Algorithmic Results in List Decoding

Algorithmic Results in List Decoding

Venkatesan Guruswami

Department of Computer Science & Engineering
University of Washington Seattle WA 98195, USA
venkat@cs.washington.edu

now
the essence of knowledge
Boston – Delft

Foundations and Trends® in Theoretical Computer Science

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

Library of Congress Control Number: 2006935853

The preferred citation for this publication is V. Guruswami, Algorithmic Results in List Decoding, Foundation and Trends® in Theoretical Computer Science, vol 2, no 2, pp 107–195, 2006

Printed on acid-free paper

ISBN: 1-60198-004-5

Foundations and Trends® in Theoretical Computer Science

Volume 2 Issue 2, 2006

Editorial Scope

Foundations and Trends® in Theoretical Computer Science will publish survey and tutorial articles in the following topics:

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations Research
- Parallel algorithms
- Quantum Computation
- Randomness in Computation

Information for Librarians

Foundations and Trends® in Theoretical Computer Science, 2006, Volume 2, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 2, No 2 (2006) 107–195
© 2006 V. Guruswami
DOI: 10.1561/0400000007

Algorithmic Results in List Decoding

Venkatesan Guruswami

*Department of Computer Science & Engineering, University of Washington
Seattle WA 98195, USA, venkat@cs.washington.edu*

Abstract

Error-correcting codes are used to cope with the corruption of data by noise during communication or storage. A code uses an encoding procedure that judiciously introduces redundancy into the data to produce an associated codeword. The redundancy built into the codewords enables one to decode the original data even from a somewhat distorted version of the codeword. The central trade-off in coding theory is the one between the data rate (amount of non-redundant information per bit of codeword) and the error rate (the fraction of symbols that could be corrupted while still enabling data recovery). The traditional decoding algorithms did as badly at correcting any error pattern as they would do for the worst possible error pattern. This severely limited the maximum fraction of errors those algorithms could tolerate. In turn, this was the source of a big hiatus between the error-correction performance known for probabilistic noise models (pioneered by Shannon) and what was thought to be the limit for the more powerful, worst-case noise models (suggested by Hamming).

In the last decade or so, there has been much algorithmic progress in coding theory that has bridged this gap (and in fact nearly eliminated it for codes over large alphabets). These developments rely on an error-recovery model called "list decoding," wherein for the pathological error patterns, the decoder is permitted to output a small list of candidates that will include the original message. This book introduces and motivates the problem of list decoding, and discusses the central algorithmic results of the subject, culminating with the recent results on achieving "list decoding capacity."

Contents

Part I

General Literature

1
Introduction

1.1 Codes and noise models

Error-correcting codes enable reliable transmission of information over a noisy communication channel. The idea behind error-correcting codes is to *encode* the message to be transmitted into a longer, *redundant* string (called a *codeword*) and then transmit the codeword over the noisy channel. The redundancy is judiciously chosen in order to enable the receiver to *decode* the transmitted codeword even from a somewhat distorted version of the codeword. Naturally, the larger the amount of noise (quantified appropriately, according to the specific channel noise model) one wishes to correct, the greater the redundancy that needs to be introduced during encoding. A convenient measure of the redundancy is the *rate* of an error-correcting code, which is the ratio of the number of information bits in the message to the number of bits in the codeword. The larger the rate, the less redundant the encoding.

The trade-off between the rate and the amount of noise that can be corrected is a fundamental one, and understanding and optimizing the precise trade-off is one of the central objectives of coding theory. The optimal rate for which reliable communication is possible on a given noisy channel is typically referred to as "capacity." The challenge is to construct codes with rate close to capacity, together with efficient algorithms for encoding and error correction (decoding).

The underlying model assumed for the channel noise crucially governs the rate at which one can communicate while tolerating noise. One of the simplest models is the binary symmetric channel; here the channel flips each bit independently with a certain cross-over probability p. It is well-known that the capacity of this channel equals $1 - H(p)$ where $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function. In other words, there are codes of rate up to $1 - H(p)$ that achieve probability of miscommunication approaching 0 (for large message lengths), and for rates above $1 - H(p)$, no such codes exist.

The above was a *stochastic* model of the channel, wherein we took an optimistic view that we knew the precise probabilistic behavior of the channel. This stochastic approach was pioneered by Shannon in his landmark 1948 paper that marked the birth of the field of information theory [65]. An alternate, more combinatorial approach, put forth by Hamming [46], models the channel as a jammer or adversary that can corrupt the codeword arbitrarily, subject to a bound on the total number of errors it can cause. This is a stronger noise model since one has to deal with *worst-case* or adversarial, as opposed to typical, noise patterns. Codes and algorithms designed for worst-case noise are more robust and less sensitive to inaccuracies in modeling the precise channel behavior (in fact, they obviate the need for such precise modeling!).

This survey focuses on the worst-case noise model. Our main objective is to highlight that even against adversarial channels, one can achieve the information-theoretically optimal trade-off between rate and fraction of decodable errors, matching the performance possible against weaker, stochastic noise models. This is shown for an error recovery model called *list decoding*, wherein for the pathological, worst-case noise patterns, the decoder is permitted to output a small list of candidate messages that will include the correct message. We next motivate the list decoding problem, and discuss how it offers the hope of achieving capacity against worst-case errors.

Remark 1. [Arbitrarily varying channel]

The stochastic noise model assumes knowledge of the precise probability law governing the channel. The worst-case model takes a conservative, pessimistic view of the power of the channel assuming only a limit on the total amount of noise. A hybrid model called *Arbitrarily Varying Channel* (AVC) has also been proposed to study communication under channel uncertainty. Here the channel is modeled as a jammer which can select from a family of strategies (corresponding to different probability laws) and the sequence of selected strategies, and hence the channel law, is not known to the sender. The strategy can in general vary arbitrarily from symbol to symbol, and the goal is to do well against the worst possible sequence. A less powerful model is that of the *compound channel* where the jammer has a choice of strategies, but the chosen channel law does not change during the transmission of various symbols of the codeword. AVCs have been the subject of much research – the reader can find a good introduction to this topic as well as numerous pointers to the extensive literature in a survey by Lapidoth and Narayan [54]. To the author's understanding, it seems that much of the work has been of a non-constructive flavor, driven by the information-theoretic motivation of determining the capacity under different AVC variants. There has been less focus on explicit constructions of codes or related algorithmic issues.

1.2 List decoding: Context and motivation

Given a received word **r**, which is a distorted version of some codeword **c**, the decoding problem strives to find the original codeword **c**. The natural error recovery approach is to place one's bet on the codeword that has the highest likelihood of being the one that was transmitted, conditioned on receiving **r**. This task is called *Maximum Likelihood Decoding* (MLD), and is viewed as the holy grail in decoding. MLD amounts to finding the codeword *closest* to **r** under an appropriate distance measure on distortions (for which a larger distortion is less likely than a smaller one). In this survey, we will measure distortion by the Hamming metric, i.e., the distance between two strings $x, y \in \Sigma^n$

is the number of coordinates $i \in \{1,2,\ldots,n\}$ for which $x_i \neq y_i$. MLD thus amounts to finding the codeword closest to the received word in Hamming distance. No approach substantially faster than a brute-force search is known for MLD for any non-trivial code family. One, therefore settles for less ambitious goals in the quest for efficient algorithms. A natural relaxed goal, called *Bounded Distance Decoding* (BDD), would be to perform decoding in the presence of a bounded number of errors. That is, we assume at most a fraction p of symbols are corrupted by the channel, and aim to solve the MLD problem under this promise. In other words, we are only required to find the closest codeword when there is a codeword not too far away (within distance pn) from the received word.

In this setting, the basic trade-off question is: What is the largest fraction of errors one can correct using a family of codes of rate R? Let $C : \Sigma^{Rn} \to \Sigma^n$ be the encoding function of a code of rate R (here n is the *block length* of the code, and Σ is the *alphabet* to which codeword symbols belong). Now, a simple pigeonholing argument implies there must exist $x \neq y$ such that the codewords $C(x)$ and $C(y)$ agree on the first $Rn - 1$ positions. In turn, this implies that when $C(x)$ is transmitted, the channel could distort it to a received word $\mathbf{r}$ that is equidistant from both $C(x)$ and $C(y)$, and differs from each of them in about a fraction $(1-R)/2$ of positions. Thus, unambiguous bounded distance decoding becomes impossible for error fractions exceeding $(1-R)/2$.

However, the above is not a compelling reason to be pessimistic about correcting larger amounts of noise. This is due to the fact that received words such as $\mathbf{r}$ reflect a pathological case. The way Hamming spheres pack in high-dimensional space, even for p much larger than $(1-R)/2$ (and in fact for $p \approx 1-R$) there exist codes of rate R (over a larger alphabet Σ) for which the following holds: for *most* error patterns $\mathbf{e}$ that corrupt fewer than a fraction p of symbols, when a codeword $\mathbf{c}$ gets distorted into $\mathbf{z}$ by the error pattern $\mathbf{e}$, there will be no codeword besides $\mathbf{c}$ within Hamming distance pn of $\mathbf{z}$.[1] Thus, for typical noise

[1] This claim holds with high probability for a random code drawn from a natural ensemble. In fact, the proof of Shannon's capacity theorem for q-ary symmetric channels can be viewed in this light. For Reed–Solomon codes, which will be our main focus later on, this claim has been shown to hold, see [19, 59, 58].

patterns one can hope to correct many more errors than the above limit faced by the worst-case error pattern. However, since we assume a worst-case noise model, we do have to deal with bad received words such as **r**. List decoding provides an elegant formulation to deal with worst-case errors without compromising the performance for typical noise patterns – the idea is that in the worst-case, the decoder may output multiple answers. Formally, the decoder is required to output a list of all codewords that differ from the received word in a fraction p of symbols.

Certainly returning a small list of possibilities is better and more useful than simply giving up and declaring a decoding failure. Even if one deems receiving multiple answers as a decoding failure, as mentioned above, for many error patterns in the target noise range, the decoder will output a unique answer, and we did not have to model the channel stochastics to design our code or algorithm! It may also be possible to pick the correct codeword from the list, in case of multiple answers, using some semantic context or side information (see [23]). Also, if in the output list, there is a unique closest codeword, we can also output that as the maximum likelihood choice. In general, list decoding is a stronger error-recovery model than outputting just the closest codeword(s), since we require that the decoder output all the close codewords (and we can always prune the list as needed). For several applications, such as concatenated code constructions and also those in complexity theory, having the entire list adds more power to the decoding primitive than deciding solely on the closest codeword(s).

Some other channel and decoding models. We now give pointers to some other relaxed models where one can perform unique decoding even when the number of errors exceeds half the minimum Hamming distance between two codewords. We already mentioned one model where an auxiliary channel can be used to send a small amount of side information which can be used to disambiguate the list [23]. Another model that allows one to identify the correct message with high probability is one where the sender and recipient share a secret random key, see [53] and a simplified version in [67].

Finally, there has been work where the noisy channel is modeled as a *computationally bounded* adversary (as opposed to an all-powerful adversary), that must introduce the errors in time polynomial in the block length. This is a very appealing model since it is a reasonable hypothesis that natural processes can be implemented by efficient computation, and therefore real-world channels are, in fact, computationally bounded. The computationally bounded channel model was put forth by Lipton [56]. Under standard cryptographic assumptions, it has been shown that in the private key model where the sender and recipient share a secret random seed, it is possible to decode correctly from error rates higher than half-the-minimum-distance bound [21,48]. Recently, similar results were established in a much simpler cryptographic setting, assuming only that one-way functions exist, and that the sender has a public key known to the receiver (and possibly to the channel as well) [60].

1.3 The potential of list decoding

The number of codewords within Hamming distance pn of the worst-case received word $\mathbf{r}$ is clearly a lower bound on the runtime of any list decoder that corrects a fraction p of errors. Therefore, in order for a polynomial time list decoding algorithm to exist, the underlying codes must have the *a priori* combinatorial guarantee of being *p-list-decodable*, namely every Hamming ball of radius pn has a small number, say $L(n)$, of codewords for some polynomially bounded function $L(\cdot)$.[2] This "packing" constraint poses a combinatorial upper bound on the rate of the code; specifically, it is not hard to prove that we must have $R \leqslant 1 - p$ or otherwise the worst-case list size will grow faster than any polynomial in the block length n.

Remarkably, this simple upper bound can actually be met. In other words, for every p, $0 < p < 1$, there exist codes of rate $R = 1 - p - o(1)$ which are p-list-decodable. That is, non-constructively we can show the existence of codes of rate R that offer the potential of list decoding up to a fraction of errors approaching $(1 - R)$. We will refer to the quantity $(1 - R)$ as the *list decoding capacity.* Note that the list decoding

[2] Throughout the survey, we will be dealing with the asymptotics of a family of codes of increasing block lengths with some fixed rate.

capacity is *twice* the fraction of errors that one could decode if we insisted on a unique answer always – quite a substantial gain! Since the message has Rn symbols, information-theoretically we need at least a fraction R of correct symbols at the receiving end to have any hope of recovering the message. Note that this lower bound applies even if we somehow *knew* the locations of the error and could discard those misleading symbols. With list decoding, therefore, we can potentially reach this information-theoretic limit and decode as long as we receive slightly more than Rn correct symbols (the correct symbols can be located arbitrarily in the received word, with arbitrary noise affecting the remaining positions).

To realize this potential, however, we need an *explicit description* of such capacity-achieving list-decodable codes, and an efficient algorithm to perform list decoding up to the capacity (the combinatorics only guarantees that every Hamming ball of certain radius has a small number of codewords, but does not suggest any efficient algorithm to actually find those codewords). The main technical result in this survey will achieve precisely this objective – we will give explicit codes of rate R with a polynomial time list decoding algorithm for a fraction $(1 - R - \varepsilon)$ of errors, for any desired $\varepsilon > 0$.

The above description was deliberately vague on the size of the alphabet Σ. The capacity $1 - R$ for codes of rate R applies in the limit of large alphabet size. It is also of interest to ask how well list decoding performs for codes over a fixed alphabet size q. For the binary $(q = 2)$ case, to correct a fraction p of errors, list decoding offers the potential of communicating at rates up to $1 - H(p)$. This is exactly the capacity of the binary symmetric channel with cross-over probability p that we discussed earlier. With list decoding, therefore, we can deal with worst-case errors without any loss in rate. For binary codes, this remains a non-constructive result and constructing explicit codes that achieve list decoding capacity remains a challenging goal.

1.4 The origins of list decoding

List decoding was proposed in the late 50s by Elias [13] and Wozencraft [78]. Curiously, the original motivation in [13] for formulating list decoding was to prove matching upper and lower bounds on the decoding error probability under maximum likelihood decoding on the binary symmetric channel. In particular, Elias showed that, when the decoder is allowed to output a small list of candidate codewords and a decoding error is declared only when the original codeword is not on the output list, the average error probability of all codes is almost as good as that of the best code, and in fact almost all codes are almost as good as the best code. Despite its origins in the Shannon stochastic school, it is interesting that list decoding ends up being the right notion to realize the true potential of coding in the Hamming combinatorial school, against worst-case errors.

Even though the notion of list decoding dates back to the late 1950s, it was revived with an algorithmic focus only recently, beginning with the Goldreich–Levin algorithm [17] for list decoding Hadamard codes, and Sudan's algorithm in the mid 1990s for list decoding Reed–Solomon codes [69]. It is worth pointing out that this modern revival of list decoding was motivated by questions in computational complexity theory. The Goldreich–Levin work was motivated by constructing hard-core predicates, which are of fundamental interest in complexity theory and cryptography. The motivation for decoding Reed–Solomon and related polynomial-based codes was (at least partly) establishing worst-case to average-case reductions for problems such as the permanent. These and other more recent connections between coding theory (and specifically, list decoding) and complexity theory are surveyed in [29, 70, 74] and [28, Chapter 12].

1.5 Scope and organization of the book

The goal of this survey is to obtain algorithmic results in list decoding. The main technical focus will be on giving a complete presentation of the recent algebraic results achieving list decoding capacity. We will only provide pointers or brief descriptions for other works on list decoding.

The survey is divided into two parts. The first part (Chapters 1–5) covers the general literature, and the second part focuses on achieving list decoding capacity. The author's Ph.D. dissertation [28] provides a more comprehensive treatment of list decoding. In comparison with

[28], most of Chapter 5 and the entire Part II of this survey discuss material developed since [28].

We now briefly discuss the main technical contents of the various chapters. The basic terminology and definitions are described in Chapter 2. Combinatorial results which identify the potential of list decoding in an existential, non-constructive sense are presented in Chapter 3. In particular, these results will establish the capacity of list decoding (over large alphabets) to be $1 - R$. We begin the quest for explicitly and algorithmically realizing the potential of list decoding in Chapter 4, which discusses a list decoding algorithm for Reed–Solomon (RS) codes – the algorithm is based on bivariate polynomial interpolation. We conclude the first part with a brief discussion in Chapter 5 of algorithmic results for list decoding certain codes based on expander graphs.

In Chapter 6, we discuss folded Reed–Solomon codes, which are RS codes viewed as a code over a larger alphabet. We present a decoding algorithm for folded RS codes that uses multivariate interpolation plus some other algebraic ideas concerning finite fields. This lets us approach list decoding capacity. Folded RS codes are defined over a polynomially large alphabet, and in Chapter 7 we discuss techniques that let us bring down the alphabet size to a constant independent of the block length. We conclude with some notable open questions in Chapter 8.

References

[1] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth, "Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs," *IEEE Transactions on Information Theory*, vol. 38, pp. 509–516, 1992.

[2] N. Alon and F. R. K. Chung, "Explicit construction of linear sized tolerant networks," *Discrete Mathematics*, vol. 72, pp. 15–19, 1988.

[3] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1732–1736, 1996.

[4] N. Alon and J. Spencer, *The Probabilistic Method*. John Wiley and Sons, Inc., 1992.

[5] S. Ar, R. Lipton, R. Rubinfeld, and M. Sudan, "Reconstructing algebraic functions from mixed data," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 488–511, 1999.

[6] E. Berlekamp, "Factoring polynomials over large finite fields," *Mathematics of Computation*, vol. 24, pp. 713–735, 1970.

[7] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding of interleaved Reed Solomon codes over noisy data," in *Proceedings of the 30th International Colloquium on Automata, Languages and Programming*, pp. 97–108, 2003.

[8] V. M. Blinovsky, "Bounds for codes in the case of list decoding of finite volume," *Problems of Information Transmission*, vol. 22, no. 1, pp. 7–19, 1986.

[9] V. M. Blinovsky, "Code bounds for multiple packings over a nonbinary finite alphabet," *Problems of Information Transmission*, vol. 41, no. 1, pp. 23–32, 2005.

[10] D. Boneh, “Finding smooth integers in short intervals using CRT decoding,” in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pp. 265–272, 2000.

[11] D. Coppersmith and M. Sudan, “Reconstructing curves in three (and higher) dimensional spaces from noisy data,” in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pp. 136–142, June 2003.

[12] I. Dumer, D. Micciancio, and M. Sudan, “Hardness of approximating the minimum distance of a linear code,” *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 22–37, 2003.

[13] P. Elias, “List decoding for noisy channels,” Technical Report 335, Research Laboratory of Electronics, MIT, 1957.

[14] P. Elias, “Error-correcting codes for list decoding,” *IEEE Transactions on Information Theory*, vol. 37, pp. 5–12, 1991.

[15] G. D. Forney, *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

[16] P. Gemmell and M. Sudan, “Highly resilient correctors for multivariate polynomials,” *Information Processing Letters*, vol. 43, no. 4, pp. 169–174, 1992.

[17] O. Goldreich and L. Levin, “A hard-core predicate for all one-way functions,” in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pp. 25–32, May 1989.

[18] O. Goldreich, D. Ron, and M. Sudan, “Chinese remaindering with errors,” *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 1330–1338, July 2000.

[19] O. Goldreich, R. Rubinfeld, and M. Sudan, “Learning polynomials with queries: The highly noisy case,” in *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pp. 294–303, 1995.

[20] O. Goldreich, R. Rubinfeld, and M. Sudan, “Learning polynomials with queries: The highly noisy case,” *SIAM Journal on Discrete Mathematics*, vol. 13, no. 4, pp. 535–570, November 2000.

[21] P. Gopalan, R. Lipton, and Y. Ding, “Error correction against computationally bounded adversaries,” *Theory of Computing Systems*, to appear.

[22] V. Guruswami, “Iterative decoding of low-density parity check codes (A Survey),” *CoRR*, cs.IT/0610022, 2006. Appears in Issue 90 of the *Bulletin of the EATCS*.

[23] V. Guruswami, “List decoding with side information,” in *Proceedings of the 18th IEEE Conference on Computational Complexity (CCC)*, pp. 300–309, July 2003.

[24] V. Guruswami, “Limits to list decodability of linear codes,” in *Proceedings of the 34th ACM Symposium on Theory of Computing*, pp. 802–811, 2002.

[25] V. Guruswami, “List decoding from erasures: Bounds and code constructions,” *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2826–2833, 2003.

[26] V. Guruswami, “Better extractors for better codes?,” in *Proceedings of 36th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 436–444, June 2004.

[27] V. Guruswami, “Error-correcting codes and expander graphs,” *SIGACT News*, pp. 25–41, September 2004.

[28] V. Guruswami, "List decoding of error-correcting codes," *Lecture Notes in Computer Science*, no. 3282, Springer, 2004.

[29] V. Guruswami, "List decoding in pseudorandomness and average-case complexity," in *IEEE Information Theory Workshop*, March 2006.

[30] V. Guruswami, J. Hastad, M. Sudan, and D. Zuckerman, "Combinatorial bounds for list decoding," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1021–1035, 2002.

[31] V. Guruswami and P. Indyk, "Expander-based constructions of efficiently decodable codes," in *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 658–667, 2001.

[32] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 812–821, 2002.

[33] V. Guruswami and P. Indyk, "Linear-time encodable and list decodable codes," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 126–135, June 2003.

[34] V. Guruswami and P. Indyk, "Linear-time list decoding in error-free settings," in *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 695–707, July 2004.

[35] V. Guruswami and P. Indyk, "Linear-time encodable/decodable codes with near-optimal rate," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3393–3400, October 2005.

[36] V. Guruswami and A. Patthak, "Correlated algebraic-geometric codes: Improved list decoding over bounded alphabets," in *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 227–238, October 2006. Journal version to appear in *Mathematics of Computation*.

[37] V. Guruswami and A. Rudra, "Explicit capacity-achieving list-decodable codes," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pp. 1–10, May 2006.

[38] V. Guruswami and A. Rudra, "Limits to list decoding Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 52, no. 8, August 2006.

[39] V. Guruswami, A. Sahai, and M. Sudan, "Soft-decision decoding of Chinese remainder codes," in *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pp. 159–168, 2000.

[40] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1757–1767, 1999.

[41] V. Guruswami and M. Sudan, "List decoding algorithms for certain concatenated codes," in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 181–190, 2000.

[42] V. Guruswami and M. Sudan, "Decoding concatenated codes using soft information," in *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC)*, pp. 148–157, 2002.

[43] V. Guruswami and M. Sudan, "On representations of algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1610–1613, May 2001.

[44] V. Guruswami, C. Umans, and S. Vadhan, "Extractors and condensers from univariate polynomials," *Electronic Colloquium on Computational Complexity,* Report TR06-134, October 2006.

[45] V. Guruswami and S. Vadhan, "A lower bound on list size for list decoding," in *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pp. 318–329, 2005.

[46] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, April 1950.

[47] R. Impagliazzo, R. Jaiswal, and V. Kabanets, "Approximately list-decoding direct product codes and uniform hardness amplification," in *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science*, October 2006.

[48] K. Jain and R. Venkatesan, "Efficient code construction via cryptographic assumptions," in *Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.

[49] R. Koetter, "On optimal weight assignments for multivariate interpolation list-decoding," in *IEEE Information Theory Workshop*, March 2006.

[50] R. Koetter and A. Vardy, "Soft decoding of Reed Solomon codes and optimal weight assignments," in *ITG Fachtagung*, January 2002. Berlin, Germany.

[51] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, 2003.

[52] V. Y. Krachkovsky, "Reed-Solomon codes for correcting phased error bursts," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2975–2984, November 2003.

[53] M. Langberg, "Private codes or succinct random codes that are (almost) Perfect," in *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pp. 325–334, 2004.

[54] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, October 1998.

[55] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications.* Cambridge University Press, Cambridge, MA, 1986.

[56] R. J. Lipton, "A new approach to information theory," in *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 699–708, 1994.

[57] C.-J. Lu, O. Reingold, S. P. Vadhan, and A. Wigderson, "Extractors: Optimal up to constant factors," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pp. 602–611, 2003.

[58] R. J. McEliece, "On the average list size for the Guruswami-Sudan decoder," in *7th International Symposium on Communications Theory and Applications (ISCTA)*, July 2003.

[59] R. J. McEliece and L. Swanson, "On the decoder error probability for Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 32, no. 5, pp. 701–703, 1986.

[60] S. Micali, C. Peikert, M. Sudan, and D. A. Wilson, "Optimal error correction against computationally bounded noise," in *Proceedings of the 2nd Theory of Cryptography Conference (TCC)*, pp. 1–16, 2005.

[61] F. Parvaresh and A. Vardy, "Multivariate interpolation decoding beyond the Guruswami-Sudan radius," in *Proceedings of the 42nd Allerton Conference on Communication, Control and Computing*, 2004.

[62] F. Parvaresh and A. Vardy, "Correcting errors beyond the Guruswami-Sudan radius in polynomial time," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pp. 285–294, 2005.

[63] W. W. Peterson, "Encoding and error-correction procedures for Bose-Chaudhuri codes," *IEEE Transactions on Information Theory*, vol. 6, pp. 459–470, 1960.

[64] J. Radhakrishnan, "Proof of q-ary Johnson bound," 2006. Personal Communication.

[65] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[66] M. Sipser and D. Spielman, "Expander codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.

[67] A. Smith, "Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes," Cryptology ePrint Archive, Report 2006/020, http://eprint.iacr.org/, 2006.

[68] D. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1723–1732, 1996.

[69] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, 1997.

[70] M. Sudan, "List decoding: Algorithms and applications," *SIGACT News*, vol. 31, pp. 16–27, 2000.

[71] M. Sudan, "Ideal error-correcting codes: Unifying algebraic and number-theoretic algorithms," in *Proceedings of AAECC-14: The 14th Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pp. 36–45, November 2001.

[72] A. Ta-Shma and D. Zuckerman, "Extractor codes," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3015–3025, 2004.

[73] L. Trevisan, "List-decoding using the XOR lemma," in *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pp. 126–135, 2003.

[74] L. Trevisan, "Some applications of coding theory in computational complexity," *Quaderni di Matematica*, vol. 13, pp. 347–424, 2004.

[75] J. H. van Lint, "Introduction to coding theory," *Graduate Texts in Mathematics*, vol. 86, 3rd Edition, Springer-Verlag, Berlin, 1999.

[76] J. von zur Gathen, *Modern Computer Algebra*. Cambridge University Press, 1999.

[77] L. R. Welch and E. R. Berlekamp, "Error correction of algebraic block codes," US Patent Number 4,633,470, December 1986.

[78] J. M. Wozencraft, "List decoding," *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, vol. 48, pp. 90–95, 1958.

[79] V. V. Zyablov and M. S. Pinsker, "List cascade decoding," *Problems of Information Transmission*, vol. 17, no. 4, pp. 29–34, (in Russian); pp. 236–240 (in English), 1982, 1981.