The Complexity of Error-Correcting Codes

Daniel A. Spielman
Massachusetts Institute of Technology

Abstract. By concatenating linear-time codes with small, good codes, it is possible to construct in polynomial time a family of asymptotically good codes that approach the Shannon bound that can be encoded and decoded in linear time. Moreover, their probability of decoder error is exponentially small in the block length of the codes. In this survey, we will explain exactly what this statement means, how it is derived, and what problems in the complexity of error-correcting codes remain open. Along the way, we will survey some key developments in the complexity of error-correcting codes.

1 Introduction

Error-correcting codes are the means by which we compensate for the corruption that occurs in communication over imperfect channels. In a coding system, a message first passes through an encoder, which transforms it into a codeword; this codeword is then transmitted over the channel. The channel modifies the codeword by adding noise, so that the received word received by the receiver may differ from the codeword that was transmitted. The received word is processed by a decoder, which uses the received word to guess which codeword was transmitted, and outputs its guess. Much of the research on error-correcting codes is devoted to improving the trade-off between the probability that the decoder’s guess is correct and the complexity of the encoders and decoders.

In this survey, we examine the software complexity of this problem from an asymptotic perspective. We will restrict our attention to communication over the binary symmetric channel (BSC), as it seems natural to most computer scientists. A channel is binary if it allows the transmission of only two symbols, which we take to be 0 and 1. The binary symmetric channel with error-probability $p$ (BSC$p$) is the channel that

1 We should point out that, at the present time, almost every implementation of error-correcting codes uses special-purpose hardware. Moreover, coding schemes that are efficient in software can be inefficient in hardware, and vice versa.

2 Results similar to those in this survey may be obtained for any reasonable memoryless channel.
transmits one bit at a time and flips the value of that bit with probability $p$. If it does not flip a bit, then it transmits the bit correctly. For each bit transmitted by the channel, the probability that it is flipped is independent of the others.

In the rest of this introduction, we will review the definitions needed to describe the types of error-correcting codes we will use and then state the complexity and coding problems we will consider. We will conclude the introduction with an overview of the rest of the paper.

The framework presented in this survey builds on those developed in [4] and [29,30]. Most of the material in Sections 2 and 3 can be found in those papers. The material in Section 2 is standard, and we refer the reader to a reference such as [21], [40], or [3] for a more thorough treatment. We also recommend the recent survey by Vardy [41]. For references on error-correcting codes that emphasize an engineer's perspective, we recommend [26], [42] and [6].

1.1 Coding Definitions

A code is an injective mapping from strings to strings. We will consider families of block codes over the alphabet $\{0,1\}$. A binary block code is one whose messages are strings over $\{0,1\}^m$ and whose codewords are strings over $\{0,1\}^n$, for some $n \geq m$; the length of the code is $n$, and its rate is $n/m$. The strings of $\{0,1\}^m$ are the possible messages, and we assume that each is equally likely. Each image of a string from $\{0,1\}^m$ is a codeword, and the word "code" is sometimes used to refer just to the set of codewords. An encoder maps a message to its codeword, and a decoder maps a word in $\{0,1\}^n$ to either a codeword or a message indicating that it cannot decide on a codeword.

A family of codes can be defined as an infinite sequence of codes, each of a different length, indexed by their length. A code constructor for a family of codes is a device that takes as input a block length, and outputs a description of an encoder and a decoder for the code in the family of that length. We do not insist that this decoder be the best possible decoder for the resulting code, as we will measure the quality of the coding system by the number of errors that the decoder can actually correct. By measuring the complexity of the code constructor, we allow ourselves

---

3 The task of adjusting one's message space so that each message is equally likely is that of compression. Compression schemes need to take advantage of the special structure of the data to which they are applied. We would like to avoid such concerns. In situations in which error-free communication is desired, it is reasonable to assume that data has been compressed before it is encoded with an error-correcting code.