SUMMER 2009 - NSERC USRA REPORT

ERROR-CORRECTING CODES AND THEIR APPLICATIONS

AL-HASAN AL-AZZAWI

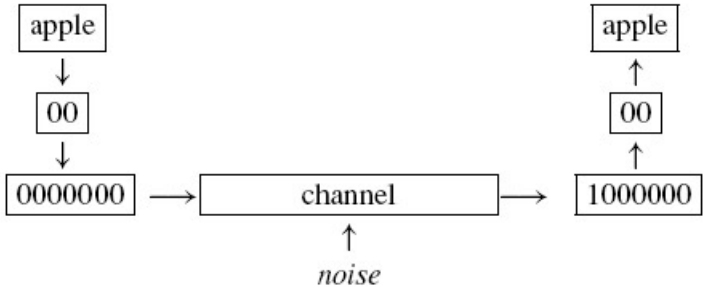
Information media, such as communication systems and storage devices of data, are not absolutely reliable in practice because of noise or other forms of introduced interference. One of the tasks in coding theory is to detect, or even correct, errors. One of the most important applications of coding theory is the Compact Disk (CD). Reed-Solomon codes are used to overcome an enormous number of errors introduced due to imperfections on the CD such as dirt and scratches. It is not an exaggeration to say that without error-correcting codes, the Compact Disk system would not be technically feasible. Under the supervision of Dr. Hamid Usefi, I undertook the task of studying and implementing the theories of this topic.

Consider the source encoding of four fruits, apple, banana, cherry, grape, as follows: $apple \to 00$, $banana \to 01$, $cherry \to 10$, $grape \to 11$. Suppose the message apple, which is encoded as 00, is transmitted over a noisy channel. The message may become distorted and may be received as 01. The receiver may not realize that the message was corrupted. This communication fails. How can we store the information so that all is not lost? On thing we can do is to encode the message by repeating every bit 3 times to create a codeword. Then to decode: in every triple of bits in the received word, we take a majority vote to determine the bit of the message, see the figure below. This is however a naive way of encoding. With the help of error-correcting codes, we can use much more efficient ways to encode and decode that would add little redundancy while achieving better error-correcting ability.

I began my research by studying topics in number theory such as finite fields and minimal polynomials since Coding Theory requires this background. Then under the guidance of Dr. Usefi, I studied several types of codes, ranging from the general and basic types such as linear codes to more complicated ones such as Reed-Solomon codes. In this project I had the opportunity to learn Maple and C++ more in depth. I have implemented several programs in both Maple and C++, including the implementation of the Reed-Solomon code with parameters [255, 251, 5]. These programs can be accessed at [2].

The Reed-Solomon code [255, 251, 5] was first implemented successfully using the high level programming language Maple. This code can correct up to 2 errors if used on its own. I chose it because of the fact that it is the same code used in Compact Discs. The error

Date: Aug, 31 2009.



locator polynomial in the program was determined using the Euclidean algorithm method. Dr. Usefi suggested using the NTL library for C++ to implement the decoding. The NTL library [6] by V. Shoup provides arithmetic modulo finite fields, and also goes well beyond that as NTL is known for having one of the fastest polynomial arithmetic. After building this decoder, I tested it against maple. Although not very user-friendly, it proved to be of better performance as the decoding was almost twice as fast as the one done with maple.

After deeply learning Reed-Solomon codes, I conducted a research on their application in the Compact Disk and the ways in which CDs are encoded and decoded. Encouraged by Dr. Usefi, I have written a report [1] that summarizes the error-correcting codes I have studied and their applications in Compact Disks. Later I indulged in more advanced topics such as list-decoding of Reed-Solomon codes [4] and studied recent research papers on the topic such as [3] and [5]. An implementation of the Guruswami-Sudan list decoder for Reed-Solomon Codes is currently a work in progress and will hopefully be finished by the end of my work term.

References

[1] A. Al-Azzawi, An introduction to Error-Correcting Codes. Maybe accessed at `http://codingprograms.googlecode.com/files/codingtheory.pdf` (2009).

[2] A. Al-Azzawi, Implementations for Coding Theory. Maybe accessed at `http://code.google.com/p/codingprograms/downloads/list`.

[3] V. Guruswami and A. Rudra, Explicit capacity-achieving list-decodable codes, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing* (2006), p. 1-10.

[4] V. Guruswami and M. Sudan, Improved decoding of Reed-Solomon codes and algebraic geometry codes, *IEEE Trans. Inform. Theory* vol. 45 no. 6 (1999), p. 17571767.

[5] F. Parvaresh and A. Vardy, Correcting errors beyond the Guruswami-Sudan radius in polynomial time, *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science* (2005), p. 285294.

[6] V. Shoup. NTL: A library for doing number theory, 19902007. Homepage at `http://www.shoup.net/ntl/`.