Lecture Notes in Computer Science 3282

Venkatesan Guruswami

List Decoding of Error-Correcting Codes

Winning Thesis of the
2002 ACM Doctoral Dissertation Competition

Springer

Author

Venkatesan Guruswami
University of Washington
Department of Computer Science and Engineering
Seattle, WA 98195-2350, USA
E-mail: venkat@cs.washington.edu

Library of Congress Control Number: 2004115727

CR Subject Classification (1998): E.4, F.2.2, H.1.1, G.2.1

ISSN 0302-9743
ISBN 3-540-24051-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 11365877 06/3142 5 4 3 2 1 0

To my parents

Foreword

How can one exchange information effectively when the medium of communication introduces errors? This question has been investigated extensively starting with the seminal works of Shannon (1948) and Hamming (1950), and has led to the rich theory of "error-correcting codes". This theory has traditionally gone hand in hand with the algorithmic theory of "decoding" that tackles the problem of recovering from the errors *efficiently.* This thesis presents some spectacular new results in the area of decoding algorithms for error-correcting codes. Specifically, it shows how the notion of "list-decoding" can be applied to recover from far more errors, for a wide variety of error-correcting codes, than achievable before.

A brief bit of background: *error-correcting codes* are combinatorial structures that show how to represent (or "encode") information so that it is resilient to a moderate number of errors. Specifically, an error-correcting code takes a short binary string, called the message, and shows how to transform it into a longer binary string, called the codeword, so that if a small number of bits of the codeword are flipped, the resulting string does not look like any other codeword. The maximum number of errors that the code is guaranteed to detect, denoted d, is a central parameter in its design. A basic property of such a code is that if the number of errors that occur is known to be smaller than $d/2$, the message is determined uniquely. This poses a computational problem, called the decoding problem: compute the message from a corrupted codeword, when the number of errors is less than $d/2$. While naive decoding algorithms run in time exponential in d, sophisticated algorithms with polynomial running time have been found for a variety of codes, enabling widespread usage of error-correcting codes.

The principal concern of this thesis is the question: "What happens when the number of errors that occur is more than $d/2$?" This question is important for practical purposes, so that one can extract more out of any given communication channel. Furthermore, the central nature of error-correcting codes in the theory of computer science makes this question an important one in this domain as well. It is well known that if the number of errors exceed $d/2$, then the message may potentially not be recoverable uniquely. However, it is conceivable that one can pin down a small list of candidate messages that include the intended message. This possibility motivated Elias

(1957) and Wozencraft (1958) to define the list-decoding problem: “Given a corrupted codeword and an error parameter e, compute a list of all codewords that differ from the corrupted word in most e places.”

Even though the list-decoding problem had been in existence for several decades, it did not meet with algorithmic success till 1997. In the last ten years or so, however, this area has seen some remarkable advances, and these results represent the original contributions of this thesis. List-decoding algorithms are presented for a wide variety of codes considered in the literature including “Reed-Solomon codes”, “algebraic-geometry codes”, “concatenated codes”, and “graph-theoretic codes”. In addition to describing new results, the thesis also serves as a valuable source of reference on list-decoding. It introduces the topic gently, re-examining the definition, explaining why it is interesting and then describing the central combinatorial and algorithmic problems in this domain. It includes a nice survey of prior combinatorial work most of which is scattered in the literature. After covering the new algorithmic results, the thesis includes an excellent survey of the many applications of list-decoding in theoretical computer science including “hardness amplification”, “extracting randomness”, and “pseudorandomness”.

The style of the exposition is crisp and the enormous amount of information is presented in a clear, structured form. This thesis will be valuable to readers interested in mathematical aspects of computer science or communication.

August 2004

Madhu Sudan
Professor of Computer Science
MIT, Cambridge, MA, USA.

Preface

Error-correcting codes are combinatorial objects designed to cope with the problem of reliable transmission of information on a noisy channel. A fundamental algorithmic challenge in coding theory and practice is to efficiently decode the original transmitted message even when a few symbols of the received word are in error. The naive search algorithm runs in exponential time, and several classical polynomial time decoding algorithms are known for specific code families. Traditionally, however, these algorithms have been constrained to output a unique codeword. Thus they faced a "combinatorial barrier" and could only correct up to $d/2$ errors, where d is the minimum distance of the code.

An alternate notion of decoding called *list decoding*, proposed independently by Elias and Wozencraft in the late 1950s, allows the decoder to output a list of *all* codewords that differ from the received word in a certain number of positions. Even when constrained to output a relatively small number of answers, list decoding permits recovery from errors well beyond the $d/2$ barrier, and opens up the possibility of meaningful error correction from large amounts of noise. However, for nearly four decades after its conception, this potential of list decoding was largely untapped due to the lack of *efficient* algorithms to list decode beyond $d/2$ errors for useful families of codes.

This book presents a detailed investigation of list decoding, and proves its potential, feasibility, and importance as a combinatorial and algorithmic concept. The results discussed in the book are divided into three parts: the first one on combinatorial results, the second on polynomial time list decoding algorithms, and the third on applications. We describe each of the parts in further detail below.

Part I deals with the combinatorics of list decoding and attempts to sharpen our understanding of the potential and limits of list decoding, and its relation to more classical coding-theoretic parameters like the rate and minimum distance. A combinatorial bound called the Johnson bound asserts that codes with large minimum distance have a large list decoding radius, and this raises algorithmic questions on list decoding such codes from a large number of errors for central codes that are known to have good distance properties. This is not the only approach to obtaining good list decodable codes, and in fact directly optimizing the list decoding radius leads to better trade-offs as a function of the rate of the code (as can be shown by applications of the probabilistic method). Part I can be summed up with the statement: *good codes with excellent combinatorial list decodability properties exist.* This sets the stage for the algorithmic results of Part II by highlighting what one can and cannot hope to do with list decoding, and poses the challenge of tapping the potential of list decoding with efficient algorithms.

Part II comprises the crux of the book, namely its algorithmic results, which were lacking in the early works on list decoding. The algorithmic results attempt to "match" the combinatorial bounds with explicit code constructions and efficient decoding algorithms. Our algorithmic results include:

- Efficient list decoding algorithms for classically studied codes such as Reed-Solomon codes and algebraic-geometric codes. In particular, building upon an earlier algorithm by Sudan, we present the *first* polynomial time algorithm to decode Reed-Solomon codes beyond $d/2$ errors for every value of the rate.
- A new *soft* list decoding algorithm for Reed-Solomon and algebraic-geometric codes, and novel decoding algorithms for concatenated codes based on it.
- New code constructions using concatenation and/or expander graphs that have good (and sometimes near-optimal) rates and are efficiently list decodable from extremely large amounts of noise.
- Error-correcting codes with good (and sometimes near-optimal rates) for list decoding from erasures.

Part II can be summed up with the statement: *there exist "explicit" constructions of "good" codes together with efficient list decoding algorithms.*

In Part III, we discuss some applications of the results and techniques from earlier chapters to domains both within and outside of coding theory. Using an expander-based construction in the same spirit as our construction for list decoding, we get a significant improvement over a prior result for *unique decoding.* Specifically, we construct *linear time* encodable and decodable codes that match the trade-off between rate and error-correction radius achieved by the best known constructions with polynomial time decoding (and in fact the trade-off is almost the best possible over large alphabets). This constitutes a vast improvement compared with previous constructions of linear time codes that could only correct a tiny fraction of errors with positive rates. The notion of list decoding turns out to be central to certain contexts in theoretical computer science outside of coding theory, for example in complexity theory, cryptography, and algorithms. For these applications unique decoding does not suffice, and moreover, for several of them one needs *efficient* list decoding algorithms.

A detailed chapter by chapter description of the contents can be found in Section 2.3.

Acknowledgments

We know too much for one man to know much.
J. Robert Oppenheimer

This monograph is a revised version of my doctoral dissertation, written under the supervision of Madhu Sudan and submitted to MIT in August 2001. I am grateful to MIT for nominating my Ph.D. thesis for the ACM Doctoral Dissertation Award competition, and to ACM and the awards committee for awarding the honor to my dissertation.

My first and foremost acknowledgment is to my advisor Madhu Sudan. When I made a decision to go to MIT for grad school in the spring of 1997, I was not aware that Madhu Sudan would be joining its faculty that Fall, so it was quite serendipitous that I got him as my advisor. While I found MIT to be every bit the wonderful place I had anticipated it to be and more, Madhu was the most important reason my academic experience at MIT was so enjoyable and fulfilling. For the wonderful collaboration which led to several of the key chapters of my thesis, for all his patient advice, help and support on matters technical and otherwise, and for all the things I learned from him during my stay at MIT and continue to do so, I will be forever grateful to Madhu.

I am most grateful to Madhu Sudan, Johan Håstad, Piotr Indyk, Amit Sahai, and David Zuckerman for their collaboration which led to several of the results discussed in this monograph. Collectively, this is as much, if not more, their book as it is mine. I also wish to thank the several other people with whom I have had useful discussions on coding theory and related topics. These include Noga Alon, Sanjeev Arora, Sasha Barg, Moses Charikar, Yevgeniy Dodis, Peter Elias, Sanjeev Khanna, Subhash Khot, Ralf Koetter, Ravi Kumar, Hendrik Lenstra, Daniele Micciancio, Jaikumar Radhakrishnan, Amin Shokrollahi, D. Sivakumar, Dan Spielman, Luca Trevisan, Salil Vadhan, and Alex Vardy, though undoubtedly I have left out several others.

A special thanks is due to the members of my thesis reading committee at MIT: Peter Elias, Dan Spielman, and Madhu Sudan. Technically, it was only appropriate that I had these three people on my committee: Peter first defined the notion of list decoding; Madhu discovered the first non-trivial efficient list decoding algorithm; and Dan constructed the first linear-time codes (the subject of Chapter 11 of this book). I regret that I will not be able to present a personal copy of the book to Peter, who sadly left us a few months after I submitted my thesis to MIT.

It is with really fond memories that I acknowledge the stimulating working atmosphere and the company of a great group of friends and colleagues that I found in MIT's theory group. The good time I had at MIT owes a lot to the wonderful student body I had the privilege of being a part of. I would like to thank Salil, Yevgeniy, Eric, Amit, Raj, Anna, Sofya, Adam K., Adam S., Maria, Matthias, Feldman, Abhi, Rocco, Daniele, Alantha, Ryan, Prahladh, and many others, for numerous conversations on all sorts of topics, and for making my life at MIT LCS so much fun. I was lucky that Luca Trevisan was at MIT the year I started; from him I learned a lot, and with him (and Danny Lewin and Madhu) I shared my first research experience in graduate school. In my last year at MIT I benefited immensely from the time I spent working and hanging out with Piotr Indyk, for which I sincerely thank him. I relish very much our continuing collaboration on expander codes. Lars Engebretsen, the other member of our espresso trio, also contributed greatly to making my final year at MIT so memorable.

My sincere thanks to the theory group staff, and in particular Joanne Talbot and Be Blackburn, for their good cheer and all their administrative and other help.

It is a pleasure to acknowledge my current academic home, University of Washington CSE, for its warm and congenial atmosphere, with special thanks to my theory colleagues Paul Beame, Anna Karlin and Richard Ladner for their support and company.

A *huge* thanks to all my friends whom I met at various junctures of my life. True friends are those who take pride in your achievements, and I am grateful that I have several who meet this definition and who are an inseparable part of my life.

I owe a lot to two professors from college: C. Pandu Rangan for encouraging me in every possible way and getting me started on research well before I started grad school; and S. A. Choudum whose wonderful Graph Theory course sparked my interest in algorithmic graph theory and eventually theoretical computer science.

My most important acknowledgment is to my close and loving family: my parents and my sister Shantha, who have filled my life with joy and who mean the world to me. Many thanks to Vaishnavi, my most fortunate discovery, for her cheer and providing useful distractions during the course of this revision.

Words cannot express my thanks to my parents for all that they have gone through and done for me. So, of all the sentences in this book none was easier to write than this one: To my parents, this book is dedicated with love.

Seattle, Washington
August 2004

Venkatesan Guruswami

I gratefully acknowledge the fellowships and grants that supported my research at MIT. My research was supported in part by funding from NSF CCR 9875511, NSF CCR 9912342, and NTT Award MIT 2001-04, and in part by an IBM Graduate Fellowship.

Contents

Part I Combinatorial Bounds

Part II Code Constructions and Algorithms

Part III Applications