8 List Decoding of Concatenated Codes

8.1 Introduction

The decoding algorithms for Reed-Solomon and AG-codes provide the first results which algorithmically exploit the potential of list decoding well beyond half the minimum distance. In addition, these codes are widely studied and used, and thus these algorithms are not only theoretically interesting, but could also have a lot of practical impact. In this chapter, we are interested in polynomial time constructible linear codes over $\mathbb{F}_q$ for a *small*, fixed q, which can be efficiently list decoded from a large, and essentially "maximum" possible, fraction of errors, and which have good rate. Codes over small alphabets are desirable for several applications. Of particular interest to us will be *binary* codes. Such small alphabet codes with high list decodability cannot be directly obtained from Reed-Solomon or algebraic-geometric codes. Recall that Reed-Solomon codes require the alphabet size to be at least as large as the blocklength of the code. While AG-codes can be defined over an alphabet of fixed size q, their performance is limited by certain algebraic barriers. In particular these rule out the existence of good binary AG-codes, and even for larger q limit their list decodability to much less than what is in general possible for q-ary codes.

The reader will recall that in Chapter 5 we had investigated trade-offs between list decodability and rate for q-ary codes. The results of this chapter can be viewed as an attempt to constructivize, to whatever extent possible, the existential bounds established in Chapter 5.

While Reed-Solomon and AG-codes do not yield good list decodable q-ary codes for small q, we show in this chapter that concatenated codes that use them as outer codes along with appropriate inner codes do achieve small alphabet size together with good algorithmic list decodability properties. The concatenated codes are decoded in two steps: in the first step, a decoding of the portions of the received word corresponding to the various inner encodings is performed. The inner code, owing to its small dimension, can be decoded by brute-force in the allowed runtime (which is polynomial in the entire blocklength). The inner decoding passes to the outer decoder, information concerning the possible symbols at each position, together with appropriate weights or confidence information. The decoding is then completed by running the soft (list) decoding algorithms for the outer Reed-Solomon

V. Guruswami: List Decoding of Error-Correcting Codes, LNCS 3282, pp. 177-207, 2004.

or AG-code from Chapter 6. This represents a novel use of the soft decoding algorithm, and is one of the few such uses where a simple worst-case analytic bound on the number of errors corrected by the algorithm can be proved. (In contrast, a large body of literature on soft decoding applies it to probabilistic channels and obtains either analytic or experimental estimates of the decoding error probability under the specific error model (cf. [59, 60, 121]).)

8.2 Context and Motivation of Results

We are interested in families of q-ary codes that can be efficiently list decoded from a large fraction of errors. Decoding from a fraction of errors beyond $(1 - 1/q)$ is information-theoretically impossible for q-ary codes (of positive rate). This is because a random received word will differ from any particular codeword in a expected fraction $(1-1/q)$ of positions. Therefore, list decoding beyond a fraction $(1 - 1/q)$ of errors will require a list size proportional to the total number of codewords, and hence an exponential list size for code families with positive rate.

Therefore, we are interested in families of q-ary codes that can be list decoded from a fraction p of errors, for $0 < p < (1 - 1/q)$. Having fixed the desired level of error-resilience, the quantity we would like to optimize is the rate of the code family. This is exactly in the spirit of the results from Chapter 5, except that we are now interested in both explicit specifications or polynomial time constructions of the code, *and* efficient list decoding algorithms (and not just a good combinatorial list decodability property).

The main tools used for the above pursuit are concatenated codes with outer Reed-Solomon or AG-codes and inner codes with good combinatorial list decodability and/or distance properties. This gives a polynomial time construction of, say, a binary code with good combinatorial list decodability. We enhance the nice combinatorial properties of concatenated codes with algorithmic ones, by presenting fairly general schemes to efficiently decode these codes to close to their Johnson radius (which is the *a priori* "list decoding capacity" of any code).

In order to present the above constructive results, we focus on the "high-noise regime", i.e., list decoding up to a fraction $(1 - 1/q - \varepsilon)$ of errors for q-ary codes (where q is thought of as small and fixed). For such codes, the results of Chapter 5 imply that the best rate achievable is $\Theta(\varepsilon^2)$. Our goal will be to approach this performance with explicit codes and efficient decoding algorithms. We loosely refer to codes that can correct such a large fraction (approaching $(1 - 1/q)$) of errors as *highly list decodable.*

We focus on the high-noise regime since it brings out the asymptotics very well. Even optimizing the exponent of ε in the rate is a non-trivial problem to begin with in this context. Hence, working in the high-noise regime implies that (at least for current results) we need not be very careful with the constant factors in the rate that are independent of ε (since the $\varepsilon^{O(1)}$

term is the dominant one in the rate). Moreover, there is a natural and well-posed goal of approaching the "optimal" rate, i.e., obtaining the best possible exponent of ε in the rate. We note here that this is a very asymptotic and computer science style perspective, and indeed the motivation comes partly from applications of list decoding to complexity theory, to be discussed in Chapter 12, where the high-noise regime is the most interesting and useful one to focus on. Coding theorists are sometimes disturbed by the low rate in the way we state some of our results. But we would like to stress that the low rate is unavoidable since we are targeting decoding from a very large fraction of errors. Moreover, we believe that optimizing our techniques for the high-noise (and *consequently* low-rate) regime is a good first step, and that the techniques can eventually be applied to a more careful, thorough investigation of the situation where we do not wish to correct such a large fraction of errors. The results of the next two chapters will also be motivated by and stated for the high-noise regime – these chapters will deal with codes over large (but constant-sized) alphabets and erasure codes, respectively.

8.3 Overview of Results

We present list decoding algorithms for several families of concatenated codes. Recall that the distance of a concatenated code whose outer code has distance D and inner code has distance d is at least Dd (and this quantity is referred to as the *designed distance* of the concatenated code). Unique decoding algorithms to decode up to the *product bound*, namely to correct fewer than $Dd/2$ errors, are known based on Generalized Minimum Distance decoding of the outer code [60, 110] (this is also discussed in detail in Appendix A). The focus of this chapter is on list decoding algorithms that permit recovery well beyond the product bound for certain families of concatenated codes. A discussion of the specific results follows.

In Section 8.4, we give list decoding algorithms for codes where the outer code is a Reed-Solomon or Algebraic-geometric code and the inner code is a Hadamard code. Our algorithms decode these codes up to the Johnson bound on list decoding radius. These algorithms also serve as a beautiful illustration of the power of our soft decoding algorithms for list decoding Reed-Solomon and AG-codes from Chapter 6. The construction with an appropriate algebraic-geometric outer code, upon picking parameters suitably, gives us a construction of q-ary codes of rate $\Omega(\varepsilon^6)$ list decodable up to a fraction $(1 - 1/q - \varepsilon)$ of errors.

In Section 8.5, we present a decoding algorithm for concatenated codes with outer Reed-Solomon or AG-code and an arbitrary inner code. The algorithm falls short of decoding up to the Johnson radius, but decodes well beyond half the distance when the rate of the outer code is small. In particular, it gives an alternative construction of q-ary codes of rate $\Omega(\varepsilon^6)$ decodable up to a fraction $(1 - 1/q - \varepsilon)$ of errors. The advantage of this construction is that one can use Reed-Solomon codes as opposed to the more complicated AG-codes necessary for the earlier result using Hadamard codes. The construction and decoding algorithms are consequently also easier and faster.

Finally, in Section 8.6, we use special purpose codes as inner codes in a concatenated construction to obtain binary linear codes of rate $\Omega(\varepsilon^4)$ efficiently list decodable from a fraction $(1/2 - \varepsilon)$ of errors. The inner codes are a more general variant of the ones guaranteed by Theorem 5.8 of Chapter 5. We should remark that we are able to obtain this result only for binary codes.[1]

We stress here that our construction that has rate $\Omega(\varepsilon^4)$ is **not** obtained by constructing a large distance binary code and then appealing to the Johnson bound to argue that the list decoding radius is at least $(1/2-\varepsilon)$. Indeed, this will require the relative distance to be at least $(1/2 - O(\varepsilon^2))$ and the best known polynomial time constructions of such codes yield a rate of only about $\Omega(\varepsilon^6)$. In fact, a polynomial time construction of binary code families of relative distance $(1/2 - O(\varepsilon^2))$ and rate $\Omega(\varepsilon^4)$ will asymptotically match the Gilbert-Varshamov bound at low rates, and will be a *major* breakthrough in coding theory.

The results of this chapter focus exclusively on linear codes. Some results in the next two chapters will resort to a certain "limited" amount of non-linearity.

8.4 Decoding Concatenated Codes with Inner Hadamard Code

We present list decoding algorithms for concatenated codes with an outer algebraic code and inner Hadamard code. The motivation of considering the Hadamard code is its nice properties which we exploit to decode the concatenated codes up to their Johnson radius. Concatenated codes with algebraic-geometric outer code and inner Hadamard code are among the best explicitly known codes in terms of the rate vs. distance trade-offs. By decoding such codes up to the Johnson radius, we will get codes of good rate and very high list decodability. This is our primary motivation for considering decoding algorithms for such concatenated codes.

Recall the definition of Hadamard codes from Chapter 2. The q-ary Hadamard code of dimension m encodes an $\mathbf{x} \in \mathbb{F}_q^m$ by $\langle x \cdot z \rangle_{z \in \mathbb{F}_q^m}$ (i.e.

[1] We know how to achieve a similar performance for general alphabets if we relax the requirement of linearity. The next chapter will discuss several non-linear code constructions with good list decodability. The codes, though not linear, will be based on "pseudolinear" codes, and will possess succinct representation and be efficiently encodable/decodable. Using random q-ary pseudolinear codes as inner codes will permit us to obtain codes of rate $\Omega(\varepsilon^4)$ list decodable up to a fraction $(1 - 1/q - \varepsilon)$ of errors, for every prime power q. We, however, do not elaborate on this point further.

by its dot product over $\mathbb{F}_q$ with every vector $\mathbf{z} \in \mathbb{F}_q^m$). It has blocklength q^m and minimum distance $(1 - 1/q) \cdot q^m$; in fact all non-zero codewords in the code have Hamming weight $(1 - 1/q) \cdot q^m$.

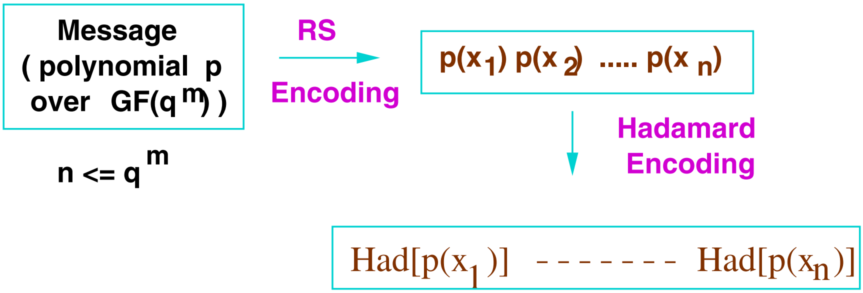

Fig. 8.1. Reed-Solomon concatenated with a Hadamard code

The codes considered in this section will be the concatenation of a Reed-Solomon or AG-code over GF(q^m) with the q-ary Hadamard code of dimension m. Note the number of outer codeword symbols (i.e., q^m) exactly equals the number of Hadamard codewords, so concatenation of these codes is well-defined. Figure 8.1 depicts the structure of a Reed-Solomon concatenated with a Hadamard code. The encoding of a message (a polynomial) p will be Had$(p(x_1))$Had$(p(x_2)) \cdots$ Had$(p(x_n))$, where $x_1, \ldots, x_n$ are distinct elements in GF(q^m) that are used in defining the Reed-Solomon code. (To encode an element $\alpha \in$ GF(q^m) using the Hadamard code, one views α as a string of length m over GF(q) using some fixed representation of GF(q^m) as vectors of length m over GF(q).) The reader might recall that we already used Reed-Solomon codes concatenated with Hadamard codes in Section 4.6 (with $q = 2$).

Jumping ahead to how our decoding will proceed, the inner decoder will "decode" the Hadamard code and pass information concerning the possible symbols at each position, together with appropriate weights. Suppose the i'th block (corresponding to the inner encoding of the i'th outer codeword symbol) of the received word is r_i. It is natural that the weight that the inner decoder gives to a symbol $\alpha \in$ GF(q^m) for position i should be a decreasing function of $\Delta(\text{Had}(\alpha), r_i)$ (where $\Delta(x, y)$ measures the Hamming distance between x and y). This is because, intuitively, the larger this distance, the smaller is the likelihood that the i'th symbol of the outer codeword was α. In fact, the inner decoder will set weights to be a decreasing linear function of this distance (the linearity makes possible a precise analysis of the number of errors corrected). Specifically, the weight for symbol α for the i'th block r_i of received word will be set to

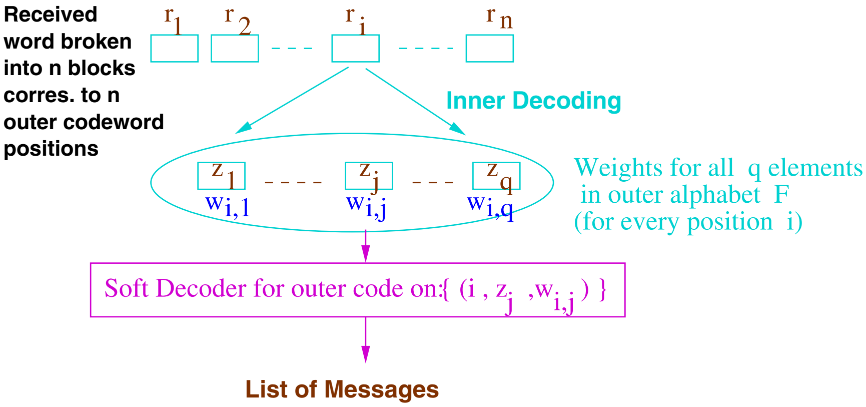
Received word broken into n blocks corres. to n outer codeword positions
r1
r2
ri
rn
Inner Decoding
z1
w i,1
zj
w i,j
zq
w i,q
Weights for all q elements in outer alphabet F (for every position i)
Soft Decoder for outer code on:{ (i , zj ,wi,j) }
List of Messages

Fig. 8.2. The basic idea behind our decoding algorithms for concatenated codes. For each position of the outer code, the inner decoding passes a weight or confidence rating for every element of the field $F = \mathrm{GF}(q)$. These are then used by a soft list decoding algorithm for the outer code to finish the decoding.

$$\left(1 - \frac{q}{q-1}\frac{\Delta(r_i, \mathrm{Had}(\alpha))}{q^m}\right) .$$

The decoding is then completed by running the soft (list) decoding algorithms for the outer Reed-Solomon or AG-code from Chapter 6 with these choice of weights. This is in fact the procedure used for decoding all of the concatenated codes in this chapter. Figure 8.2 illustrates the basic structure of our decoding schemes for concatenated codes.

Recalling the statements of Theorems 6.26 and 6.41, the sum of the squares of the weights is an important quantity that governs the performance of the decoding algorithm. Good upper bounds on this sum will permit a good analysis of the error-correction performance of the algorithm. Below, we provide such an upper bound for the choice of weights made by the inner decoder in decoding the Hadamard code.

Proposition 8.1. *Let q be a prime power and let m be a positive integer. Let* Had $: \mathbb{F}_{q^m} \to \mathbb{F}_q^{q^m}$ *be the q-ary Hadamard code of dimension m and blocklength q^m. Let $f \in \mathbb{F}_q^{q^m}$ be an arbitrary vector of length q^m over $\mathbb{F}_q$. Then*

$$\sum_{\alpha \in \mathrm{GF}(q^m)} \left(1 - \frac{q}{q-1} \cdot \frac{\Delta(f, \mathrm{Had}(\alpha))}{q^m}\right)^2 \leq 1 . \tag{8.1}$$

Remark: For the case $q = 2$, $\big(1 - 2\Delta(f, \mathrm{Had}(\alpha))/2^m\big)$ equals the *Fourier coefficient* $\hat{f}_\alpha$ of f with respect to $\mathrm{Had}(\alpha)$, viewed as a linear function mapping $\mathbb{F}_q^m$ to $\mathbb{F}_q$. In this case, the statement of the Proposition in fact holds with equality, and is simply the standard Plancherel's identity $\sum_\alpha \hat{f}_\alpha^2 = 1$. The result for the non-binary case appears in [120], and the proof there is based on the MacWilliams-Sloane identities for the weight distribution of dual codes; we give a more elementary proof below.

Proof: The proof works by embedding any string $f \in \mathbb{F}_q^{q^m}$ as a q^{m+1}-dimensional *real unit vector*. The embedding will be such that for every $\alpha \neq \beta \in \mathrm{GF}(q^m)$, the vectors associated with the Hadamard codewords $\mathrm{Had}(\alpha)$ and $\mathrm{Had}(\beta)$ will be orthogonal (in the usual real dot product over $\mathbb{R}^{q^m \cdot q}$). Furthermore, the embedding will be such that the quantity

$$\Bigl(1 - \frac{q}{q-1} \cdot \frac{\Delta(f,g)}{q^m}\Bigr)$$

for every two functions $f, g \in \mathbb{F}_q^{q^m}$ will simply be the dot product of the vectors associated with f, g. The result will then follow since the sum of the squares of the projections of a unit vector along pairwise orthogonal vectors can be at most 1.

Suppose the q elements of $\mathbb{F}_q$ are $\gamma_1, \gamma_2, \ldots, \gamma_q$. Associate a q-dimensional vector e_i with γ_i as follows (e_{il} denotes the l'th component of e_i): $e_{ii} = \sqrt{(q-1)/q}$ and $e_{il} = -1/\sqrt{q(q-1)}$ for $l \neq i$. Note that this definition satisfies $\langle e_i, e_i \rangle = 1$ and $\langle e_i, e_j \rangle = -1/(q-1)$ for $i \neq j$. For a string $f \in \mathbb{F}_q^{q^m}$, we view f as the q^{m+1}-dimensional vector obtained in the obvious way by juxtaposing the q-dimensional vectors for each of the q^m values which f takes on its domain, and then normalizing it to a unit vector (by dividing every component by $\sqrt{q^m}$). By abuse of notation, we will denote the real vector associated with f also by f.

Note that when we take the inner product $\langle f, g \rangle$, we get a contribution of $1/q^m$ corresponding to the positions where f, g agree, and a contribution of $\frac{-1}{(q-1)} \cdot q^{-m}$ corresponding to places where f, g differ. Hence we have

$$\begin{aligned}\langle f, g \rangle &= (q^m - \Delta(f,g)) \cdot q^{-m} + \Delta(f,g) \cdot \Bigl(\frac{-1}{q-1}\Bigr) \cdot q^{-m} \\ &= 1 - \frac{q}{q-1} \cdot \frac{\Delta(f,g)}{q^m} \ .\end{aligned}$$

Now, for $\alpha \neq \beta \in \mathrm{GF}(q^m)$, $\Delta(\mathrm{Had}(\alpha), \mathrm{Had}(\beta)) = (1 - 1/q) \cdot q^m$ (recall that two distinct codewords in the Hadamard code corresponding to $\mathbb{F}_q^m$ agree on exactly q^{m-1} places and differ at $q^{m-1}(q-1)$ places). Thus, for $\alpha \neq \beta$, we have $\langle \mathrm{Had}(\alpha), \mathrm{Had}(\beta) \rangle = 0$. Also by our choice of vectors, $\langle \mathrm{Had}(\alpha), \mathrm{Had}(\alpha) \rangle = 1$. Hence the q^m vectors associated with the Hadamard codewords are pairwise orthogonal unit vectors. Using this fact the result follows since

$$\sum_{\alpha \in \mathbb{F}_q^m} \Bigl(1 - \frac{q}{q-1} \frac{\Delta(f, \mathrm{Had}(\alpha))}{q^m}\Bigr)^2 = \sum_\alpha \langle f, \mathrm{Had}(\alpha) \rangle^2 \leq \langle f, f \rangle = 1 \ . \qquad \square$$

8.4.1 Reed-Solomon Concatenated with Hadamard Code

We now use the above result to analyze the error-correction capability of Reed-Solomon codes concatenated with Hadamard code, when using the soft decoding algorithm for Reed-Solomon together with the weights passed by the Hadamard decoding.

Theorem 8.2. *Let C be q-ary code of blocklength n and relative distance δ, that is obtained by concatenation of a Reed-Solomon code over* $\mathrm{GF}(q^m)$ *with the Hadamard code of dimension m, for some m. Then, there is a polynomial time list decoding algorithm for C that decodes up to E errors where*

$$E = n\Big(1 - \frac{1}{q}\Big)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) - O(1) \ .$$

(In other words, one can decode such a code up to, essentially, the q-ary Johnson bound on list decoding radius.)

Proof: The relative distance of a q-ary Hadamard code is $(1 - 1/q)$, and in fact all non-zero codewords have the same Hamming weight. Hence, it follows that in order for the relative distance of the concatenated code C to be δ, the relative distance of the outer Reed-Solomon code, call it C_{RS}, must be $q\delta/(q-1)$. Let the blocklength of C_{RS} be $n_0 \leq q^m$, and its dimension be $(k_0 + 1)$, where

$$k_0 = n_0\Big(1 - \frac{q\delta}{q-1}\Big) \ . \tag{8.2}$$

Let $x_1, x_2, \ldots, x_{n_0}$ be distinct elements of $\mathrm{GF}(q^m)$ that are used to define C_{RS}. Thus, the messages of C_{RS} (and hence C, too) are degree k_0 polynomials over $\mathrm{GF}(q^m)$, and a polynomial p is encoded under C_{RS} as $\langle p(x_1), p(x_2), \ldots, p(x_{n_0})\rangle$. The blocklength n of the overall concatenated code C satisfies $n = n_0 q^m$, and its dimension equals $(k_0 + 1)m$.

Let $y \in \mathbb{F}_q^n$ be a "received word"; the task of list decoding that we wish to solve is to obtain a list of all codewords of C within a Hamming distance of E from y. For $1 \leq i \leq n_0$, denote by y_i the portion of y in block i of the codeword (i.e., the portion corresponding to the Hadamard encoding of the i^{th} symbol of the outer code).

We now perform the "decoding" of each of the n_0 blocks y_i as follows. For $1 \leq i \leq n_0$ and $\alpha \in \mathrm{GF}(q^m)$, compute the Hamming distance $e_{i,\alpha}$ between y_i and $\mathrm{Had}(\alpha)$, and then compute the *weight* $w_{i,\alpha}$ as:

$$w_{i,\alpha} \stackrel{\mathrm{def}}{=} \max\Big\{\Big(1 - \frac{q}{q-1} \cdot \frac{e_{i,\alpha}}{q^m}\Big),\ 0\Big\} \ . \tag{8.3}$$

Note the computation of all these weights can be done by a straightforward brute-force computation in $O(n_0(q^m)^2) = O(n^2/n_0)$ time. Thus all the inner decodings can be performed efficiently in at most quadratic time.

The key combinatorial property of these weights, that follows from Proposition 8.1 above, is that

$$\sum_{\alpha} w_{i,\alpha}^2 \leq 1 \ , \tag{8.4}$$

for every i, $1 \leq i \leq n_0$. These weights will now be "passed" to the outer Reed-Solomon decoder as the confidence information about the various symbols of the Reed-Solomon codeword. For the outer decoder, we will use the soft decoding algorithm from Chapter 6. Specifically, we will use the result of Theorem 6.26. Applied to this context, the result implies that, for any desired tolerance parameter $\varepsilon > 0$, we can find in time polynomial in n_0 and $1/\epsilon$, a list of all polynomials p over $\mathrm{GF}(q^m)$ of degree at most k_0 that satisfy

$$\sum_{i=1}^{n_0} w_{i,p(x_i)} \geq \Big(k_0 \cdot \sum_{\substack{1 \leq i \leq n_0 \\ \alpha \in \mathrm{GF}(q^m)}} w_{i,\alpha}^2\Big)^{1/2} + \epsilon \max_{i,\alpha} w_{i,\alpha} \ . \tag{8.5}$$

Applied to the choice of weights (8.3) and using Equation (8.4), the decoding algorithm can thus retrieve all codewords corresponding to degree k_0 polynomials p for which

$$\sum_{i=1}^{n_0} \Big(1 - \frac{q}{q-1} \cdot \frac{e_{i,p(x_i)}}{q^m}\Big) \geq \sqrt{k_0 n_0} + \epsilon \ . \tag{8.6}$$

Note that $w_{i,p(x_i)} \geq (1 - \frac{q}{q-1} \cdot \frac{e_{i,p(x_i)}}{q^m})$, and hence if the above condition is satisfied then so is Condition (8.5).

Now, recall that $e_{i,p(x_i)} = \Delta(y_i, \mathrm{Had}(p(x_i)))$. Hence, (8.6) above implies that we can find all codewords at a distance E from the received word y provided

$$\begin{aligned} & n_0 - \frac{qE}{(q-1) \cdot q^m} \geq \sqrt{k_0 n_0} + \epsilon \text{ or} \\ & \qquad \frac{qE}{q-1} \leq n\Big(1 - \sqrt{\frac{k_0}{n_0}} - \frac{\epsilon}{\sqrt{n}}\Big) \quad (\text{since } n = n_0 q^m) \\ \Longleftarrow \qquad & E \leq n\Big(\frac{q-1}{q}\Big)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) - \epsilon q^m \ , \end{aligned}$$

where in the last step we use the value of k_0 from Equation (8.2). If we pick $\epsilon \leq 1/n$, this implies we can list decode up to

$$E = n\Big(\frac{q-1}{q}\Big)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) - O(1)$$

errors, as desired. □

8.4.2 AG-code Concatenated with Hadamard Code

The result of Theorem 8.2 decodes the concatenated code up to the Johnson radius, and thus has very good error-correction performance for the concerned code. However, while interesting for a variety of reasons, from a coding standpoint, the Reed-Solomon concatenated with Hadamard codes are not very attractive. This is because they have very low rate, since the inner Hadamard code maps m symbols into q^m symbols, and thus has very poor, vanishing, rate for large m. In particular, the family of codes is not asymptotically good, and has rate rapidly tending to 0 in the limit of large blocklengths. It is thus way off our pursuit of codes list decodable to a fraction $(1-1/q-\varepsilon)$ of errors with rate somewhat close to $\Theta(\varepsilon^2)$.

In this section, we will adapt the result of Theorem 8.2 to concatenated codes with outer AG-code (instead of Reed-Solomon code). The inner code will be the Hadamard code as before. The rate of the overall code will once again not be great, since it will inherit the poor rate of the Hadamard code. But since AG-codes with good parameters exist over a fixed alphabet of size independent of the blocklength, the inner Hadamard code will now be a constant-sized code, and thus will have some fixed, albeit small, rate. Thus, we will be able to achieve positive rate (i.e. rate which is at least r for some fixed constant $r > 0$ that is independent of the blocklength) for the overall code. As a corollary, in the next section, we will plug in the best-known AG-codes (those discussed in Section 6.3.9) to obtain constructions of codes which are list decodable up to a fraction $(1-1/q-\varepsilon)$ of errors and have rate $\Omega(\varepsilon^6)$.

The formal result concerning list decoding AG-codes concatenated with Hadamard codes is stated below. The hypothesis about a suitable representation of the code is necessary in the statement of the theorem, since the decoding algorithms of Chapter 6 also made this assumption.

Theorem 8.3. *Let $C_{\mathrm{AG-Had}}$ be q-ary code of blocklength n and relative distance at least δ, that is obtained by concatenation of an algebraic-geometric code over $\mathrm{GF}(q^m)$ of relative designed distance $q\delta/(q-1)$ with the q-ary Hadamard code of dimension m, for some m. Then, there exists a representation of the code of size polynomial in n under which a polynomial time list decoding algorithm exists to list decode $C_{\mathrm{AG-Had}}$ up to E errors, where*

$$E = n\Big(1-\frac{1}{q}\Big)\Big(1-\sqrt{1-\frac{q\delta}{q-1}}\Big) - O(1) \ .$$

(In other words, one can decode such a code up to, essentially, the q-ary Johnson bound on list decoding radius.)

Proof: The proof parallels that of the earlier result (Theorem 8.2) where the outer code was a Reed-Solomon code. The inner decodings of the various Hadamard codes proceeds exactly as before, passing weights to the outer decoder. Now, for the outer decoder we can make use of the soft list decoding algorithm for AG-codes developed in Theorem 6.41, instead of the Reed-Solomon soft decoder. This is really the only change necessary to the proof of Theorem 8.2, and the claimed bound on the number of errors corrected follows as before. We omit the details. The soft decoding algorithm for AG-codes from Theorem 6.41 works in polynomial time only assuming a specific (non-standard) representation of the AG-code, which necessitates the hypothesis about the representation of the code in the statement of the theorem. □

8.4.3 Consequence for Highly List Decodable Codes

We now apply Theorem 6.41 with AG-codes that achieve the best known trade-off between rate and distance (from Section 6.3.9 of Chapter 6). This gives us codes list decodable up to a fraction $(1-1/q-\varepsilon)$ of errors and which have reasonably good rate.

Corollary 8.4. *For every fixed prime power q, the following holds: For every $\varepsilon > 0$, there exists a family of linear codes over $\mathbb{F}_q$ with the following properties:*

(i) *The family is polynomial time constructible in that the generator matrix of a code of blocklength n in the family can be computed in time a fixed polynomial in n.*
(ii) *Its rate is $\Omega(\varepsilon^6 \cdot \log(1/\varepsilon))$.*
(iii) *For each code in the family, there exists a polynomial amount of advice information given which there is a polynomial time list decoding algorithm that decodes the code up to a fraction $(1-1/q-\varepsilon)$ of errors.*

Proof: We will employ the concatenated code construction of Theorem 8.3 applied with the outer code being AG-codes that meet the Drinfeld-Vlădut bound (as guaranteed by Fact 6.43). By picking m even, we know there exist AG-codes over GF(q^m) of relative designed distance δ' and rate $R \geq 1 - 1/(q^{m/2}-1) - \delta'$. The fraction of errors corrected by the algorithm of Theorem 8.3 is $(1-1/q)(1-\sqrt{1-\delta'})$. Picking $\delta' = 1 - O(\varepsilon^2)$, we can get a list decoding radius of $(1-1/q-\varepsilon)$. For such a value of δ', the rate R of the AG-code can be $\Omega(\varepsilon^2)$, provided $q^{m/2} = \Omega(\varepsilon^{-2})$. This can be achieved with $m = \Theta(\log(1/\varepsilon))$ (since q is fixed, we absorb constant terms that depend on q into the Θ-notation). The rate of the concatenated code is the rate of the AG-code multiplied by the rate of the Hadamard code, and is thus $R \cdot (m/q^m)$. Since $R = \Omega(\varepsilon^2)$, $m = \Theta(\log(1/\varepsilon))$ and $q^m = O(\varepsilon^{-4})$, the rate is $\Omega(\varepsilon^6 \log(1/\varepsilon))$. □

8.5 Decoding a General Concatenated Code with Outer Reed-Solomon or AG-code

The concatenated codes in the previous section used the Hadamard code as inner code. This permitted an elegant analysis of the decoding algorithms based on the combinatorial identity of Proposition 8.1 and the soft decoding algorithms from Chapter 6. However, the Hadamard code has very poor rate which makes these codes not so attractive from a coding theory viewpoint.

In this section, we present an algorithm to decode concatenated codes with outer Reed-Solomon or AG-codes when the inner code is an arbitrary q-ary code. The idea behind the decoding will remain the same (recall Figure 8.2) — in the first step, the inner decoding will pass weights which are linear functions of the distance between the received word and the concerned inner codeword. These weights will then be used in a soft decoding of the outer code. The key technical step in making this work when the inner code is not Hadamard but arbitrary is to prove an analog of the combinatorial bound of Proposition 8.1 for a general q-ary code. We do so next.

8.5.1 A Relevant Combinatorial Result

To motivate the exact statement of the combinatorial result, we jump ahead to give a hint of how the decoding will exactly proceed. When presented a received word $\mathbf{r}$, the inner decoder will simply search for and output all codewords which lie in a Hamming ball of a certain radius R around $\mathbf{r}$. The weight associated with a codeword $\mathbf{c}$ at a distance $e_{\mathbf{c}} = \Delta(\mathbf{r}, \mathbf{c}) \leq R$ from $\mathbf{r}$ will be set to be $(R - e_{\mathbf{c}})$. These weights will be used in a soft decoding of the outer code as before. We now state and prove a combinatorial result that gives an upper bound on the sum of squares of the weights $(R - e_{\mathbf{c}})$. Some readers may prefer to take the result below on faith and jump right ahead to the decoding algorithm and its analysis in Section 8.5.2.

Proposition 8.5. *Let $C \subseteq [q]^n$ be a q-ary code (not necessarily linear), and let d be the minimum distance of C, and $\delta = d/n$ its relative distance. Let $\mathbf{r} \in [q]^n$ be arbitrary, and let*

$$R = n\Big(1 - \frac{1}{q}\Big)\Big(1 - \sqrt{1 - \frac{\delta}{(1 - 1/q)}}\Big) \tag{8.7}$$

be the q-ary Johnson radius of the code. Then we have

$$\sum_{\mathbf{c} \in C} \Big(\max\{(R - \Delta(\mathbf{r}, \mathbf{c})), 0 \ \}\Big)^2 \leq \delta n^2 \tag{8.8}$$

Proof: The proof follows essentially the same approach as in the proof of the Johnson bound (Theorems 3.1 and 3.2) from Chapter 3. Instead of bounding the number of codewords within a distance R from $\mathbf{r}$, we now require an upper bound on the sum of squares of linear functions of the distance over all such codewords. The proof will be identical to that of Theorem 3.1 for the most part, with a change towards the end. For purposes of readability, we give the full proof here. The reader familiar with the proof of Theorem 3.1

can jump to just past Equation (8.14) since the proof is identical till that stage.[2]

We identify elements of $[q]$ with vectors in $\mathbb{R}^q$ by replacing the symbol i $(1 \le i \le q)$ by the unit vector of length q with a 1 in position i. We then associate elements in $[q]^n$ with vectors in $\mathbb{R}^{nq}$ by writing down the vectors for each of the n symbols in sequence. This allows us to embed the codewords of C as well as the received word $\mathbf{r}$ into $\mathbb{R}^{nq}$. Let $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_M}$ be all the codewords that satisfy $\Delta(\mathbf{r}, \mathbf{c_i}) \le R$, where R is a parameter that will be set shortly (it will end up being set to the Johnson radius as in Equation (8.7)). By abuse of notation, let us denote by $\mathbf{c_i}$ also the nq-dimensional real vector associated with the codeword $\mathbf{c_i}$, for $1 \le i \le M$ (using the above mentioned identification), and by $\mathbf{r}$ the vector corresponding to $\mathbf{r} \in [q]^n$. Let $\mathbf{1} \in \mathbb{R}^{nq}$ be the all 1's vector. Now define $\mathbf{v} = \alpha \mathbf{r} + \frac{(1-\alpha)}{q}\mathbf{1}$ for a parameter $0 \le \alpha \le 1$ to be specified later in the proof.

The idea behind the rest of the proof is the following. We will pick α so that the nq-dimensional vectors $\mathbf{d_i} = (\mathbf{c_i} - \mathbf{v})$, for $1 \le i \le M$, have all pairwise dot products less than 0. Geometrically speaking, we shift the origin O to O' where $OO' = \mathbf{v}$, and require that relative to the new origin the vectors corresponding to the codewords have pairwise angles which are greater than 90 degrees. We will then exploit the geometric fact that for such vectors $\mathbf{d_i}$, for any vector $\mathbf{w}$, the sum of the squares of its projections along the $\mathbf{d_i}$'s is at most $\langle \mathbf{w}, \mathbf{w} \rangle$ (this is proved in Lemma 8.6). This will then give us the required bound (8.8).

For $1 \le i \le M$, let $e_i = \Delta(\mathbf{r}, \mathbf{c_i})$ be the Hamming distance between $\mathbf{c_i}$ and $\mathbf{r}$. Note by the way we associate vectors with elements of $[q]^n$, we have $\langle \mathbf{c_i}, \mathbf{r} \rangle = n - e_i$. Now

$$\langle \mathbf{c_i}, \mathbf{v} \rangle = \alpha \langle \mathbf{c_i}, \mathbf{r} \rangle + \frac{(1-\alpha)}{q} \langle \mathbf{c_i}, \mathbf{1} \rangle = \alpha(n - e_i) + (1-\alpha)\frac{n}{q} \tag{8.9}$$

$$\langle \mathbf{v}, \mathbf{v} \rangle = \alpha^2 n + 2(1-\alpha)\alpha \frac{n}{q} + (1-\alpha)^2 \frac{n}{q} = \frac{n}{q} + \alpha^2 \Big(1 - \frac{1}{q}\Big) n \tag{8.10}$$

$$\langle \mathbf{c_i}, \mathbf{c_j} \rangle = n - \Delta(\mathbf{c_i}, \mathbf{c_j}) \le n - d \ . \tag{8.11}$$

Using (8.9), (8.10) and (8.11), we get for $i \ne j$

$$\begin{aligned} \langle \mathbf{d_i}, \mathbf{d_j} \rangle = \langle \mathbf{c_i} - \mathbf{v}, \mathbf{c_j} - \mathbf{v} \rangle &\le \alpha e_i + \alpha e_j - d + \Big(1 - \frac{1}{q}\Big)(1-\alpha)^2 n \\ &\le 2\alpha R - d + \Big(1 - \frac{1}{q}\Big)(1-\alpha)^2 n \qquad (8.12) \end{aligned}$$

Hence we have $\langle \mathbf{d_i}, \mathbf{d_j} \rangle \le 0$ as long as

$$R \le (1 - 1/q)n - \Big((1-1/q)\frac{\alpha n}{2} + \frac{(1-1/q)n - d}{2\alpha}\Big) \ .$$

[2] We prove this result here and not in Chapter 3 due to the local nature of its context and use.

Picking $\alpha = \sqrt{1 - \frac{d/n}{(1-1/q)}} = \sqrt{1 - \frac{\delta}{(1-1/q)}}$ maximizes the "radius" R for which our bound will apply. Hence we pick

$$\alpha = \Big(1 - \frac{\delta}{(1-1/q)}\Big)^{1/2} . \tag{8.13}$$

and

$$R = n\Big(1 - \frac{1}{q}\Big)\Big(1 - \sqrt{1 - \frac{\delta}{(1-1/q)}}\Big) = n\big(1 - \frac{1}{q}\big)(1-\alpha) . \tag{8.14}$$

For this choice of α, R, we have $\langle \mathbf{d_i}, \mathbf{d_j} \rangle \le 0$ for every $1 \le i < j \le M$. Now a simple geometric fact, proved in Lemma 8.6 at the end of this proof, implies that for any vector $\mathbf{x} \in \mathbb{R}^{nq}$ that satisfies $\langle \mathbf{x}, \mathbf{d_i} \rangle \ge 0$ for $i = 1, 2, \ldots, M$, we have

$$\sum_{i=1}^{M} \frac{\langle \mathbf{x}, \mathbf{d_i} \rangle^2}{\langle \mathbf{d_i}, \mathbf{d_i} \rangle} \le \langle \mathbf{x}, \mathbf{x} \rangle . \tag{8.15}$$

We will apply this to the choice $\mathbf{x} = \mathbf{r}$. Straightforward computations show that

$$\langle \mathbf{r}, \mathbf{r} \rangle = n \tag{8.16}$$

$$\langle \mathbf{d_i}, \mathbf{d_i} \rangle = \langle \mathbf{c_i} - \mathbf{v}, \mathbf{c_i} - \mathbf{v} \rangle = 2\alpha e_i + (1-\alpha)^2(1 - \frac{1}{q})n \tag{8.17}$$

$$\langle \mathbf{r}, \mathbf{d_i} \rangle = (1-\alpha)\Big(1 - \frac{1}{q}\Big)n - e_i = R - e_i . \tag{8.18}$$

Since each $e_i \le R$, we have $\langle \mathbf{r}, \mathbf{d_i} \rangle \ge 0$ for each i, $1 \le i \le M$, and therefore we can apply Equation (8.15) above. For $1 \le i \le M$, define

$$W_i = \frac{\langle \mathbf{r}, \mathbf{d_i} \rangle}{\sqrt{\langle \mathbf{d_i}, \mathbf{d_i} \rangle}} = \frac{R - e_i}{\sqrt{2\alpha e_i + (1-\alpha)R}} \tag{8.19}$$

(the second step follows using (8.14), (8.17) and (8.18)). Since each $e_i \le R$, we have

$$W_i = \frac{R - e_i}{\sqrt{2\alpha e_i + (1-\alpha)R}} \ge \frac{R - e_i}{\sqrt{(1+\alpha)R}} = \frac{R - e_i}{\sqrt{\delta n}} , \tag{8.20}$$

where the last equality follows by substituting the values of α and R from (8.13) and (8.14). Now combining (8.16), (8.17) and (8.18), and applying Equation (8.15) to the choice $\mathbf{x} = \mathbf{r}$, we get

$$\sum_{i=1}^{M} W_i^2 \le n . \tag{8.21}$$

Now from (8.20) and (8.21) it follows that

$$\sum_{i=1}^{M} (R - \varDelta(\mathbf{r}, \mathbf{c_i}))^2 \le \delta n^2 . \tag{8.22}$$

This clearly implies the bound (8.8) claimed in the statement of the proposition, since the codewords $\mathbf{c_i}$, $1 \le i \le M$, include *all* codewords $\mathbf{c}$ that satisfy $\Delta(\mathbf{r}, \mathbf{c}) \le R$, and the remaining codewords contribute zeroes to the left hand side of Equation (8.8). □

We now prove the geometric fact that was used in the above proof. Once again the reader should feel to skip its proof and move on to the decoding algorithm in the next section, since there is no harm taking its statement on faith.

Lemma 8.6. *Let* $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_M}$ *be* distinct *unit vectors in* $\mathbb{R}^N$ *such that* $\langle \mathbf{v_i}, \mathbf{v_j} \rangle \le 0$ *for* $1 \le i < j \le M$*. Further, suppose* $\mathbf{x} \in \mathbb{R}^N$ *is a vector such that* $\langle \mathbf{x}, \mathbf{v_i} \rangle \ge 0$ *for each* i*,* $1 \le i \le M$*. Then*

$$\sum_{i=1}^{m} \langle \mathbf{x}, \mathbf{v_i} \rangle^2 \le \langle \mathbf{x}, \mathbf{x} \rangle \tag{8.23}$$

Proof: Note that if $\langle \mathbf{v_i}, \mathbf{v_j} \rangle = 0$ for every $i \ne j$, then the $\mathbf{v_i}$'s form a linearly independent set of pairwise orthogonal unit vectors. They may thus be extended to an orthonormal basis. The bound (8.23) then holds since the sum of squares of projection of a vector on vectors in an orthonormal basis *equals* the square of its norm, and hence the sum of squares when restricted to the $\mathbf{v_i}$'s cannot be larger than $\langle \mathbf{x}, \mathbf{x} \rangle$. We need to show this holds even if the $\mathbf{v_i}$'s are more than 90 degrees apart.

Firstly, we can assume $\langle \mathbf{x}, \mathbf{v_i} \rangle > 0$ for $i = 1, 2, \ldots, M$. This is because if $\langle \mathbf{x}, \mathbf{v_i} \rangle = 0$, then it does not contribute to the left hand side of Equation (8.23) and may therefore be discarded. In particular, this implies that we may assume $(\mathbf{v_i} \ne -\mathbf{v_j})$ for any $1 \le i, j \le M$. Since the $\mathbf{v_i}$'s are distinct unit vectors, this means that $|\langle \mathbf{v_i}, \mathbf{v_j} \rangle| < 1$ for all $i \ne j$.

We will prove the claimed bound (8.23) by induction on M. When $M = 1$ the result is obvious. For $M > 1$, we will project the vectors $\mathbf{v_1}, \ldots, \mathbf{v_{M-1}}$, and also $\mathbf{x}$, onto the space orthogonal to $\mathbf{v_M}$. We will then apply the induction hypothesis to the projected vectors and conclude our final bound using the analog of (8.23) for the set of projected vectors. The formal details follow.

For $1 \le i \le M - 1$, define $\mathbf{v_i'} = \mathbf{v_i} - \langle \mathbf{v_i}, \mathbf{v_M} \rangle \mathbf{v_M}$. Since $\mathbf{v_i}$ is different from $\mathbf{v_M}$ and $-\mathbf{v_M}$, each $\mathbf{v_i'}$ is a non-zero vector. Let $\mathbf{u_i}$ be the unit vector associated with $\mathbf{v_i'}$. Let us also define $\mathbf{x'} = \mathbf{x} - \langle \mathbf{x}, \mathbf{v_M} \rangle \mathbf{v_M}$. We wish to apply the induction hypothesis to the vectors $\mathbf{u_1}, \ldots, \mathbf{u_{M-1}}$ and $\mathbf{x'}$.

Now, for $1 \le i < j \le M - 1$, we have $\langle \mathbf{v_i'}, \mathbf{v_j'} \rangle = \langle \mathbf{v_i}, \mathbf{v_j} \rangle - \langle \mathbf{v_i}, \mathbf{v_M} \rangle \langle \mathbf{v_j}, \mathbf{v_M} \rangle \le \langle \mathbf{v_i}, \mathbf{v_j} \rangle \le 0$, since all pairwise dot products between the $\mathbf{v_i}$'s are non-positive. Hence the pairwise dot products $\langle \mathbf{u_i}, \mathbf{u_j} \rangle$, $1 \le i < j \le M - 1$, are all non-positive. To apply the induction hypothesis we should also verify that $\langle \mathbf{x'}, \mathbf{u_i} \rangle > 0$ for $i = 1, 2, \ldots, (M - 1)$. It will be enough to verify that $\langle \mathbf{x'}, \mathbf{v_i'} \rangle > 0$ for each i. But this is easy to check since

$$\begin{aligned}\langle \mathbf{x}', \mathbf{v}'_\mathbf{i} \rangle &= \langle \mathbf{x}, \mathbf{v}_\mathbf{i} \rangle - \langle \mathbf{x}, \mathbf{v}_\mathbf{M} \rangle \cdot \langle \mathbf{v}_\mathbf{i}, \mathbf{v}_\mathbf{M} \rangle \\ &\geq \langle \mathbf{x}, \mathbf{v}_\mathbf{i} \rangle \qquad (8.24) \\ &> 0\end{aligned}$$

where (8.24) follows since $\langle \mathbf{x}, \mathbf{v}_\mathbf{M} \rangle > 0$ and $\langle \mathbf{v}_\mathbf{i}, \mathbf{v}_\mathbf{M} \rangle \leq 0$.

We can therefore apply the induction hypothesis to the $(M-1)$ unit vectors $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_{M-1}}$ and the vector $\mathbf{x}'$. This gives

$$\sum_{i=1}^{M-1} \langle \mathbf{x}', \mathbf{u}_\mathbf{i} \rangle^2 \leq \langle \mathbf{x}', \mathbf{x}' \rangle \ . \qquad (8.25)$$

Now, $\|\mathbf{v}'_\mathbf{i}\|^2 = \langle \mathbf{v}'_\mathbf{i}, \mathbf{v}'_\mathbf{i} \rangle = \langle \mathbf{v}_\mathbf{i}, \mathbf{v}_\mathbf{i} \rangle - \langle \mathbf{v}_\mathbf{i}, \mathbf{v}_\mathbf{M} \rangle^2 \leq \|\mathbf{v}_\mathbf{i}\|^2 = 1 = \|\mathbf{u}_\mathbf{i}\|^2$. This implies that $\langle \mathbf{x}', \mathbf{v}'_\mathbf{i} \rangle \leq \langle \mathbf{x}', \mathbf{u}_\mathbf{i} \rangle$, for $1 \leq i \leq M-1$. Also, by (8.24) $\langle \mathbf{x}', \mathbf{v}'_\mathbf{i} \rangle \geq \langle \mathbf{x}, \mathbf{v}_\mathbf{i} \rangle$, and therefore

$$\langle \mathbf{x}, \mathbf{v}_\mathbf{i} \rangle \leq \langle \mathbf{x}', \mathbf{u}_\mathbf{i} \rangle \ , \qquad (8.26)$$

for $i = 1, 2, \ldots, (M-1)$. Also, we have

$$\langle \mathbf{x}', \mathbf{x}' \rangle = \langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{x}, \mathbf{v}_\mathbf{M} \rangle^2 \ . \qquad (8.27)$$

The claimed result now follows by using (8.26) and (8.27) together with the inequality (8.25). □

8.5.2 The Formal Decoding Algorithm and Its Analysis

We are now ready to state and prove our result about decoding concatenated codes with a general inner code.

Theorem 8.7. *Consider a family of linear q-ary concatenated codes where the outer codes belong to a family of Reed-Solomon codes of relative distance Δ over a field of size at most polynomial in the blocklength, and the inner codes belong to any family of q-ary linear codes of relative distance δ. There is a polynomial time decoding procedure to list decode codes from such a family up to a fractional radius of*

$$\Big(1 - \frac{1}{q}\Big)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) - \sqrt{\delta(1-\Delta)} \ . \qquad (8.28)$$

Proof: (Sketch) Consider a concatenated code C with outer code a Reed-Solomon code over $\mathrm{GF}(q^m)$ of blocklength n_0, relative distance Δ and dimension $(1-\Delta)n_0 + 1$. We assume $q^m \leq n_0^{O(1)}$, so that the field over which the Reed-Solomon code is defined is of size polynomial in the blocklength. Let the inner code C_{in} be any q-ary linear code of dimension m, blocklength n_1 and relative distance δ. Messages of C correspond to polynomials of degree at most $k_0 = (1-\Delta)n_0$ over $\mathrm{GF}(q^m)$, and a polynomial p is encoded as $\langle C_{\text{in}}(p(x_1)), \ldots, C_{\text{in}}(p(x_{n_0}))\rangle$ where $x_1, x_2, \ldots, x_{n_0}$ are distinct elements of $\mathrm{GF}(q^m)$ that are used to define the Reed-Solomon encoding.

The proof parallels that of the earlier result (Theorem 8.2) where the inner code was the Hadamard code. Let $y \in \mathbb{F}_q^n$ be a received word. For $1 \le i \le n_0$, denote by y_i the portion of y in block i of the codeword (namely, the portion corresponding to the encoding by C_{in} of the i^{th} symbol of the outer Reed-Solomon code).

We now perform the "decoding" of each of the n_0 blocks y_i as follows. Let

$$R = n_1(1 - 1/q)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) \tag{8.29}$$

be the Johnson radius of the inner code C_{in}. For $1 \le i \le n_0$ and $\alpha \in \mathrm{GF}(q^m)$, compute the Hamming distance $e_{i,\alpha}$ between y_i and the codeword $C_{\text{in}}(\alpha)$, and then compute the *weight* $w_{i,\alpha}$ as:

$$w_{i,\alpha} \overset{\text{def}}{=} \max\{(R - e_{i,\alpha}), 0\}\ . \tag{8.30}$$

Note the computation of all these weights can be done by a straightforward brute-force computation in $O(n_0 n_1 q^m) = O(n_1 n_0^{O(1)}) = \mathrm{poly}(n)$ time. Thus all the inner decodings can be performed efficiently in polynomial time.

By Proposition 8.5 applied to the y_i's, for $1 \le i \le n_0$, we know that the above weights have the crucial combinatorial property

$$\sum_\alpha w_{i,\alpha}^2 \le \delta n_1^2\ , \tag{8.31}$$

for $i = 1, 2, \ldots, n_0$. We will then run the soft decoding algorithm for Reed-Solomon codes from Theorem 6.26 for this choice of weights. Now, arguing exactly as in the proof of Theorem 8.2 that and using (8.31) above, we conclude that we can find in time polynomial in n and $1/\epsilon$, a list of all polynomials p over $\mathrm{GF}(q^m)$ of degree at most k_0 for which the condition

$$\sum_{i=1}^{n_0} (R - e_{i,p(x_i)}) \ge \sqrt{k_0 n_0 \delta n_1^2} + \epsilon n_1 \tag{8.32}$$

holds. Recalling the definition of R (Equation (8.29)) and using $k_0 = (1 - \varDelta)n_0$, we conclude that we can find a list of all codewords that are at a Hamming distance of at most

$$n\big(1 - \frac{1}{q}\big)\Big(1 - \sqrt{1 - \frac{q\delta}{q-1}}\Big) - n\sqrt{\delta(1-\varDelta)} - \epsilon n_1\ ,$$

from y. Picking $\epsilon < 1/n_1$, we get decoding up to the claimed fraction of errors. □

Comment on the error-correction performance of above: The bound of (8.28) is attractive only for very large values of $\varDelta$, or in other words when the rate of the outer Reed-Solomon code is rather small. For example, for the binary case $q = 2$, even for $\varDelta = 3/4$, the bound does not even achieve the product bound (namely, $\varDelta\delta/2$), for *any* value of δ in the range $0 < \delta < 1/2$ (in fact, the bound as stated in (8.28) is negative unless $\varDelta$ is quite large). However, the merit of the bound is that as $\varDelta$ gets very close to 1, the bound (8.28) approaches the quantity $(1-1/q)(1-\sqrt{1-\frac{q\delta}{q-1}})$, and since the relative designed distance of the concatenated code is $\varDelta \cdot \delta \to \delta$, it approaches the Johnson bound on list decoding radius. Therefore for $\varDelta \to 1$, the result of Theorem 8.7 performs very well and decodes almost up to the Johnson bound, and hence beyond the product bound, for almost the entire range of the inner code distances $0 < \delta < 1/2$. In particular, for $\varDelta \to 1$ and $\delta \to (1 - 1/q)$, the bound tends to $(1 - 1/q)$, permitting us to list decode up to close to the maximum possible fraction $(1 - 1/q)$ of errors.

Alternative Decoding Bound By slightly modifying the analysis used in proving the combinatorial bound of Proposition 8.5, one can prove the following alternative bound instead of (8.8).

$$\sum_{\mathbf{c}\in C}\Big(\max\Big\{\Big(1-\frac{\varDelta(\mathbf{r},\mathbf{c})}{\tilde{R}}\Big),0\ \Big\}\Big)^2 \le \frac{q}{q-1}\ , \tag{8.33}$$

where we use the same notation as in the statement of Proposition 8.5 and $\tilde{R}$ is defined as

$$\tilde{R} \overset{\text{def}}{=} \left(1-\sqrt{1-\frac{q\delta}{q-1}}\right)^2\Big(1-\frac{1}{q}\Big)n\ .$$

(The only change required in the proof is to replace the lower bound on W_i from Equation (8.20) with the alternative lower bound $W_i \ge \big(1 - \frac{e_i}{\tilde{R}}\big)\sqrt{n(q-1)/q}$, which follows easily from the definition of W_i in Equation (8.19).)

Now, replacing the choice of weights in Equation (8.30) in the proof of Theorem 8.7 by

$$w_{i,\alpha} \overset{\text{def}}{=} \max\Big\{\Big(1-\frac{e_{i,\alpha}}{\tilde{R}}\Big),0\Big\}\ ,$$

and then using (8.33), we obtain a decoding algorithm to decode up to a fraction

$$\Big(1-\frac{1}{q}\Big)\left(1-\sqrt{1-\frac{q\delta}{q-1}}\right)^2\left(1-\sqrt{\frac{1-\varDelta}{(1-1/q)}}\right) \tag{8.34}$$

of errors. This bound is positive whenever $\varDelta > 1/q$, and in general appears incomparable to that of (8.28). However, note that even for $\varDelta$ very close to 1, the bound (8.34) does not approach the Johnson bound, except for δ very close to $(1-1/q)$. But as with the bound (8.28), for $\Delta \to 1$ and $\delta \to (1-1/q)$, the above tends to a fraction $(1-1/q)$ of errors. In particular, it can also be used, instead of (8.28), to obtain the results outlined in the next section for highly list decodable codes.

8.5.3 Consequence for Highly List Decodable Codes

We now apply Theorem 8.7 with a suitable choice of parameters to obtain an alternative construction of codes list decodable to a fraction $(1-1/q-\varepsilon)$ of errors and which have rate $\Omega(\varepsilon^6)$. Compared to the construction of Corollary 8.4 that was based on a concatenation of AG-codes with Hadamard codes, the rate is slightly worse – namely by a factor of $O(\log(1/\varepsilon))$. But the following construction offers several advantages compared to that of Corollary 8.4. Firstly, it is based on outer Reed-Solomon codes, and hence does not suffer from the high construction and decoding complexity of AG-codes. In particular, the claim of polynomial time decoding is unconditional and does not depend on having access to precomputed advice information about the outer code. Secondly, the inner code can be *any* linear code of large minimum distance, and not necessarily the Hadamard code. In fact, picking a random code as inner code will give a highly efficient probabilistic construction of the code that has the desired list decodability properties with high probability.

In the next section (Section 8.6) we will present a construction of highly list decodable codes of rate $\Omega(\varepsilon^4)$. Even with this substantial improvement, the bound proved in this section is not strictly subsumed. This is for two reasons. Firstly, the results of Section 8.6 apply only to *binary* linear codes, where as the result below applies to linear codes over any finite field $\mathbb{F}_q$. Secondly, while the deterministic construction complexity of both the constructions in this section and the one with rate $\Omega(\varepsilon^4)$ are almost similar (both of them being fairly high), the codes of this section have very efficient probabilistic constructions, where as we do not know a faster probabilistic construction for the codes of Section 8.6. In conclusion, despite the improvement in rate that will be obtained in Section 8.6, the construction presented next remains interesting.

Theorem 8.8. *For every fixed prime power q, the following holds: For every $\varepsilon > 0$, there exists a family of linear codes over $\mathbb{F}_q$ with the following properties:*

(i) A description of a code of blocklength, say n, in the family can be constructed deterministically in $n^{O(1/\varepsilon^4)}$ time. For probabilistic constructions, a Las Vegas construction can be obtained in time which with high probability will be $O(n \log n/\varepsilon^4)$, or a Monte Carlo construction that has the claimed properties with high probability can be obtained in $O(\log n/\varepsilon^4)$ time.

(ii) Its rate is $\Omega(\varepsilon^6)$ and its relative minimum distance is $(1-1/q-O(\varepsilon^2))$.

(iii) There is a polynomial time list decoding algorithm for every code in the family to perform list decoding up to a fraction $(1 - 1/q - \varepsilon)$ *of errors.*

Proof: We will use Theorem 8.7 with the choice of parameters $\varDelta = 1 - O(\varepsilon^2)$ and $\delta = 1 - 1/q - O(\varepsilon^2)$. Substituting in the bound (8.28), the fraction of errors corrected by the decoding algorithm from Section 8.5.2 will be $(1 - 1/q - \varepsilon)$, which handles Property (iii) claimed above. Also, the relative distance of the code is at least $\varDelta \cdot \delta$, and is thus $(1 - 1/q - O(\varepsilon^2))$, verifying the distance claim in (ii) above. The outer Reed-Solomon code has rate $1 - \varDelta = \varOmega(\varepsilon^2)$. For the inner code, if we pick a random linear code, then it will meet the Gilbert-Varshamov bound $(R = 1 - H_q(\delta))$ with high probability (cf. [193, Chapter 5]). Therefore, a random inner code of rate $\varOmega(\varepsilon^4)$ will have relative distance $\delta = 1 - 1/q - O(\varepsilon^2)$, exactly as we desire. The overall rate of the concatenated code is just the product of the rates of the Reed-Solomon code and the inner code, and is thus $\varOmega(\varepsilon^2 \cdot \varepsilon^4) = \varOmega(\varepsilon^6)$, proving Property (ii).

We now turn to Property (i) about the complexity of constructing the code. We may pick the outer Reed-Solomon code over a field of size at most $O(n)$. Hence, the inner code has at most $O(n)$ codewords and thus dimension at most $O(\log_q n)$. The inner code can be specified by its $O(\log_q n) \times O(\log_q n/\varepsilon^4)$ generator matrix G. To construct an inner code that has relative distance $(1 - 1/q - O(\varepsilon^2))$, we can pick such a generator matrix G at *random*, and then check, by a brute-force search over the at most $O(n)$ codewords, that the code has the desired distance. Since the distance property holds with high probability, we conclude that the generator matrix an inner code with the required rate and distance property can be found in $O(n \log^2 n/\varepsilon^4)$ time with high probability. Allowing for a small probability for error, a Monte Carlo construction can be obtained in $O(\log^2 n/\varepsilon^4)$ probabilistic time by picking a random linear code as inner code (the claimed distance and list decodability properties (ii), (iii) will then hold with high probability). As the outer Reed-Solomon code is explicitly specified, this implies that the description of the concatenated code can be found within the same time bound.

A naive derandomization of the above procedure will require time which is quasi-polynomial in n. But the construction time can be made polynomial by reducing the size of the sample space from which the inner codes is picked. For this, we note that, for every prime power q, there is a small sample space of q-ary linear codes of any desired rate, called a "Wozencraft ensemble" in the literature, with the properties that: (a) a random code can be drawn from this family using a linear (in the blocklength) number of random elements from $\mathbb{F}_q$, and (b) such a code will meet the Gilbert-Varshamov bound with high probability. We record this fact together with a proof as Proposition 8.10 at the end of this section. Applying Proposition 8.10 for the choice of parameters $b = O(\varepsilon^{-4})$, $k = O(\log_q n)$, and using the fact that for small γ, $H_q^{-1}(1 - O(\gamma^2))$ is approximately $(1 - 1/q - O(\gamma))$, we obtain a sample space of linear codes of size $q^{O(\log_q n/\varepsilon^4)} = n^{O(1/\varepsilon^4)}$ which includes a code of rate $\varOmega(\varepsilon^4)$

and relative distance $(1-1/q-O(\varepsilon^2))$. One can simply perform a brute-force search for the desired code in such a sample space. Thus one can find an inner code of rate $\Omega(\varepsilon^4)$ and relative distance $(1-1/q-O(\varepsilon^2))$ deterministically in $n^{O(1/\varepsilon^4)}$ time. Moreover, picking a random code from this sample space, which works just as well as picking a general random linear code, takes only $O(\log n/\varepsilon^4)$ time. This reduces the probabilistic construction times claimed earlier by a factor of $\log n$. Hence a description of the overall concatenated code can be obtained within the claimed time bounds. This completes the verification of Property (i) as well. □

Obtaining an explicit construction: The high deterministic construction complexity or the probabilistic nature of construction in Theorem 8.8 can be removed at the expense of a slight worsening of the rate of the code. One can pick for inner code an *explicitly specified* q-ary code of relative distance $(1-1/q-O(\varepsilon^2))$ and rate $\Omega(\varepsilon^6)$. A fairly simple explicit construction of such codes is known [6] (see also [164]). This will give an *explicit construction* of the overall concatenated code with rate $\Omega(\varepsilon^8)$. We record this below.

Theorem 8.9. *For every fixed prime power q, the following holds: For every $\varepsilon > 0$, there exists a family of* explicitly specified *linear codes over $\mathbb{F}_q$ with the following properties:*

(i)Its rate is $\Omega(\varepsilon^8)$ and its relative minimum distance is $(1-1/q-O(\varepsilon^2))$.
(ii)There is a polynomial time list decoding algorithm for every code in the family to perform list decoding up to a fraction $(1-1/q-\varepsilon)$ of errors.

A Small Space of Linear Codes Meeting the Gilbert-Varshamov Bound We now turn to the result about a small space of linear codes meeting the Gilbert-Varshamov bound. Such an ensemble of codes is referred to as a "Wozencraft ensemble" in the literature. Recall that we made use of such a result in the proof of Theorem 8.8.

Proposition 8.10 (cf. [197]). *For every prime power q, and every integer $b \geq 1$, the following holds. For all large enough k, there exists a sample space, denoted $S_q(b,n)$ where $n \overset{\text{def}}{=} (b+1)k$, consisting of $[n,k]_q$ linear codes of rate $1/(b+1)$ such that:*

(i) There are at most $q^{bn/(b+1)}$ codes in $S_q(b,n)$. In particular, one can pick a code at random from $S_q(b,n)$ using at most $O(n\log q)$ random bits.
(ii) A random code drawn from $S_q(b,n)$ meets the Gilbert-Varshamov bound, i.e. has minimum distance $n\cdot H_q^{-1}\big(\frac{b}{b+1}-o(1)\big)$, with overwhelming (i.e. $1-o(1)$) probability.

Proof: The fact that a code that meets the Gilbert-Varshamov bound can be picked by investing a linear amount of randomness is by now a folklore result. The proof we present here follows the construction due to Weldon [197], which in turn was a generalization of a construction for the rate 1/2 case that

Justesen used in the first explicit construction of a family of asymptotically good binary codes [110].

Let α be a primitive element of the finite field $\mathrm{GF}(q^k)$, so that $\{\alpha^i : 0 \leq i < q^k - 1\}$ are all the non-zero elements of $\mathrm{GF}(q^k)$. A code in $S_q(b,n)$ will be specified by a b-tuple $\mathcal{I}_b = (i_0, i_1, \ldots, i_{b-1})$ where each i_s, $0 \leq s \leq b-1$, is an integer that satisfies $0 \leq i_s \leq q^k - 1$. Note that there are $q^{kb} = q^{bn/(b+1)}$ codes in the sample space $S_q(b,n)$, since there are exactly so many b-tuples. A random code in $S_q(b,n)$ can be picked by choosing a random b-tuple $\mathcal{I}_b$. Hence the sample space $S_q(b,n)$ meets the requirement (i).

A message $\mathbf{a} \in \mathbb{F}_q^k$, will be encoded by a code indexed by a b-tuple $(i_0, i_1, \ldots, i_{b-1})$ as follows: view $\mathbf{a}$ as a field element $\gamma \in \mathrm{GF}(q^k)$ (using some fixed representation of $\mathrm{GF}(q^k)$ over $\mathrm{GF}(q)$), then encode it as $\langle \gamma, \gamma\alpha^{i_0}, \gamma\alpha^{i_1}, \ldots, \gamma\alpha^{i_{b-1}} \rangle$. This gives a $(b+1)$-tuple over $\mathrm{GF}(q^k)$ or equivalently a word of length $(b+1)k = n$ over $\mathrm{GF}(q)$, as desired.

The crucial observation used to prove (ii) is the following. Any non-zero vector $\mathbf{v} \in \mathbb{F}_q^n$ can belong to at most one of the codes in $S_q(b,n)$. Indeed, it is easily checked that the b-tuple associated with a code containing $\mathbf{v}$ can be uniquely reconstructed from $\mathbf{v}$. Property (ii) is now a simple consequence of this fact. Indeed, the number of vectors $\mathbf{v} \in \mathbb{F}_q^n$ of Hamming weight at most w is at most $q^{H_q(w/n)n}$ (see for example [193, Chapter 1]). Applying this to $w = d \overset{\text{def}}{=} n \cdot H_q^{-1}\big(\frac{b}{b+1} - \zeta\big)$, the number of vectors of Hamming weight less than or equal to d is at most $q^{(\frac{b}{b+1} - \zeta)n}$. Since a non-zero vector belongs to at most one code among those in $S_q(b,n)$, this implies that the fraction of codes in $S_q(b,n)$ that have some non-zero codeword of weight less than or equal to d is at most $q^{-\zeta n}$. Picking $\zeta = o(1)$, say $1/\sqrt{n}$, we conclude that a random code from $S_q(b,n)$ has minimum distance greater than $n \cdot H_q^{-1}\big(\frac{b}{b+1} - o(1)\big)$ with very high (i.e., $(1 - o(1))$) probability. □

8.6 Improved Rate Using Tailor-Made Concatenated Code

We now proceed to a construction of highly list decodable codes that improves over the rate of ε^6 that was achieved by Theorem 8.8 (and also by Corollary 8.4). The results of this section apply *only* to binary linear codes. Recall that binary codes that can be list decoded from $(1/2 - \varepsilon)$ errors using polynomial sized lists can have rate at best $\Omega(\varepsilon^2)$. We will be able to attain a rate of $\Omega(\varepsilon^4)$. The formal result is stated below.

Theorem 8.11. *There exist absolute constants $b, d > 0$ such that for each fixed $\varepsilon > 0$, there exists a polynomial time constructible* binary linear *code family $\mathcal{C}$ with the following properties:*

1. *A code of blocklength N from the family $\mathcal{C}$ can be constructed in $N^{O(1/\varepsilon^2)}$ time deterministically.*

2. *The rate $R(\mathcal{C})$ of $\mathcal{C}$ is at least $\frac{\varepsilon^4}{b}$, and its relative distance $\delta(\mathcal{C})$ is at least $(1/2 - \varepsilon)$.*
3. *There is a* polynomial time list decoding algorithm *that can list decode codes in $\mathcal{C}$ from up to a fraction $(1/2 - \varepsilon)$ of errors, using lists of size at most d/ε^2.* □

The above theorem will follow from Theorem 8.14, which is stated and proved in Section 8.6.2. The basic idea is to use a concatenated code with the outer code being a Reed-Solomon code and the inner code being a "tailor-made" one. The inner code will be chosen so that it possesses a rather peculiar looking combinatorial property, which is formalized in Lemma 8.12. This property will be very useful when it is used in conjunction with the soft decoding algorithm for Reed-Solomon codes (Theorem 6.26). We first turn to the existence and construction of the necessary inner code.

8.6.1 The Inner Code Construction

Existence of a "Good" Code We now prove the existence of codes that will serve as excellent inner codes in our later concatenated code construction. The proof is an adaptation of that of Theorem 5.8. We will then show how such a code can be constructed in $2^{O(n)}$ time (where n is the blocklength) using an iterative greedy procedure.

Lemma 8.12. *There exist absolute constants $\sigma, A > 0$ such that for any $\varepsilon > 0$ there exists a binary linear code family $\mathcal{C}$ with the following properties:*

1. *The rate of the family satisfies $R(\mathcal{C}) = \sigma\varepsilon^2$*
2. *For every code $C \in \mathcal{C}$ and every $x \in \{0,1\}^n$ where n is the blocklength of C, we have*

$$\sum_{\substack{\mathbf{c} \in C \\ \Delta(x,\mathbf{c}) \le (1/2-\varepsilon)n}} \left(1 - \frac{2\Delta(x,\mathbf{c})}{n}\right)^2 \le A\ . \tag{8.35}$$

Proof: For every large enough n, we will prove the existence of a binary linear code C_k of blocklength n and dimension $k \ge \sigma\varepsilon^2 n$ which satisfies Condition (8.35) for every $x \in \{0,1\}^n$.

The proof will follow very closely the proof of Theorem 5.8 and in particular we will again build the code C_k iteratively in k steps by randomly picking the k linearly independent basis vectors $b_1, b_2, \ldots, b_k$ in turn. Define $C_i = \mathrm{span}(b_1, \ldots, b_i)$ for $1 \le i \le k$ (and define $C_0 = \{\mathbf{0}\}$). The key to our proof is the following potential function W_C defined for a code C of blocklength n (compare with the potential function (5.12) from the proof of Theorem 5.8):

$$W_C = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \exp_2\left(\frac{n}{A} \cdot \sum_{\mathbf{c} \in C : \Delta(x,\mathbf{c}) \le (1/2-\varepsilon)n} \left(1 - \frac{2\Delta(x,\mathbf{c})}{n}\right)^2\right)\ , \tag{8.36}$$

where, for readability, we used $\exp_2(z)$ to denote 2^z. (The constant A will be fixed later in the proof, and we assume that $A > \ln 4$.) Denote the random variable W_{C_i} by the shorthand W_i. For $x \in \{0,1\}^n$, define

$$R_x^i = \sum_{\substack{\mathbf{c} \in C_i \\ \Delta(x,\mathbf{c}) \le (1/2-\varepsilon)n}} \Big(1 - \frac{2\Delta(x,\mathbf{c})}{n}\Big)^2 , \tag{8.37}$$

so that

$$W_i = 2^{-n} \sum_x \exp_2\Big(\frac{n}{A} \cdot R_x^i\Big) .$$

Now, exactly as in the proof of Theorem 5.8, we have $R_x^{i+1} = R_x^i + R_{x+b_{i+1}}^i$ when b_{i+1} is picked outside the span of $\{b_1, b_2, \ldots, b_i\}$. Now, arguing as in the proof of Theorem 5.8, one can deduce that

$$\mathbf{E}[W_{i+1} | W_i = \hat{W}_i] \le \frac{\hat{W}_i^2}{1 - 2^{i-n}} . \tag{8.38}$$

when the expectation is taken over a random choice of b_{i+1} outside $\mathrm{span}(b_1, \ldots, b_i)$.

Applying (8.38) repeatedly for $i = 0, 1, \ldots, k-1$, we conclude that there exists an $[n,k]_2$ binary linear code $C = C_k$ with

$$W_C = W_k \le \frac{W_0^{2^k}}{1 - k2^{k-n}} . \tag{8.39}$$

If we could prove, for example, that $W_C = O(1)$, then this would imply, using (8.36), that $R_x^k \le A$ for every $x \in \{0,1\}^n$ and thus C would satisfy Condition (8.35), as desired. To show this, we need an estimate of (upper bound on) W_0, to which we turn next.

Define $a = (1/2 - \varepsilon)n$. Since C_0 consists of only the all-zeroes codeword, we have $R_x^0 = (1 - 2\mathrm{wt}(x)/n)^2$ if $\mathrm{wt}(x) \le a$ and $R_x^0 = 0$ otherwise (here we use $\mathrm{wt}(x) = \Delta(x, \mathbf{0})$ to denote the Hamming weight of x). We now have

$$\begin{aligned} W_0 &= 2^{-n} \sum_{x \in \{0,1\}^n} \exp_2\big(\frac{n}{A} R_x^0\big) \\ &\le 1 + 2^{-n} \sum_{i=0}^{a} \binom{n}{i} \exp_2\Big(\frac{n}{A}\Big(1 - \frac{2i}{n}\Big)^2\Big) \\ &\le 1 + n2^{-n} \exp_2\Big(\max_{0 \le i \le a} \Big\{ H\Big(\frac{i}{n}\Big)n + \frac{4n}{A}\Big(\frac{1}{2} - \frac{i}{n}\Big)^2 \Big\} \Big) \\ &\le 1 + n2^{un} \end{aligned} \tag{8.40}$$

where $u \overset{\text{def}}{=} \max_{0 \le y \le (1/2-\varepsilon)} \big\{ H(y) - 1 + \frac{4}{A}(\frac{1}{2} - y)^2 \big\}$. We now claim that for every y, $0 \le y \le 1/2$, we have $H(y) \le 1 - \frac{2}{\ln 2}(\frac{1}{2} - y)^2$. One way to prove this is to consider the Taylor expansion around $1/2$ of $H(y)$, which is valid for the range $0 \le y \le 1/2$. We have $H'(1/2) = 0$ and $H''(1/2) = -4/\ln 2$. Also it is easy to check that all odd derivatives of $H(y)$ at $y = 1/2$ are non-negative while the even derivatives are non-positive. Thus $H(y) \le H(1/2) - H''(1/2)\frac{(1/2-y)^2}{2} = 1 - \frac{2}{\ln 2}(\frac{1}{2} - y)^2$. Therefore

$$u \le \max_{0 \le y \le (1/2-\varepsilon)} \Big(\frac{4}{A} - \frac{2}{\ln 2}\Big)\Big(\frac{1}{2} - y\Big)^2 = -4\Big(\frac{1}{\ln 4} - \frac{1}{A}\Big)\varepsilon^2 \ , \qquad (8.41)$$

since $A > \ln 4$. Combining (8.39), (8.40) and (8.41), it is now easy to argue that we will have $W_C = W_k = O(1)$ as long as $k < -un$, which will be satisfied if $k < 4(\frac{1}{\ln 4} - \frac{1}{A})\varepsilon^2 n$. Thus the statement of the lemma holds, for example, with $A = 2$ and $\sigma = 0.85$. □ *(Lemma 8.12)*

Remark: Arguing exactly as in the remark following the proof of Theorem 5.8, one can also add the condition $\delta(\mathcal{C}) \ge (1/2 - \varepsilon)$ to the claim of Lemma 8.12. The proof will then pick b_{i+1} randomly from among all choices such that $\mathrm{span}(b_1, b_2, \ldots, b_{i+1}) \cap B(\mathbf{0}, (\frac{1}{2} - \varepsilon)n) = \emptyset$.

A Greedy Construction of the "Inner" Code We now discuss how a code guaranteed by Lemma 8.12 can be constructed in a greedy fashion. We will refer to some notation that was used in the proof of Lemma 8.12. The algorithm works as follows:

Algorithm GREEDY-INNER:

Parameters: Dimension k; $\varepsilon, A > 0$ (where A is the absolute constant from Lemma 8.12)

Output: A binary linear code $C = $ GREEDY(k, ε) with dimension k, block-length $n = O(k/\varepsilon^2)$ and minimum distance at least $(1/2 - \varepsilon)n$ such that for every $x \in \{0,1\}^n$, Condition (8.35) holds.

```
1. Start with $b_0 = \mathbf{0}$.
2. For $i = 1, 2, \ldots, k$:
   – Let $U_i = \{x \in \{0,1\}^n : \mathrm{span}(b_1, b_2, \ldots, b_{i-1}, x) \cap B(\mathbf{0}, (1/2-\varepsilon)n) = \emptyset\ \}$.
   – Pick $b_i \in U_i$ that *minimizes* the potential function $W_i = 2^{-n} \sum_x 2^{\frac{n}{A} \cdot R_x^i}$,
     where $R_x^i$ is as defined in Equation (8.37) (break ties arbitrarily)
3. Output $C = \mathrm{span}(b_1, b_2, \ldots, b_k)$.
```

The following result easily follows from the proof of Lemma 8.12 since each of the k iterations of the for loop above can be implemented to run in $2^{O(n)}$ time.

Lemma 8.13. *Algorithm* GREEDY-INNER *constructs a code* GREEDY(k, ε) *with the desired properties in* $k \cdot 2^{O(n)}$ *time.*

8.6.2 The Concatenated Code and the Decoding Algorithm

The statement of Theorem 8.11 that we set out to prove, follows immediately from the concatenated code construction guaranteed by the following theorem.

Theorem 8.14. *There exist absolute constants $b, d > 0$ such that for every integer K and every $\varepsilon > 0$, there exists a concatenated code $C_K \overset{\text{def}}{=} \text{RS} \oplus$ $\textsc{Greedy}(m, \varepsilon/2)$ (for a suitable parameter m) that has the following properties:*

1. *C_K is a linear code of dimension K, blocklength $N \leq \frac{bK}{\varepsilon^4}$, and minimum distance at least $(\frac{1}{2} - \varepsilon)N$.*
2. *The generator matrix of C_K can be constructed in $N^{O(\varepsilon^{-2})}$ time.*
3. *C_K is $((\frac{1}{2} - \varepsilon)N, d/\varepsilon^2)$-list decodable; i.e. any Hamming ball of radius $(1/2 - \varepsilon)N$ has at most $O(\varepsilon^{-2})$ codewords of C_K.*
4. *There exists a polynomial time list decoding algorithm for C_K that can correct up to $(1/2 - \varepsilon)N$ errors.*

Proof: The code C_K is constructed by concatenating an outer Reed-Solomon code C_{RS} over $\text{GF}(2^m)$ of blocklength $n_0 = 2^m$ and dimension $k_0 = K/m$ (for some integer m which will be specified later in the proof) with an inner code $C_{\text{inner}} = \textsc{Greedy}(m, \varepsilon/2)$ (as guaranteed by Lemma 8.13). Since the blocklength of C_{inner} is $n_1 = O(\frac{m}{\varepsilon^2})$, the concatenated code C_K has dimension K and blocklength

$$N = n_0 n_1 = O\Big(\frac{n_0 m}{\varepsilon^2}\Big) . \tag{8.42}$$

and minimum distance D at least

$$D \geq N\Big(1 - \frac{K}{mn_0}\Big)\Big(\frac{1}{2} - \frac{\varepsilon}{2}\Big) . \tag{8.43}$$

For ease of notation, we often hide constants using the big-Oh notation in what follows, but in all these cases the hidden constants will be absolute constants that do not depend upon ε. By Lemma 8.13, C_{inner} is constructible in $2^{O(n_1)} = 2^{O(m/\varepsilon^2)}$ time, and since $m = \lg n_0$, the generator matrix for C_K can be constructed in $N^{O(\varepsilon^{-2})}$ time. This proves Property 2 claimed in the theorem.

We will now present a polynomial time list decoding algorithm for C_K to correct a fraction $(1/2 - \varepsilon)$ of errors using lists of size $O(1/\varepsilon^2)$. This will clearly establish both Properties 3 and 4 claimed in the theorem.

The decoding algorithm will follow the same approach as that of Theorems 8.2 and 8.7. Let $y \in \{0, 1\}^N$ be any received word. We wish to find a list of all codewords $\mathbf{c} \in C_K$ such that $\Delta(y, \mathbf{c}) \leq (1/2 - \varepsilon)N$. For $1 \leq i \leq n_0$, denote by y_i the portion of y in block i of the codeword (i.e. the portion corresponding to the encoding by C_{inner} of the i^{th} Reed-Solomon symbol).

Now, consider the following decoding algorithm for C_K. First, the inner codes are decoded by a brute force procedure that goes over all codewords. Specifically, for each position i, $1 \le i \le n_0$, of the outer Reed-Solomon code, and for each $\alpha \in \mathrm{GF}(2^m)$, the inner decoder computes a set of weights $w_{i,\alpha}$ defined by:

$$w_{i,\alpha} = \max\Big\{\Big(\frac{1}{2} - \frac{\varepsilon}{2} - \Delta\big(y_i, C_{\mathrm{inner}}(\alpha)\big)\Big), 0\Big\} \tag{8.44}$$

Once again all the n_0 inner decodings can be performed in $O(n_0 \cdot 2^m \cdot m/\varepsilon^2) = O(n_0^2 m/\varepsilon^2)$ time, and thus certainly in $O(N^2)$ time.

These weights are then passed to the soft decoding algorithm for Reed-Solomon codes from Theorem 6.26. To analyze the performance of the soft decoding algorithm, we will make use of the crucial combinatorial property of C_{inner} which is guaranteed by Lemmas 8.12 and 8.13. Using this property of C_{inner}, we have, for each i, $1 \le i \le n_0$,

$$\sum_{\alpha \in \mathrm{GF}(2^m)} w_{i,\alpha}^2 \le B' \ , \tag{8.45}$$

for some *absolute* constant B'.

Using the soft decoding algorithm to complete the decoding implies that one can find, in time polynomial in n_0 and $1/\gamma$, a list of *all* codewords $\mathbf{c} \in C_K$ that satisfy

$$\sum_{i=1}^{n_0} w_{i,\mathbf{c}_i} \ge \sqrt{\Big(n_0 - \frac{n_0 - K/m + 1}{1+\gamma}\Big) \cdot \sum_{i,\alpha} w_{i,\alpha}^2} \ . \tag{8.46}$$

In the above, $\gamma > 0$ is a parameter to be set later, and we have abused notation to denote $w_{i,\mathbf{c}_i} = w_{i,\alpha_i}$ where $\alpha_i \in \mathrm{GF}(2^m)$ is such that $C_{\mathrm{inner}}(\alpha_i) = \mathbf{c}_i$.

The soft decoding algorithm, used as stated in Theorem 6.26, can decode even with the choice $\gamma = 0$ in the above Condition (8.46). However, with a positive value of γ, we can appeal to the weighted Johnson bounds from Chapter 3, specifically the result stated in Part (ii) of Corollary 3.7, to conclude that there will be at most $(1 + 1/\gamma)$ codewords $\mathbf{c}$ that satisfy Condition (8.46) for *any* choice of weights $w_{i,\alpha}$. Hence, our decoding algorithm, too, will output only a list of at most $O(1/\gamma)$ codewords.

We now analyze the number of errors corrected by the algorithm. Using (8.44) and (8.45), we notice that Condition (8.46) will be satisfied if

$$\begin{aligned} \sum_{i=1}^{n_0}\Big(\frac{1}{2} - \frac{\varepsilon}{2} - \frac{\Delta(y_i, \mathbf{c}_i)}{n_1}\Big) &\ge \sqrt{\Big(\gamma n_0 + \frac{K}{m}\Big) \cdot n_0 B'} \\ \Longleftarrow \quad \Delta(y, \mathbf{c}) &\le N\Big(\frac{1}{2} - \frac{\varepsilon}{2} - \sqrt{B'\Big(\gamma + \frac{K}{m n_0}\Big)}\Big) \\ \Longleftarrow \quad \Delta(y, \mathbf{c}) &\le \Big(\frac{1}{2} - \varepsilon\Big)N \ , \end{aligned}$$

where the last step holds as long as we pick $\gamma \leq \frac{\varepsilon^2}{8B'}$ and m such that

$$\frac{K}{mn_0} = \frac{K}{m2^m} \leq \frac{\varepsilon^2}{8B'} \ . \tag{8.47}$$

Thus we have a decoding algorithm that outputs a list of all $O(1/\gamma) = O(\varepsilon^{-2})$ codewords that differ from y in at most $(1/2-\varepsilon)N$ positions. This establishes Properties 3 and 4 claimed in the theorem.

Also, by (8.47), we have $mn_0 = O(K/\varepsilon^2)$. Plugging this into (8.42) and (8.43), we have that the blocklength N of C_K satisfies $N = O(K/\varepsilon^4)$ and the distance D satisfies $D \geq (1/2-\varepsilon)N$. This establishes Property 1 as well, and completes the proof of the theorem. □

Discussion: The time required to construct a code with the properties claimed in Theorem 8.14, though polynomial for every fixed ε, grows as $N^{O(\varepsilon^{-2})}$. It is desirable to obtain a construction time of the form $O(f(\varepsilon)n^c)$ where c is a fixed constant independent of ε, for some arbitrary function f. A family whose codes can be constructed within such time bounds is often referred to as being *uniformly constructive* (see [6] for a formal definition).

If one uses the best known algebraic-geometric codes (namely those discussed in Section 6.3.9) as the outer code instead of Reed-Solomon codes, one can carry out the code construction of Theorem 8.14 in $2^{O(\varepsilon^{-2}\log(1/\varepsilon))}N^c$ time for a fixed constant c (the constant c will depend upon the time required to construct the outer algebraic-geometric code). This is not entirely satisfying since the construction complexity of the necessary algebraic-geometric codes is still quite high. A further drawback is that the promise of a polynomial time decoding algorithm will hinge on assumptions about specific representations of the AG-code.

The construction of Theorem 8.8 had a similar drawback in terms of high deterministic construction time. Nevertheless, it had a highly efficient probabilistic construction that had the claimed properties with high probability. A similar probabilistic construction for the codes of Theorem 8.11 is not known. The reason for this is that the existence result of Lemma 8.12 is not known to hold with high probability for a *random* code (unlike the situation in Theorem 8.8 where it is known that the rate vs. distance trade-off of a random linear code meets the Gilbert-Varshamov bound with high probability). Thus the following is an interesting open question:

Question 8.15. 1. Is there a randomized (Monte Carlo) construction of a family of binary linear codes of rate $\Omega(\varepsilon^4)$ list decodable up to a fraction $(1/2-\varepsilon)$ of errors, that runs in, say, quadratic time in the blocklength?

2. Is there a uniformly constructive family of binary linear codes which can be list decoded efficiently from a fraction $(1/2-\varepsilon)$ errors and which have rate $\Omega(\varepsilon^4)$ or better?

8.7 Open Questions

In addition to the above, there are two central open questions regarding the contents of this chapter. These are listed below.

Question 8.16. Let C be a q-ary concatenated code of designed distance $\varDelta \cdot \delta$ with the outer code being a Reed-Solomon code of relative distance $\varDelta$, and the inner code being an arbitrary q-ary code of relative distance δ. Is there a polynomial time list decoding algorithm for C to decode up to its Johnson radius? In other words, is there a polynomial time algorithm to list decode up to a fraction $(1-1/q)\big(1-\sqrt{1-\frac{\varDelta\cdot\delta}{(1-1/q)}}\big)$ of errors?

In fact the following "easier" question is also open. As mentioned earlier, the GMD algorithm can be used to unique decode such codes up to the product bound (i.e. a fraction $\varDelta\delta/2$ of errors) in polynomial time [59, 110]. The question below simply asks if one can always, for every concatenated code with an outer Reed-Solomon code, perform efficient list decoding beyond the product bound.

Question 8.17. Let C be a q-ary concatenated code of designed distance $\varDelta \cdot \delta$ with the outer code being a Reed-Solomon code of relative distance $\varDelta$, and the inner code being an arbitrary q-ary code of relative distance δ. Is there a polynomial time list decoding algorithm for C to decode up to a fraction $f(\varDelta, \delta)$ of errors, where f is a real-valued function that takes values in $[0, 1-1/q)$ and which satisfies $f(\varDelta, \delta) > \frac{\varDelta\delta}{2}$ in the entire range $0 < \varDelta < 1$ and $0 < \delta < 1 - 1/q$? In other words, is there a polynomial time algorithm to always list decode such concatenated codes beyond the product bound ?

Finally, we state the open question concerning the best rate of a constructive family of binary codes with very high list decodability.

Question 8.18. Is there a polynomial time constructible family of binary codes which have rate $\varOmega(\varepsilon^a)$ for some $a < 4$ and which have a polynomial time list decoding algorithm to decode up to a fraction $(1/2 - \varepsilon)$ of errors ?

We know that existentially $a = 2$ is achievable and that this is the best possible.

We note that even if Question 8.16 is answered in the affirmative, the rate achievable for a list decoding radius of $(1 - 1/q - \varepsilon)$ is only $O(\varepsilon^6 \log(1/\varepsilon))$. This is because we need to have $\varDelta = 1 - O(\varepsilon^2)$ and $\delta = (1 - 1/q - O(\varepsilon^2))$ in order for the Johnson radius to be $(1 - 1/q - \varepsilon)$. The former implies that the rate of the Reed-Solomon code is $O(\varepsilon^2)$ and the latter, by appealing to the linear programming bounds [139], implies that the rate of the inner code is $O(\varepsilon^4 \log(1/\varepsilon))$. The overall rate is thus at most $O(\varepsilon^6 \log(1/\varepsilon))$. An answer in the affirmative to Question 8.18, therefore, has to either not be based on concatenation at all, or must use a special purpose construction, akin to the one in Section 8.6, which can be list decoded beyond its Johnson radius. In the next chapter, we will present a probabilistic construction with $\Omega(\varepsilon^3)$ rate, but the decoding time will be sub-exponential as opposed to polynomial.

8.8 Bibliographic Notes

Concatenated codes were defined and studied extensively in the seminal Ph.D. work of Forney [59], and by now have deservedly become standard textbook material. Forney [60] developed a Generalized Minimum Distance (GMD) decoding algorithm for Reed-Solomon codes, and used it as a soft decoding algorithm to decode concatenated schemes with outer Reed-Solomon code. He presented a detailed estimation of the probability of decoding error for such a scheme. Justesen [110] used a concatenated scheme to give the first explicit construction of an asymptotically good binary code family, thereby refuting the popular myth existing at that time that explicitly specified codes would probably never be asymptotically good. Justesen also gave an algorithm using GMD decoding to decode his concatenated codes up to the product bound (i.e. half the designed distance). In fact, his result implicitly shows that any concatenated code whose outer code has an efficient errors-and-erasures decoding algorithm (which in turn implies a GMD algorithm by results of Forney [60]) can be uniquely decoded up to the product bound. The GMD based algorithm for unique decoding concatenated codes up to the product bound is also described in detail in Appendix A of this book.

The inner decoding stage in all these algorithms passed to the outer Reed-Solomon decoder at most one field element together with an associated weight (confidence information) for each outer codeword position. This was also the case in a recent work of Nielsen [145] who investigated in detail decoding algorithms for concatenated codes where the inner code is decoded uniquely but instead of the GMD algorithm, the weighted list decoding algorithm (from Chapter 6) is used for decoding the outer Reed-Solomon code. In contrast, in the algorithms discussed in this chapter, the inner decoders pass to the outer Reed-Solomon decoder not one, but several field elements, each with an associated weight, as candidate symbols for each position. We should mention that Nielsen [145] also considers a decoding algorithm where the inner codes are list decoded beyond half the minimum distance, but does not present a quantitative analysis of such an algorithm. Indeed to perform such an analysis one needs at least a partial knowledge of the weight distribution of cosets of the inner code, which is a highly non-trivial task in itself. The result of Proposition 8.5 from this chapter provides a non-trivial, and apparently new, bound on the weight distribution of cosets given the knowledge of *only* the minimum distance of the code. We believe, though, that to really reap the benefits of the soft Reed-Solomon decoder in concatenated code constructions, one must use special purpose inner codes for which we have good bounds on the weight distributions of cosets. In fact, our results in Section 8.6 follow this approach, but we believe there is still lots of improvements to be made.

The decoding algorithms from Section 8.4 when the inner code is the Hadamard code appear in [89]. The results of Section 8.5 appear in [90]. The code construction and decoding algorithm of Section 8.6 appear in [80].