics and computer science at Yale University, New Haven, CT.