Coding Theory: Tutorial & Survey

Madhu Sudan*

Abstract

Coding theory has played a central role in the theoretical computer science. Computer scientists have long exploited notions, constructions, theorems and techniques of coding theory. More recently, theoretical computer science has also been contributing to the theory of error-correcting codes - in particular in making progress on some fundamental algorithmic connections. Here we survey some of the central goals of coding theory and the progress made via algebraic methods. We stress that this is a very partial view of coding theory and a lot of promising combinatorial and probabilistic approaches are not covered by this survey.

1 Introduction, Disclaimers

This is a hastily written article intended to go along with a 2-hour tutorial whose mandate was to introduce coding theory to theoretical computer scientists. Limited by my (lack of) knowledge, time, and energy, I have chosen to focus the survey on some of the classical algebraic results, and some recent algorithmic progress in this direction. This article is just a transcription of my notes. They lack polish, and more relevantly, rigor in many (most?) places. The hope is that they will still provide sufficiently many pointers to help the interested reader gain more information about the area.

Coding theory is the umbrella term used to cover the study of two related, but distinguishable, topics: The theory of "error correcting codes" and the mathematics behind "reliable communication in the presence of noise". The latter term is self-explanatory while the former could use some elaboration. Error-correcting codes are collections of sequences of elements from a finite set (or words over a finite alphabet) with some nice extremal properties — namely, any two words in the collection disagree on many coordinates. Thus the theory of error-correcting codes could be considered an area of combinatorial mathematics (with some natural algorithmic tasks that arise naturally). The theory of communication in the presence of noise, on the other hand often leads to information theory and/or statistics. The two aspects of coding theory enjoy a symbiotic relationship from the days of their origin.

Coding theory, and the dichotomy within, owes its origins to two roughly concurrent seminal works by Hamming [45] and Shannon [80], in the late 1940s.

Hamming was studying devices to store information and wanted to design simple schemes to protect from the corruption of a small number of bits. Hamming realized the explicit need to consider sets of words or "codewords" where every pair differs in a large number of coordinates. Hamming extracted the combinatorial underpinnings of the theory of error-correcting codes. He defined the notion of distance between two sequences over finite alphabets - now called the Hamming distance, observed this was a metric and thus led to interesting properties. This leads to the notion of distance we work with in coding theory, named the *Hamming* distance. Hamming also constructed an explicit family of codes that acheived non-trivial distance. We will see these codes shortly. Hamming's paper was eventually published in 1950.

Slightly prior to Hamming's publication, Shannon (1948) wrote an treatise of monumental impact formalizing the mathematics behind the theory of communication. In this treatise he developed probability and statistics to formalize a notion of information. He then applied this notion to study how a sender can communicate "efficiently" over different media, or more generally, channels of communication to a receiver. The channel under consideration were of two distinct types: Noiseless or Noisy. In the former case, the goal is to *compress* information at the senders end so as to, say, minimize the total number of "symbols" communicated while allowing the receiver to recover the transmitted information correctly. The noisy case, which is more important to the topic of this article, considers a channel that alters the signal being sent by adding to it a

*MIT Laboratory for Computer Science, 200 Technology Square, Cambridge, MA 02143, USA. `http://theory.lcs.mit.edu/~madhu`. This work was supported in part by NSF Career Award CCR 9875511, NSF Award 9912342, and MIT-NTT Award MIT 2001-04.

"noise". The goal in this case is to add some redundancy to the message being sent so that a few erroneous "symbols" at the receiver's end still allow the receiver to recover the sender's intended message. Shannon's work showed, somewhat surprisingly, that the same underlying notions captured the "rate" at which one could communicate over either class of channels. Even more surprisingly, Shannon's work showed that, contrary to popular belief then, that when transmitting information at a fixed "feasible" rate, longer messages were more likely to be recovered (completely correctly) than short ones - so asymptotics actually help! Shannon's methods roughly involved encoding messages using long random strings, and the theory relied on the fact that long messages chosen at random (over a fixed alphabet) tend to be *far* away from each other. The distance measure under which these strings tend to be far away were not focussed on explicitly in Shannon's work.

Shannon's and Hamming's works were chronologically and technically deeply intertwined. Technically these works complement each other perfectly. Hamming focusses on the mathematical properties behing these combinatorial objects, while Shannon creates the perfect setting for their application. Hamming focusses on combinatorial aspects, more based on an "adversarial" perspective, while Shannon bases his theory on "probabilistic" models (of error, message etc.). Hamming's results were highly constructive and he pays special attention to the small finite cases. Shannon's results were highly non-constructive (and a stellar example of the probabilistic method in action), and his theory pays speecial attention to the asymptotic, rather than finite behavior. These complementary influences leads Berlekamp [11] to characterize Shannon's work as "statistical and existential" and Hamming's work as "combinatorial and constructive". (A brief digression: The edited volume by Berlekamp [11] is a personal favorite of mine, as a source book of ideas, their history, the mood surrounding the research, as well as for the refreshingly candid opinions of the editor on a broad variety of topics. Reading, at least the editorials, is highly recommended.) It may be pointed out that the statistical/combinatorial distinction seems to point to differences in their goals, while the existential/combinatorial distinction seems to be a consequence of the state of their knowledge/capability. Shannon would have probably loved to see his results being more constructive, but the field was not yet developed enough to prove such results. (He definitely lived to see his results being transformed into more constructive ones.) Hamming, on the other hand, started with some constructive results and build the theory around these - and thus came this distinction. In the sequel we will refer to the Hamming/Shannon dichotomy, but we will use this to refer to the combinatorial/statistical distinction in their goals.

Moving on to the chronology: While Shannon's paper appeared first, he borrows one code from Hamming's construction and cites Hamming for this code. So he was certainly aware of Hamming's work. Hamming presumably was also aware by the time of publication of his work, but does not cite Shannon's work explicitly. However he cites a short (2/3rd page) article by Golay [34], who in turn cites Shannon's article (so Hamming was likely aware of Shannon's work). Both papers seem to regard the other as far away from their own work. Shannon's paper never explicitly refers to error-correcting codes or even the distance property in his main technical results — they only occur between the lines of the proofs of his main results! (He only uses Hamming's codes as an example.) Hamming, in turn, does not mention the applicability of his notion to reliable communication. He goes to the extent of enumerating all prior related works - and mentions Golay's work [34] as the only prior work on error-correcting codes.

Both works however, were immediately seen to be of monumental impact. Initially both works were cited almost equally often (with a fair number of works that would cite Hamming's paper but not Shannon's). Over time the distinctions between the two perspectives have been overwhelmed by the common themes. Further the asymptotic theory of Shannon started driving the practice of communication and storage of information. This, in turn, became the primary motivator for much research in the theory of error-correcting codes. As a result, today it is very common to see articles that ascribe origins of the entire theory to Shannon's work, cf. [49, Chapter 1], [96]. (The Handbook on Coding Theory [49], for instance, introduces Shannon's work on the first page and waits for about ten pages before mentioning Hamming's work.)

1.1 Basic definitions

We now move on to some of the basic notions of coding theory. To define codes, messages, and other such notions, let us follow a message on its path from the source to the receiver as it goes over a noisy channel and let us see what are the relevant features that emerge. (In what follows, emphasization is used to indicate that a phrase is a reserved one, at least in this article if not all of coding theory. Parenthesized portions of emphasized texts indicate portions that are typically omitted in usage.)

As already mentioned there are three entities involved in this process - the *sender*, the *receiver*, and the *noisy channel* seperating the two. The goal of the sender is to communicate one element, or *message* $\mathbf{m}$, of a finite set $\mathcal{M}$, the *message space*, to the receiver. Both sender and receiver know this space of possible messages $\mathcal{M}$. This space of messages is "large" - exponentially larger than the time we hope to spend in computing or communication at either the sender's or receiver's end.

The noisy channel is capable of communicating arbitrarily long sequences of *symbols* from a finite *alphabet* Σ. While both $\mathcal{M}$ and Σ are "finite", Σ should be thought of as small (the reader may even think of this as $\{0,1\}$ for the initial part of this article), while $\mathcal{M}$ is thought of as being significantly large. So, e.g., running times that are polynomial or even exponential in Σ are reasonable, while running times that are polynomial in $\mathcal{M}$ are typically unreasonable. To send messages over this channel, the sender and receiver agree upon the length of the sequences to be transmitted - termed *block length*, usually denoted n. Thus the space of words being transmitted over the channel is Σ^n, and is referred to as the *ambient space*. The sender and receiver agree also upon the *encoding* E, an injective map from the message space to the ambient space ($E : \mathcal{M} \to \Sigma^n$), to be used to encode the messages before transmitting. Thus the sender, to communicate the message m to the receiver, instead transmits $E(\mathbf{m})$ over the channel. The image of the encoding function $\{E(\mathbf{m})|\mathbf{m} \in \mathcal{M}\}$ is the *error-correcting code*, or simply code. To enable easy quantitative comparisons on the performance or redundancy of the code it is useful to identify the message space with a space of sequences from Σ of length k. This parameter k does not seem to have a well-defined name in the coding theory literature, so we'll start calling it the *message length*. The ratio k/n then measures the *(message) rate* of the code - a fundamental parameter in coding theory.

Now moving on to the channel, the channel inserts some noise into the transmitted symbol. Very broadly, the noise of the channel is just a map from the ambient space to itself. To make this definition slightly more interesting, we ascribe a group structure to the alphabet Σ with operator $+$. This endows the ambient space Σ^n also with a group structure naturally with $\langle a_1, \ldots, a_n\rangle + \langle b_1, \ldots, b_n\rangle = \langle a_1 + b_1, \ldots, a_n + b_n\rangle$. In these terms, we can say that the channel produces a noise vector, commonly termed *error pattern*, $\eta \in \Sigma^n$ and that the receiver receives the corruption $\mathbf{y} = E(\mathbf{m}) + \eta$ as the *received word*. The receiver now employs some *decoding* function $D : \Sigma^n \to \mathcal{M}$. Hopefully, D runs efficiently and produces $D(\mathbf{y}) = \mathbf{m}$.

The information-theoretic focus of coding theory revolves around questions of the form:

> Given a distribution $\mathcal{P}$ on the error inserted be the channel (i.e, $\mathcal{P} : \Sigma^n \to [0,1]$), what are the best encoding and decoding functions, or more formally what is the smallest *error probabability*
>
> $$\max_{E,D} \{\mathbf{E}_{m\in\mathcal{M}} [\mathbf{Pr}_{\eta\in\mathcal{P}} [D(E(\mathbf{m}) + \eta) = \mathbf{m}]]\}?$$

If the error induced by the channel can be described uniformly as a function of n, then this enables asymptotic studies of the channel. E.g., Shannon studies the case where the channel distribution was a just an n-wise Cartesian product of the distribution over individual coordinates. In this case he showed that the channel has a capacity $C_0 \in [0,1]$ such that if one picks a $C < C_0$ and $\epsilon > 0$, then for every large enough n there exists an encoding/decoding pair with rate at least C which acheives an error probability of at most ϵ. (Thus, asymptotics help.) He also showed that it is not feasible to communicate at rates larger than C_0. It is known that the error-probability drops off exponentially with n for every $C < C_0$, but the exact exponent with which this error drops of, is still not known and is one of the fundamental questions of the existential aspects of Shannon's theory. Of course, once one requires the code and the associated functions to be efficiently computable, a vast host of open questions emerge. We will not dwell on these. The texts of Blahut [14] or Cover and Thomas [18] are possibly good starting points to read more about this direction of work. Let us move on to the Hamming notions.

Recall Σ is a group - let zero be the identity element of this group.: The *(Hamming) weight* of an sequence $\eta \in \Sigma^n$, denoted $\mathbf{wt}(\eta)$, is the number of non-zero symbols in the sequence. The usage of the letter η is suggestive in that this notion is supposed to be applied to the error pattern. It is reasonable to expect (from the noisy channel) that the noise should affect relatively small number of symbols of the transmitted word and thus error patterns are restricted in their Hamming weight. The *(Hamming) distance* between two sequences $\mathbf{x}$ and $\mathbf{y}$, denoted $\Delta(\mathbf{x}, \mathbf{y})$, is simply the Hamming weight of $\mathbf{x} - \mathbf{y}$. Hamming defined this notion (explicitly) and noticed that it is a metric and thus other metric concepts could be brought into play here. For example, the notion of a Hamming *ball* of radius r around a sequence $\mathbf{y}$ is the set $\{\mathbf{x} \in \Sigma^n | \Delta(\mathbf{y}, \mathbf{x}) \leq r\}$ will be of interest to us.

Hamming introduced the notion of *error-correcting* and *error-detecting* codes and used these to motivate the concept of the *(minimum) distance* of a code. A code $\mathcal{C} \subseteq \Sigma^n$ corrects e errors if for every sequence $\mathbf{y}$ in the ambient space it is the case that there is at most one codeword in $\mathcal{C}$ in the Hamming ball of radius e around $\mathbf{y}$. (Thus if a transmission is corrupted by an error pattern of weight at most e then error-recovery is feasible, at least, if we ignore computational efficiency issues.) Using the metric properties this is equivalent to saying that the Hamming balls of radius e around the codewords are non-overlapping. Similarly, a code $\mathcal{C}$ detects e errors if no error pattern of weight at most e maps a codeword into another one. Thus the test, "Is the received vector a codeword?" is guaranteed to detect any occurence of up to e errors. Finally, the minimum distance of a code $\mathcal{C}$, denoted $\Delta(\mathcal{C})$, is the minimum Hamming distance between any pair of distinct codewords in it, i.e.,

$$\Delta(\mathcal{C}) = \min_{\mathbf{x},\mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}} \{\Delta(\mathbf{x}, \mathbf{y})\}.$$

The metric properties lead to the immediate observations: For every integer e, $\mathcal{C}$ is an e-error-correcting code if and only if it is a $(2e)$-error-detecting code if and only if it has a minimum distance of at least $2e+1$. The minimum distance of a code is another fundamental parameter associated with a code, and it makes sense to study this as a fraction of the length of the code. This quantity d/n is usually referred to as a *distance rate* of a code. Since rate is already a reserved word, this phrasing can lead to awkwardness — and we avoid it by referring to this quantity as the *relative distance*.

The fundamental questions of coding theory as arising from the Hamming theory is based on the construction of codes with good rate and good relative distance. E.g. the most elementary question is:

> Given an alphabet Σ of size q, and integers n and k, what is the largest minimum distance d that a code $\mathcal{C} \subseteq \Sigma^n$ of message length k can have?

(Note that the actual alphabet Σ itself does not really play a role in this question, once its size is fixed. So q will be an important parameter in the questions below, but not Σ.) The question raised above remains wide open for small q. When $q \geq n$, this question has satisfactory answers. Even getting with a constant (absolute) distance, while maintaining a rate close to 1, is non-trivial. Getting codes of distance 1 is trivial (the identity function does this)! Hamming points out that appending the parity of all bits gives a code of minimum distance 2 over the binary alphabet. His main result is a family of codes of minimum distance 3 that are now named the Hamming codes. Hamming also gives codes of distance 4 as part of a general result showing how to go from a code of odd minimum distance to a code of even minimum distance (by increasing the block length by 1).

The question raised above remains open even in an asymptotic sense. In this setting we look at families of codes $\mathcal{C} = \{\mathcal{C}_i | i \in \mathbb{Z}^+\}$, with $\mathcal{C}_i$ having block length n_i, where the n_i's form an increasing sequence. The rate of the family is the infimum of the rates of the codes in the family. The relative distance the family is the infimum of the relative distances of the codes in the family. The asymptotic version of the question asks:

> Given an alphabet Σ of size q and rate R, what is the largest relative distance δ for which there exists an infinite family of codes of rate $\geq R$ and minimum distance $\geq \delta$?

A family of codes is said to be *asymptotically good* if both its rate and minimum distance are greater than zero. While early results showed asymptotically good codes do exist (see below), explicit constructions of such families of codes took significantly longer. Study has since turned to the explicit relation between R and δ above and is the core question driving much of the Hamming-based theory.

Before going on to describing some of the early results of coding theory, we mention one broad subclass of codes that are essentially most of what is investigated. These are the class of *linear codes*. In these codes, one associates the alphabet Σ with some finite field of size q. Sequences of length n over Σ can now be thought of as n-dimensional vectors. If a code forms a linear subspace of the ambient space, then the code is said to be linear. While there are no theorems indicating linear codes should be as good (asymptotically) as general codes - this has been the empirical observation so far. Since these codes bring together a host of nice properties with them, study of these codes has been significantly more popular than the general case. For example, Hamming suggested the notion of a *systematic code* - codes whose coordinates can be split into k message symbols and $n-k$ check symbols. Hamming noted that every linear code can be made systematic without loss of performance. Linear codes are special in that such a code $\mathcal{C}$ can be described succintly by giving a *generator matrix* $G \in \Sigma^{k \times n}$ such that $\mathcal{C} = \{\mathbf{x}G | \mathbf{x} \in \Sigma^k\}$.

1.2 Rest of this article

The results of coding theory can be classified into one the following three broad categories (obviously with plenty of scope for further subdivisions): (1) Constructions/Existence results for codes (2) Limitations of codes (non-existence results). (3) Efficient Algorithms (for encoding and decoding). A fourth category which can be added, but whose contents would vary widely depending on the audience is (4) Applications of codes to other areas. In this article we start by describing the limits to the performance first (a non-standard choice), and using this to motivate some of the algebraic constructions. We then move on to the task of decoding. Notable omissions from this article are the non-algebraic constructions, and the work on the side of highly efficient encoding. Applications of codes to theoretical computer science abound (cf. [90, Lecture 9]) and new ones continue to emerge at a regular rate (see the thesis of Guruswami [40] for many recent connections). However we will not describe any here.

2 Bounds and constructions for codes

2.1 The random linear code

Perhaps the most classical "code" is the random code and the random linear code. Hints that the random code are likely to be quite good in performance were implicit in the work of Shannon, but he does not make this part explicit. Gilbert [33] was the first to make this explicit. Later Varshamov showed that the bounds actually hold even for random linear codes making the search space significantly smaller - singly exponential instead of being doubly exponential in the code parameters. Finally, Wozencraft reduced the complexity even further for codes of rate $1/2$ (extensions of the technique works for codes of rate $1/l$ for any integer l), making the complexity of searching for such a code simply exponential ($2^{O(n)}$ instead of $2^{n^{O(1)}}$).

To describe the construction and performance of these codes, let us introduce some notation. Let $V_q(n,r)$ denote the volume of the Hamming ball of radius r in Σ^n, where $|\Sigma| = q$. Gilbert's bound is based on the following greedy process. Pick $K = q^k$ random sequences greedily as follows: Having picked $v_1, \ldots, v_i$, pick v_{i+1} so that it does not lie within a distance of d from any of $v_1, \ldots, v_i$. When does such a vector v_{i+1} exist? Note that each of the vectors v_j rules out $V_q(n,d)$ vectors from Σ^n. But this only rules out $i \cdot V_q(n,d)$ vectors in all from Σ^n. Thus if $(K-1) \cdot V_q(n,d) < q^n$ then the greedy process can always make progress. Varshamov picks the generator matrix of the linear code in a similar greedy process. It is a simple exercise show that this process also acheives the same bound. Wozencraft's construction (personal communication in [65, Section 2.5]) is quite different, but acheives the same bounds for rate $1/2$ codes is as follows. Let $n = 2k$ and associate Σ^k with a finite field $\mathbb{F}_{q^k}$ of size q^k. For every member $\alpha \in \mathbb{F}_{q^k}$, let the code $\mathcal{C}_\alpha$ be given by $\{(x, \alpha x) | x \in \mathbb{F}_{q^k}\}$. (Note that the messages are now being viewed as elements of $\mathbb{F}_{q^k}$ while the encodings are two elements of the same field.) Wozencraft shows that one of the codes $\mathcal{C}_\alpha$ also achieves the same performance as guaranteed by the results of Gilbert and Varshamov. (Wozencraft's ensemble is actually a wider class of constructions, of which the above is one specific example. This is the example given explicitly in Justesen [51].)

To understand the bound ($q^k \cdot V_q(n,d) \leq q^n$) in slightly clearer light, let us simplify it slightly. For $0 < \delta < 1$, let

$$H_q(\delta) = \delta \log_q \left(\frac{q-1}{\delta}\right) + (1-\delta)\log_q\left(\frac{1}{1-\delta}\right).$$

Then $V_q(n, \delta n)$ is well approximated by $q^{H_q(\delta)n}$. The results of Gilbert, Varshamov (and Wozencraft) show that for every choice of n and δ, their sample space contains a code of length n, minimum distance δn, and message length $n - \log_q(V_q(n,\delta n)) \approx n(1 - H_q(\delta))$. Thus asymptotically one can obtain codes of relative distance δ of rate $1 - H_q(\delta)$. This result bound is usually termed the Gilbert-Varshamov bound, which states that there exist codes of rate R and relative distance δ satisfying

$$R \geq 1 - H_q(\delta) \quad \text{(Gilbert-Varshamov Bound)}$$

(This bound is syntactically similar to Shannon's bound for the capacity of the "q-ary symmetric channel" of error rate δ. Such a channel acts on the coordinates of the transmitted word independently, and preserves every coordinate with probaliltiy $1-\delta$ and flips it to a random different symbol with probability $\delta/(q-1)$. The Shannon capacity of such a channel is $1 - H_q(\delta)$. This is not a coincidence!)

2.2 Non-existence results for codes

Given a choice of n, k, d and q, does a q-ary code of length n, message length k and minimum distance d exist? We already know that if $n > k+?d$, then such a code does exist as proven by the probabilistic method. The main question driving much of this section is whether the performance of the random code can be beaten in an asymptotic sense, and if not — how can we prove this? In other words, what techniques can we use to prove that certain choices of code parameters are impossible to acheive. (These are the sort of results that us computer scientists are used to calling "lower bounds", except in the coding context these are really upper bounds on the rate (as a function of the relative distance).)

We will start by describing some simple ideas used to prove these upper bounds and then give pointers to state of the art results.

We will start with a simple bound due to Singleton (cf. [63]) Suppose $\mathcal{C}$ is a code of length n and message length k. Project $\mathcal{C}$ down to any $k-1$ coordinates. By the pigeonhole principle two codewords must project to the same string. Thus these codewords differed in at most $n-k+1$ coordinates. This proves that the minimum distance of the code d satisfies:

$$d \leq n-k+1 \qquad \text{(Singleton bound)}$$

The bound seems naive - yet it is tight in that some codes achieve it! Codes that match the Singleton bound are called *Maximum Distance Separable (MDS)* codes and we will see an example in the next section.

The Singleton bound is independent of the alphabet size q. Indeed the codes that acheive this use large values of q. In the following we will get bounds that improve if q is small. We start with a bound from Hamming's original paper. (Note that in the very first paper, Hamming had formalized codes to the extent that he could prove negative results!) This bound is based on the following simple observation. If we consider a code of minimum distance d, then Hamming balls of radius $d/2$ around the codewords are disjoint. Since there are at least q^k such balls, we get $q^k V_q(n, d/2) \leq q^n$. Written asymptotically, we see that if a code of rate R has relative distance δ, then:

$$R \leq 1 - H_q(\delta/2) \qquad \text{(Hamming Bound)}$$

Once again, there do exist codes that meet the Hamming bound (the exact version, not the asymptotic version). such codes are called perfect. However they don't exist for all choices of parameters.

In fact, while the Hamming bound is quite good as $\delta \to 0$, it deteriorates considerably for larger δ and ends up proving obviously weak bounds for large δ. In particular, the random codes only get a relative distance of $1-\frac{1}{q}$ in the best of cases, while the Hamming bound allows a distance of twice this quantity, which is clearly impossible (since the relative distance should not be greater than one!). This bound was improved upon by Elias (unpublished, but reported later in [81]) and independently by Bassalygo [7], building upon proofs of Plotkin [74] and Johnson [50] respectively. The principal idea behind these bounds is the following. We know that no Hamming ball in Σ^n of radius $d/2$ includes two codewords from a code of distance d. We can do slightly better: It is possible to show that any Hamming ball of radius $r \stackrel{\text{def}}{=} \frac{q-1}{q}\left(n - \sqrt{n(n - \frac{qd}{q-1})}\right)$ has more than qn codewords. This quantity r is between $d/2$ and d, and tends to d as $d/n \to 1-\frac{1}{q}$. Thus if we take the q^k balls of radius r around the codewords, we are overcounting the points of the ambient space only by a factor of at most qn. Thus we get $q^k V_q(n,r) \leq qnq^n$. Asymptotically this shows that if a q-ary code of rate R and relative distance δ exists then:

$$R \leq 1 - H_q\left(\left(1-\frac{1}{q}\right)\cdot\left(1-\sqrt{1-\frac{q}{q-1}\delta}\right)\right)$$

(Elias-Bassalygo Bound)

The Elias-Bassalygo bounds already give decent asymptotic bounds on the rate to relative distance performance of codes. (In particular, they meet the Gilbert-Varshamov bound at the two ends, i.e., $R=0$ and $R=1$.) In the interim region they are bounded away from each other, but it is hard to get a qualitative sense of the difference in this regime. However a closer look at the two endpoints does reveal a qualitative difference. For instance, if we fix an absolute distance d and ask what is the limit of $n-k$ as $n,k \to \infty$, then the Hamming/Elias-Bassalygo bounds require $n-k \geq d/2 \log_q n$ while the Gilbert-Varshamov bound acheives $n-k \leq d \log_q n$ - so these bounds are only off by a factor of 2 (in some sense), but it turns out the non-existence result bound is the right one (and the Gilbert-Varshamov bound is not!).

On the other hand, at the other end, if we consider codes of relative minimum distance $1-\frac{1}{q}-\epsilon$ and ask what is the largest rate they can acheive: The Elias-Bassalygo bound shows that $R \leq O(\epsilon)$, while the Gilbert-Varshamov bound only acheives $R \geq \Omega(\epsilon^2)$. So here again the two bounds are qualitatively off from each other. In this case, it turns out the Elias-Bassalygo bound is the one that is too weak. A better bound does exist in the literature — termed the linear programming bound. This bound was introduced by Delsarte [19] who showed that a certain linear program on n variables with $O(n)$ inequalities could be used to give an upper bound on the message length of a code code of minimum distance d. (This linear program was in turn inspired by a series of identities due to MacWilliams [62] which establish some quantitative relationships between a linear code and its "dual".) Going from the linear program to an asymptotic bound on the performance of a code turns out to be highly nontrivial. Such an analysis (not known to be tight) was given by McEliece, Rudemich, Rumsey and Welch [68] which shows, in our example, that a code of minimum distance $1-\frac{1}{q}-\epsilon$ has rate at most $O(\epsilon^2 \log \frac{1}{\epsilon})$ and thus the Gilbert-Varshamov bound is almost tight at this extreme. A complete description of the analysis leading to the [68] bound can be found in the text of van Lint [59]. Levenshtein [57]'s article on this subject is a source of more information.

2.3 Codes

We will introduce various codes to the reader in approximately the historical order, also constrasting them with various bounds acheived so far. All codes of this section are linear unless otherwise noted. The associated alphabet will be $\mathbb{F}_q$, the field of cardinality q. All vectors will be row vectors (standard convention of coding theory.) Almost all beat the Gilbert-Varshamov bound at some point or the other.

Hamming codes. These are linear codes obtained as follows. For any integer l, consider the $q^l \times l$ matrix H' whose rows include all l-dimensional vectors over $\mathbb{F}_q$. Delete the row corresponding to the all zeroes vector from H'. Next delete scalar multiples from H': I.e., if $v_i = \alpha v_j$ for some $\alpha \in \mathbb{F}_q$, then delete one of the rows v_i or v_j arbitrarily. Continue this process till H' has no scalar multiples left. This leaves us with a $n \times l$ matrix, let us call it H, with $n = \frac{q^l - 1}{q-1}$. The Hamming code of length n is the collection of vectors $\{x \in \mathbb{F}_q^n | xH = 0\}$. Note that this forms a linear subspace of $\mathbb{F}_q^n$. The message length equals $n - l \approx n - \log_q n$. It can be established fairly easily that the minimum distance of this code is at least 3 (after some reasoning this corresponds to establishing that no two rows of H are linearly dependent). The Hamming codes are *perfect* in that they meet the Hamming bound exactly. The earlier mentioned paper of Golay mentions two other codes (one binary and one ternary) which meets the Hamming bound. After a significant amount of work, van Lint [58] and Tietavainen [94], eastablished that these were the only perfect codes!

(Generalized) Reed Solomon codes. These codes, due to Reed and Solomon [76], are defined for $q \geq n$. Let $\alpha_1, \ldots, \alpha_n$ be n distinct elements of $\mathbb{F}_q$. The Reed Solomon code of message length k, with parameters $\alpha_1, \ldots, \alpha_n$ is defined as follows: Associate with a message $\mathbf{m} = \langle m_0, \ldots, m_{k-1} \rangle$ a polynomial $M(x) = \sum_{j=0}^{k-1} m_j x^j$. The encoding of $\mathbf{m}$ is the evaluation of M at the n given points i.e., $E(\mathbf{m}) = \langle M(\alpha_1), \ldots, M(\alpha_n) \rangle$. By construction, the Reed Solomon codes have message length k and block length n. Their minimum distance is $n - k + 1$ which is an immediate consequence of the fact that two degree $k-1$ polynomials can agree at no more than $k-1$ points. Thus the Reed Solomon codes give a simple class of codes that meet the Singleton bound. Such codes are called *Maximum Distance Seperable (MDS)* codes. These are essentially the only such codes that are known.

BCH and alternant codes. These codes are named after their co-discoverers. Hocquenghem [47] and Bose and Chaudhari [17] proposed these binary codes essentially around the same time as the Reed-Solomon codes were proposed. We won't describe the codes explicitly here, but instead describe a slightly more general class of codes called alternant codes. Essentially these codes can be obtained from Reed Solomon codes as we describe next. For n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ (over some field $\mathbb{F}$), let $\mathbf{x} * \mathbf{y}$ be the coordinate wise product of $\mathbf{x}$ and $\mathbf{y}$. Pick a finite field $\mathbb{F}_{2^l}$ such that $2^l \geq n$ and $\alpha_1, \ldots, \alpha_n$ distinct from $\mathbb{F}_{2^l}$ and let $\mathcal{C}_1$ be the Reed Solomon code of dimension k over this field. Now, pick a vector $\mathbf{v} = \langle v_1, \ldots, v_n \rangle$ with the $v_i \in \mathbb{F}_{2^l} \setminus \{0\}$. Consider the code $\mathcal{C}_2 = \{\mathbf{v} * \mathbf{c} | \mathbf{c} \in \mathcal{C}_1\}$. Note that $\mathcal{C}_2$ has the same message length and distance $\mathcal{C}_1$. Now take the intersection of $\mathcal{C}_2$ with $\mathbb{F}_2^n$ (i.e., take only the binary vectors from this code). Often it is possible to make non-trivial choices of vectors α and $\mathbf{v}$ so that the resulting code has large dimension. BCH codes are one such family of codes. The BCH codes yield codes of block length n, minimum distance d and dimension $k = n - ((d-1)/2) \log_2 n$ and thus asymptotically match the Hamming/Elias-Bassalygo bounds. described above. (In particular, these codes generalize the Hamming codes.)

Reed-Muller codes. These codes are named after Muller [69], its discoverer, and Reed [75] who gave a decoding algorithm for these codes. Appropriately generalized, these codes are based on evaluations of multivariate polynomials. Consider a l-variate domaing over a q-ary field. Consider the space of all polynomials of degree at most m over this field, subject to the condition that no variable takes on a degree of q or more. If $m < q$, then this is a vector space of dimension $\binom{m+l}{m}$. For $m \geq q$, the dimension of this space does not have a closed form, but can be easily lower bounded by at least $\binom{l}{m}$. We identify this space with the vector space of messages and let their encodings be their evaluations at all points in the space. Thus we get $n = q^l$. The relative distance of these codes is at least $1 - \frac{m}{q}$ if $m < q$, and at least $q^{-m'}$ where $m' = \lceil m/q \rceil$ for $m \geq q$. (The first part follows from the familiar Schwartz-Zippel lemma in theoretical computer science. Of course, the Reed-Muller codes predate this lemma by 25 years.) The parameter m above (degree) is referred to as the order of the Reed-Muller code.

Hadamard codes. Reed-Muller codes of order one are special in coding theory since they form a subclass of the "Hadamard" codes or simplex codes. Binary Hadamard codes are obtained from integer $n \times n$

matrices H with entries from $\{+1, -1\}$ that are self-orthogonal (i.e., $H^T H = nI$). The rows of this matrix (interpreted as binary vectors) make a binary code (not necessarily linear) in which every pair of codewords differ from each other in half the places. Codes over a q-ary alphabet with a relative distance of $1 - 1/q$ are what we term as Hadamard codes.

Algebraic-geometry codes. No description of algebraic coding theory is complete without mention of the famed "algebraic-geometry codes" or the "geometric-goppa codes". These codes may be thought of as codes obtained by starting with Reed-Muller codes of some order and then removing a large number of coordinates (or projecting onto a small number of coordinates). This process may identify many codewords. A very careful choice of the coordinates onto which one projects, based on deep insights offered by algebraic geometry, lead to codes over q-ary alphabets of dimension k and block length n, with minimum distance being at least $n - k - \frac{n}{\sqrt{q}-1}$, when q is a square. The most interesting aspect of these codes is that over constant sized alphabets, these codes can beat the Gilbert-Varshamov bound with a rate and minimum distance bounded away from 1. (Note that all other codes of this section also beat the random codes - however they either require a large alphabet (Reed Solomon codes) or they do better only for rate being $1 - o(1)$ (Hamming codes, BCH codes), or they do better only for rate being $o(1)$ (Hadamard codes).

Concatenated codes. Concatenated codes don't refer to any fixed family of codes, but rather to codes obtained by a certain mechanism for combining two families of codes. This construction and some immediate applications are extremely useful and so we describe these in this section. Suppose we have two codes: $\mathcal{C}_1$ of length n_1, dimension k_1 and distance d_1, and $\mathcal{C}_2$ of length n_2, dimension k_2 and distance d_2. Suppose further that the alphabet of the second code is q and the alphabet of the first code is carefully chosen to be q^{k_2}. We call the first code the outer code and the second one the inner code. Thus the two codes are not in a symmetric relationship. In particular the number of codewords of the inner code correspond exactly to the number of letters in the outer alphabet. Then this gives a new way to encode messages of the outer code. First encode it using the outer code. Then encode each letter of the ensuing codeword in the outer alphabet using the encoding provided by the inner code. It is easy to prove that this leads to a q-ary code of length $n_1 n_2$ with message length $k_1 k_2$ of minimum distance at least $d_1 d_2$.

This process was proposed by Forney [27], who noted that if one used Reed Solomon codes as the outer code and searched for the best possible inner code (in polynomial time), then one gets a code with very good error-correcting properties. In particular for appropriate choice of parameters one get asymptotically good codes that are constructible in polynomial time — the first such construction! Forney did not mention such a result (of the Hamming type) explicitly. Instead he cast his results in the Shannon model (correcting random errors) and in this setting Forney's results were even more impressive. He gave a decoding algorithm for concatenated codes by a natural combination of the inner and outer decoding - and using this got an algorithmic version of Shannon's theorem - where both the encoder and decoder worked in polynomial time and offered exponentially small probability of error! In some senses Forney closed two of the most significant open questions in coding theory - one in the Shannon style and one in the Hamming style, using a very elementary but deeply insightful idea.

These ideas were built upon further by Justesen who suggested the use of n_1 different inner codes, one for each coordinate of the outer code. He noticed that it sufficed if most of the inner codes exhibited good distance. He then applied this idea with the Wozencraft ensemble as the n_1 inner codes to get a fully explicit (logspace constructible?) family of codes that were asymptotically good. More recently Katsman, Tsfasman and Vladuts [53] concatenated algebraic geometry codes with Hadamard codes to get binary codes of rate $O(\epsilon^3)$ of relative distance $\frac{1}{2} - \epsilon$ giving the best codes at such high relative distance.

Brief history of developments. The above codes were mentioned in roughly chronological order, but with a few inversions. Here we clarify the exact order and some interim history. Hamming had already introduced linear-algebra (at least over $\mathbb{F}_2$) in the context of coding theory. Slepian [85] was the first to systematically explore the role of linear algebra in coding and thus introduce linear error-correcting codes as a field of their own.

The first truly "algebraic" codes were the Reed-Muller codes - and this is where field multiplication seems to have been used in a strong form. The Reed-Solomon and BCH codes were all developed around the same time. The description of BCH codes was not like the description here and it was not known/felt to be related to RS codes in any way. BCH codes were an immediate hit, since they worked over the binary alphabet and generalized Hamming codes in the right way. Peterson analyzed these codes and gave a polynomial decoding algorithm - a highly non-trivial achievement. Later Gorenstein and Zierler [38] extended the BCH codes to non-binary alphabets and also extended the decoding algorithm to these codes. They also noticed that Reed-Solomon codes exhibited an interesting duality property (under appropriate choice and ordering of the α_i's) and this allowed them to be treated as a subfamily of BCH codes. Since then the study of RS codes and description has been merged into that of the BCH codes in the common texts (a pity, in my opinion). RS codes do not seem to have enjoyed the same attention as combinatorial objects because of their large alphabet. Forney's work changed some of this by finding use for codes over large alphabets. Also for some storage media it made sense to view the channel as naturally offering a large alphabet. People also noticed that large alphabets provide a natural way to cope with bursty error patterns. So Reed-Solomon codes eventually became quite popular and got widely deployed - more so than any other algebraic code family (perhaps excluding the parity code!).

We move on to the algebraic geometry codes. These codes were an "invention" of V.D. Goppa [37] — who suggested this route way before it was clear that these codes could offer any interesting possibilities. It took a significant amount of time and effort before the full potential of this idea was realized. The full potential was realized in the work of Tsfasman, Vladuts, and Zink [95] which builds upon some deep work in algebraic number theory. Proofs of the properties of these codes is being simplified significantly nowadays, with the works of Garcia and Stichtenoth [30] being major steps in this direction. For the time being, the best references are [87, 48].

Non algebraic codes.

As already mentioned we won't be covering such codes. Yet we should mention that there is an extremely elegant body of work based on binary codes whose parity check matrices are of "low-density" (i.e., have a small number of non-zero entries). These codes, called Low Density Parity Check codes (LDPC codes), were innovated by Gallager [28] who showed that a random matrix with this property produced codes that were asymptotically good, and allowed for efficient (linear time) decoding for a constant fraction of errors. This result completely belies the intuition that suggests that for a code to be decodable efficiently it must also be constructible efficiently. Subsequent work by Tanner [92] started to make a lot of these constructions explicit. Finally the works of Sipser and Spielman [84] and Spielman [86] culminated in explicitly constructible linear time encodable and decodable codes in this family whose constructions were fully explicit. LDPC codes remain an active area of study today with the works of Luby et al. [61, 60] and Richardson et al. [77, 78] showing the potential that these codes have for acheiving Shannon capacity of various channels even at small values of the block length. The work of Barg and Zemor [6] shows how these codes lead to linear time decodable codes that achieve Shannon capacity!

3 Decoding algorithms

3.1 Algorithmic Tasks

There are three essential algorithmic tasks associated with error-correcting codes: (1) Construction of the code; (2) Encoding and (3) Decoding. The first of these tasks was implicitly dealt with in the previous section. For linear codes, the task of encoding takes only $O(n^2)$ time, once the code is specified by its generator matrix. So this part is not too hard. Faster algorithms can, should, and have been sought - but we will not dwell on this part here. A somewhat harder task - with naive algorithms running in exponential time - is the task of correcting errors, or decoding.

The decoding problem is not entirely trivial to formulate. Numerous variants of this problem exist in the literature. The first question raised by these problems is: How should the code be specified? If it is part of the input, almost immediately hardness results crop up. The first such results were due to Berlekamp, McEliece and van Tilborg [13]. Subsequently many variants have been shown to remain hard — e.g., approximation [2, 20], to within error bounded by distance [23], fixed number of errors [21]. (See also the survey by Barg [5].) In this section we will not deal with such problems, but focus on this problem for fixed classes of (algebraic) codes.

Even after we fix the code (family) to be decoded, the decoding problem is not completely fixed. The literature includes a multitude of decoding problems: Maximum likelihood decoding (MLD), Nearest codeword problem (NCP), Bounded Distance Decoding (BDD), Unambiguous decoding, List decoding etc. Every pair of these problems differ from each other (sometimes very slightly). Here we attempt to fix these terms. In all problems the input is a vector $\mathbf{r}$ from the ambient space Σ^n. In all cases the output is a (possibly empty) set of messages $\mathbf{m}$ (or equivalently, a codeword $\mathbf{c}$). The main question is what is expected of $\mathbf{c}$. The following criteria distinguish the above problems:

MLD Here we are given a distribution $\mathcal{D}$ on the error patterns and we wish to output a single codeword $\mathbf{c}$ that maximizes the probability of $\mathbf{r} - \mathbf{c}$.

NCP Here we wish to output a single codeword $\mathbf{c}$ that is closest to $\mathbf{r}$ in Hamming distance. (Note that this is essentially the MLD problem when the distribution on error is that of the q-ary symmetric channel - i.e., error is i.i.d. on coordinates, with the error on any coordinate being equally likely to be any non-zero symbol).

List decoding Here again an error bound e is fixed and the goal is to output the set of all codewords within Hamming distance e from the received vector.

Bounded Distance Decoding Here along with the code an error bound e is fixed and one is expected to output a single vector $\mathbf{c}$ within Hamming distance e from $\mathbf{r}$ if one exists or an empty set otherwise.

Unambiguous decoding This is the Bounded Distance Decoding problem with e set to $(d-1)/2$ where d is the minimum distance of the code $\mathcal{C}$.

Informally, the problems roughly decrease in complexity as one goes down the list (though of course this depends on the choice of the error parameter e in the 3rd and 4th problems above). The classically studied problem is that of unambiguous decoding. This is the problem that Peterson [73] solved in polynomial time for BCH codes, and the problem that led to the famed Berlekamp-Massey algorithm. We will discuss such an algorithm in the following section. Attempting to solve either the bounded distance decoding problem, or the potentially harder list decoding problem for larger error values was not really pursued actively till recently. Recently the author, and several others have found list decoding algorithms for a varied family of codes, that solve the problem for e larger than $d/2$. We will discuss later. The MLD or NCP problem have not really been solved exactly in any non-trivial cases. (Two trivial cases are the Hamming code, where it is possible to enumerate all error-patterns in polynomial time, and the Hadamard code where it is possible to enumerate all codewords in polynomial time.) But reasonably good approximations to this problem can be solved efficiently, as was shown by Forney [27]. We will discuss some of this later.

3.2 Unambiguous decoding for algebraic codes

Unambiguous decoding algorithms for almost all algebraically (or number-theoretically) defined codes including BCH, Reed-Solomon, Reed-Muller, Algebraic-Geometry codes are now known. Furthermore these algorithms have been unified to the extent that a single decoding algorithm, abstracted appropriately, actually works for all these cases. Such an algorithm is described in [48, Section 6]. Here we will be slightly less ambitious and describe an algorithm that works for all code families above except the Reed-Muller case. We will describe the algorithm in general and show how it applies to the Reed Solomon case. This algorithm is essentially from the thesis of Duursma [24] and is attributed to Duursma, Kötter [54] and Pellikaan [72].

Recall that $\mathbf{x} * \mathbf{y}$ denotes the coordinatewise product of $\mathbf{x}$ and $\mathbf{y}$. For sets of vectors A and B, let $A * B = \{\mathbf{x} * \mathbf{y} | \mathbf{x} \in A, \mathbf{y} \in B\}$.

Definition 1 *A pair of linear codes A, B are said to be a t-error-correcting pair for a linear code C of block length n over Σ if (1) $A * C \subseteq B$, (2) dimension of A is at least $t+1$, (3) the minimum distance of A is at least $n - \Delta(C) + 1$, and (4) the minimum distance of B is at least $t+1$.*

The above definition is almost entirely consisting of basic notions from linear codes, except for the first condition which involves "higher algebra" (and sets it up so that quadratic constraints may be made to look linear).

Theorem 2 *For $t \le n$, let A, B, C be linear codes over F of block length n, such that (A, B) form a t-error correcting pair for C. Then, given generator matrices for A, B, C and a "received word" $r \in \mathbb{F}^n$, a codeword $c \in C$ that differs from r in at most t locations can be found in $O(n^3)$ time.*

Proof: The algorithm for finding the codeword $\mathbf{c}$ works in two steps: (Step 1) Find $\mathbf{a}, \mathbf{b}$ satisfying:

$$\mathbf{a} \in A - \{\mathbf{0}\}, \mathbf{b} \in B \text{ and } \mathbf{a} * \mathbf{r} = \mathbf{b} \tag{1}$$

(Step 2) Let $I \stackrel{\text{def}}{=} \{i : a_i \neq 0\}$. Find $\mathbf{c} \in C$ s.t. $c_i = r_i$ for every $i \in I$, if such a c exists, and output it.

Let G_A, G_B and G_C be the generating matrices of A, B and C respectively. Let k_A, k_B, k_C denote their dimensions. Then Step 1 above amounts to finding vectors $x_A \in \mathbb{F}^{k_A}$ and $x_B \in \mathbb{F}^{k_B}$ such that $(G_A \cdot x_A)_i r_i = (G_B \cdot x_B)_i$. This amounts to solving a linear system with n equations and $k_A + k_B \le 2n$ unknowns and can certainly be solved using $O(n^3)$ field operations. Similarly, Step 2 amounts to finding $x_C \in \mathbb{F}^{k_C}$ such that $(G_C \cdot x_C)_i = r_i$ for $i \in I$. Again these form at most n linear equations in $r_C \le n$ unknowns and can be solved in $O(n^3)$ time. This yields the claim about the running time.

The correctness is proved by three claims. (We enumerate the claims below but omit the details of the proof. The reader is referred to [36] for full details.)

We first claim that if $\mathbf{r}$ is in fact close to some codeword $\mathbf{c}$, then a pair satisfying Equation (1) must exist — a claim that follows from a rank argument (the linear system is homogenous and has greater rank than number of constraints). Next, we claim that for any pair $\mathbf{a}, \mathbf{b}$ satisfying Equation (1), every codeword $\mathbf{c}$ that is close to $\mathbf{r}$ will satisfy the conditions required in Step 2. This claim follows from the fact that $A * C \subseteq B$ and B has large minimum distance. Finally, we claim that for any pair $\mathbf{a}, \mathbf{b}$, there is at most one solution to Step 2. This step follows from the fact that C has large minimum distance. This completes the proof that $\mathbf{c}$ is the unique answer output by our algorithm. ▌

The above paradigm for decoding unifies most of the known (unique) decoding algorithms. For example, the Welch-Berlekamp algorithm [98, 12] (as described in [32]) can be obtained as a special case of the above. To decode a Reed-Solomon code C of dimension k (with the message space being all polynomials of degree less than k), we use as the code A the Reed Solomon code consisting of all polynomials of degree at most $\lfloor \frac{n-k}{2} \rfloor$. We note that $A*C$ is contained in the space B of polynomials of degree at most $\lfloor \frac{n+k}{2} \rfloor$. Note that A, B now satisfy all conditions required of a t-error correcting pair for $t = \lfloor \frac{n-k}{2} \rfloor$ and thus give a t-error correcting algorithm.

3.2.1 Decoding concatenated codes

In this section we describe an extremely elegant polynomial time algorithm for decoding concatenated codes under fairly minimal assumptions about the codes involved in concatenation. Specifically, the assumptions are that the alphabet size of the outer code is polynomially bounded in the block length of the concatenated code, and that the outer code has a polynomial time "errors and erasures decoding" algorithm. The latter assumption needs some clarification and we will get to this shortly.

First, let us recall the notion of concatenation. If $\mathcal{C}_1$ is a code of block length n_1, message length k_1 and minimum distance d_1 over an alphabet of size q^{k_2}; and $\mathcal{C}_2$ is a code of block length n_2, message length k_2 and minimum distance d_2 over an alphabet of size q; then $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2$, the concatenation of $\mathcal{C}_1$ and $\mathcal{C}_2$ is a code of block length $n = n_1 n_2$, message length $k = k_1 k_2$ and minimum distance $d = d_1 d_2$ over an alphabet of size q. This code is obtained by encoding a message (coming from a space of size $(q^{k_2})^{k_1}$) first using the encoding function associated with $\mathcal{C}_1$, and then encoding each coordinate of the resulting word using the encoding function associated with $\mathcal{C}_2$.

Going on to the decoding problem: Let us use Σ to denote the q-ary alphabet of $\mathcal{C}_2$ and $\mathcal{C}$, and Γ to denote the q^{k_2}-ary alphabet of $\mathcal{C}_1$ (as also the message space of $\mathcal{C}_2$). Let E_1 and E_2 denote the encoding functions used by $\mathcal{C}_1$ and $\mathcal{C}_2$. We receive a vector $\mathbf{r} = \langle \mathbf{r_1}, \ldots, \mathbf{r_{n_1}} \rangle$, where $\mathbf{r_i}$ are elements of Σ^{n_2}. A natural algorithm to decode $\mathbf{r}$ is the following two-stage process. (1) First, for each $i \in \{1, \ldots, n_1\}$, compute $y_i \in \Gamma$ that minimizes the Hamming distance between $\mathbf{r_i}$ and y_i. (2) Let $\mathbf{y} = \langle y_1, \ldots, y_n \rangle \in \Gamma^{n_1}$, Compute the nearest codeword $\mathbf{c} \in \mathcal{C}_1$ to $\mathbf{y}$ and output $\mathbf{c}$.

Let us first examine the computational complexity of the above algorithm: (1) The first step runs in time polynomial in Γ by a brute force implementation - and this is not an unreasonable running time. (2) The second step is a decoding step of the code $\mathcal{C}_1$. So if we assume that the outer code has a small alphabet and a fast, say, unambiguous decoding algorithm, then the above algorithm runs efficiently. Now let use examine the question of how many errors does the above algorithm correct? Say the transmitted word is $\mathbf{m}$ and let $E_1(\mathbf{m}) = \langle z_1, \ldots, z_{n_1} \rangle$. For the second stage to output $E_1(\mathbf{m})$, we need to ensure that $y_i \neq z_i$ for less than $d_1/2$ values of i. In turn this forces us to make sure that on at most $d_1/2$ of the blocks we have a Hamming distance of more than $d_2/2$ between $E_2(z_i)$ and $\mathbf{r_i}$. All in all, this quick calculation reveals that the algorithm can correct essentially $d_1 d_2/4 = d/4$ errors. This is a reasonable achievement under minimal assumptions, but does not yet achieve the goal of unambiguous decoding (which would like to correct $d/2$ errors).

We now describe the elegant Generalized Minimum Distance decoding algorithm of Forney [27] which accomplishes unambiguous decoding under slightly stronger assumptions. Forney considers the following issue. When decoding the inner codes on the various coordinates, one gets more information than just the fact that the outer symbol could have been y_i in the ith coordinate. We actually get some reliability information telling us that if y_i is the symbol, then a certain number of errors must have occured, and if it is not, then a (larger) number of errors must have occured. What if we simply throw out the coordinates where in either case a large number of errors have occured? This should help the outer decoding process in principle. Forney makes this assumption precise by using the notion of an error and erasure decoding algorithm - an algorithm for which symbols are allowed to be erased and such an erasure only accounts for half an error. This notion is formalized below.

Definition 3 *For a code $\mathcal{C}_1 \subseteq \Gamma^{n_1}$, of minimum distance d_1, we say that an algorithm $\mathcal{A}$ is an unambiguous error and erasure decoding algorithm if it takes as input $\mathbf{y} \in (\Gamma \cup \{?\})^{n_1}$, (where ? is a special symbol* *not in* Γ*) and outputs a codeword* $\mathbf{c} \in \mathcal{C}_1$ *such that* $2t + s < d_1$*, where* $s \stackrel{\text{def}}{=} |\{i | y_i = ?\}|$ *and* $t \stackrel{\text{def}}{=} |\{i | c_i \neq y_i\}|$*, if such a codeword exists.*

Forney's algorithm using such an unambiguous decoder is now the following (on input $\mathbf{r} \in \Sigma^{n_1 n_2}$).

Stage 1 For $i \in \{1, \ldots, n_1\}$ do: Let y_i be the message of Γ minimizing $e_i \stackrel{\text{def}}{=} \Delta(E_2(y_i), \mathbf{r_i})$.

Stage 2 For $\tau \in \{0, \ldots, n_2\}$ do: Let $y'_i = y_i$ if $e_i \leq \tau$ and let $y'_i = ?$ otherwise. Use an unambiguous error and erasure decoder to decode $\mathbf{y}'$. If the received codeword is close enough to $\mathbf{r}$ in Hamming distance, then output $\mathbf{c}$, else continue.

Forney analyzes the algorithm and shows, somewhat surprisingly that it decodes upto half the minimum distance of the concatenated code. We conclude by observing that the algebraic decoding algorithm of the previous section naturally provides an unambiguous errors and erasures decoding algorithm, and so this assumption is a very reasonable one.

Before moving on, we make a small aside. Forney [27] actually used the algorithms above (the naive one, or the more sophisticated GMD) to actually achieve Shannon capacity of say the binary symmetric channel with polynomial time algorithms. This is a significantly stronger result than the unambiguous decoder given above since it needs to correct about twice as many errors as the number guaranteed above. However in this case the errors are random. How does he take advantage of this fact? Forney notices that when errors occur at random wiTh say probability p, then most of the inner blocks have at most the $(p+\epsilon)$-fraction of errors. So the inner decoding actually gets rid of most errors. Hence the outer code, and decoder only need to correct a small number of errors — not proportional to p, but rather a function of ϵ. So one does not lose much in the outer decoding, even though the outer decoding algorithm (and the loss in performance due to the fact that it corrects only half as many errors as the outer distance) do not adversely affect the overall performance. Careful formalization of this intuition and analysis lead him to the algorithmic version of Shannon's coding theorem. (As should be evident, Forney's thesis [27] is another personal favorite - again reading this monumental work is highly recommended. Some of the sketchy arguments above are also described in more details in my notes on my webpage [90].)

3.3 List decoding

We now move beyond the scope of unambiguous decoding. The main motivation of this part is to just solve the bounded distance decoding problem for $e > d/2$, where d is the minimum distance of the code we are given. However, thus far, we have only found algorithms that solve this problem by enumerating all codewords lying within a Hamming ball of radius e around the received vector, and thus solving the seemingly harder *list-decoding problem*.

The notion of list-decoding was introduced by Elias [25] and Wozencraft [99] who considered this as a weak form of recovery from errors, in the Shannon model. In this weak form, error recovery was considered successful if the decoding algorithm output a small set of codewords that included the transmitted message. This notion led to tighter analysis of notions such as error-probability and the error-exponent of a channel etc. It may also be noted that the Elias-Bassalygo upper bound (on the rate of a code given its minimum distance) revolves around the combinatorial notion of list-decoding. Thus list-decoding was evidently a useful notion in both the Shannon-based and the Hamming-based theories of error-correcting codes.

However, till recently no non-trivial algorithmic results were achieved in this direction. A work of Goldreich and Levin [35] re-exposed the importance of this concept to theoretical computer science (they exploited this notion to get hardcore predicates for one-way functions). They also construct a very non-trivial list-decoder for a Hadamard code - unfortunately this result is non-trivial only in a non-standard model of computation. We won't get further into a description of their result. The first result for list-decoding in the standard models is due to this author [88]. This work built upon previous work of work of Ar, Lipton, Rubinfeld and Sudan [1] to get a list-decoding algorithm for Reed-Solomon codes. Since this algorithm was discovered, it has been extended to many classes of codes, improved significantly in terms of efficiency and the number of errors it corrects, and used to design list-decoding algorithms for other families of codes. Below we describe this algorithm and some of the subsequent work.

3.3.1 List-decoding of Reed Solomon codes

The problem we are interested in is the following: Given vectors $\alpha, \mathbf{y} \in \mathbb{F}_q^n$ find all polynomials of degree less than k that agree with the point pairs (α_i, y_i) for many values of i. The unambiguous decoding algorithm for this problem could be explained in terms of the following idea: Suppose the polynomial being encoded is M. First construct an algebraic explanation for the error points, by constructing a polynomial $E(x)$ such that $E(\alpha_i) = 0$ if there is an error at location i. Next notice that for every coordinate i, the following condition holds: $y_i = M(\alpha_i)$ "or" $E(\alpha_i) = 0$. The logical "or" can be arithmetized using simple multiplication, and we get for every coordinate i, $E(\alpha_i) \cdot (y_i - M(\alpha_i)) = 0$. Thus the unambiguous decoding algorithm actually finds an algebraic relation in the (x, y)-plane which is satisfied by all the points (α_i, y_i), and then factors the algebraic relation to extract the polynomial M.

The algorithm of [1, 88] implements this idea and works in two phases: (1) Find a non-zero bivariate polynomial $Q(x, y)$ of small total degree such that $Q(\alpha_i, y_i) = 0$ for all $i \in \{1, \ldots, n\}$. (2) Factor the polynomial $Q(x, y)$ into irreducible polynomials and report every polynomial M of degree less than k such that $y - M(x)$ is a factor of $Q(x, y)$ and $y_i = M(\alpha_i)$ for sufficiently many coordinates $i \in \{1, \ldots, n\}$.

The first thing to notice about the above algorithm is that once the parameters are fixed, it can be implemented in polynomial time. The first step just amounts to finding a non-trivial solution to a system of homogenous linear equations. The second step is non-trivial, but fortunately solved due to the (independent) works of Grigoriev [39], Kaltofen [52], and Lenstra [56] who gave polynomial time algorithms to factor multivariate polynomials. (Of course, the seminal work of Berlekamp [10] lies at the core of all these results.)

Now we think about correctness. Ar et al. [1] use a standard result from algebra (Bezout's theorem in the plane) to note that that if Q is a polynomial of total degree D and M has degree k and $Q(a, b) = a - M(b) = 0$ for more than Dk pairs (a, b), then $y - M(x)$ must divide (and hence be an irreducible factor of) $Q(x, y)$. So the question reduces to the following: How small can we make the degree of Q? Turns out we can assume D is at most $2\sqrt{n}$, as pointed out in [88]. This is seen easily via a counting argument: A bivariate polynomial of total degree $2\sqrt{n}$ has more than n coefficients, while the conditions $Q(\alpha_i, y_i) = 0$ only give n homogenous linear constraints. Thus a non-trivial solution does exist to this linear system. Putting the above together, we find the algorithm described above gives a Reed-Solomon decoding algorithm that can recover the polynomial M provided the number of agreements (non-error locations) is at least $2k\sqrt{n}$!

Is this an improvement? Not necessarily! In fact for some choice of code parameters the above algorithm, as analyzed above, is even worse than the unambiguous decoding algorithm. But that is just because we have been sloppy in our analysis. If we are careful and pick the right set of coefficients of Q to be non-zero, then at the very least we don't do worse that the unambiguous decoding algorithm (in particular if interpreted properly this algorithm is a strict generalization of to the Welch-Berlekamp algorithm). But for other choices of code parameters this algorithm does significantly better. To see such a choice consider a case where $k = n^{1/3}$. In such a case, the number of errors the algorithm can decode from is $n - O(n^{5/6}) = n - o(n)$, while the unambiguous decoder could not allow more than $n/2$ error. So in this kind of a scenario, this algorithm can recover from errors, even when the amount of error is far larger than the true information. A careful analysis in [89] shows that this list-decoding algorithm can do better than the unambiguous decoding algorithms provided $k < n/3$. Even better results are known from subsequent work and we describe some of these in the sequel.

3.3.2 Further results in list-decoding

List-decoding in algebraic settings. Since the discovery of the list-decoding algorithm for the Reed Solomon codes, list-decoding algorithms for a number of other families of codes have been discovered. Many of these work in the spirit of the algorithm described above. The first such algorithm was for algebraic-geometry codes due to Shokrollahi and Wasserman [82]. Later Goldreich, Ron, and Sudan [36] gave an algorithm decoding a number theoretic analog of the Reed Solomon code that they call the Chinese Remainder Theorem (CRT) code, (These codes are studied in the coding theory literature under the name Redundant Residue Number System (RRNS) codes [64].) Finally, Guruswami, Sahai, and Sudan [44] showed how to unify the above codes and algorithms using an "ideal-theoretic framework" and how the above-mentioned algorithms could be expressed as special cases of this ideal-theoretic algorithm. (Unfortunately, we are not yet aware of any interesting ideal-theoretic codes other than the ones already mentioned!)

New paradigms for list-decoding. The above results show how to adapt a paradigm for a decoding algorithm to work for different classes of codes. A second strain of results show how to use the Reed-Solomon decoding algorithm can be used as a subroutine to construct list-decoding algorithms for other families of codes. These results are highly non-trivial in that the resulting algorithms (and sometimes also the codes) are very distinct from the algorithm described above and potentially generate new paradigms in decoding algorithms. One such result, due to Arora and Sudan [3] with improvements by Sudan, Trevisan and Vadhan [91], produces a list-decoding algorithm for Reed-Muller codes. This result is a culmination of a long series of works on the "random self-reducibility" of multivariate polynomials [8, 31, 32, 26]. These results are typically stated and studied in very different contexts — however, they are essentially just randomized decoding algorithms for Reed-Muller codes, whose efficiency can be highlighted even further if one looks into the non-standard models of computing used in these papers. A second such result is a new and exciting construction of codes based on "extractors", due to Ta-Shma and Zuckerman [93], whose constructions lead to new codes with interesting "soft-decision decoding" capabilities. Unfortunately, we won't be able to discuss the terms "extractors", or "soft-decision decoding" in this article, so we won't dwell on this result. Nevertheless this result gives one of only four distinctive flavors of list-decoding algorithms, and thus is important for future study. Finally, Guruswami and Indyk [41] construct several families codes with efficient list-decoding algorithms. An example of the many results given there is a binary code of positive rate which is list-decodable upto nearly $1/2$ fraction of errors in nearly linear time. Formally, for every $\beta, \epsilon > 0$, they show there exists an $R > 0$ and an infinite family of codes of rate R which is list-decodable to within $1/2 - \epsilon$ fraction of errors in time $O(n^{1+\beta})$. (Previously it was known how to achieve such error-correction capability in polynomial time [91, 43].)

Improvements to error-correction capability. As noted above, the decoding algorithm for the Reed Solomon codes only improved upon the error-correction capability for codes of rate less than $1/3$. (The case of other codes gets even worse!) Focussing on this aspect, Guruswami and Sudan [42], proposed an extension to the Reed-Solomon decoding algorithm that overcame this limitation. This algorithm has the feature that it can decode Reed Solomon codes for the largest error bound e for which it is known that a ball of radius e has only polynomially many radius (see the discussion leading to the Elias-Bassalygo bound). Thus any improvements to this algorithm would require new combinatorial insights about Reed Solomon codes. [42] give similar results also for algebraic geometry codes where the best results are due to Kötter and Vardy [55]. Similar improvements were also obtained for the case of CRT codes by Boneh [16] and Guruswami, Sahai, and Sudan [44].

Improving the running time. Most of the decoding algorithms described above (with the exception of the result by Guruswami and Indyk [41]) run in polynomial time but not a particularly small polynomial! Fortunately, a fair amount of effort has been expended in the coding theory literature on making these running times more respectable. For instance, [70, 79] show how to solve the interpolation problem defined in the first stage of the list-decoding algorithm for Reed Solomon codes (as well as the variant that comes up in the case of algebraic geometry codes) in $O(n^2)$ time, when the output list size is a constant. Similar running times are also known for the root finding problem (which suffices for the second step in the algorithms above) [4, 29, 67, 70, 79, 100]. Together these algorithms further enhance the practical potential for list-decoding (at least they turn the focus away from algorithmic issues and move them towards combinatorial ones).

The wealth of results revolving around list-decoding is growing even as we write this article, shedding further light on its combinatorial significance, algorithmic capability, and use in theoretical computer science. A good starting point for further reading on this topic may be the thesis of Guruswami [40].

4 Conclusions

In summary we have hope to have introduced coding theory to a wide audience and convinced them that this field contains a wealth of results, while also offering the potential for new research problems. In particular some central algorithmic questions of coding theory, both in the Shannon sense and in the Hamming sense are open today, and theoretical computer scientists can (and are) contributing. Readers seeking further material are encouraged to check out the website of the author [90]. More stable sources of information include the classical text of MacWilliams and Sloane [63], the concise text of van Lint on algebraic coding theory [59], the out-of-print, but highly recommended, book by Blahut [15] which is an excellent source for some of the algorithmic works, and the highly detailed (and not-so-handy) handbook of coding theory [49].

Acknowledgments

Many thanks to Venkatesan Guruswami for many conversations, and pointers to the early history.

References

[1] Sigal Ar, Richard Lipton, Ronitt Rubinfeld, and Madhu Sudan. Reconstructing algebraic functions from mixed data. *SIAM Journal on Computing*, 28(2):488–511, 1999.

[2] Sanjeev Arora, Laszlo Babai, Jacques Stern and Elizabeth Z. Sweedyk. The hardness of ap-

proximating problems defined by linear constraints. *Journal of Computer and System Sciences*, 54(2):317–331, April 1997.

[3] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 485–495, 1997.

[4] Daniel Augot and Lancelot Pecquet. A Hensel lifting to replace factorization in list decoding of algebraic-geometric and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 46:2605–2613, November 2000.

[5] Alexander Barg. Complexity issues in coding theory. In [49, Chapter 7].

[6] Alexander Barg and Gillés Zémor. Linear-time decodable, capacity achieving binary codes with exponentially falling error probability. *IEEE Transactions on Information Theory*, 2001.

[7] L. A. Bassalygo. New upper bounds for error correcting codes. *Problems of Information Transmission*, 1(1):32–35, 1965.

[8] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. *Proceedings of the 7th Symposium on Theoretical Aspects of Computer Science*, Lectures Notes in Computer Science v. 415, pp. 37–48, February 1990.

[9] Elwyn R. Berlekamp. *Algebraic Coding Theory.* McGraw Hill, New York, 1968.

[10] Elwyn R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computations*, 24:713–735, 1970.

[11] Elwyn R. Berlekamp. *Key papers in the development of coding theory.* IEEE Press, New York, 1974.

[12] Elwyn R. Berlekamp. Bounded distance +1 soft-decision Reed-Solomon decoding. *IEEE Transactions on Information Theory*, 42(3):704–720, 1996.

[13] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems (Corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[14] Richard E. Blahut. *Principles and Practice of Information Theory.* Addison-Wesley, Reading, Massachusetts, 1987.

[15] Richard E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley, Reading, Massachusetts, 1983.

[16] Dan Boneh. Finding smooth integers in short intervals using CRT decoding. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 265–272, 2000.

[17] R. C. Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3:68–79, 1960.

[18] T.M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Publishing, New York, 1991.

[19] Ph. Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Research Reports*, Suppl. 10, 1973.

[20] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating-CVP to within almost-polynomial factors is NP-hard. *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.

[21] Rod G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. The parametrized complexity of some fundamental problems in coding theory. *SIAM Journal on Computing* 29(2):545–570, 1999.

[22] Ilya I. Dumer. Two algorithms for the decoding of linear codes. *Problems of Information Transmission*, 25(1):24–32, 1989.

[23] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 475-484, New York City, New York, 17-19 October, 1999.

[24] Iwan M. Duursma. *Decoding codes from curves and cyclic codes.* Ph.D. Thesis, Eindhoven, 1993.

[25] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[26] Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Computational Complexity*, 6(2):101–132, 1997.

[27] G. David Forney. *Concatenated Codes.* MIT Press, Cambridge, MA, 1966.

[28] Robert G. Gallager. *Low Density Parity Check Codes.* MIT Press, Cambridge, Massachusetts, 1963.

[29] Shuhong Gao and M. Amin Shokrollahi. Computing roots of polynomials over function fields of curves. *Coding Theory and Cryptography: From Enigma and Geheimschreiber to Quantum Theory (D. Joyner, Ed.), Springer*, pages 214–228, 2000.

[30] Arnaldo Garcia and Henning Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlădut bound. *Inventiones Mathematicae*, 121:211–222, 1995.

[31] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing,* pages 32-42, New Orleans, Louisiana, 6-8 May 1991.

[32] Peter Gemmell and Madhu Sudan. Highly resilient correctors for multivariate polynomials. *Information Processing Letters*, 43(4):169–174, 1992.

[33] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504–522, May 1952.

[34] Marcel J. E. Golay. Notes on digital coding. *Proceedings of the IRE*, v. 37, page 657, June 1949.

[35] Oded Goldreich and Leonid Levin. A hard-core predicate for all one-way functions. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, May 1989.

[36] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46(5):1330–1338, July 2000. (Fuller version available as TR98-062 (Revision 4), Electronic Colloquium on Computational Complexity `http://www.eccc.uni-trier.de/eccc`.)

[37] V. D. Goppa. Codes on algebraic curves. *Soviet Math. Doklady*, 24:170–172, 1981.

[38] Daniel Gorenstein and Neal Zierler. A class of error-correcting codes in p^m symbols. *Journal of Society of Industrial and Applied Mathematics*, 9:207–214, June 1961.

[39] Dima Grigoriev. Factorization of polynomials over a finite field and the solution of systems of algebraic equations. *Translated from Zapiski Nauchnykh Seminarov Lenningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR*, 137:20–79, 1984.

[40] Venkatesan Guruswami. *List decoding of Error-correcting codes.* Ph.D. Thesis. Massachusetts Institute of Technology, 2001.

[41] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, to appear, October 2001.

[42] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[43] Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 181–190, 2000.

[44] Venkatesan Guruswami, Amit Sahai, and Madhu Sudan. Soft-decision decoding of Chinese Remainder codes. *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 159–168, 2000.

[45] Richard W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29:147–160, April 1950.

[46] H. J. Helgert. Alternant codes. *Information and Control*, 26:369–380, 1974.

[47] A. Hocquenghem. Codes correcteurs d'erreurs. *Chiffres (Paris)*, 2:147–156, 1959.

[48] Tom Høholdt, Jacobus H. van Lint, Ruud Pellikaan. Algebraic geometry codes. In [49].

[49] C. Huffman and V. Pless. *Handbook of Coding Theory*, volumes 1 & 2. Elsevier Sciences, 1998.

[50] S. M. Johnson. A new upper bound for error-correcting codes. *IEEE Transactions on Information Theory*, 8:203–207, 1962.

[51] Jørn Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18:652–656, 1972.

[52] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM Journal on Computing*, 14(2):469–489, 1985.

[53] G. L. Katsman, Michael Tsfasman, and Serge Vlădut. Modular curves and codes with a polynomial construction. *IEEE Transactions on Information Theory*, 30:353–355, 1984.

[54] Ralf Kötter. A unified description of an error locating procedure for linear codes. *Proceedings of Algebraic and Combinatorial Coding Theory*, Voneshta Voda, Bulgaria, 1992.

[55] Ralf Kötter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, pages 625–635, October 2000.

[56] Arjen K. Lenstra. Factoring multivariate polynomials over finite fields. *Journal of Computer and System Sciences*, 30(2):235–248, April 1985.

[57] V. I. Levenshtein. Universal bounds for codes and designs. [49, Chapter 6], 1998.

[58] Jacobus H. van Lint. Nonexistence theorems for perfect error-correcting codes. *Proceedings of the Symposium on Computers in Algebra and Number Theory*, New York, 1970, pages 89–95, G. Birkhoff and M. Hall Jr. (Eds), SIAM-AMS Proceedings vol IV, American Mathematical Society, Providence, RI, 1971.

[59] Jacobus H. van Lint. *Introduction to Coding Theory.* Graduate Texts in Mathematics **86**, (Third Edition) Springer-Verlag, Berlin, 1999.

[60] M. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. STOC 1998.

[61] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.

[62] Jessie MacWilliams. A theorem on the distribution of weights in a systematic code. *Bell Systems Technical Journal*, 42:79–94, January 1963.

[63] F. J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes.* Elsevier/North-Holland, Amsterdam, 1981.

[64] D. M. Mandelbaum. On a class of arithmetic codes and a decoding algorithm. *IEEE Transactions on Information Theory*, 21:85–88, 1976.

[65] J. L. Massey. *Threshold decoding.* MIT Press, Cambridge, 1963.

[66] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15:122–127, 1969.

[67] R. Matsumoto. On the second step in the Guruswami-Sudan list decoding algorithm for ag-codes. *Technical Report of IEICE*, pages 65–70, 1999.

[68] Robert J. McEliece, E. R. Rodemich, H. C. Rumsey, and L. R. Welch. New upper bounds on the rate of a code via the Delsarte-Macwilliams inequalities. *IEEE Transactions on Information Theory*, 23:157–166, 1977.

[69] D. E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IRE Transactions on Electronic Computation*, vol. EC-3, pp 6–12, Sept. 1954.

[70] Rasmus R. Nielsen and Tom Høholdt. Decoding Hermitian codes with Sudan's algorithm. *Proceedings of AAECC-13, LNCS 1719*, pages 260–270, 1999.

[71] Vadim Olshevsky and M. Amin Shokrollahi. A displacement structure approach to efficient list decoding of algebraic geometric codes. *Proccedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999.

[72] Ruud Pellikaan. On decoding linear codes by error correcting pairs. *Eindhoven University of Technology*, preprint, 1988.

[73] W. W. Peterson. Encoding and error-correction procedures for Bose-Chaudhuri codes. *IRE Transactions on Information Theory*, 6:459–470, 1960.

[74] M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, IT-6:445–450, 1960.

[75] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IRE Transactions on Information Theory*, vol. PGIT-4, pp. 38–49, September 1954.

[76] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. SIAM*, 8:300–304, 1960.

[77] Tom Richardson and Rudiger Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, March 2001.

[78] Tom Richardson, Amin Shokrollahi and Rudiger Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, March 2001.

[79] Ronny Roth and Gitit Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, January 2000.

[80] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[81] Claude E. Shannon, Robert G. Gallager, and Elwyn R. Berlekamp. Lower bounds to error probability for coding on discrete memoryless channels. *Information and Control*, 10:65–103 (Part I), 522–552 (Part II), 1967.

[82] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

[83] V. M. Sidelnikov. Decoding Reed-Solomon codes beyond $(d-1)/2$ and zeros of multivariate polynomials. *Problems of Information Transmission*, 30(1):44–59, 1994.

[84] Michael Sipser and Daniel Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[85] David Slepian. A class of binary signalling alphabets. *Bell System Technical Journal*, 35:203–234, January 1956.

[86] Daniel Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1732, 1996.

[87] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, Berlin, 1993.

[88] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[89] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction diameter. *Proceedings of the 35th Annual Allerton Conference on Communication, Control and Computing*, 1997.

[90] Madhu Sudan. *A Crash Course in Coding Theory*. Slides available from `http://theory.lcs.mit.edu/~madhu`, November 2000.

[91] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2): 236–266, March 2001.

[92] R. Michael Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27:533–547, September 1981.

[93] Amnon Ta-Shma and David Zuckerman. Extractor Codes. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 193–199, July 2001.

[94] Aimo Tietavainen. On the nonexistence of perfect codes over finite fields. *SIAM Journal of Applied Mathematics*, 24(1):88-96, January 1973.

[95] Michael A. Tsfasman, Serge G. Vlădut, and Thomas Zink. Modular curves, Shimura curves, and codes better than the Varshamov-Gilbert bound. *Math. Nachrichten*, 109:21–28, 1982.

[96] Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. STOC, pages 92-109, 1997.

[97] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Akad. Nauk SSSR*, 117:739–741, 1957.

[98] Lloyd Welch and Elwyn R. Berlekamp. Error correction of algebraic block codes. *US Patent Number 4,633,470*, December 1986.

[99] J. M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[100] Xin-Wen Wu and P. H. Siegel. Efficient list decoding of algebraic geometric codes beyond the error correction bound. *Proceedings of the International Symposium on Information Theory*, June 2000.