1 Definitions

In the sequences reconstruction problem, a codeword \( x \) is transmitted through multiple channels. Then, a decoder receives all channel outputs and generates an estimation on the transmitted word, while it is guaranteed that all channel outputs are different from each other, see Fig. 1. If \( x \) belongs to a code \( C \) with minimum distance \( d \) and in every channel there can be at most some \( t > \left\lfloor \frac{d-1}{2} \right\rfloor \) errors, then Levenshtein studied the minimum number of channels that guarantees the existence of a successful decoder. This number has to be greater than

\[
\max_{x_1, x_2 \in C, x_1 \neq x_2} |B_t(x_1) \cap B_t(x_2)|, \tag{1}
\]

where \( B_t(x) \) is the ball of radius \( t \) surrounding \( x \). To see that, notice that if the intersection of the radius-\( t \) balls of \( x_1 \) and \( x_2 \) contains \( N \) words and the channel outputs are these \( N \) words, then a decoder cannot determine what the transmitted word is. However, if the number of channel outputs is greater than the maximum size of the intersection of two balls, then there is only one codeword of distance at most \( t \) from all received channel outputs.

2 The Size of the Largest Intersection

Since the value in (1) does not depend on the code, but only its minimum distance, it is denoted by \( N(t, d) \).

Figure 1: Channel model for the sequences reconstruction problem.
Lemma 1. Let $d, t$ be two positive integers such that $t > \lceil \frac{d-1}{2} \rceil$, then the value of $N(t, d)$ satisfies

\[
N(t, d) = \sum_{i=0}^{t-[\frac{d}{2}]} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k}.
\]

and

\[
N(t, d) = \sum_{k=0}^{\min\{d,t\}} \binom{d}{k} \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}.
\]

Proof. Assume $d_H(x, y) = d$ and the goal is to find the cardinality of the set

\[ S_t(\{x,y\}) \triangleq \{z \in \{0,1\}^n : d_H(z, x), d_H(z, y) \leq t\}. \]

For any word $z \in S_t(\{x,y\})$, let $S_{0,0}, S_{0,1}, S_{1,0}, S_{1,1}$ be the following four sets:

- $S_{0,0} = \{i : y_i = z_i = x_i\}$,
- $S_{0,1} = \{i : y_i = x_i, z_i = \overline{x}_i\}$,
- $S_{1,0} = \{i : y_i = \overline{x}_i, z_i = x_i\}$,
- $S_{1,1} = \{i : y_i = z_i = \overline{x}_i\}$.

Note that $|S_{0,0}| + |S_{0,1}| = n - d$ and $|S_{1,0}| + |S_{1,1}| = d$. Since $d_H(z, x) \leq t$ and $d_H(z, y) \leq t$ we get that

\[ |S_{0,1}| + |S_{1,1}| \leq t, \quad |S_{0,1}| + |S_{1,0}| \leq t, \]

or

\[ |S_{0,1}| + |S_{1,1}| \leq t, \quad |S_{0,1}| + d - |S_{1,1}| \leq t. \]

Denote $|S_{0,1}| = i$ and $|S_{1,1}| = k$ so we get

\[ i + k \leq t, \quad i + d - k \leq t, \]

or

\[ 0 \leq i \leq t - \lfloor d/2 \rfloor, \quad i + d - t \leq k \leq t - i. \]

Therefore, the number of words in the intersection of these two spheres is given by

\[
|S_t(\{x,y\})| = \sum_{i=0}^{t-[\frac{d}{2}]} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k},
\]

where $\binom{a}{b} = 0$ if $b < 0$ or $b > a$. Note that if $d_H(x, y) > d$ then the size of the set $S_t(\{x,y\})$ does not increase and thus

\[
N(t, d) = \sum_{i=0}^{t-[\frac{d}{2}]} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k}.
\]

Lastly, if we substitute the order of $i, k$ in the last term, we get $0 \leq k \leq \min\{d,t\}, 0 \leq i \leq t - \max\{k,d-k\}$, and

\[
N(t, d) = \sum_{k=0}^{\min\{d,t\}} \binom{d}{k} \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}.
\]

\[ \square \]
Corollary 2. Let \( t, d \) be two positive integers such that \( d \) is even, then
\[
N(t, d) = N(t, d - 1).
\]

**Proof.** According to (3) we have
\[
N_t(2, d) = \sum_{k=0}^{\min\{d,t\}} \binom{d}{k} \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}.
\]
We use the identity \( \binom{a-1}{b-1} + \binom{a-1}{b} = \binom{a}{b} \) to get
\[
\binom{d}{k} = \binom{d-1}{k-1} + \binom{d-1}{k},
\]
and thus,
\[
N_t(2, d) = \sum_{k=0}^{\min\{d,t\}} \left( \binom{d-1}{k-1} + \binom{d-1}{k} \right) \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}
\]
\[
= \sum_{k=0}^{\min\{d,t\}-1} \binom{d-1}{k} \sum_{i=0}^{t-\max\{k+1,d-1-k\}} \binom{n-d}{i} + \sum_{k=0}^{\min\{d,t\}} \left( \binom{d-1}{k-1} + \binom{d-1}{k} \right) \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}.
\]
Next, we use the following identity:
\[
\sum_{i=0}^{t} \binom{n}{i} + \sum_{i=0}^{t+1} \binom{n}{i} = \sum_{i=0}^{t+1} \binom{n}{i-1} + \sum_{i=0}^{t+1} \binom{n}{i} = \sum_{i=0}^{t+1} \binom{n+1}{i},
\]
to get
\[
N_t(2, d) = \sum_{k=0}^{d-1} \binom{d-1}{k} \sum_{i=0}^{t-(d-1-k)} \binom{n-(d-1)}{i} + \sum_{k=0}^{\min\{d,t\}-1} \binom{d-1}{k} \sum_{i=0}^{t-k} \binom{n-(d-1)}{i} + \left( \binom{d-1}{\min\{d,t\}} \right).
\]
If \( t \geq d \) then \( \min\{d, t\} - 1 = \min\{d - 1, t\} \) and \( \binom{d-1}{\min\{d,t\}} = 0 \) so we get
\[
N_d(2, d) = \sum_{k=0}^{\min\{d-1,t\}} \binom{d-1}{k} t^{\max\{k,d-1-k\}} \sum_{i=0}^{\min\{d-1,t\}-1} \binom{n-(d-1)}{i} = N_d(2, d-1).
\]
Otherwise, \( t \leq d - 1 \), so \( \min\{d, t\} = \min\{d - 1, t\} = t \) and we get
\[
N_t(2, d) = \sum_{k=0}^{\min\{d-1,t\}-1} \binom{d-1}{k} t^{\max\{k,d-1-k\}} \sum_{i=0}^{\min\{d-1,t\}-1} \binom{n-(d-1)}{i} + \binom{d-1}{\min\{d-1,t\}} = N_t(2, d-1).
\]

3 Sequences Reconstruction Decoders

In this section, we show how to construct decoders for substitution errors, where the decoder has to output the transmitted word.

The case \( d = 1 \) was solved by Levenshine where the majority algorithm on each bit successfully decodes the transmitted word. The majority algorithm receives the estimations on each bit from every channel and simply decodes the bit according to a majority vote among all the channel estimations. According to Corollary 2, this algorithm works for \( d = 2 \) as well since the number of channels has to be the same. However, if \( d \) is greater than two, then the majority algorithm on each bit does not necessarily work. Furthermore, even decoding the output of the majority decoder using a decoder of the code which can correct at most \( \lfloor \frac{d-1}{2} \rfloor \) errors will not work in the worst case. The next example demonstrates this undesirable property.

**Example 1.** Assume that the transmitted word belongs to the Hamming code of length 7, there are at most two errors in every channel, and the zero word is the transmitted word \( c \). According to (2), there are \( N(2, 3) + 1 = 7 \) channels, and assume that the channel outputs are the following seven words:

\[
\begin{align*}
y_1 &= (1, 0, 1, 0, 0, 0, 0) \\
y_2 &= (1, 0, 0, 1, 0, 0, 0) \\
y_3 &= (1, 0, 0, 0, 1, 0, 0) \\
y_4 &= (1, 1, 0, 0, 0, 0, 0) \\
y_5 &= (0, 1, 1, 0, 0, 0, 0) \\
y_6 &= (0, 1, 0, 1, 0, 0) \\
y_7 &= (0, 1, 0, 0, 1, 0, 0)
\end{align*}
\]

Then, the output of the majority decoder on these seven words is the word \( (1, 1, 0, 0, 0, 0, 0) \). Thus, even this word suffers two errors with respect to the transmitted codeword and therefore the Hamming decoder would fail in its decoding. \( \square \)
In general, according to Corollary 2, if \( d \) is even then the number of channels for a code with minimum distance \( d \) or \( d - 1 \) is the same. Hence, we only need to solve here the case of odd minimum distance.

Assume that the transmitted word \( c \) belongs to a code \( C \) with odd minimum distance \( d \), there are at most \( t \) errors in every channel, where \( t > \frac{d-1}{2} \), and the number of channels is \( N = N(t, d) + 1 \). Assume also that \( n \) is relatively large enough with respect to \( d \) and \( t \). The \( N \) channel outputs are denoted by \( y_1, \ldots, y_N \) and are assumed to be different. Furthermore, the code \( C \) has a decoder \( D_C \), which can successfully correct at most \( \frac{d-1}{2} \) errors. We assume that this decoder is complete in the sense that for every input, it outputs a decoded word, while we only know that if the number of errors is at most \( \frac{d-1}{2} \), then the decoded word is the transmitted one.

A first observation in constructing a decoder is that it is always possible to detect whether the output word is the transmitted one. This can simply be done by checking if the distance between the output word and every channel output is at most \( t \).

**Lemma 3.** If \( c \) is the transmitted word, then for any \( \tilde{c} \in C, \tilde{c} = c \) if and only if

\[
\max_{1 \leq i \leq N} \{ d_H(\tilde{c}, y_i) \} \leq t.
\]

**Proof.** If \( \tilde{c} = c \) then every channel suffers at most \( t \) errors and thus \( \max_{1 \leq i \leq N} \{ d_H(\tilde{c}, y_i) \} \leq t \). In case \( \tilde{c} \neq c \), assume in the contrary that \( \max_{1 \leq i \leq N} \{ d_H(\tilde{c}, y_i) \} \leq t \). Then the set \( S_i(\{\tilde{c}, c\}) \) contains at least \( N = N(t, d) + 1 \) words in contradiction to the definition of \( N(t, d) \).

A naive algorithm can choose any of the channel outputs and add all error vectors of weight at most \( t - \frac{d-1}{2} \). For at least one of these error vectors we will get a word with at most \( \frac{d-1}{2} \) errors which can be decoded by the decoder of the code \( C \) and can be verified to be the correct transmitted word according to Lemma 3. The main drawback of this algorithm is its complexity. The number of vectors of weight at most \( t - \frac{d-1}{2} \) is of order \( \Theta \left( n^{t-(d-1)/2} \right) \). The number of channels \( N \) is of order \( \Theta \left( n^{t-(d+1)/2} \right) \) and thus the verification procedure in Lemma 3 has complexity \( \Theta \left( n^{t-(d+1)/2+1} \right) \).

Hence, all together the complexity of this decoder is \( \Theta \left( n^{2t-d+1} \cdot D(n) \right) \), where \( D(n) \) is the decoding complexity of the decoder \( D_C \).

The complexity of this algorithm can be improved as follows. Assume for example that \( t = \frac{d-1}{2} + 1 \). Then, there are two channel outputs, say \( y_1 \) and \( y_2 \), that are different in at least one bit location. If we flip this bit in both \( y_1 \) and \( y_2 \), then in exactly one of them the number of errors reduces by one and thus is at most \( \frac{d-1}{2} \), which can be decoded by \( D_C \). This idea can be generalized for arbitrary \( t \). Let \( \rho = t - \frac{d-1}{2} \), which corresponds to the number of additional errors on top of the error-correction capability.

**Lemma 4.** There exist two channel outputs \( y_i, y_j \) such that \( d_H(y_i, y_j) \geq 2\rho - 1 \).

**Proof.** Assume to the contrary that there are no such words. Then, the words \( y_1, y_2, \ldots, y_N \) form an anticode of diameter \( 2\rho - 2 \). The maximum size of such an anticode is \( \beta_{\rho-1, n} = \sum_{i=0}^{\rho-1} \binom{n}{i} \),
while according to (2) the value of $N$ satisfies

$$N > N(t, d) = \sum_{i=0}^{t-d+1} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k} = \sum_{i=0}^{\rho-1} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k} > \sum_{i=0}^{\rho-1} \binom{n}{i} = b_{\rho-1,n}.$$

The decoding algorithm works as follows.

**Algorithm 1: Decoding**

The input to the decoder is the set of all $N$ channel outputs $y_1, \ldots, y_N$ and it returns an estimation $\hat{c}$ on $c$.

1. Find two words $y_i, y_j$ such $d_H(y_i, y_j) \geq 2\rho - 1$, and let $I = \{i_1, i_2, \ldots, i_{2\rho-1}\}$ be a set of $2\rho - 1$ different indices that the two vectors are different from each other.

2. For all vectors $e$ of weight $\rho$ on the $2\rho - 1$ indices of the set $I$,
   
   (a) $D(y_i + e) = \hat{c}_1$, $D(y_j + e) = \hat{c}_2$.
   
   (b) If $\max_{1 \leq i \leq N} \{d_H(\hat{c}_1, y_i)\} \leq t$, $\hat{c} = \hat{c}_1$.
   
   (c) If $\max_{1 \leq i \leq N} \{d_H(\hat{c}_2, y_i)\} \leq t$, $\hat{c} = \hat{c}_2$.

**Theorem 5.** The output of Algorithm 1 satisfies $\hat{c} = c$.

**Proof.** The success of Step 1 is guaranteed according to Lemma 4. For every index $i_j, 1 \leq j \leq 2\rho - 1$, exactly one of the channel outputs $y_i$ or $y_j$ has an error. Therefore, either $y_i$ or $y_j$ has at least $\rho$ errors on these indices. Without loss of generality assume it is $y_i$ and let $E \subseteq \{i_1, \ldots, i_{2\rho-1}\}$ be a subset of its error locations, where $|E| = \rho$. In Step 2 we exhaustively search over all error vectors $e$ of weight $\rho$ on these $2\rho - 1$ indices. For every error vector $e$ let $E_e = \{i : e_i = 1\}$. Therefore, there exists an error vector $e_1$ such that $E_{e_1} = E$. Hence, $d_H(c, y_i + e_1) \leq \frac{d-1}{2}$, so the decoder in Step 2.b succeeds. Finally, the algorithm succeeds and $\hat{c} = c$. \qed

A naive implementation of Step 1 results with complexity $\Theta(N^2n) = \Theta(n^{2t-d})$. The number of vectors $e$ in Step 2 is constant since $\rho$ is constant as well, and thus the complexity of this step is $\Theta(Nn)$, since we assume that the complexity of the decoder $D_C$ is less than the order of channel outputs $N$. Together, we deduce that the complexity order of Algorithm 1 is $\Theta(n^{2t-d})$, which improves upon the decoding complexity of the naive algorithm.