An Optimal Sequence Reconstruction Algorithm for Reed-Solomon Codes

Shubhransh Singhvi[*], **Roni Con**[‡], **Han Mao Kiah**[§] and **Eitan Yaakobi**[‡§]
[*]Signal Processing & Communications Research Center, IIIT Hyderabad, India
[‡]Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel
[§]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

***Abstract*—The *sequence reconstruction problem*, introduced by Levenshtein in 2001, considers a scenario where the sender transmits a codeword from some codebook, and the receiver obtains N noisy outputs of the codeword. We study the problem of *efficient* reconstruction using N outputs that are corrupted by substitutions. Specifically, for the ubiquitous Reed-Solomon codes, we adapt the Koetter-Vardy soft-decoding algorithm, presenting a reconstruction algorithm capable of correcting beyond Johnson radius. Furthermore, the algorithm uses $\mathcal{O}(nN)$ field operations, where n is the codeword length.**

I. Introduction

The *sequence reconstruction* problem introduced by Levenshtein [1], [2] corresponds to a model in which a sequence from some codebook is transmitted over several noisy channels. The channels are assumed to be independent, except it is required that their outputs are different. The main problem under this paradigm has been to determine the minimum number of channels required to uniquely reconstruct the transmitted sequence. Levenshtein proved that for unique reconstruction, the number of channels in the worst case has to be greater than the maximum intersection size between two balls of any possible two inputs. Here, the ball of an input refer to all possible channel outputs of the specific input.

Also, of interest is the task to design an *efficient* decoder that correctly reconstructs a codeword from these noisy outputs. While Levenshtein introduced this problem in his seminal work [1], efficient decoders are less studied and to the best of authors' knowledge, only [3]–[6] designed efficient decoders.

This problem was first motivated by the fields of biology and chemistry, however it is also relevant for applications in wireless sensor networks. Recently this model has received significant attention due to its applicability to DNA storage, where the same information is read multiple times and thereby several channel estimations of the data are provided [7]–[9]. Solving the reconstruction problem was studied in [1] with respect to several channels such as the Hamming distance, the Johnson graphs, and other metric distances. In [10]–[12], it was analyzed for permutations, and in [13], [14] for other general error graphs. Later, the problem was studied in [15] for permutations with the Kendall's τ distance and the Grassmann graph, and in [5], [17], [18] for insertions and deletions, respectively. The connection between the reconstruction problem and associative memories was proposed in [3] and more results were then derived in [19]–[21]. This problem was also studied in [22] for the purpose of asymptotically improving the Gilbert–Varshamov bound.

In this work, we consider the t-substitution model, where every channel introduces at most t substitution errors. To describe the problem formally, we introduce some notation. For $\boldsymbol{v} \in \mathbb{F}_q^n$, let $B_t(\boldsymbol{v})$ be the radius-t ball surrounding a word $\boldsymbol{v}$. Assume that the transmitted words belong to some code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with minimum distance d. Denote by $N_{n,q}(t,d) \triangleq \max_{\boldsymbol{c}_1,\boldsymbol{c}_2 \in \mathcal{C}} |B_t(\boldsymbol{c}_1) \cap B_t(\boldsymbol{c}_2)|$, the maximum intersection between two balls of radius t of any possible pair of codewords. In general, computing $N_{n,q}(t,d)$, is not straightforward and in some settings the exact value is not known. However, for substitution errors, Levenstein computed this quantity

Lemma 1 ([1]). *Let $e \triangleq \left\lfloor \frac{d-1}{2} \right\rfloor$ and $t \triangleq e + \ell$. Then*

$$N_{n,q}(t,d) = \sum_{i=0}^{t-\lceil \frac{d}{2} \rceil} \binom{n-d}{i} (q-1)^i \cdot \left(\sum_{a=d-t+i}^{t-i} \sum_{b=d-t+i}^{t-i} \binom{d}{a} \binom{d-a}{b} (q-2)^{d-a-b} \right) .$$

We note that in order to read the entire input, one has to read $\mathcal{O}(n \cdot N_{n,q}(t,d))$ elements in $\mathbb{F}_q$. Hence, a decoder is said to be *optimal* if it takes $\mathcal{O}(nN_{n,q}(t,d))$ $\mathbb{F}_q$ field operations to output the correct codeword. Note that we measure time complexity in terms of field operations over $\mathbb{F}_q$. Thus, we omit $\mathrm{poly}(\log q)$ factors in our complexity notations.

A. Existing Reconstruction Decoders

Let $\mathrm{Vol}_q(\ell, n) \subset {\mathbb{F}_q}^n$ denote the volume of the q-ary Hamming ball of radius ℓ. For brevity, we let $N \triangleq N_{n,q}(t,d) + 1$. Let the set of N channel outputs (reads) be denoted by $Y \triangleq \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\} \subseteq B_t^S(\boldsymbol{c})$ for some $\boldsymbol{c} \in \mathcal{C}$. We briefly describe possible solutions to the reconstruction problem, with their complexity.

1) Decoder based on majority-logic-with-threshold: Levenshtein in [1], showed that the majority-logic decoder is optimal when the code $\mathcal{C}$ is the entire space. However, when $\mathcal{C}$ is not the entire space, a unique reconstruction decoder for $q = 2$ was designed in [26] and was recently extended to arbitrary q in [6]. Furthermore, it was shown that the decoder has a runtime complexity of $\mathcal{O}(q^{\min\{n, t(e+2)\}} nN)$, which is optimal

when $q^{t(e+2)}$ is a constant. However, for $q = \mathcal{O}(n)$, which is the case for RS codes, the decoder is far from optimality.

2) Brute-force decoder: This decoder iterates through all the codewords in $\mathcal{C}$ to find the word $\boldsymbol{c}$ such that $Y \subseteq B_t(\boldsymbol{c})$. Note that it takes $\mathcal{O}(|\mathcal{C}|Nn)$ time to find the correct word.

3) List-decoding using a single read: The decoder selects any one out of the N reads and generates a list of codewords, $\mathcal{L}$, in radius t using a list decoder for the code. Next, the decoder iterates through all the codewords in the list to find the word $\boldsymbol{c}$ such that $Y \subseteq B_t(\boldsymbol{c})$. Note that the run-time complexity of this decoder is the complexity that it takes to produce the list plus the time that it iterates over all the codewords in the list, which is again far from optimality.

In summary, in the regime where $t = \mathcal{O}(n)$ and $|C| = q^{Rn}$ for some $R > 0$, both the majority-logic-with-threshold and brute-force decoders have complexities significantly higher than $\mathcal{O}(n \cdot N)$. On the other hand, the complexity of the list-decoder (using a single read) relies on the availability of an efficient list-decoding algorithm for the code $\mathcal{C}$. Hence, our focus is on codes equipped with efficient list-decoding algorithms, aiming to augment the error-correcting radius in the context of the sequence reconstruction problem. A natural candidate is the marvelous *Reed-Solomon codes*.

B. Our Decoder

Reed-Solomon codes (RS codes) [23] are the most widely used family of codes in theory and practice and have found many applications (some applications include QR codes, secret sharing schemes, space transmission, data storage and more. The ubiquity of these codes can be attributed to their simplicity as well as to their efficient encoding and decoding algorithms. We next give a definition of RS codes

Definition 1. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct points in a finite field $\mathbb{F}_q$ of order $q \geq n$. For $k \leq n$, the $[n,k]_q$ RS code, defined by the evaluation vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, is the set of codewords*

$$\{c_f = (f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg f < k\} \ .$$

RS codes are efficiently unique decodable up to half their minimum distance (using, e.g., the Berlekamp–Welch algorithm). They are also efficiently list decodable up to the Johnson radius [24]. Namely, let $\mathcal{C}$ be an $[n,k]$ RS code of rate $R := k/n$. Then, for $\rho \leq 1 - \sqrt{R}$ (the Johnson radius), there is a polynomial time algorithm that given $y \in \mathbb{F}_q^n$ outputs a list of codewords $\mathcal{L}$, where $\mathcal{L}$ is such that (i) $d(c,y) \leq \rho n$ for all $c \in \mathcal{L}$ and (ii) $|\mathcal{L}| = \text{poly}(n)$.

In this paper, we provide a reconstruction decoder for RS codes that can decode from any N corrupted reads at distance at most t from the transmitted codeword. The running time of our decoder is $\mathcal{O}(n \cdot N)$, which is the order of the input size and is thus optimal. We also show that in some settings, the value of t for which our decoder works, exceeds the Johnson bound significantly. Formally, we prove the following theorem.

Theorem 1. *Let $\varepsilon > 0$. Let $\mathcal{C}$ be an $[n,k]_q$ RS code and let t be an integer such that*

$$\frac{t}{n} \leq 1 - \sqrt{\frac{k}{n} \cdot \left(1 - \frac{\ell}{2n} + \varepsilon\right)} \ , \tag{1}$$

where $\ell = t - \left\lfloor \frac{d-1}{2} \right\rfloor$. Let $\boldsymbol{c} \in \mathcal{C}$ be a codeword and $Y \subseteq B_t(\boldsymbol{c})$ with $|Y| = N \geq N_{n,q}(t, n-k+1) + 1$. Then there exists an algorithm that takes the received set Y as its input and outputs $\boldsymbol{c}$ in time $\mathcal{O}(n \cdot N + n^3 \varepsilon^{-6})$.

We note that by slightly changing the algorithm constructed in Theorem 1, we can improve the number of errors that we can recover from, t, at the expense of higher complexity. Formally,

Theorem 2. *Let $\varepsilon > 0$. Let $\mathcal{C}$ be an $[n,k]_q$ RS code and let t be an integer such that*

$$\frac{t}{n} \leq 1 - \sqrt{\frac{k}{n} \cdot \left(1 - \frac{\ell}{n} + \varepsilon\right)} \ , \tag{2}$$

where $\ell = t - \left\lfloor \frac{d-1}{2} \right\rfloor$. Let $\boldsymbol{c} \in \mathcal{C}$ be a codeword and let $Y \subseteq B_t(\boldsymbol{c})$ with $|Y| = N \geq N_{n,q}(t, n-k+1) + 1$. Then there exists an algorithm that takes the received set Y as its input and outputs $\boldsymbol{c}$ in time $\mathcal{O}(n \cdot N^2 + n^3 \varepsilon^{-6})$.

Remark 1. *We note that our decoding radius is beyond the Johnson bound. Indeed, denote $\rho = t/n$, $R = k/n$ and observe that (1) becomes*

$$\rho \leq 1 - \sqrt{R\left(1 - \left(\frac{\rho}{2} - \frac{1-R}{4}\right) + \varepsilon\right)} \ , \tag{3}$$

and (2) becomes

$$\rho \leq 1 - \sqrt{R\left(1 - \left(\rho - \frac{1-R}{2}\right) + \varepsilon\right)} \ . \tag{4}$$

A graphical comparison with the Johnson radius is given in Figure 1.

Remark 2. *We compare the performance of our decoder with the one suggested above that performs list decoding on a single read. List decoding up to* almost *the Johnson radius, i.e., up to $1 - \sqrt{R(1+\varepsilon)}$ takes poly$(n, 1/\varepsilon)$ time and produces a list of constant size [24]. Thus, the total complexity of this algorithm is $\mathcal{O}(n \cdot N + \text{poly}(n, 1/\varepsilon))$, as in Theorem 1. However, this solution works only for values of t such that $t/n \leq 1 - \sqrt{(k/n) \cdot (1+\varepsilon)}$ whereas our decoder works for larger values of t, as implied by (1).*

C. Soft Decoding á la Koetter and Vardy

We briefly recall the soft-decision list-decoding algorithm of Koetter and Vardy (KV) [25] which is an extension of the Guruswami-Sudan (GS) list decoding for RS codes. The interested reader is refered to [24] and [25].

Koetter and Vardy [25] extended the GS algorithm to the case where the decoder is supplied with probabilistic reliability information concerning the received symbols. In particular, the

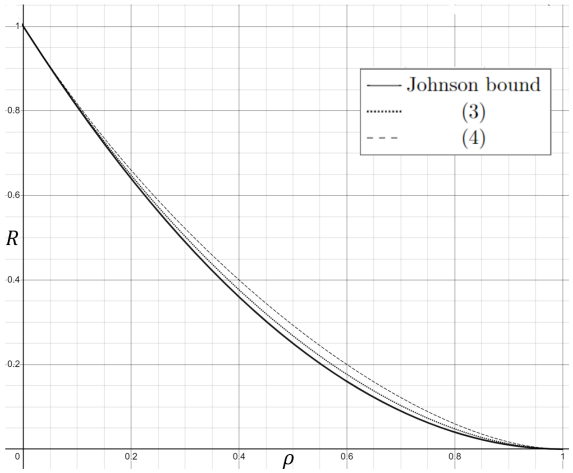

Fig. 1: Tradeoff between rate R and the fraction of errors that can be corrected. The algorithm that achieves the tradeoff corresponding to (3) has complexity $\mathcal{O}(nN)$, while the algorithm achieving (4) has complexity $\mathcal{O}(nN^2)$.

Koetter-Vardy algorithm performs a soft-decision decoding by assigning unequal multiplicities to points according to this extra information. A convenient way to keep track of the interpolation points and their different multiplicities is by means of a multiplicity matrix.

Definition 2. *Let $\delta_0, \delta_1, \ldots, \delta_{q-1}$ be some ordering of $\mathbb{F}_q$. A* multiplicity matrix*, denoted by M, is a $(q \times n)$-matrix with entries $m_{i,j}$ denoting the multiplicity of (δ_i, α_j).*

We provide a high-level description. Given a multiplicity matrix M, the KV algorithm computes a non-trivial bivariate polynomial $Q_M(X,Y)$ of minimal $(1, k-1)$-weighted degree that has a zero of multiplicity at least $m_{i,j}$ at the point (α_j, δ_i) for every (i,j) such that $m_{i,j} \neq 0$. Then, the algorithm factorizes this polynomial to get a list of candidate codewords. (we refer again to [25]).

The cost of constructing $Q_M(X,Y)$ for a given multiplicity matrix M, denoted by $C(M)$, is the number of linear equations that needs to be satisfied for the interpolation. Specifically,

Definition 3. *The* cost *for a multiplicity matrix M is defined as follows:*

$$C(M) = \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} \binom{m_{i,j}+1}{2} .$$

For $\boldsymbol{v} \in \mathbb{F}_q^n$, let $[\boldsymbol{v}]$ denote the $(q \times n)$-matrix representation of $\boldsymbol{v}$, i.e., $[\boldsymbol{v}]_{i,j} = 1$ if $\boldsymbol{v}_j = \delta_i$, and $[\boldsymbol{v}]_{i,j} = 0$ otherwise.

Definition 4. *The score of a vector $\boldsymbol{v} \in \mathbb{F}_q^n$ with respect to a given multiplicity matrix M is defined as the inner product $\mathcal{S}_M(\boldsymbol{v}) = \langle M, [\boldsymbol{v}] \rangle$.*

With these definitions, we are ready to give a black box description of the KV algorithm

Theorem 3 ([25])**.** *Let $\mathcal{C}$ be an $[n,k]_q$ RS code defined with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$. Let M be an $q \times n$ matrix, which denotes a multiplicity matrix. Given the input M, the KV algorithm outputs a list $\mathcal{L}$ such that the following holds.*

(i) A codeword $c \in \mathcal{C}$ is in the list $\mathcal{L}$ if we have $\mathcal{S}_M(\boldsymbol{c}) \geq \sqrt{2(k-1)C(M)}$.

(ii) It holds that $|\mathcal{L}| \leq \sqrt{\frac{2C(M)}{k-1}}$.

(iii) The algorithm runs in time $\mathcal{O}((C(M))^3)$.

II. Constructing the Multiplicity Matrix in Our Settings

In Algorithm 1, we describe how we construct the multiplicity matrix from a set of reads $Y' \subseteq Y$. Then, we shall compute the cost of the constructed multiplicity matrix and the score of a transmitted codeword with respect to the generated multiplicity matrix.

Algorithm 1: Multiplicity matrix constructor

input : A set Y' of reads and an integer μ
output: A multiplicity matrix $M \in \mathbb{F}_q^{q \times n}$

```
Set M = 0_{q×n}
for j ∈ [n] do
    for y ∈ Y' do
        Set i such that δ_i = y_j
        (M)_{i,j} = (M)_{i,j} + μ
    end
end
Return M
```

Lemma 2. *Let M be the multiplicity matrix generated by Algorithm 1 when given $Y' \subset Y$ and μ as input. Then, the score of the transmitted codeword $\boldsymbol{c}$ with respect to M is*

$$\mathcal{S}_M(\boldsymbol{c}) \geq \mu \cdot |Y'| \cdot (n-t).$$

Proof. Since each word can have at most t erroneous positions, the result follows. □

Next, we give a combinatorial lemma, whose proof we defer to the Appendix.

Lemma 3. *Let $a_0, a_1, \ldots, a_{b-1}$ be positive integers such that $\sum_{i=0}^{b-1} a_i = c$. Then*

$$\sum_{i=0}^{b-1} \binom{a_i+1}{2} = \binom{c+1}{2} - \frac{1}{2} \sum_{i=0}^{b-1} a_i(c-a_i).$$

Lemma 3 then allows us to analyze the cost of the multiplicity matrix.

Lemma 4. *Let M be the matrix returned by Algorithm 1 with input Y' and μ. The cost of M is*

$$C(M) = n\binom{\mu|Y'|+1}{2} - \frac{\mu^2}{2} \sum_{\boldsymbol{v},\boldsymbol{u} \in Y'} d(\boldsymbol{v},\boldsymbol{u}) . \qquad (5)$$

Proof. From Definition 3, it follows that,

$$C(M) = \sum_{j=0}^{n-1} \left(\sum_{i=0}^{q} \binom{m_{i,j}+1}{2} \right) .$$

Note that the sum of each column of M is exactly $\mu|Y'|$. Therefore, for any j, by Lemma 3, we have that

$$\sum_{i=0}^{q}\binom{m_{i,j}+1}{2}=\binom{\mu|Y'|+1}{2}-\frac{1}{2}\sum_{i=0}^{q-1}m_{i,j}(\mu|Y'|-m_{i,j}).$$

Now, by the definition of $m_{i,j}$ and the process by which we construct the matrix, we have (recall that $\{\delta_0,\ldots,\delta_{q-1}\}=\mathbb{F}_q$)

$$\begin{aligned}\sum_{i=0}^{q-1}m_{i,j}(\mu|Y'|-m_{i,j})&=\sum_{i=0}^{q-1}\left(\sum_{\boldsymbol{u}\in Y'}\mu\mathbb{1}_{\boldsymbol{u}_j=\delta_i}\right)\left(\sum_{\boldsymbol{v}\in Y'}\mu\mathbb{1}_{\boldsymbol{v}_j\neq\delta_i}\right)\\&=\mu^2\sum_{\boldsymbol{u},\boldsymbol{v}\in Y'}\left(\sum_{i=0}^{q-1}\mathbb{1}_{\boldsymbol{u}_j=\delta_i}\cdot\mathbb{1}_{\boldsymbol{u}_j\neq\delta_i}\right)\\&=\mu^2\left(\sum_{\boldsymbol{u},\boldsymbol{v}\in Y'}\mathbb{1}_{\boldsymbol{v}_j\neq\boldsymbol{u}_j}\right),\end{aligned}$$

where $\mathbb{1}$ is the indicator function. Therefore, putting it together, we have

$$\begin{aligned}C(M)&=\sum_{j=0}^{n-1}\left(\binom{\mu|Y'|+1}{2}-\frac{\mu^2}{2}\sum_{\boldsymbol{u},\boldsymbol{v}\in Y'}\mathbb{1}_{\boldsymbol{v}_j\neq\boldsymbol{u}_j}\right)\\&=n\cdot\binom{\mu|Y'|+1}{2}-\frac{\mu^2}{2}\sum_{\boldsymbol{u},\boldsymbol{v}\in Y'}d(\boldsymbol{u},\boldsymbol{v})\,,\end{aligned}$$

as desired. □

Remark 3. *From (5), we observe that for a given number of reads and an integer μ, the cost of the multiplicity matrix is dominated by the total distance of the reads in Y'. Concretely, the larger the total distance is, the smaller the cost is.*

The reconstruction decoder is given in Algorithm 2.

Algorithm 2: Reconstruction decoder

input : A set of N distinct reads Y, a subset $Y'\subset Y$, and an integer μ

output: A codeword c

1 Generate a multiplicity matrix M using Algorithm 1 with input Y' and μ

2 Run the KV algorithm with the multiplicity matrix M

3 Return $\boldsymbol{c}\in\mathcal{L}$ such that $Y\subseteq B_t(\boldsymbol{c})$

We summarize the correctness of this algorithm in the following proposition.

Proposition 1. *Let $\mathcal{C}$ be an $[n,k]_q$ RS code. Let $\boldsymbol{c}\in\mathcal{C}$ and let $Y\subseteq B_t(\boldsymbol{c})$ with $|Y|\geq N:=N_{n,q}(t,n-k+1)+1$. Suppose that M is the matrix generated in the first step of Algorithm 2 when given inputs Y, Y', and μ. Then Algorithm 2 outputs the codeword $\boldsymbol{c}$ if $\mu\cdot|Y'|\cdot(n-t)>\sqrt{2(k-1)C(M)}$.*

Proof. At the first step, we construct M using the reads in Y' by invoking Algorithm 1. By Lemma 2, the score of $\boldsymbol{c}$ is at least $\mu\cdot|Y'|\cdot(n-t)$. Since by our assumption, $\mu\cdot|Y'|\cdot(n-t)>\sqrt{2(k-1)C(M)}$, according to Theorem 3, $\boldsymbol{c}$ is inside the list returned by the KV algorithm. Now, by the Levenshtein's reconstruction problem settings, all the reads are at distance t from $\boldsymbol{c}$ and thus, the last step uniquely identifies $\boldsymbol{c}$, the transmitted codeword. □

III. Constructing the Multiplicity Matrix Using only Two Reads

In this section, we examine the performance of a multiplicity matrix constructed from only two reads. We will observe that the distance between these two reads significantly impacts the error correction capability of the KV algorithm. Specifically, a larger distance results in a greater decoding radius. In Section III-A, we shall consider a general case first, where the two reads did not necessarily come from the Levenshtein's reconstruction problem. In Section III-B, we present our reconstruction decoder for the Levenshtein's reconstruction problem.

A. Connecting the Distance between the Two Reads with the Decoding Radius

In the following proposition we are applying the KV algorithm with a multiplicuty matrix based on two reads. We show that if we are guaranteed that the two reads are far from each other, then the correction capability exceeds the Johnson radius and that the error correction capability increases in correlation with the distance between the two reads. Formally,

Proposition 2. *Let $\delta\in[0,2]$, $\rho\in(0,1)$, and $\varepsilon>0$. Let C be an $[n,k]_q$ RS code of rate $R=k/n$ such that*

$$1-\rho\geq\sqrt{R\cdot\left(1-\frac{\delta\rho}{2}+\varepsilon\right)}\,. \tag{6}$$

Let $\boldsymbol{v}$ and $\boldsymbol{u}$ be two corrupted reads of a codeword $\boldsymbol{c}$ such that

- *$d(\boldsymbol{v},\boldsymbol{c})\leq\rho n$ and $d(\boldsymbol{u},\boldsymbol{c})\leq\rho n$.*
- *$d(\boldsymbol{v},\boldsymbol{u})=\delta\cdot\rho n$.*

Let M be the multiplicity matrix obtained by Algorithm 1 with input $Y=Y'=\{\boldsymbol{v},\boldsymbol{u}\}$ and $\mu=1/\varepsilon$. Then, the KV Algorithm with multiplicity matrix M outputs a list of codewords $\mathcal{L}$ such that $\boldsymbol{c}\in\mathcal{L}$ and $|\mathcal{L}|=O(1/\varepsilon\sqrt{R})$.

Proof. The condition for a codeword $\boldsymbol{c}$ to be in the list, according to the KV algorithm analysis is that $S_M(\boldsymbol{c})>\sqrt{2(k-1)C(M)}$. According to Lemma 2, $S_M(\boldsymbol{c})\geq(n-\rho n)\cdot2\mu$ and according to Lemma 4,

$$C(M)=n\binom{2\mu+1}{2}-\mu^2\cdot\delta\rho n=n\mu^2\left(2+\frac{1}{\mu}-\delta\rho\right)\,.$$

Thus, we have to ensure that

$$(n-\rho n)\cdot2\mu>\sqrt{2(k-1)\cdot n\mu^2\cdot\left(2+\frac{1}{\mu}-\delta\rho\right)}\,.$$

Divide by $n\cdot2\mu$ to get

$$1-\rho>\sqrt{\left(R-\frac{1}{n}\right)\cdot\left(1-\frac{\delta\rho}{2}+\frac{\varepsilon}{2}\right)}\,,$$

and note that this inequality clearly holds according to (6). As for the list size, according to Theorem 3, the list size is upper bound by

$$\sqrt{\frac{n\mu^2\left(4-2\delta\rho+\frac{2}{\mu}\right)}{k-1}}=O\left(\frac{1}{\varepsilon\sqrt{R}}\right)\,.$$

□

Remark 4. *We consider the two extreme points of δ.*

- *When $\delta = 0$, the two reads are equal. In this case, inequality (6) yields exactly the Johnson bound.*
- *When $\delta = 2$, inequality (6) gives*

$$1 - \rho \geq \sqrt{R \cdot (1 - \rho + \varepsilon)} \, ,$$

which implies

$$\rho < 1 - R - \varepsilon \, .$$

B. Back to Levenshtein's Reconstruction Problem

In this section, we prove Theorem 1. In Levenstein's reconstruction problem, given a codeword $\boldsymbol{c} \in \mathcal{C}$, an adversary will output a set Y of N reads such that (i) all the reads are at distance t from $\boldsymbol{c}$ and (ii) the N reads are sufficient to identify $\boldsymbol{c}$ uniquely. According to Proposition 2, our goal is to find two reads of maximal distance inside the set Y. We first start with the following lemma whose proof for the binary case ($q = 2$) was given in [3, Lemma 17]

Lemma 5. *Let d be odd, let $e = \frac{d-1}{2}$, and let $t = e + \ell$. Also, assume that $\ell < \frac{d}{2}$. Let Y be a set containing $N_{n,q}(t, d) + 1$ reads. Then, there exist two reads in Y with distance at least $2\ell - 1$.*

Proof. We will prove that $N_{n,q}(t, d) \geq \mathrm{Vol}_q(\ell - 1, n)$ where $\mathrm{Vol}_q(\ell - 1, n)$ is the size of the Hamming ball of radius $\ell - 1$. Note that this proves the claim. Indeed, assume that the distance of every pair of reads is at most $2\ell - 2$. It means that all the $N_{n,q}(t, d) + 1$ reads fit inside a ball of radius $\ell - 1$ which leads to a contradiction.

Let x, y be two codewords such that $d(x, y) = d$, the minimum distance of the code. Let $u \in \mathbb{F}_q^n$ be such that $d(x, u) = d(y, u) = (d + 1)/2$. Consider, $B_{\ell-1}(u)$, the Hamming ball around u of radius $\ell - 1$. We will prove that $B_{\ell-1}(u) \subseteq B_t(x)$ and $B_{\ell-1}(u) \subseteq B_t(y)$. This would imply that $\mathrm{Vol}_q(\ell - 1, n) = |B_{\ell-1}(u)| \leq |B_t(x) \cap B_t(y)| = N_{n,q}(t, d)$.

Note that by the triangle inequality, for any $v \in B_{\ell-1}(u)$, it holds that $d(v, x) \leq d(v, u) + d(u, x) = \ell - 1 + \frac{d+1}{2} = t$ and an identical argument also yields that $d(v, x) \leq t$ for any $v \in B_{\ell-1}(u)$. Thus, we have proved that $B_{\ell-1}(u) \subseteq B_t(x) \cap B_t(y)$, as desired. □

Clearly, we can perform a search that would take $\mathcal{O}(n \cdot N^2)$ time to find two reads that are at distance $2\ell - 1$ apart. We note again that N is relatively big compared to n, so we would like to reduce our dependency on N. A simple observation shows that to find two reads that are at distance ℓ, we can perform only $\mathcal{O}(n \cdot N)$ iterations. Indeed, set $\boldsymbol{u}$ to be the first read. Then, evaluate the distance of all the reads from $\boldsymbol{u}$ and take the read that is the farthest apart. The correctness of this claim is given in the following simple claim

Claim 1. *Let Y be a set of reads such that there are two reads with distance at least $2\ell - 1$. Let $\boldsymbol{u} \in Y$. There exists a $\boldsymbol{v} \in Y$ such that $d(\boldsymbol{u}, \boldsymbol{v}) \geq \ell$.*

Proof. Let $\boldsymbol{w}$ and $\boldsymbol{z}$ be two reads with $d(\boldsymbol{w}, \boldsymbol{z}) \geq 2\ell - 1$. Then, by the triangle inequality, $d(\boldsymbol{u}, \boldsymbol{w}) + d(\boldsymbol{u}, \boldsymbol{z}) \geq d(\boldsymbol{w}, \boldsymbol{z}) \geq 2\ell - 1$. Thus, either $d(\boldsymbol{u}, \boldsymbol{w}) \geq \ell$ or $d(\boldsymbol{u}, \boldsymbol{z}) \geq \ell$. □

Therefore, our final algorithm is given in Algorithm 3.

Algorithm 3: Reconstruction using two reads

input : A set of N distinct reads Y, and an integer r
output: A codeword c

1 Set $\boldsymbol{u} = \boldsymbol{y}_1$ and set $\boldsymbol{v} = \max_{\boldsymbol{y} \in Y} d(\boldsymbol{u}, \boldsymbol{y})$
2 Execute Algorithm 2 with Y, $\{\boldsymbol{u}, \boldsymbol{v}\}$, and r

Proposition 3. *Let $\varepsilon > 0$ and Let n, k, t be integers such that*

$$\frac{t}{n} \leq 1 - \sqrt{\frac{k}{n} \cdot \left(1 - \frac{\ell}{2n} + \varepsilon\right)} \, .$$

Let $\mathcal{C}$ be an $[n, k]_q$ RS code. Let $\boldsymbol{c} \in \mathcal{C}$ be a codeword and let Y be a set containing $N := N_{q,n}(t, n - k + 1) + 1$ vectors such that $\forall \boldsymbol{y} \in Y$, we have $d(\boldsymbol{c}, \boldsymbol{y}) \leq t$. Then, applying Algorithm 3 on Y and $\lceil 1/\varepsilon \rceil$ we get back $\boldsymbol{c}$. Furthermore, the time complexity of the algorithm is $\mathcal{O}(n \cdot N + n^3 \varepsilon^{-6})$.

Proof. By Lemma 5, we know that Y consists of two reads such that their distance is at least $2\ell - 1$ and by Claim 1, we know that the first step in Algorithm 3 finds two reads that are ℓ distance apart in $\mathcal{O}(n \cdot N)$ time. To ensure that the KV algorithm indeed works and the transmitted codeword is inside the list, we check that $S_M(\boldsymbol{c}) > \sqrt{2(k-1)C(M)}$, where M is the multiplicity matrix constructed by two reads that are ℓ-distance apart. Indeed, in this case, by plugging in Proposition 2 $\rho = t/n$, $\delta = \ell/t$, we get that

$$\frac{t}{n} \leq 1 - \sqrt{\frac{k}{n} \cdot \left(1 - \frac{\frac{t}{n} \cdot \frac{\ell}{t}}{2} + \varepsilon\right)} = 1 - \sqrt{\frac{k}{n} \cdot \left(1 - \frac{\ell}{2n} + \varepsilon\right)} \, ,$$

and that running the KV algorithm on M produces a list of size $\mathcal{O}(1/(\sqrt{R}\varepsilon))$ with $\boldsymbol{c}$ in the list. Note that as $C(M) = \mathcal{O}(n \cdot \varepsilon^{-2})$, then the KV algorithm runs in time $\mathcal{O}(n^3 \cdot \varepsilon^{-6})$. The final step of Algorithm 2 iterates over all codewords in the list and for each one, checks if it is at distance $\leq t$ from all the reads. Thus, since the list is of constant size, this takes $\mathcal{O}(n \cdot N)$ time. Overall, the complexity of Algorithm 3 is $\mathcal{O}(n \cdot N + n^3 \cdot \varepsilon^{-6})$ as desired. □

REFERENCES

[1] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan 2001.

[2] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *J. Comb. Theory Ser. A*, vol. 93, no. 2, pp. 310–332, Feb. 2001.

[3] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," *IEEE Trans. Inform. Theory,* vol. 65, pp. 2155–2165. 2018.

[4] M. Abu-Sini and E. Yaakobi, "On Levenshtein's reconstruction problem under insertions, deletions, and substitutions." *IEEE Trans on Inf. Theory*, vol. 67, no. 11, pp. 7132–7158, 2021.

[5] V. L. P. Pham, K. Goyal and H. M. Kiah, "Sequence Reconstruction Problem for Deletion Channels: A Complete Asymptotic Solution," *Proc. IEEE Int. Symp. on Inf. Theory*, Espoo, Finland, pp. 992-997, 2022.

[6] V. Junnila, T. Laihonen and T. Lehtilä, "Levenshtein's Reconstruction Problem with Different Error Patterns," *Proc. IEEE Int. Symp. on Inf. Theory*, Taipei, Taiwan, pp. 1300-1305, 2023.

[7] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012.

[8] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[9] S.H.T. Yazdi, H.M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Trans. Mol., Bio. and Multi-Scale Com.*, vol. 1, pp. 230–248, 2015.

[10] E. Konstantinova, "On reconstruction of signed permutations distorted by reversal errors," *Discrete Math.*, vol. 308, no. 5-6, pp. 974–984, Mar. 2008.

[11] E. Konstantinova, "Reconstruction of permutations distorted by reversal errors," *Discrete Applied Math.*, vol. 155, no. 18, pp. 2426–2434, 2007.

[12] E. Konstantinova, V. I. Levenshtein, and J. Siemons, "Reconstruction of permutations distorted by single transposition errors," *arXiv:0702.191v1*, Feb. 2007.

[13] V. Levenshtein, E. Konstantinova, E. Konstantinov, and S. Molodtsov, "Reconstruction of a graph from 2-vicinities of its vertices," *Discrete Appl. Math.*, vol. 156, no. 9, pp. 1399–1406, May 2008.

[14] V. I. Levenshtein and J. Siemons, "Error graphs and the reconstruction of elements in groups," *J. Comb. Theory Ser. A*, vol. 116, no. 4, pp. 795–815, May 2009.

[15] E. Yaakobi, M. Schwartz, M. Langberg, and J. Bruck, "Sequence reconstruction for grassmann graphs and permutations," *Proc. IEEE Int. Symp. on Inf. Theory*, pp. 874–878, July 2013.

[16] F. Sala, R. Gabrys, C. Schoeny, and L. Dolecek, "Exact reconstruction from insertions in synchronization codes," *IEEE Trans on Inf. Theory*, vol. 63, no. 4, pp. 2428–2445, 2017.

[17] R. Gabrys and E. Yaakobi, "Sequence reconstruction over the deletion channel," *Proc. Int. Symp. on Inf. Theory*, pp. 1596–1600, Jul. 2016.

[18] F. Sala, R. Gabrys, C. Schoeny, and L. Dolecek, "Exact reconstruction from insertions in synchronization codes," *IEEE Trans on Inf. Theory*, vol. 63, no. 4, pp. 2428–2445, 2017.

[19] V. Junnila and T. Laihonen, "Codes for information retrieval with small uncertainty," *IEEE Trans. on Inf. Theory*, vol. 60, no. 2, pp. 976–985, Feb. 2014.

[20] V. Junnila and T. Laihonen, "Information retrieval with unambiguous output," *Inf. and Computation*, vol. 242, pp. 354–368, Jun. 2015.

[21] V. Junnila and T. Laihonen, "Information retrieval with varying number of input clues," *IEEE Trans. on Inf. Theory*, vol. 62, no. 2, pp. 625–638, Feb. 2016.

[22] T. Jiang and A. Vardy, "Asymptotic improvement of the Gilbert-Varshamov bound on the size of binary codes," *IEEE Trans on Inf. Theory*, vol. 50, no. 8, pp. 1655–1664, Aug. 2004.

[23] I. S. Reed and G. Solomon. "Polynomial codes over certain finite fields." *Journal of the Society for Industrial and Applied Mathematics*,vol. 8, no. 2, pp. 300–304, 1960.

[24] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans on Inf. Theory*, vol. 45, no. 6, pp. 1757-1767, Sept. 1999.

[25] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans on Inf. Theory*, vol. 49, no. 11, pp. 2809-2825, Nov. 2003.

[26] M. Abu-Sini and E. Yaakobi, "On Levenshtein's reconstruction problem under insertions, deletions, and substitutions," *IEEE Trans on Inf. Theory*, vol. 67, no. 11, pp. 7132-7158, Nov. 2021.

APPENDIX

Lemma 3. *Let $a_0, a_1, \ldots, a_{b-1}$ be positive integers such that $\sum_{i=0}^{b-1} a_i = c$. Then*

$$\sum_{i=0}^{b-1} \binom{a_i+1}{2} = \binom{c+1}{2} - \frac{1}{2}\sum_{i=0}^{b-1} a_i(c-a_i).$$

Proof. For $g, h \in \mathbb{Z}_+$, we know that

$$\binom{g+h+1}{2} = \binom{g+1}{2} + \binom{h+1}{2} + gh.$$

Therefore,

$$\begin{aligned}
&\sum_{i=1}^{b-1} \left(\binom{1+\sum_{j=0}^{i-1} a_j}{2} + \binom{1+a_i}{2} \right) \\
&= \sum_{i=1}^{b-1} \left(\binom{1+\sum_{j=0}^{i} a_j}{2} - a_i \left(\sum_{j=0}^{i-1} a_j \right) \right).
\end{aligned}$$

Note that

$$\sum_{i=1}^{b-1} a_i \sum_{j=0}^{i-1} a_j = \sum_{i=0}^{b-2} a_i \left(c - \sum_{j=0}^{i} a_j \right).$$

Hence, we have that $\sum_{i=0}^{b-1} \binom{a_i+1}{2}$ is

$$\begin{aligned}
&\binom{c+1}{2} - \sum_{i=1}^{b-1} a_i \sum_{j=0}^{i-1} a_j \\
&= \binom{c+1}{2} - \frac{1}{2}\left(\sum_{i=1}^{b-1} a_i \sum_{j=0}^{i-1} a_j + \sum_{i=0}^{b-2} a_i \left(c - \sum_{j=0}^{i} a_j \right) \right) \\
&= \binom{c+1}{2} - \frac{1}{2}\sum_{i=0}^{b-1} a_i(c-a_i).
\end{aligned}$$

□