Decoding Concatenated Codes using Soft Information

Venkatesan Guruswami*
University of California at Berkeley
Computer Science Division
Berkeley, CA 94720.
venkat@lcs.mit.edu

Madhu Sudan†
MIT Laboratory for Computer Science
200 Technology Square
Cambridge, MA 02139.
madhu@mit.edu

Abstract

We present a decoding algorithm for concatenated codes when the outer code is a Reed-Solomon code and the inner code is arbitrary. *"Soft" information on the reliability of various symbols is passed by the inner decodings and exploited in the Reed-Solomon decoding. This is the first analysis of such a soft algorithm that works for* arbitrary inner codes*; prior analyses could only handle some special inner codes. Crucial to our analysis is a combinatorial result on the coset weight distribution of codes given only its minimum distance. Our result enables us to decode essentially up to the "Johnson radius" of a concatenated code when the outer distance is large (the* Johnson radius *is the "a priori list decoding radius" of a code as a function of its distance). As a consequence, we are able to present simple and efficient constructions of q-ary linear codes that are list decodable up to a fraction* $(1 - 1/q - \varepsilon)$ *of errors and have rate* $\Omega(\varepsilon^6)$*. Codes that can correct such a large fraction of errors have found numerous complexity-theoretic applications. The previous constructions of linear codes with a similar rate used algebraic-geometric codes and thus suffered from a complicated construction and slow decoding.*

1. Introduction

Concatenation of codes presents a simple, yet powerful tool to construct good codes over small (eg., binary) alphabets by combining an "outer" code over a large alphabet, often an algebraic code with nice properties, with a good "inner" code over a small alphabet. The basic idea behind code concatenation is to first encode the message using the outer code and then encode each of the symbols of the outer codeword further using the inner code. The size of the inner code is small enough to permit efficient constructions of good codes and efficient encoding/decoding, and this translates into similar claims for the concatenated code.

Since its discovery by Forney [For66a], code concatenation has become one of the most widely used tools in all of coding theory. Forney's original motivation was to prove that one could approach Shannon capacity with polynomial decoding complexity, and he did so by using suitable concatenated codes with an outer Reed-Solomon code. One of the most fundamental problems in coding theory is the construction of asymptotically good code families whose rate and relative distance are both positive, and until the recent work of Sipser and Spielman [SS96], concatenated codes comprised the *only known* explicit (or even polynomial time) constructions of asymptotically good codes. The explicit binary code constructions with the current best known trade-offs between rate and minimum distance are still based on concatenation.

In light of the pervasiveness of concatenated codes, the problem of decoding them efficiently is an important one which has received considerable attention; a thorough discussion of some of the basic results in this area can be found in [Dum98]. A concatenated code whose outer and inner codes have distance at least D and d respectively, has minimum distance at least Dd. Forney [For66b] presented an elegant and general algorithm based on Generalized Minimum Distance (GMD) decoding to correct such a code from fewer than $Dd/2$ errors (this bound is called the *product bound*). The GMD algorithm is efficient as long as the inner code has polynomially many codewords and the outer code has an efficient errors-and-erasures decoding algorithm.

Recently, with the spurt of activity on the subject of *list decoding*, there is a renewed interest in decoding important families of codes beyond the half-the-distance bound. Specifically the goal is to find a list of all codewords that differ from the received word in at most e places for some bound e which is much bigger than $d/2$ (but is still small enough so that the list that the decoding algorithm has to output is guaranteed to be small). Following the powerful list decoding algorithms for Reed-Solomon and algebraic-

*Supported by a Miller Postdoctoral Fellowship.
†Supported by MIT-NTT Award MIT2001-04, and NSF CCR-9875511 and NSF CCR-9912342.

geometric codes [Sud97, GS99, SW99], list decoding algorithms with good error-correction performance have been proposed for certain concatenated codes with outer Reed-Solomon or AG-codes [GS00, STV01, Nie00, GHSZ00, KV01].

In this paper, we present a list decoding algorithm for concatenated codes when the outer code is a Reed-Solomon code and the inner code is *arbitrary*. Quantitatively, ours is the first algorithm which decodes up to essentially the *Johnson radius* of a concatenated code when the inner code is arbitrary and the outer code is a Reed-Solomon code with large distance. (The Johnson radius is the "a priori list decoding potential" of a code and is the largest general bound on number of errors for which list decoding with "small" lists is possible.) To explain what is new in our result compared to the several previous algorithms and analyses, we next elaborate on the basic operational structure of the known list decoding algorithms for concatenated codes.

Let n_0 be the block length of the outer code. The received word of a concatenated code is broken up into n_0 blocks which correspond to the inner encodings of the n_0 outer codeword positions. Each of these n_0 blocks is decoded by an inner decoder to produce some information about the i'th symbol of the outer codeword, for every i, $1 \leq i \leq n_0$. This information from the inner decodings is then passed to the outer decoder which uses it to list decode the outer code and produce a final list of answers. This is the general scheme which all known list decoding algorithms for concatenated codes, including ours in this paper, follow. The main difference between the various algorithms is in the exact inner decoding used and the nature of information passed to the outer decoding stage.

Our goal is to correct as many errors as possible by a careful choice of the nature of information passed by the inner decoder. In most previously known decoding algorithms, the inner decodings have either passed a *single* symbol corresponding to each position of the outer code, possibly together with a "weight" which is a quantitative estimate of the decoder's confidence on that symbol being the correct one (eg. in [For66b, Nie00]), or they have passed a list of potential symbols (without any confidence information to rank them) for each position (eg. in [STV01, GS00]). The exceptions to this are the algorithms in [GS00] for the case when the inner code is an Hadamard code, where every possible symbol is returned by the inner code along with an associated weight. Such an algorithm is referred as decoding with *soft information* in coding theory parlance, since the inner decoder qualifies its vote with a "soft" reliability information. More recently, Koetter and Vardy [KV01] gave such an algorithm when the inner code is the extended binary Golay code. In both these cases, the special nature of the inner code (specifically, its so-called *coset weight distribution*) was crucially exploited in setting the weights and analyzing the performance of the overall algorithm. Our algorithm follows the spirit of these algorithms in [GS00, KV01], and each inner decoding returns a list of symbols each with its associated weight. These weights are then passed to the outer decoder which, as is by now standard, is the weighted (or, *soft*) Reed-Solomon list decoding algorithm due to the authors [GS99]. The novelty in our approach is that we are able to present a suitable setting of weights in the inner decoding and quantitatively analyse the performance of the entire algorithm for *every choice of the inner code*, using only the distance of the Reed-Solomon and inner codes in the analysis. Along the way, we also prove a combinatorial property concerning the coset weight distribution of a code given only information about its minimum distance.

We now discuss a concrete goal which motivated much of the prior work (eg. those in [STV01, GS00, GHSZ00]) on decoding concatenated codes, and where our result yields a quantitative improvement over the previous results. The goal is to construct simple and explicit linear code families over small finite fields $\mathbb{F}_q$ that can be efficiently list decoded from up to a fraction $(1 - 1/q - \varepsilon)$ of errors (think of ε as an arbitrarily small positive constant). This represents the "high-noise" regime and is information-theoretically the largest fraction of errors one can hope to decode from (since a random received word will differ from each codeword in an expected fraction $(1 - 1/q)$ of positions). Having fixed the desired error-resilience of the code, the goal is to achieve good rate together with fast construction and decoding times.

One of the main motivations for the study of such codes which can correct a fraction $(1 - 1/q - \varepsilon)$ of errors comes from complexity theory where they have applications in hardness amplification (and as a consequence in derandomization) [STV01], constructions of generic hardcore predicates from one-way permutations (see [Sud00]), constructions of pseudorandom generators [SU01, Uma02], constructions of efficiently decodable extractor codes and hardcore functions that output several bits [TZ01], hardness of approximation results [MU01], etc. (see [Sud00] or [Gur01, Chap. 12] for a discussion of these applications). Though *qualitatively* the codes necessary for all of these applications are now known, these diverse complexity-theoretic applications nevertheless motivate the pursuit for constructions of such codes with better quantitative parameters and in particular constructions that approach the optimal rate of $\Omega(\varepsilon^2)$ (which is achieved by random codes and exponential time decoding).

We first review the known results concerning this problem that appear in [GS00, GHSZ00]. For the binary case, the construction in [GHSZ00] gives the currently best known rate of $\Omega(\varepsilon^4)$; their construction, however, is not explicit and takes deterministic $n^{O(1/\varepsilon)}$ time. It also applies only to *binary* linear codes. An earlier result by [GS00] achieves a rate of $\Omega(\varepsilon^6)$ and works over larger alphabets as well, but the outer code has to be an algebraic-geometric (AG) code which makes the construction complicated and the decoding slow and caveat-filled. The previous best construction of reasonable complexity that worked for all alphabet sizes had a rate of $\Omega(\varepsilon^8)$ (the construction was a Reed-Solomon code concatenated with a large distance inner code). Though the deterministic construction time is high (namely, $n^{O(1/\varepsilon)}$), these codes have a fast probabilistic construction (either a logarithmic time Monte Carlo construction that will have the claimed list decodability property with high probability, or a near-linear time Las Vegas construction that is guaranteed to have the claimed list decodability property). If this is not good enough, and one needs completely explicit constructions,[1] then this can be achieved by lowering the rate to $\Omega(\varepsilon^{10})$.

As a corollary of our decoding algorithm for concatenated codes with arbitrary inner codes, we get a fast probabilistic construction that achieves a rate of $\Omega(\varepsilon^6)$. This matches the best known bound that works for all alphabet sizes, and further, unlike the previous AG-codes based construction, has a very fast probabilistic construction and a near-quadratic time list decoding algorithm. The construction can be made completely explicit by lowering the rate to $\Omega(\varepsilon^8)$, and once again this is the best known for completely explicit codes (which do not even involve search for constant-sized objects).

Organization of the paper: We begin in Section 2 with some basic definitions. Section 3 states and proves the combinatorial result that will be used in decoding the inner codes in our algorithms. In Section 4 we use the combinatorial result to design an algorithm for decoding concatenated codes that makes good use of soft information. In Section 5, we focus on the "high-noise" situation when we wish to correct a large (namely, $(1 - 1/q - \varepsilon)$) fraction of errors, and state and prove what our algorithm implies in this situation. Finally, we conclude by mentioning a few open questions in Section 6.

2. Preliminaries and Notation

A linear code C over a q-ary alphabet is simply a subspace of $\mathbb{F}_q^n$ ($\mathbb{F}_q$ being the finite field with q elements) for some n; we call n the *block length* of the code. We identify the elements of a q-ary alphabet with the integers $1, 2, \ldots, q$ in some canonical way. Let $[q] = \{1, 2, \ldots, q\}$. For $\mathbf{x}, \mathbf{y} \in [q]^n$, the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, denoted $\Delta(\mathbf{x}, \mathbf{y})$, is the number of positions where $\mathbf{x}$ and $\mathbf{y}$ differ. The *minimum distance*, or simply *distance* of a code C, is defined to the minimum Hamming distance between a pair of distinct codewords of C. A q-ary linear code of block length n, dimension k and minimum distance d is usually denoted as an $[n, k, d]_q$ code. The *relative distance* of such a code is defined to be the normalized quantity d/n.

For $\mathbf{r} \in [q]^n$ and $0 \le e \le n$, the Hamming ball of radius e around $\mathbf{r}$ is defined by $B_q(\mathbf{r}, e) = \{\mathbf{x} \in [q]^n : \Delta(\mathbf{r}, \mathbf{x}) \le e\}$. For a pair of real vectors $\mathbf{v}$ and $\mathbf{w}$, we denote by $\langle \mathbf{v}, \mathbf{w} \rangle$ their usual dot product over the reals.

Given an $[N, K, D]_{q^m}$ linear code C_1 and an $[n, m, d]_q$ linear code C_2, their *concatenation* $\tilde{C} = C_1 \oplus C_2$ is a code which first encodes the message according to C_1 and then encodes each of symbols of the codeword of C_1 further using C_2 (since each symbol is an element of $\mathbb{F}_{q^m}$ and C_2 is a q-ary code of dimension m, this encoding is well-defined). The concatenated code $\tilde{C}$ is an $[Nn, Kk, \geq Dd]_q$ linear code. In particular, its distance is at least the product of the outer and inner distances (Dd is called the *designed distance* of the concatenated code). The codes C_1 and C_2 as above are respectively referred to as the *outer* and *inner* codes of the concatenation.

A Reed-Solomon code is a linear code whose messages are degree k polynomials over a finite field $\mathbb{F}_q$, and a message is encoded by its evaluations at n distinct elements of the field. This gives an $[n, k+1, n-k]_q$ linear code. Note that the definition of the code requires that $q \geq n$, thus these are codes over a large alphabet. By concatenating these codes with, say a binary code of dimension $\log_2 q$, we can get binary codes.

3. A combinatorial result

In this section we prove a combinatorial result that will be used in the analysis of the error-correction performance of our decoding algorithm for concatenated codes. To motivate the exact statement of the combinatorial result, we jump ahead to give a hint of how the inner codes will be decoded in our algorithms. When presented with 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 $e_{\mathbf{c}} = \Delta(\mathbf{r}, \mathbf{c}) \le R$ from $\mathbf{r}$ will be set to be $(R - e_{\mathbf{c}})$. (This is a fairly natural choice for the weights, since the larger the distance between $\mathbf{r}$ and $\mathbf{c}$, intuitively the less likely it is that $\mathbf{r}$ will be received when $\mathbf{c}$ is transmitted.) These weights, for each of the outer codeword positions, will be passed to a soft decoding algorithm for Reed-Solomon codes which will then use the weights to complete the list decoding. We now state and prove a combinatorial result that gives an upper bound on the sum of squares of the weights $(R - e_{\mathbf{c}})$.

[1] While we do not try to define the term "explicit" formally, a construction would be considered explicit if it can be specified by a mathematical formula of an acceptable form.

Proposition 1 *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*

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

(this quantity is the so-called q-ary Johnson radius of the code, the maximum bound up to which we are guaranteed to have a "small" number of codewords within distance R of $\mathbf{r}$). Then we have

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

Proof: The proof follows essentially the same approach as in the proof of the "Johnson bound" (see, for example, [Gur01, Chap. 3]) which gives an upper bound on the 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.

We identify elements of $[q]$ with vectors in $\mathbb{R}^q$ by replacing the symbol i ($1 \leq i \leq q$) 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 $\Delta(\mathbf{r}, \mathbf{c_i}) \leq R$, where R is a parameter that will be set shortly (it will end up being set as in Equation (1)). 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 $1 \leq i \leq M$ (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 $0 \leq \alpha \leq 1$ 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 $1 \leq i \leq M$, have all pairwise dot products less than 0. Geometrically speaking, we shift the origin by $\mathbf{v}$ to a new point relative to which 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 2). This will then give us the required bound (2).

For $1 \leq i \leq M$, 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, straightforward calculations yield:

$$\langle \mathbf{c_i}, \mathbf{v} \rangle = \alpha(n - e_i) + (1-\alpha)\frac{n}{q} \quad (3)$$

$$\langle \mathbf{v}, \mathbf{v} \rangle = \frac{n}{q} + \alpha^2\Big(1 - \frac{1}{q}\Big)n \quad (4)$$

$$\langle \mathbf{c_i}, \mathbf{c_j} \rangle = n - \Delta(\mathbf{c_i}, \mathbf{c_j}) \leq n - d\,. \quad (5)$$

Using (3), (4) and (5), we get for $i \neq j$

$$\begin{aligned} \langle \mathbf{d_i}, \mathbf{d_j} \rangle &= \langle \mathbf{c_i} - \mathbf{v}, \mathbf{c_j} - \mathbf{v} \rangle \\ &\leq \alpha e_i + \alpha e_j - d + \Big(1 - \frac{1}{q}\Big)(1-\alpha)^2 n \\ &\leq 2\alpha R - d + \Big(1 - \frac{1}{q}\Big)(1-\alpha)^2 n \quad (6) \end{aligned}$$

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

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

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

$$\alpha = \Big(1 - \frac{\delta}{(1-1/q)}\Big)^{1/2}\,. \quad (7)$$

and

$$R = n\Big(1 - \frac{1}{q}\Big)\Big(1 - \sqrt{1 - \frac{\delta}{(1-1/q)}}\Big) = n\big(1 - \frac{1}{q}\big)(1-\alpha)\,. \quad (8)$$

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

$$\sum_{i=1}^{M} \frac{\langle \mathbf{x}, \mathbf{d_i} \rangle^2}{\langle \mathbf{d_i}, \mathbf{d_i} \rangle} \leq \langle \mathbf{x}, \mathbf{x} \rangle\,. \quad (9)$$

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

$$\langle \mathbf{r}, \mathbf{r} \rangle = n \quad (10)$$

$$\begin{aligned} \langle \mathbf{d_i}, \mathbf{d_i} \rangle &= \langle \mathbf{c_i} - \mathbf{v}, \mathbf{c_i} - \mathbf{v} \rangle \\ &= 2\alpha e_i + (1-\alpha)^2(1 - \frac{1}{q})n \quad (11) \end{aligned}$$

$$\langle \mathbf{r}, \mathbf{d_i} \rangle = (1-\alpha)\Big(1 - \frac{1}{q}\Big)n - e_i = R - e_i\,. \quad (12)$$

Since each $e_i \leq R$, we have $\langle \mathbf{r}, \mathbf{d_i} \rangle \geq 0$ for each i, $1 \leq i \leq M$, and therefore we can apply Equation (9) above. For $1 \leq i \leq M$, define

$$W_i = \frac{\langle \mathbf{r}, \mathbf{d_i} \rangle}{\sqrt{\langle \mathbf{d_i}, \mathbf{d_i} \rangle}} = \frac{R - e_i}{\sqrt{2\alpha e_i + (1-\alpha)R}} \quad (13)$$

(the second step follows using (8), (11) and (12)). Since each $e_i \leq R$, we have

$$W_i = \frac{R - e_i}{\sqrt{2\alpha e_i + (1-\alpha)R}} \geq \frac{R - e_i}{\sqrt{(1+\alpha)R}} = \frac{R - e_i}{\sqrt{\delta n}}\,, \quad (14)$$

where the last equality follows by substituting the values of α and R from (7) and (8). Now combining (10), (11) and (12), and applying Equation (9) to the choice $\mathbf{x} = \mathbf{r}$, we get $\sum_{i=1}^{M} W_i^2 \leq n$. Together with Equation (14), this gives

$$\sum_{i=1}^{M} (R - \Delta(\mathbf{r}, \mathbf{c_i}))^2 \leq \delta n^2 \ . \tag{15}$$

This clearly implies the bound (2) claimed in the statement of the proposition, since the codewords $\mathbf{c_i}$, $1 \leq i \leq M$, include *all* codewords $\mathbf{c}$ that satisfy $\Delta(\mathbf{r}, \mathbf{c}) \leq R$, and the remaining codewords contribute zeroes to the left hand side of Equation (2). □

The geometric fact that was used in the above proof is stated below. A proof may be found in Appendix A.

Lemma 2 *Let* $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_M}$ *be* distinct *unit vectors in* $\mathbb{R}^N$ *such that* $\langle \mathbf{v_i}, \mathbf{v_j} \rangle \leq 0$ *for* $1 \leq i < j \leq M$. *Further, suppose* $\mathbf{x} \in \mathbb{R}^N$ *is a vector such that* $\langle \mathbf{x}, \mathbf{v_i} \rangle \geq 0$ *for each* i, $1 \leq i \leq M$. *Then*

$$\sum_{i=1}^{m} \langle \mathbf{x}, \mathbf{v_i} \rangle^2 \leq \langle \mathbf{x}, \mathbf{x} \rangle \ . \tag{16}$$

4. The formal decoding algorithm and its analysis

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

Theorem 3 *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 block length, and the inner codes belong to any family of q-ary linear codes of relative distance δ. Then, there is a polynomial time decoding procedure to list decode codes from such a family up to a fraction*

$$\left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right) - \sqrt{\delta(1-\Delta)} \tag{17}$$

of errors.

Proof: (Sketch) Consider a concatenated code C with outer code a Reed-Solomon code over $\mathrm{GF}(q^m)$ of block length 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 block length. Let the inner code C_{in} be any q-ary linear code of dimension m, block length 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 $\langle C_{\mathrm{in}}(p(x_1)), \ldots, C_{\mathrm{in}}(p(x_{n_0}))\rangle$, 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.

We now present and analyze the algorithm that decodes the concatenated code up to the claimed fraction (17) of errors. Let $y \in \mathbb{F}_q^n$ be a received word. For $1 \leq i \leq n_0$, 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

$$R = n_1\left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right) \tag{18}$$

be the "Johnson radius" of the inner code C_{in}. 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 the codeword $C_{\mathrm{in}}(\alpha)$, and then compute the *weight* $w_{i,\alpha}$ as:

$$w_{i,\alpha} \stackrel{\mathrm{def}}{=} \max\{(R - e_{i,\alpha}), 0\} \ . \tag{19}$$

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

$$\sum_{\alpha} w_{i,\alpha}^2 \leq \delta n_1^2 \ , \tag{20}$$

for $i = 1, 2, \ldots, n_0$. We will now run the soft decoding algorithm for Reed-Solomon codes from [GS99] for this choice of weights. In the form necessary to us, which is stated for example in [Gur01, Chap. 6], this result states the following for decoding a Reed-Solomon over $\mathrm{GF}(Q)$ of block length n_0: Let $\epsilon > 0$ be an arbitrary constant. For each $i \in [n]$ and $\alpha \in \mathrm{GF}(Q)$, let $w_{i,\alpha}$ be a non-negative rational number. Then, there exists a deterministic algorithm with run-time $\mathrm{poly}(n, Q, 1/\epsilon)$ that, when given as input the weights $w_{i,\alpha}$ for $i \in [n]$ and $\alpha \in \mathrm{GF}(Q)$, finds a list of *all* polynomials $p \in \mathrm{GF}(Q)[x]$ of degree at most k that satisfy

$$\sum_{i=1}^{n} w_{i,p(x_i)} \geq \sqrt{k \sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} w_{i,\alpha}^2} + \epsilon \max_{i,\alpha} w_{i,\alpha} \ . \tag{21}$$

Now, combining the above result with Equation (20) and the definition of the weights from Equation (19), 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

$$\sum_{i=1}^{n_0} (R - e_{i,p(x_i)}) \geq \sqrt{k_0 n_0 \delta n_1^2} + \epsilon n_1 \tag{22}$$

holds. Recalling the definition of R (Equation (18)) and using $k_0 = (1-\Delta)n_0$, we conclude that we can find a list of all codewords that are at a Hamming distance of at most

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

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

4.1. Comment on the error-correction performance of Theorem 3

The bound of (17) is attractive only for very large values of Δ, 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 $\Delta = 3/4$, the bound does not even achieve the product bound (namely, $\Delta\delta/2$), for *any* value of δ in the range $0 < \delta < 1/2$ (in fact, the bound as stated in (17) is negative unless Δ is quite large). However, the merit of the bound is that as Δ gets very close to 1, the bound (17) 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 $\Delta\cdot\delta \to \delta$, it approaches the so-called *Johnson bound on list decoding radius*. This bound states that for a q-ary code of relative distance γ and block length n, every Hamming ball of radius at most

$$e_J(\gamma,q,n) \stackrel{\text{def}}{=} n(1-1/q)\Big(1-\sqrt{1-\frac{\gamma}{1-1/q}}\Big) \qquad (23)$$

has at most a polynomial number of codewords (in fact, at most $O(nq)$) codewords; cf. [Gur01, Chap. 3]). Therefore, for $\Delta \to 1$, the result of Theorem 3 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 $0 < \delta < (1-1/q)$. In particular, for $\Delta \to 1$ 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 (this will be expanded upon in Section 5).

4.2. Alternative decoding bound

By slightly modifying the analysis used in proving the combinatorial bound of Proposition 1, one can prove the following alternative bound instead of (2).

$$\sum_{\mathbf{c}\in C}\Big(\max\Big\{\Big(1-\frac{\Delta(\mathbf{r},\mathbf{c})}{\tilde{R}}\Big),0\Big\}\Big)^2 \le \frac{q}{q-1} \ , \qquad (24)$$

where we use the same notation as in the statement of Proposition 1 and $\tilde{R}$ is defined as

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

(The only change required in the proof is to replace the lower bound on W_i from Equation (14) with the alternative lower bound $W_i \ge \big(1-\frac{e_i}{\tilde{R}}\big)\sqrt{\frac{n(q-1)}{q}}$, which follows easily from the definition of W_i in Equation (13).)

Now, replacing the choice of weights in Equation (19) in the proof of Theorem 3 by

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

and then using (24), we obtain a decoding algorithm to decode up to a fraction

$$\Big(1-\frac{1}{q}\Big)\Big(1-\sqrt{1-\frac{q\delta}{q-1}}\Big)^2\Big(1-\sqrt{\frac{1-\Delta}{(1-1/q)}}\Big) \qquad (25)$$

of errors. This bound is positive whenever $\Delta > 1/q$, and in general appears incomparable to that of (17). However, note that even for Δ very close to 1, the bound (25) does not approach the Johnson bound, except for δ very close to $(1-1/q)$. But as with the bound (17), 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 (17), to obtain the results outlined in the next section for highly list decodable codes.

5. Consequence for highly list decodable codes

We now apply Theorem 3 with a suitable choice of parameters to obtain an alternative construction of codes list decodable up to a fraction $(1-1/q-\varepsilon)$ of errors and which have rate $\Omega(\varepsilon^6)$. Compared to the construction of [GS00] 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 the AG+Hadamard based construction in [GS00]. 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 pre-computed advice information about the outer code, which is necessary for decoding AG-codes, see [GS01]. 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.

For binary linear codes, a construction of codes list decodable from a fraction $(1/2-\varepsilon)$ of errors and having rate $\Omega(\varepsilon^4)$ is known [GHSZ00]. Even with this substantially better rate, the result of [GHSZ00] does not strictly subsume the result proved in this section. This is for two reasons. Firstly, the stated result from [GHSZ00] applies 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 rate $\Omega(\varepsilon^4)$ codes of [GHSZ00].

Theorem 4 *For every fixed prime power q, the following holds: For every small enough $\varepsilon > 0$, there exists a family of linear codes over $\mathbb{F}_q$ with the following properties:*

(i) A description of a code of block length, 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 $O(n \log^2 n/\varepsilon^4)$, 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 3 with the choice of parameters $\Delta = 1 - O(\varepsilon^2)$ and $\delta = 1 - 1/q - O(\varepsilon^2)$. Substituting in the bound (17), the fraction of errors corrected by the decoding algorithm from Section 4 will be $(1 - 1/q - \varepsilon)$, which handles Property (iii) claimed above. Also, the relative distance of the code is at least $\Delta \cdot \delta$, and is thus $(1-1/q-O(\varepsilon^2))$, verifying the distance claim in (ii) above. The outer Reed-Solomon code has rate $1 - \Delta = \Omega(\varepsilon^2)$. 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. [vL99, Chapter 5]). Therefore, a random inner code of rate $\Omega(\varepsilon^4)$ 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 $\Omega(\varepsilon^2 \cdot \varepsilon^4) = \Omega(\varepsilon^6)$, 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 code 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 block length) 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 as Proposition 6 at the end of this section. Applying Proposition 6 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 $\Omega(\varepsilon^4)$ 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 4 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 [ABN+92] (see also [She93]). This will give an *explicit construction* of the overall concatenated code with rate $\Omega(\varepsilon^8)$. We record this below.

Theorem 5 *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 4. The proof of the following result is implicit in [Wel73].

Proposition 6 *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 $n \stackrel{\text{def}}{=} (b+1)k$, 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.

6. Concluding Remarks

The Johnson bound on list decoding radius (defined in Equation (23)) gives a lower bound on the number of errors one can hope to correct from with small lists, in any q-ary code of a certain relative distance. Accordingly, for a q-ary concatenated code whose outer (say, Reed-Solomon) and inner codes have relative distance at least Δ and δ respectively, a natural target would be to decode up to a fraction $(1 - 1/q)\big(1 - \sqrt{1 - \frac{\Delta\delta}{1-1/q}}\big)$ of errors. In this paper, we presented an algorithm whose decoding performance approached this bound for the case when the outer distance Δ was very close to 1. This is ideally suited for the situation when we wish to tolerate a very large fraction of errors, as in such a case one has to use an outer codes with large relative distance. However, it is an extremely interesting question to decode up to the Johnson bound for *every* choice of outer and inner distances Δ, δ. A good first target, one that already appears quite challenging, would be to decode beyond the "product bound", namely beyond a fraction $\Delta\delta/2$ of errors, for every choice of Δ, δ. Decoding up to the product bound will result in a unique codeword and can be accomplished in polynomial time using Generalized Minimum Distance (GMD) decoding for most interesting settings (including when the outer code is a Reed-Solomon code over a polynomially large field).

While an algorithm which decodes any concatenated code with outer Reed-Solomon code up to the Johnson bound will be a remarkable achievement, it will still not imply a construction with rate better than $\Omega(\varepsilon^6)$ for codes decodable up to a fraction $(1-1/q-\varepsilon)$ of errors. Therefore, obtaining a rate better than $\Omega(\varepsilon^6)$ together with fast constructions is another challenge in this area. One promising route to achieving this would be to prove that inner codes with the property required in the construction of [GHSZ00] (which achieves a rate of $\Omega(\varepsilon^4)$ for the binary case) exist in "abundance" for every alphabet size. This will imply that a random code picked from an appropriate ensemble can be used as the inner code, thereby yielding a fast probabilistic construction with the same properties as in [GHSZ00].

References

[ABN+92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ronny Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[Dum98] Ilya I. Dumer. Concatenated codes and their multilevel generalizations. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding Theory*, volume 2, pages 1911–1988. North Holland, 1998.

[For66a] G. David Forney. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

[For66b] G. David Forney. Generalized Minimum Distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[Gur01] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. Ph.D thesis, Massachusetts Institute of Technology, August 2001.

[GHSZ00] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, pages 603–612, October 2000.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[GS00] 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.

[GS01] Venkatesan Guruswami and Madhu Sudan. On representations of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 47(4):1610–1613, May 2001.

[KV01] Ralf Koetter and Alexander Vardy. Decoding of Reed-Solomon codes for additive cost functions. *Manuscript*, October 2001.

[MU01] Elchanan Mossel and Chris Umans. On the complexity of approximating the VC dimension. *Proceedings of the 16th IEEE Conference on Computational Complexity*, pages 220-225, 2001.

[Nie00] Rasmus R. Nielsen. Decoding concatenated codes using Sudan's algorithm. *Manuscript submitted for publication*, May 2000.

[SU01] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudo-random generator. *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 648-657, October 2001.

[She93] Ba-Zhong Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory*, 39:239–242, 1993.

[SW99] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

[SS96] Michael Sipser and Daniel Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[Sud97] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[Sud00] Madhu Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31:16–27, 2000.

[STV01] 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.

[TZ01] Amnon Ta-Shma and David Zuckerman. Extractor Codes. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 193–199, July 2001.

[Uma02] Chris Umans. Pseudo-random generators for all hardnesses. *Proceedings of the 34th ACM Symposium on Theory of Computing*, to appear, May 2002.

[vL99] J. H. van Lint. *Introduction to Coding Theory*. Graduate Texts in Mathematics **86**, (Third Edition) Springer-Verlag, Berlin, 1999.

[Wel73] Edward J. Weldon, Jr. Justesen's construction — the low-rate case. *IEEE Transactions on Information Theory*, 19:711–713, 1973.

A. Proof of a geometric fact

Proof of Lemma 2: Note that if $\langle \mathbf{v_i}, \mathbf{v_j} \rangle = 0$ for every $i \neq j$, 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 (16) 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 (16) and may therefore be discarded. In particular, this implies that we may assume $(\mathbf{v_i} \neq -\mathbf{v_j})$ for any $1 \leq i, j \leq M$. Since the $\mathbf{v_i}$'s are distinct unit vectors, this means that $|\langle \mathbf{v_i}, \mathbf{v_j} \rangle| < 1$ for all $i \neq j$.

We will prove the claimed bound (16) 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 (16) for the set of projected vectors. The formal details follow.

For $1 \leq i \leq M-1$, 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 $1 \leq i < j \leq M-1$, we have $\langle \mathbf{v'_i}, \mathbf{v'_j} \rangle = \langle \mathbf{v_i}, \mathbf{v_j} \rangle - \langle \mathbf{v_i}, \mathbf{v_M} \rangle \langle \mathbf{v_j}, \mathbf{v_M} \rangle \leq \langle \mathbf{v_i}, \mathbf{v_j} \rangle \leq 0$, since all pairwise dot products between the $\mathbf{v_i}$'s are non-positive. Hence the pairwise dot products $\langle \mathbf{u_i}, \mathbf{u_j} \rangle$, $1 \leq i < j \leq M-1$, 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

$$
\begin{aligned}
\langle \mathbf{x'}, \mathbf{v'_i} \rangle &= \langle \mathbf{x}, \mathbf{v_i} \rangle - \langle \mathbf{x}, \mathbf{v_M} \rangle \cdot \langle \mathbf{v_i}, \mathbf{v_M} \rangle \\
&\geq \langle \mathbf{x}, \mathbf{v_i} \rangle \qquad (26) \\
&> 0
\end{aligned}
$$

where (26) follows since $\langle \mathbf{x}, \mathbf{v_M} \rangle > 0$ and $\langle \mathbf{v_i}, \mathbf{v_M} \rangle \leq 0$.

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

$$\sum_{i=1}^{M-1} \langle \mathbf{x}', \mathbf{u_i} \rangle^2 \leq \langle \mathbf{x}', \mathbf{x}' \rangle . \tag{27}$$

Now, $\|\mathbf{v_i'}\|^2 = \langle \mathbf{v_i'}, \mathbf{v_i'} \rangle = \langle \mathbf{v_i}, \mathbf{v_i} \rangle - \langle \mathbf{v_i}, \mathbf{v_M} \rangle^2 \leq \|\mathbf{v_i}\|^2 = 1 = \|\mathbf{u_i}\|^2$. This implies that $\langle \mathbf{x}', \mathbf{v_i'} \rangle \leq \langle \mathbf{x}', \mathbf{u_i} \rangle$, for $1 \leq i \leq M-1$.

Also, by (26) $\langle \mathbf{x}', \mathbf{v_i'} \rangle \geq \langle \mathbf{x}, \mathbf{v_i} \rangle$, and therefore

$$\langle \mathbf{x}, \mathbf{v_i} \rangle \leq \langle \mathbf{x}', \mathbf{u_i} \rangle , \tag{28}$$

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

$$\langle \mathbf{x}', \mathbf{x}' \rangle = \langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{x}, \mathbf{v_M} \rangle^2 . \tag{29}$$

The claimed result now follows by using (28) and (29) together with the inequality (27). $\square$