On the Uncertainty of Information Retrieval in Associative Memories

Eitan Yaakobi, *Senior Member, IEEE*, and Jehoshua Bruck, *Fellow, IEEE*

***Abstract*— We (people) are memory machines. Our decision processes, emotions, and interactions with the world around us are based on and driven by associations to our memories. This natural association paradigm will become critical in future memory systems, namely, the key question will not be "How do I store more information?" but rather "Do I have the relevant information? How do I retrieve it?" The focus of this paper is to make a first step in this direction. We define and solve a very basic problem in associative retrieval. Given a word W, the words in the memory, which are t-associated with W, are the words in the ball of radius t around W. In general, given a set of words, say W, X, and Y, the words that are t-associated with $\{W, X, Y\}$ are those in the memory that are within distance t *from all the three words.* Our main goal is to study the maximum size of the t-associated set as a function of the number of input words and the minimum distance of the words in memory—we call this value *the uncertainty of an associative memory*. In this paper, we consider the Hamming distance and derive the uncertainty of the associative memory that consists of all the binary vectors with an arbitrary number of input words. In addition, we study the retrieval problem, namely, how do we get the t-associated set given the inputs? We note that this paradigm is a generalization of the sequences reconstruction problem that was proposed by Levenshtein (2001). In this model, a word is transmitted over multiple channels. A decoder receives all the channel outputs and decodes the transmitted word. Levenshtein computed the minimum number of channels that guarantee a successful decoder—this value happens to be the uncertainty of an associative memory with two input words.**

***Index Terms*— Associative memories, reconstruction of sequences, list decoding, anticodes.**

I. Introduction

ONE of the interpretations of the term *association*, especially in the context of psychology, is the connection between two or more concepts. Throughout our life, we remember and store an enormous amount of information. However, while we are not aware of the method this information is stored, amazingly, it can be accessed and retrieved with relative ease. The way we think and process information is mainly performed by associations. Our memory content retrieval process is done by stimulating it with external or internal inputs. That is, the knowledge of some information gives rise to related information that was stored earlier. Mathematically speaking, assume the memory is a set of words $\mathcal{M} = \{m_1, \ldots, m_S\}$. Then, given an arbitrary word $\boldsymbol{x}$ as an input to the memory M, its output is another word or a set of words from the memory M, or from the space of all words, that are related or close to the input word $\boldsymbol{x}$. Here, the term "close" can be interpreted as using any distance metric between words, for example the Hamming distance, which is the distance we consider in this work. A word is associated with another word or words which again can be associated with more words and so on, resulting in a sequence of associations.[1]

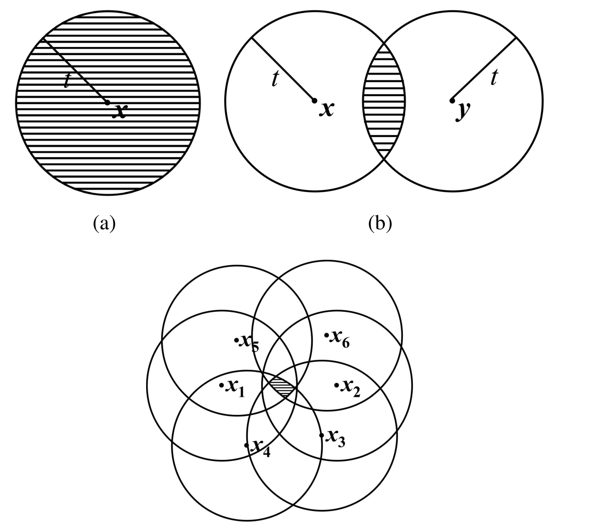

Fig. 1. Three example of associated words: (a) associated words with a single word $\boldsymbol{x}$, (b) associated words with two words $\boldsymbol{x}$ and $\boldsymbol{y}$, (c) associated words with the six words $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6$.

From the information theory perspective, we say that the words associated with an input word $\boldsymbol{x}$ are those in distance at most t (a prescribed value) from $\boldsymbol{x}$. This set comprises a ball of radius t, see Fig. 1(a). We generalize this paradigm and consider a *set of words* that are presented as an input

[1]Note that the model of associativity we study in this work is essentially different than the one studied by Hopfield for neural networks [9], [26] and is motivated by the association between words in a given memory.

Manuscript received September 12, 2017; revised July 20, 2018; accepted October 20, 2018. Date of publication October 30, 2018; date of current version March 15, 2019. This work was supported in part by the ISEF Foundation, in part by the Lester Deutsch Fellowship, in part by the National Science Foundation under Grant CCF-1116739, and in part by the NSF Expeditions in Computing Program under Grant CCF-0832824. This paper was presented at the 2012 IEEE International Symposium on Information Theory [31].

E. Yaakobi was with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA. He is now with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

J. Bruck is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: bruck@caltech.edu).

Communicated by J. Kliewer, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2018.2878750

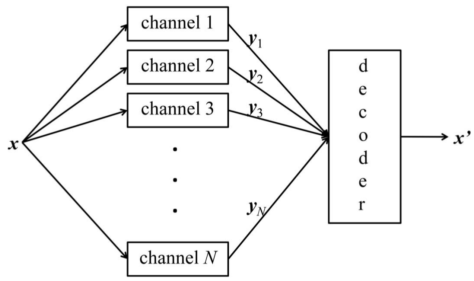

Fig. 2. Channel model of the sequences reconstruction problem.

to the memory. For example, for two input words $\boldsymbol{x}, \boldsymbol{y}$, their set of associated words are the ones that their distance from both $\boldsymbol{x}$ and $\boldsymbol{y}$ is at most t, see Fig. 1(b). Clearly, this set of associations is likely to be smaller than the ball of radius t. For more than two input words, the set of associated words is usually getting even smaller, and in general, the larger the set of input words is, the smaller the set of associated words is, see Fig. 1(c). For example, assume the input word is "tall", then many words can be associated with it, such as "tree", "mountain", "tower", "ladder", etc. But if the input words are "tall" and "fruit", then out of these four associated words, only the word "tree" will be associated with "tall" and "fruit".

Assume that the memory is the set of all binary vectors, $\mathcal{M} = \{0, 1\}^n$. For any input word $\boldsymbol{x}$, it is immediate to see that if its set of associated words are the ones of distance at most t, then there are $\sum_{i=0}^{t} \binom{n}{i}$ such words. However, for two input words, $\boldsymbol{x}, \boldsymbol{y}$, the problem of finding the set of associated words of distance at most t from both $\boldsymbol{x}$ and $\boldsymbol{y}$ becomes more complex. In fact, this problem was proposed and solved by Levenshtein [21], [22]. The motivation came from a completely different scenario in the context of the *sequences reconstruction problem*. In this model, a codeword $\boldsymbol{x}$ 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. 2. If $\boldsymbol{x}$ belongs to a code $\mathcal{C}$ with minimum distance d and in every channel there can be at most some $t > \lfloor \frac{d-1}{2} \rfloor$ errors, then Levenshtein studied the minimum number of channels that guarantees the existence of a successful decoder. This number has to be greater than

$$N = \max_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}, \boldsymbol{x}_1 \neq \boldsymbol{x}_2} |B_t(\boldsymbol{x}_1) \cap B_t(\boldsymbol{x}_2)|, \qquad (1)$$

where $B_t(\boldsymbol{x})$ is the ball of radius t surrounding $\boldsymbol{x}$. To see that, notice that if the intersection of the radius-t balls of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ 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.

The motivation to the model studied by Levenshtein came from fields such as chemistry and biology, where the redundancy in the codewords is not sufficient to construct a successful decoder. Thus, the only way to combat errors is by repeatedly transmitting the same codeword. Recently, this model was shown to be also relevant in storage technologies [4], [5], [32]. Due to the high capacity and density of today's and future's storage medium, it is no longer possible to read individual memory elements, but, rather, only a multiple of them at once. Hence, every memory element is read multiple times, which is translated into multiple estimations of the same information stored in the memory.

Finding the maximum intersection problem in (1) was studied in [22] with respect to the Hamming distance and other metric distances, such as the Johnson graph, asymmetric errors, and more general metric distances. In [16]–[18], it was analyzed over permutations, and in [24] and [25] for error graphs. The case of permutations with the Kendall's τ distance was also investigated in [33] as well as the Grassmann graph case. In [23], the equivalent problem for insertions and deletions was studied. These results were later extended in [29] for insertions and in [6] for deletions. Furthermore, reconstruction algorithms for this model were given in [3], [14], and [30]. The case of only deletions was solved in [8] in the context of trace reconstruction and in [27] and [28] an information-theoretic study was carried for a special case of deletions which is applied to DNA sequences. This problem was also studied under a different context in [10] for the purpose of asymptotically improving the Gilbert-Varshamov bound.

Returning to our original problem, the set of associated words with $\boldsymbol{x}$ and $\boldsymbol{y}$ is $B_t(\boldsymbol{x}) \cap B_t(\boldsymbol{y})$ and the maximum intersection is the value N in (1). The generalized problem of finding the maximum size of associated words of $m \geqslant 2$ input words with mutual distance at least d is expressed as

$$N_t(m, d) = \max_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, d_H(\boldsymbol{x}_i, \boldsymbol{x}_j) \geqslant d} \left\{ \left| \cap_{i=1}^{m} B_t(\boldsymbol{x}_i) \right| \right\}. \qquad (2)$$

The value of $N_t(m, d)$ will be referred in the paper as the *uncertainty of the memory*, and the main goal in this paper is to analyze the value of $N_t(m, d)$ with respect to the Hamming distance for different values of t, m, d. In particular, we show that if $A(D)$ is the size of a maximal anticode of diameter D [2], that is, the largest set of words with maximum distance D, then $N_t(A(D), 1) = A(2t - D)$.

Extensions of the problem proposed in this work and in [31] were studied later in [7], [11]–[13], [19], and [20]. Junnila and Laihonen [11] studied conditions on the associative memory to have small uncertainty and in particular uncertainty equals 1. They carried this task for the Hamming space as well as the infinite square grid. This work was extended in [13] to find conditions on the memory such that its uncertainty equals 1 when the input clues belong to a ball of some radius t. A similar model was studied in [12], where the authors considered the setup of uncertainty 1 while the radius for associations is 1 as well. Yet another model was studied in [19] for the average number of input clues which guarantee uncertainty 1. In [20], connections to the problem of list decoding of the reconstruction problem by Levenshtein using the majority voting decoder were investigated. Gripon and Rabbat [7] studied associative memories which are built on the maximum likelihood principle.

The rest of the paper is organized as follows. In Section II, we define the concept of associative memories, describe the connection to the sequences reconstruction problem, and review some results for $m = 2$. In Section III we extend these results for $m = 3$. In Section IV, we solve the problem stated in (2) for the case $d = 1$. Extensions for arbitrary d are given in Section V. In Section VI, we give efficient decoders to the reconstruction problem studied by Levenshtein. Finally, Section VII concludes the paper.

II. Definitions and Basic Properties

In this work, the words are binary vectors of length n. The Hamming distance between two words $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $d_H(\boldsymbol{x},\boldsymbol{y})$ and the Hamming weight of a word $\boldsymbol{x}$ is denoted by $w_H(\boldsymbol{x})$. For a word $\boldsymbol{x} \in \{0,1\}^n$, $B_t^n(\boldsymbol{x})$ is its surrounding ball of radius t, $B_t^n(\boldsymbol{x}) = \{\boldsymbol{y} \in \{0,1\}^n \ : \ d_H(\boldsymbol{x},\boldsymbol{y}) \leqslant t\}$. The size of $B_t^n(\boldsymbol{x})$, comprising of length-n words, is $b_{t,n} = \sum_{i=0}^{t}\binom{n}{i}$. If the length of the words is clear from the context, we use the notation $B_t(\boldsymbol{x})$. For $1 \leqslant i \leqslant n$, $\boldsymbol{e}_i$ is the unit vector where only its i-th bit is one, and $\boldsymbol{0}$ is the all-zero vector. Two words $\boldsymbol{x},\boldsymbol{y} \in \{0,1\}^n$ are called *t-associated* if $d_H(\boldsymbol{x},\boldsymbol{y}) \leqslant t$.

*Definition: The **t-associated set** of the words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ is denoted by $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\})$ and is defined to be the set of all words $\boldsymbol{y}$ that are t-associated with $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$,*

$$S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big) = \{\boldsymbol{y} : d_H(\boldsymbol{y},\boldsymbol{x}_i) \leqslant t,\ 1 \leqslant i \leqslant m\} = \bigcap_{i=1}^{m} B_t(\boldsymbol{x}_i).$$

Note that for a single word $\boldsymbol{x}$, we have $S_t\big(\{\boldsymbol{x}\}\big) = B_t(\boldsymbol{x})$. Given an associative memory $\mathcal{M}$, we define the maximum size of a t-associated set of any m words from the memory.

*Definition: Let $\mathcal{M}$ be an associative memory and m,t be two positive integers. The **uncertainty of the associative memory** $\mathcal{M}$ for m and t, denoted by $N_t(m,\mathcal{M})$, is the maximum size of a t-associated set of m different input words from $\mathcal{M}$. That is,*

$$N_t(m,\mathcal{M}) = \max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m \in \mathcal{M},\boldsymbol{x}_i \neq \boldsymbol{x}_j} \left\{\left|S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big)\right|\right\}. \quad (3)$$

In case the associative memory $\mathcal{M}$ is a code with minimum Hamming distance d, we will use the notation $N_t(m,d)$ instead of $N_t(m,\mathcal{M})$, where the former term refers to the uncertainty value among all possible codes with minimum Hamming distance d. That is, we will study the value

$$N_t(m,d) = \max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m, d_H(\boldsymbol{x}_i,\boldsymbol{x}_j) \geqslant d} \left\{\left|S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big)\right|\right\}. \quad (4)$$

For example, $N_t(m,1)$ refers to $N_t(m,\{0,1\}^n)$.

We now give the definitions that establish the connection with the channel model by Levenshtein [22]. Assume a codeword $\boldsymbol{x}$ is transmitted over N channels. The channel outputs, denoted by $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N$, are all different from each other and belong to the ball $B_t(\boldsymbol{x})$ (Fig. 2). A *list decoder* $\mathcal{D}_{\mathcal{L}}$ receives the N channel outputs and returns a list of at most $\mathcal{L}$ words $\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_\ell$, where $\ell \leqslant \mathcal{L}$. We call it an *$\mathcal{L}$-decoder* $\mathcal{D}_{\mathcal{L}}$. The $\mathcal{L}$-decoder $\mathcal{D}_{\mathcal{L}}$ is said to be *successful* if the transmitted word $\boldsymbol{x}$ belongs to the decoded output list, i.e.,

$$\boldsymbol{x} \in \mathcal{D}_{\mathcal{L}}(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N) = \{\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_\ell\}.$$

Note that here the model of list decoding is slightly different than the traditional one in the sense that the decoder receives and outputs a list of words, as opposed to the well-studied model in which the decoder receives a single word. In fact, the latter case of $\mathcal{L} = 1$ is the model studied by Levenshtein [22]. In this case, if the word $\boldsymbol{x}$ belongs to a code $\mathcal{C}$ with minimum distance d and $t \leqslant (d-1)/2$ then one channel is sufficient to construct a successful $\mathcal{L}$-decoder for all $\mathcal{L} \geqslant 1$. However, if $t > (d-1)/2$ then more than one channel is necessary.

The next theorem establishes the connection between the value of $N_t(m,d)$ and the decoding success of an $\mathcal{L}$-decoder.

Theorem 1: Assume the transmitted word $\boldsymbol{x}$ belongs to a code $\mathcal{C}$ of minimum distance d. Then, there exists a successful $\mathcal{L}$-decoder with N channels if and only if

$$N \geqslant N_t(\mathcal{L}+1,d)+1.$$

Proof: Assume to the contrary that the number of channels is $N_t(\mathcal{L}+1,d)$ and let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}$ be $\mathcal{L}+1$ words such that the set $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}\})$ contains $N_t(\mathcal{L}+1,d)$ words. If one of these $\mathcal{L}+1$ words is the transmitted one and the received channel outputs are the $N_t(\mathcal{L}+1,d)$ words in $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}\})$, then any of the $\mathcal{L}+1$ words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}$ could be the transmitted one and thus can belong to the decoder's output list. Hence, the transmitted word may not belong to the output list.

On the other hand, if there are $N_t(\mathcal{L}+1,d)+1$ channels, then for any transmitted word $\boldsymbol{x}$, there are at most $\mathcal{L}$ words in $\mathcal{C}$, all of distance at least d from each other, such that the $N_t(\mathcal{L}+1,d)+1$ channel outputs are located in the intersection of their radius-t balls. ■

For $\mathcal{L} = 1$, the value $N_t(2,d)$ was studied by Levenshtein [22]. For the completeness of the results in the paper and since we show a slightly different presentation of the value $N_t(2,d)$, we prove this result in the next lemma.

Lemma 2: Let d,t be two positive integers such that $t > \lfloor\frac{d-1}{2}\rfloor$, then the value of $N_t(2,d)$ satisfies

$$N_t(2,d) = \sum_{i=0}^{t-\lceil\frac{d}{2}\rceil}\binom{n-d}{i}\sum_{k=d-t+i}^{t-i}\binom{d}{k}, \quad (5)$$

and

$$N_t(2,d) = \sum_{k=0}^{\min\{d,t\}}\binom{d}{k}\sum_{i=0}^{t-\max\{k,d-k\}}\binom{n-d}{i}. \quad (6)$$

Proof: Assume $d_H(\boldsymbol{x},\boldsymbol{y}) = d$ and the goal is to find the cardinality of the set

$$S_t(\{\boldsymbol{x},\boldsymbol{y}\}) = \{\boldsymbol{z} \in \{0,1\}^n \ : \ d_H(\boldsymbol{z},\boldsymbol{x}), d_H(\boldsymbol{z},\boldsymbol{y}) \leqslant t\}.$$

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

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

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

$$|S_{0,1}| + |S_{1,1}| \leqslant t, \quad |S_{0,1}| + |S_{1,0}| \leqslant t,$$

or

$$|S_{0,1}| + |S_{1,1}| \leqslant t, \quad |S_{0,1}| + d - |S_{1,1}| \leqslant t.$$

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

$$i + k \leqslant t, \quad i + d - k \leqslant t,$$

or

$$0 \leqslant i \leqslant t - \lceil d/2 \rceil, \quad i + d - t \leqslant k \leqslant t - i.$$

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

$$|S_t(\{\boldsymbol{x}, \boldsymbol{y}\})| = \sum_{i=0}^{t-\lceil d/2\rceil} \binom{n-d}{i} \sum_{k=i+d-t}^{t-i} \binom{d}{k},$$

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

$$N_t(2, d) = \sum_{i=0}^{t-\lceil d/2\rceil} \binom{n-d}{i} \sum_{k=i+d-t}^{t-i} \binom{d}{k}.$$

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

$$N_t(2, d) = \sum_{k=0}^{\min\{d,t\}} \binom{d}{k} \sum_{i=0}^{t-\max\{k,d-k\}} \binom{n-d}{i}.$$

■

The value of $N_t(2, d)$ was given by Levenshtein according to equation (5). However, we use the second presentation of this term in equation (6) in order to prove the following property. We will later use it in Section VI when we construct explicit decoders for the reconstruction problem. The proof is given in Appendix A.

Corollary 3: Let t, d be two positive integers such that d is even, then

$$N_t(2, d) = N_t(2, d-1).$$

III. Largest Intersection of Three Balls

In this section, we focus on the case where there are three words, i.e. $m = 3$, and we find the value of $N_t(3, d)$ for all t and d. We show that in a similar fashion to the way $N_t(2, d)$ was calculated in [22] and was reviewed in Section II.

Let us first start with the case where d is even and we find the size of the set $S_t(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})$ where $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{y}, \boldsymbol{z}) = d$. This is proved in the next Lemma.

Lemma 4: Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \{0, 1\}^n$ be such that $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{y}, \boldsymbol{z}) = d$ and d is even, then

$$|S_t(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})| = \sum_{i_1,i_2,i_3,i_4} \binom{n-\frac{3d}{2}}{i_1} \binom{\frac{d}{2}}{i_2} \binom{\frac{d}{2}}{i_3} \binom{\frac{d}{2}}{i_4},$$

where i_1, i_2, i_3, i_4 satisfy the following constraints:

1) $0 \leqslant i_1 \leqslant t - \frac{d}{2}$,
2) $i_1 + \frac{d}{2} - t \leqslant i_4 \leqslant t - \frac{d}{2} - i_1$,
3) $d - t + i_1 \leqslant i_3 \leqslant t - (i_1 + i_4)$,
4) $\max\{i_1 - i_3 - i_4 + \frac{3d}{2} - t, i_1 + i_3 + i_4 + \frac{d}{2} - t\} \leqslant i_2 \leqslant t - (i_1 + i_4 + \frac{d}{2} - i_3)$.

Proof: We find the cardinality of the set

$$S_t(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}) = \{\boldsymbol{u} \in \{0, 1\}^n : d_H(\boldsymbol{u}, \boldsymbol{x}), d_H(\boldsymbol{u}, \boldsymbol{y}), d_H(\boldsymbol{u}, \boldsymbol{z}) \leqslant t\}.$$

Let us define the following eight sets:

$$S_{b_1,b_2,b_3} = \{i : y_i = x_i + b_1, z_i = x_i + b_2, u_i = x_i + b_3\},$$

for $b_1, b_2, b_3 \in \{0, 1\}$. Since $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{y}, \boldsymbol{z}) = d$ we get the following equalities

$$|S_{1,0,0}| + |S_{1,0,1}| + |S_{1,1,0}| + |S_{1,1,1}| = d,$$
$$|S_{0,1,0}| + |S_{0,1,1}| + |S_{1,1,0}| + |S_{1,1,1}| = d,$$
$$|S_{0,1,0}| + |S_{0,1,1}| + |S_{1,0,0}| + |S_{1,0,1}| = d,$$

so we derive that

$$|S_{0,1,0}| + |S_{0,1,1}| = |S_{1,0,0}| + |S_{1,0,1}| = |S_{1,1,0}| + |S_{1,1,1}| = \frac{d}{2}, \tag{7}$$

and

$$|S_{0,0,0}| + |S_{0,0,1}| = n - \frac{3d}{2}. \tag{8}$$

Define

$$|S_{0,0,1}| = i_1, \quad |S_{0,1,0}| = i_2, \quad |S_{1,0,0}| = i_3, \quad |S_{1,1,0}| = i_4,$$

and we also have the following inequalities:

$$d_H(\boldsymbol{u}, \boldsymbol{x}) = i_1 + \frac{d}{2} - i_2 + \frac{d}{2} - i_3 + \frac{d}{2} - i_4 \leqslant t,$$
$$d_H(\boldsymbol{u}, \boldsymbol{y}) = i_1 + \frac{d}{2} - i_2 + i_3 + i_4 \leqslant t,$$
$$d_H(\boldsymbol{u}, \boldsymbol{z}) = i_1 + i_2 + \frac{d}{2} - i_3 + i_4 \leqslant t,$$

where $0 \leqslant i_2, i_3, i_4 \leqslant \frac{d}{2}$ from (7) and $0 \leqslant i_1 \leqslant n - \frac{3d}{2}$ from (8). Finally, the cardinality of the set $S_t(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})$ is the number of solutions to the above inequalities including the choices of the coordinates we can choose for every selection of i_1, i_2, i_3, i_4. That is,

$$\sum_{i_1,i_2,i_3,i_4} \binom{n-\frac{3d}{2}}{i_1} \binom{\frac{d}{2}}{i_2} \binom{\frac{d}{2}}{i_3} \binom{\frac{d}{2}}{i_4},$$

where i_1, i_2, i_3, i_4 satisfy the following constraints.

1) $0 \leqslant i_1 \leqslant t - \frac{d}{2}$,
2) $i_1 + \frac{d}{2} - t \leqslant i_4 \leqslant t - \frac{d}{2} - i_1$,
3) $d - t + i_1 \leqslant i_3 \leqslant t - (i_1 + i_4)$,
4) $\max\{i_1 - i_3 - i_4 + \frac{3d}{2} - t, i_1 + i_3 + i_4 + \frac{d}{2} - t\} \leqslant i_2 \leqslant t - (i_1 + i_4 + \frac{d}{2} - i_3)$.

■

In case d is odd then the sum of all three distances $d_H(\boldsymbol{x}, \boldsymbol{y}), d_H(\boldsymbol{x}, \boldsymbol{z}), d_H(\boldsymbol{y}, \boldsymbol{z})$ has to be even. Therefore, one of the three distances is $d+1$. We state the equivalent of Lemma 4 for d odd and its proof appears in Appendix B.

Lemma 5: Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \{0, 1\}^n$ such that $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{z}) = d, d_H(\boldsymbol{y}, \boldsymbol{z}) = d + 1$ and d is odd, then

$$|S_t(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})| = \sum_{i_1,i_2,i_3,i_4} \binom{n-\frac{3d+1}{2}}{i_1} \binom{\frac{d+1}{2}}{i_2} \binom{\frac{d+1}{2}}{i_3} \binom{\frac{d-1}{2}}{i_4},$$

where i_1, i_2, i_3, i_4 satisfy the following constraints:

1) $0 \leqslant i_1 \leqslant t - \frac{d+1}{2}$,
2) $i_1 + \frac{d-1}{2} - t \leqslant i_4 \leqslant t - \frac{d+1}{2} - i_1$,
3) $d + 1 - t + i_1 \leqslant i_3 \leqslant t - (i_1 + i_4)$,
4) $\max\{i_1 - i_3 - i_4 + \frac{3d+1}{2} - t, i_1 + i_3 + i_4 + \frac{d+1}{2} - t\} \leqslant i_2 \leqslant t - (i_1 + i_4 + \frac{d+1}{2} - i_3)$,

According to Lemma 4 and Lemma 5, we can show the following theorem.

Theorem 6: Let t, d be such that $t > \lceil \frac{d-1}{2} \rceil$, and n large enough. The value of $N_t(3, d)$ is given by

$$N_t(3, d) = \sum_{i_1, i_2, i_3, i_4} \binom{n - \lceil \frac{3d}{2} \rceil}{i_1} \binom{\lceil \frac{d}{2} \rceil}{i_2} \binom{\lceil \frac{d}{2} \rceil}{i_3} \binom{\lfloor \frac{d}{2} \rfloor}{i_4},$$

where i_1, i_2, i_3, i_4 satisfy the following constraints:

1) $0 \leqslant i_1 \leqslant t - \lceil \frac{d}{2} \rceil$,
2) $i_1 + \lfloor \frac{d}{2} \rfloor - t \leqslant i_4 \leqslant t - \lceil \frac{d}{2} \rceil - i_1$,
3) $2\lceil \frac{d}{2} \rceil - t + i_1 \leqslant i_3 \leqslant t - (i_1 + i_4)$,
4) $\max\{i_1 - i_3 - i_4 + \lceil \frac{3d}{2} \rceil - t, i_1 + i_3 + i_4 + \lceil \frac{d}{2} \rceil - t\} \leqslant i_2 \leqslant t - (i_1 + i_4 + \lceil \frac{d}{2} \rceil - i_3)$.

Proof: We only need to show that $N_t(3, d)$ cannot be greater than the values specified in Lemma 4 for d even and in Lemma 5 for d odd. Let us start with the case where d is even. If there are two words of distance at least $d + 1$, say $d_H(\boldsymbol{x}, \boldsymbol{y}) \geqslant d + 1$, then according to the expression we calculated in Lemma 2, the cardinality of the $S_t(\{\boldsymbol{x}, \boldsymbol{y}\})$ is at most of order $O(n^{t-\frac{d}{2}-1})$, while the value we found in Lemma 4 has order $O(n^{t-\frac{d}{2}})$.

For d odd, if there are two words at distance $d + 2$ then we prove as in the first case by the difference of the orders of the two sets. Hence, all the distances are between d and $d+1$ and since the sum of the three distances is even we only need to consider the case $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{y}) = d + 1$. However, it is possible to verify that that the cardinality of the set $S(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})$ in Lemma 4 for $d + 1$ is not greater than the cardinality of the set $S(\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\})$ in Lemma 5 for d. ■

IV. Intersection of Multiple Balls

In this section, we analyze the value of $N_t(m, d)$ for $d = 1$. A first observation on the value of $N_t(m, 1)$ is stated in the next lemma.

Lemma 7: For $m, t \geqslant 1$, if $N_t(m, 1) \geqslant \ell$ and $N_t(m+1, 1) < \ell$, then $N_t(\ell, 1) = m$.

Proof: Since $N_t(m, 1) \geqslant \ell$, there exist m different words $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ such that

$$|S_t(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\})| = |B_t(\boldsymbol{x}_1) \cap \cdots \cap B_t(\boldsymbol{x}_m)| \geqslant \ell$$

and assume $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell$ are ℓ words which belong to this intersection. Therefore, $d_H(\boldsymbol{x}_i, \boldsymbol{y}_j) \leqslant t$ for all $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant \ell$, and thus

$$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subseteq S_t(\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell\}) = B_t(\boldsymbol{y}_1) \cap \cdots \cap B_t(\boldsymbol{y}_\ell),$$

and hence $N_t(\ell, 1) \geqslant m$.

Assume to the contrary that $N_t(\ell, 1) \geqslant m + 1$ and let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell$ be ℓ words such that

$$|S_t(\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell\})| = |B_t(\boldsymbol{z}_1) \cap \cdots \cap B_t(\boldsymbol{z}_\ell)| \geqslant m + 1.$$

As in the first part, we get that $N_t(m + 1, 1) \geqslant \ell$, which is a contradiction. Hence, $N_t(\ell, 1) = m$. ■

In general, for a given set of words $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, the closer the words are, the larger the size of the set $S_t(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\})$ is, however we also note that the size of this set can also remain the same. In case $d = 1$, we look for a set of words that are all close to each other, or equivalently - the maximum distance between all pairs of words is minimized.

An *anticode* of *diameter* D is a set $A \subseteq \{0, 1\}^n$ of words such that the maximum distance between every two words in A is at most D. That is, for all $x, y \in A$, $d_H(x, y) \leqslant D$. For $D \geqslant 1$, $A(D)$ is the size of the largest anticode of diameter D. It was shown in [15] that the value of $A(D)$ is given by

$$A(D) = \begin{cases} b_{\frac{D}{2}, n} & \text{if } D \text{ is even,} \\ 2b_{\frac{D-1}{2}, n-1} & \text{if } D \text{ is odd.} \end{cases}$$

Our next goal is to show that for all $D \geqslant 1$,

$$N_t(A(D), 1) = A(2t - D).$$

That is, the t-associated set of a maximum anticode of diameter D is a maximum anticode of diameter $2t - D$.

Lemma 8: For all $0 \leqslant D \leqslant 2t \leqslant n$,

$$N_t(A(D), 1) \geqslant A(2t - D).$$

Proof: Assume that D is even. We take the $A(D)$ words in $B_{D/2}(\mathbf{0})$ and consider the set

$$S_t\left(B_{D/2}(\mathbf{0})\right) = \bigcap_{\boldsymbol{x} \in B_{D/2}(\mathbf{0})} B_t(\boldsymbol{x}).$$

Then, $B_{t-D/2}(\mathbf{0}) \subseteq S_t\left(B_{D/2}(\mathbf{0})\right)$ and hence $N_t(A(D), 1) \geqslant A(2t - D)$ for even D.

In case that D is odd, let $i = (D - 1)/2$. Let us start with a maximal anticode of diameter $2i + 1$. Let X be the set

$$X = B_i(\mathbf{0}) \cup B_i(\boldsymbol{e}_1) = \{a\boldsymbol{w} : a \in \{0, 1\}, \boldsymbol{w} \in B_i^{n-1}(\mathbf{0})\},$$

and let

$$Y = B_{t-i-1}(\mathbf{0}) \cup B_{t-i-1}(\boldsymbol{e}_1) = \{b\boldsymbol{u} : b \in \{0, 1\}, \boldsymbol{u} \in B_{t-i-1}^{n-1}(\mathbf{0})\}.$$

Then, for every $\boldsymbol{x} \in X, \boldsymbol{y} \in Y$, $d_H(\boldsymbol{x}, \boldsymbol{y}) \leqslant t$. Therefore, $N_t(A(D), 1) \geqslant A(2t - D)$ for odd D as well. ■

The equivalent upper bound is proved in the next two lemmas.

Lemma 9: For all $0 \leqslant D \leqslant 2t$ and $n \geqslant (t - \frac{D}{2})(2^{D+1} + 1)$, where D is even,

$$N_t(A(D) + 1, 1) < A(2t - D).$$

Proof: Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{A(D)+1}\}$ be a set of $A(D) + 1$ words. Since the largest anticode with diameter D has size $A(D)$, there exist two words, say $\boldsymbol{x}_1, \boldsymbol{x}_2$, where $d_H(\boldsymbol{x}_1, \boldsymbol{x}_2) \geqslant D + 1$. Hence, the size of $S_t(X)$ is no greater than the size of $S_t(\{\boldsymbol{x}_1, \boldsymbol{x}_2\})$, which, according to (5), is at most

$$M = \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n - D - 1}{j} \sum_{k=D+1-t+j}^{t-j} \binom{D+1}{k}.$$

Note that

$$M < \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n}{j} 2^{D+1}.$$

For $0 \leqslant j \leqslant t-(\frac{D}{2}+1)$ and $(t-\frac{D}{2})(2^{D+1}+1)$, we have $\binom{n}{j}2^{D+1} \leqslant \binom{n}{j+1}$ and hence,

$$M < \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n}{j+1} < \sum_{j=0}^{t-\frac{D}{2}} \binom{n}{j} = A(2t-D).$$

■

An equivalent property for D odd is proved in the next lemma. In this proof and the rest of the paper we let $\boldsymbol{e}_i^j$ be the vector where its ℓ-th bit is one if and only if $\ell \in \{i, \ldots, j\}$.

Lemma 10: For all $0 \leqslant D \leqslant 2t$, where D is odd, and n large enough,

$$N_t(A(D)+1, 1) < A(2t-D).$$

Proof: Let $D = 2i+1$. For every set X with $A(D)+1$ words there are at least two words of distance at least $D+1 = 2i+2$. We first note that $|X| = \Theta(n^i)$ and $|S_t(X)| = \Theta(n^{t-i-1})$ and according to this observation the following three claims follow from cardinality arguments and using the previous results on the intersection of two or three balls.

Claim 1: There are no two words $\boldsymbol{x}, \boldsymbol{y} \in X$, such that $d_H(\boldsymbol{x}, \boldsymbol{y}) \geqslant 2i+3$.

Claim 2: There are no three words $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in X$, such that $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{y}, \boldsymbol{z}) = 2i+2$.

Claim 3: There are no three words $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ in X such that $d_H(\boldsymbol{x}, \boldsymbol{y}) = 2i+2, d_H(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{y}, \boldsymbol{z}) = 2i+1$.

Let $\boldsymbol{x}, \boldsymbol{y} \in X$ be two words such that $d_H(\boldsymbol{x}, \boldsymbol{y}) = 2i+2$. Without loss of generality we can assume that $\boldsymbol{x} = \boldsymbol{e}_1^{i+1}$, and $\boldsymbol{y} = \boldsymbol{e}_{i+2}^{2i+2}$. Let $I_0' = \{1, \ldots, i+1\}$, $I_1' = \{i+2, \ldots, 2i+2\}$, $I_1 = I_0' \cup I_1'$ and $I_2 = \{1, \ldots, n\} \setminus I_1$. For a vector $\boldsymbol{x}$ and a set $I = \{i_1, \ldots, i_\ell\}$, such that $i_1 < \cdots < i_\ell$, the vector $\boldsymbol{x}_I$ is defined by $\boldsymbol{x}_I = (x_{i_1}, \ldots, x_{i_\ell})$.

We start with some simple observations on the weight of words from X on the set I_2. There is no word $\boldsymbol{z} \in X$ such that $w_H(\boldsymbol{z}_{I_2}) \geqslant i+2$. Otherwise, $\max\{d_H(\boldsymbol{z}, \boldsymbol{x}), d_H(\boldsymbol{z}, \boldsymbol{y})\} \geqslant 2i+3$ in contradiction to Claim 1. Similarly, there is no word $\boldsymbol{z} \in X$ such that $w_H(\boldsymbol{z}_{I_2}) = i+1$. Otherwise, $d_H(\boldsymbol{z}, \boldsymbol{x}) = d_H(\boldsymbol{z}, \boldsymbol{y}) = 2i+2$ in contradiction to Claim 2 or $\max\{d_H(\boldsymbol{z}, \boldsymbol{x}), d_H(\boldsymbol{z}, \boldsymbol{y})\} \geqslant 2i+3$ in contradiction to Claim 1.

If there is no word $\boldsymbol{z} \in X$ such that $w_H(\boldsymbol{z}_{I_2}) = i$, then there are at most $2^{2i+2} \cdot b_{i-1, n-2(i+1)}$ words in X, which is in the order of $\Theta(n^{i-1})$, while the size of X has order $\Theta(n^i)$. Thus, there are words in $\boldsymbol{z} \in X$ such that $w_H(\boldsymbol{z}_{I_2}) = i$.

Let $\boldsymbol{z} \in X$ be such that $w_H(\boldsymbol{z}_{I_2}) = i$. Then,

$$d_H(\boldsymbol{z}, \boldsymbol{x}) + d_H(\boldsymbol{z}, \boldsymbol{y}) = 4i+2.$$

From Claim 1 we have that $d_H(\boldsymbol{z}, \boldsymbol{x}), d_H(\boldsymbol{z}, \boldsymbol{y}) \leqslant 2i+2$, and thus there are only three options for the values of $d_H(\boldsymbol{z}, \boldsymbol{x})$ and $d_H(\boldsymbol{z}, \boldsymbol{y})$. In case $d_H(\boldsymbol{z}, \boldsymbol{x}) = d_H(\boldsymbol{z}, \boldsymbol{y}) = 2i+1$, we get a contradiction to Claim 3. Hence, every such a word $\boldsymbol{z}$ satisfies $d_H(\boldsymbol{z}, \boldsymbol{x}) = 2i$ and $d_H(\boldsymbol{z}, \boldsymbol{y}) = 2i+2$ or $d_H(\boldsymbol{z}, \boldsymbol{x}) = 2i+2$ and $d_H(\boldsymbol{z}, \boldsymbol{y}) = 2i$. Let X_1, X_2 be the set of words $\boldsymbol{z} \in X$ such that $w_H(\boldsymbol{z}_{I_2}) = i$ and they satisfy the first, second condition, respectively, and assume without loss of generality that $|X_1| \geqslant |X_2|$. From cardinality arguments we also conclude that $|X_1| = \Theta(n^i)$.

For every word $\boldsymbol{z} \in X_1$, from the condition $d_H(\boldsymbol{z}, \boldsymbol{x}) = 2i$ and $d_H(\boldsymbol{z}, \boldsymbol{y}) = 2i+2$, we have that $w_H(\boldsymbol{z}_{I_0'}) = w_H(\boldsymbol{z}_{I_1'}) + 1$. In particular, the weight of $\boldsymbol{z}$ on I_1 is odd. Next, we partition the set X_1 into disjoint sets according to their projection on the first $2i+2$ positions of the set I_1. That is, for all vectors $\boldsymbol{a} \in \{0,1\}^{2i+2}$ of odd weight we define

$$X_{\boldsymbol{a}} = \{\boldsymbol{z} \in X_1 \ : \ \boldsymbol{z}_{I_1} = \boldsymbol{a}\}.$$

We claim that there exists only one value of $\boldsymbol{a}$ for which $|X_{\boldsymbol{a}}| = \Theta(n^i)$. If there are no such sets then we get a contradiction to $|X_1| = \Theta(n^i)$. Now assume that there exist $\boldsymbol{a}_1, \boldsymbol{a}_2$ for which $X_{\boldsymbol{a}_1}$ and $X_{\boldsymbol{a}_2}$ have order $\Theta(n^i)$. Then, we can find two words $z_1 \in X_{\boldsymbol{a}_1}$ and $z_2 \in X_{\boldsymbol{a}_2}$ such that $d_H((z_1)_{I_2}, (z_2)_{I_2}) = 2i$ and since $\boldsymbol{a}_1 \neq \boldsymbol{a}_2$ and they have odd weight we also get $d_H((z_1)_{I_1}, (z_2)_{I_1}) \geqslant 2$. Together we conclude that $d_H(z_1, z_2) \geqslant 2i+2$ and since $d_H(\boldsymbol{y}, z_1) = d_H(\boldsymbol{y}, z_2) = 2i+2$ we get a contradiction to Claim 2.

Now, we can conclude that the number of vectors in the set X is at most

$$2|X_1| + \Theta(n^{i-1}) \leqslant 2b_{i, n-2i-2} + \Theta(n^{i-1}) < |X|,$$

for n large enough, in contradiction. ■

We summarize this result in the following corollary.

Corollary 11: For all $0 \leqslant D \leqslant 2t$ and n large enough,

$$N_t(A(D), 1) = A(2t-D).$$

Proof: From Lemma 8 we get that $N_t(A(D), 1) \geqslant A(2t-D)$ and from Lemma 9 and Lemma 10 $N_t(A(D)+1, 1) < A(2t-D)$. The conditions of Lemma 7 hold and thus $N_t(A(2t-D), 1) = A(D)$, or $N_t(A(D), 1) = A(2t-D)$. ■

We note that the result shown by Levenshtein for $d = 1$ is a special case of Corollary 11 for $D = 1$.

V. Intersections of Multiple Balls With Distance Between Their Centers

Our goal in this section is to use the results found in Section IV in order to derive bounds on $N_t(m, d)$ for arbitrary d. That is, here we require to have a minimum distance d between the centers of the balls.

We first start with the case where the number of balls m is constant. In this case, the order of the largest intersection size remains the same as the one for two balls. This property is proved in the next theorem.

Theorem 12: For any fixed m, t, d and n large enough, such that $m \geqslant 3$ and $t \geqslant \lceil \frac{d}{2} \rceil$, $N_t(m, d) = \Theta(n^{t-\lceil \frac{d}{2} \rceil})$.

Proof: Since $N_t(m, d) \leqslant N_t(2, d)$ and $N_t(2, d) = \Theta(n^{t-\lceil \frac{d}{2} \rceil})$, then $N_t(m, d)$ is at most $O(n^{t-\lceil \frac{d}{2} \rceil})$.

To show the other direction of this equality, we show an example of a set X such that the cardinality of $S_t(X)$ is $O(n^{t-\lceil \frac{d}{2} \rceil})$. For $1 \leqslant i \leqslant m$, let $i_0 = (i-1)\lceil \frac{d}{2} \rceil + 1$ and $i_1 = i\lceil \frac{d}{2} \rceil$, and $\boldsymbol{x}_i = \boldsymbol{e}_{i_0}^{i_1}$. Then, for all $i \neq j$, $d_H(\boldsymbol{x}_i, \boldsymbol{x}_j) = 2\lceil \frac{d}{2} \rceil \geqslant 2d$. For any vector $\boldsymbol{y}$ of weight at most $t - \lceil \frac{d}{2} \rceil$, such that its first $m\lceil\frac{d}{2}\rceil$ bits are zero, we have that $\boldsymbol{y} \in S_t(X)$. Since there are $\sum_{\ell=0}^{t-\lceil\frac{d}{2}\rceil}\binom{n-m\lceil\frac{d}{2}\rceil}{\ell}$ such vectors we get that for n large enough, $N_t(m,d)$ is at least $O(n^{t-\lceil\frac{d}{2}\rceil})$. Together we conclude that $N_t(m,d) = \Theta(n^{t-\lceil\frac{d}{2}\rceil})$. ■

Next we move to the case where m is not a constant. We will use the result we derived in the previous section on $N_t(m,1)$ in order to show a lower bound in this case on $N_t(m,d)$ for some values of m. First, we state a useful Theorem from [1].

Theorem 13 [1]: Let $\mathcal{C} \subseteq \{0,1\}^n$ be a code with distances from $\mathcal{D} = \{d_1,\dots,d_s\} \subseteq \{1,\dots,n\}$. Further let $L_{\mathcal{D}}(B)$ be a maximal code in $B \subseteq \{0,1\}^n$ with distances from $\mathcal{D}$. Then, the following inequality holds

$$\frac{|\mathcal{C}|}{2^n} \leqslant \frac{|L_{\mathcal{D}}(B)|}{|B|}.$$

For all $1 \leqslant d \leqslant n$, we denote by $\rho_{d,n}$ the maximal ratio of the size of a code $\mathcal{C}$ with minimum distance d and length n, that is,

$$\rho_{d,n} = \max_{\mathcal{C}, d_{\min}(\mathcal{C}) \geqslant d}\left\{\frac{|\mathcal{C}|}{2^n}\right\}.$$

Theorem 13 will serve us to prove the next lemma.

Lemma 14: Let $B \subseteq \{0,1\}^n$ be a set and let $L(B)$ be a maximal code in B with minimum distance d, then

$$\rho_{d,n} \cdot |B| \leqslant |L(B)|.$$

Proof: We take the code $\mathcal{C}$ in Theorem 13 to be a code with minimum distance d and maximal ratio $\rho_{d,n}$, which has at least the same set of distance of B. Then, we get

$$\rho_{d,n} = \frac{|\mathcal{C}|}{2^n} \leqslant \frac{|L(B)|}{|B|}.$$

■

Now, we are ready to derive a connection between the values of $N_t(m,d)$ and $N_t(m,1)$.

Theorem 15: For all m, t, d and n large enough,

$$N_t(\lceil\rho_{d,n}m\rceil, d) \geqslant N_t(m,1).$$

Proof: Assume that X is a set of m words such that $S_t(X)$ has size $N_t(m,1)$. Using Lemma 14, we let $L(X)$ be a code in X of minimum distance d and size $\lceil\rho_{d,n}m\rceil$. Then, $S_t(X) \subseteq S_t(L(X))$ and thus $N_t(\lceil\rho_{d,n}m\rceil, d) \geqslant N_t(m,1)$. ■

The result of Theorem 15 claims that in order to study the value of $N_t(m,d)$ one needs to consider the case of $d=1$ and then derive a corresponding lower bound on $N_t(m,d)$.

VI. Sequences Reconstruction Decoders

The main goal in [22] was to find the necessary and sufficient number of channels in order to have a successful decoder for a code with minimum distance d, while t, the maximum number of errors in every channel, is greater than $\lfloor(d-1)/2\rfloor$. This number was studied for different error models in [16]–[18] and [21]–[25], however the only decoder constructions, which we are aware of, were given in [3] and [14] for channels with insertion and deletions, and in [8] for deletions only, all for the probabilistic model. In this section, we show how to construct decoders for substitution errors, where the decoder has to output the transmitted word (and not a list of words).

The case $d = 1$ was solved in [22] 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 3, 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 $\lceil(d-1)/2\rceil$ 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 $\boldsymbol{c}$. According to (5), there are $N_2(2,3)+1 = 7$ channels, and assume that the channel outputs are the following seven words:

$$\begin{aligned}
\boldsymbol{y}_1 &= (1,0,1,0,0,0,0)\\
\boldsymbol{y}_2 &= (1,0,0,1,0,0,0)\\
\boldsymbol{y}_3 &= (1,0,0,0,1,0,0)\\
\boldsymbol{y}_4 &= (1,1,0,0,0,0,0)\\
\boldsymbol{y}_5 &= (0,1,1,0,0,0,0)\\
\boldsymbol{y}_6 &= (0,1,0,1,0,0,0)\\
\boldsymbol{y}_7 &= (0,1,0,0,1,0,0)
\end{aligned}$$

Then, the output of the majority decoder on these sever 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.

In general, according to Corollary 3, 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.

For the rest of this section, we assume that the transmitted word $\boldsymbol{c}$ belongs to a code $\mathcal{C}$ 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 $N = N_t(2,d)+1$. We also assume that n is relatively large enough with respect to d and t. The N channel outputs are denoted by $\boldsymbol{y}_1,\dots,\boldsymbol{y}_N$ and are assumed to be different. Furthermore, the code $\mathcal{C}$ has a decoder $\mathcal{D}_{\mathcal{C}}$, 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 we can always 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. We prove that in the next lemma.

Lemma 16: If $\boldsymbol{c}$ *is the transmitted word, then for any* $\widehat{\boldsymbol{c}} \in \mathcal{C}$, $\widehat{\boldsymbol{c}} = \boldsymbol{c}$ *if and only if*

$$\max_{1\leqslant i\leqslant N}\{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t.$$

Proof: If $\widehat{\boldsymbol{c}} = \boldsymbol{c}$ then every channel suffers at most t errors and thus $\max_{1\leqslant i\leqslant N}\{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t$. In case $\widehat{\boldsymbol{c}} \neq \boldsymbol{c}$, let us assume to the contrary that $\max_{1\leqslant i\leqslant N}\{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t$. Then the set $S_t(\{\widehat{\boldsymbol{c}}, \boldsymbol{c}\})$ contains at least $N = N_t(2, d) + 1$ words in contradiction to the definition of $N_t(2, d)$. ■

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 $\mathcal{C}$ and can be verified to be the correct transmitted word according to Lemma 16. 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 verifying step in Lemma 16 has complexity $\Theta\left(n^{t-(d+1)/2+1}\right)$. Hence, all together the complexity of this decoder is $\Theta\left(n^{2t-d+2} \cdot D(n)\right)$, where $D(n)$ is the decoding complexity of the decoder $\mathcal{D}_{\mathcal{C}}$.

We will show how to modify and improve the complexity of this algorithm. Assume for example that $t = \frac{d-1}{2} + 1$. Then, there are two channel outputs, say $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, that are different in at least one bit location. If we flip this bit in both $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, 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 $\mathcal{D}_{\mathcal{C}}$. We show how to generalize this idea for arbitrary t. We let $\rho = t - \frac{d-1}{2}$, which corresponds to the number of additional errors on top of the error-correction capability. First, we prove the following Lemma.

Lemma 17: There exist two channel outputs $\boldsymbol{y}_i, \boldsymbol{y}_j$ *such that* $d_H(\boldsymbol{y}_i, \boldsymbol{y}_j) \geqslant 2\rho - 1$.

Proof: Assume to the contrary that there are no such words. Then, the words $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$ form an anticode of diameter $2\rho - 2$. According to [15], the maximum size of such an anticode is $b_{\rho-1,n} = \sum_{i=0}^{\rho-1}\binom{n}{i}$, while according to (5) the value of N satisfies

$$\begin{aligned} N > N_t(2, d) &= \sum_{i=0}^{t-\frac{d+1}{2}}\binom{n-d}{i}\sum_{k=d-t+i}^{t-i}\binom{d}{k} \\ &= \sum_{i=0}^{\rho-1}\binom{n-d}{i}\sum_{k=d-t+i}^{t-i}\binom{d}{k} > \sum_{i=0}^{\rho-1}\binom{n}{i} \\ &= b_{\rho-1,n}. \end{aligned}$$

■

We are now ready to present the algorithm.

Algorithm 18: The input to the decoder is the set of all N *channel outputs* $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ *and it returns an estimation* $\widehat{\boldsymbol{c}}$ *on* $\boldsymbol{c}$.

Step 1. Find two words $\boldsymbol{y}_i, \boldsymbol{y}_j$ *such* $d_H(\boldsymbol{y}_i, \boldsymbol{y}_j) \geqslant 2\rho - 1$, *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.*

Step 2. For all vectors $\boldsymbol{e}$ *of weight* ρ *on the* $2\rho - 1$ *indices of the set* I,

a) $\mathcal{D}(\boldsymbol{y}_i + \boldsymbol{e}) = \widehat{\boldsymbol{c}}_1$, $\mathcal{D}(\boldsymbol{y}_j + \boldsymbol{e}) = \widehat{\boldsymbol{c}}_2$.

b) If $\max_{1\leqslant i\leqslant N}\{d_H(\widehat{\boldsymbol{c}}_1, \boldsymbol{y}_i)\} \leqslant t$, $\widehat{\boldsymbol{c}} = \widehat{\boldsymbol{c}}_1$.

c) If $\max_{1\leqslant i\leqslant N}\{d_H(\widehat{\boldsymbol{c}}_2, \boldsymbol{y}_i)\} \leqslant t$, $\widehat{\boldsymbol{c}} = \widehat{\boldsymbol{c}}_2$.

Theorem 19: The output of Algorithm 18 satisfies $\widehat{\boldsymbol{c}} = \boldsymbol{c}$.

Proof: The success of Step 1 is guaranteed according to Lemma 17. For every index i_j, $1 \leqslant j \leqslant 2\rho - 1$, exactly one of the channel outputs $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$ has an error. Therefore, either $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$ has at least ρ errors on these indices. Without loss of generality assume it is $\boldsymbol{y}_1$ and let $I \subseteq \{i_1, \ldots, i_{2\rho-1}\}$ be a subset of its error locations, where $|I| = \rho$. In Step 2 we exhaustively search over all error vectors $\boldsymbol{e}$ of weight ρ on these $2\rho - 1$ indices. For every error vector $\boldsymbol{e}$ let $I_{\boldsymbol{e}} = \{i : e_i = 1\}$. Therefore, there exists an error vector $\boldsymbol{e}_1$ such that its the set of indices with value one is covered by the set I, i.e. $I_{\boldsymbol{e}_1} \subseteq I$. Hence, $d_H(\boldsymbol{c}, \boldsymbol{y}_1 + \boldsymbol{e}_1) \leqslant \frac{d-1}{2}$, so the decoder in Step 2.b succeeds. Hence the algorithm succeeds and $\widehat{\boldsymbol{c}} = \boldsymbol{c}$. ■

A naive implementation of Step 1 results with complexity $\Theta\left(N^2 n\right) = \Theta\left(n^{2t-d}\right)$. The number of vectors $\boldsymbol{e}$ in Step 2 is constant since ρ is constant as well, and thus the complexity of this step is $\Theta(Nn)$, since we assume that the complexity of the decoder $\mathcal{D}_{\mathcal{C}}$ is less than the order of channel outputs N. Together, we deduce that the complexity order of Algorithm 18 is $\Theta\left(n^{2t-d}\right)$, which improves upon the decoding complexity of the naive algorithm.

We see that the complexity of Algorithm 18 is significantly better than the naive approach. However, the larger the value of ρ is, the larger the algorithm's complexity is. Let us finish this section by showing an example of how to improve the complexity of Algorithm 18 for $d = 3$ and arbitrary $t \geqslant 3$.

According to (6), the number of channels in this case is

$$N = N_t(2, 3) + 1 = 2\sum_{i=0}^{t-3}\binom{n-3}{i} + 6\sum_{i=0}^{t-2}\binom{n-3}{i} + 1.$$

Given the N channel outputs $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$, a word $\boldsymbol{y} \in \{0, 1, ?\}^n$ will be generated as follows. For all $1 \leqslant i \leqslant n$, let $n_{i,0}, n_{i,1}$ be the number of times the i-th bit is zero, one, respectively. That is,

$$\begin{aligned} n_{i,0} &= \{\ell \ : \ 1 \leqslant \ell \leqslant N, y_{\ell,i} = 0\}, \\ n_{i,1} &= \{\ell \ : \ 1 \leqslant \ell \leqslant N, y_{\ell,i} = 1\}. \end{aligned}$$

Note that $n_{i,0}$ and $n_{i,1}$ can also be seen as the type of the vector $(y_{1,i}, y_{2,i}, \ldots, y_{N,i})$. By saying that we decode the symbol y_i according to the majority algorithm with threshold τ, we mean that if $|n_{i,0} - n_{i,1}| \leqslant \tau$ then $y_i =?$. Otherwise, if $n_{i,0} > n_{i,1}$ then $y_i = 0$ and if $n_{i,1} > n_{i,0}$ then $y_i = 1$. Assume that y_i is decoded according to the majority algorithm with threshold τ, and the i-th bit is in error e_i times. Then, $y_i =?$ if $\frac{N-\tau}{2} \leqslant e_i \leqslant \frac{N+\tau}{2}$, and y_i is in error if $e_i > \frac{\tau+N}{2}$. Let $Y = \sum_{i=0}^{t-2}\binom{n-2}{i}$ and we set the two threshold values τ_1, τ_2 to be $\tau_1 = Y$, $\tau_2 = 3Y$. We are now ready to present the decoder algorithm for $d = 3$. We assume that $\mathcal{D}_H$ is a decoder algorithm for a Hamming code, which the transmitted codeword $\boldsymbol{c}$ belongs to. This decoder can correct a single error or two erasures.

Algorithm 20: The input to the decoder is the set of all N *words* $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ *and it returns an estimation* $\widehat{\boldsymbol{c}}$ *on* $\boldsymbol{c}$.

Step 1. For $1 \leqslant i \leqslant n$, *let*

$$n_{i,0} = \{\ell \ :\ 1 \leqslant \ell \leqslant N, y_{\ell,i} = 0\},$$
$$n_{i,1} = \{\ell \ :\ 1 \leqslant \ell \leqslant N, y_{\ell,i} = 1\}.$$

Step 2. Set the word $\boldsymbol{y} \in \{0,1,?\}^n$ *as follows*

a) *If there exists* j *such that* $|n_{j,0} - n_{j,1}| \leqslant \tau_1$, *then decode all symbols* y_i *according to the majority algorithm with threshold* τ_2.

b) *Otherwise, for all* i $|n_{i,0} - n_{i,1}| > \tau_1$, *then decode all symbols* y_i *according to the majority algorithm with threshold* τ_1.

Step 3. If $\boldsymbol{y}$ *has no ? then decode the word* $\boldsymbol{y}$ *by the Hamming decoder, that is* $\hat{\boldsymbol{c}} = \mathcal{D}_H(\boldsymbol{y})$. *Otherwise, for any possible codeword which matches on the ?'s in* $\boldsymbol{y}$, *check if it satisfies the condition in Lemma 16 and output the correct one.*

The complexity of Algorithm 20 is $O(Nn) = O(n^{t-1})$, and the next theorem proves its correctness.

Theorem 21: The word $\boldsymbol{y}$ *in Algorithm 20 has either one error or at most* $8t$ *erasures and thus the output of the algorithm* $\hat{\boldsymbol{c}}$ *is* $\boldsymbol{c}$.

Proof: For all $1 \leqslant i \leqslant n$, let $0 \leqslant e_i \leqslant N$ be the number of times the i-th bit is in error. Assume first that there exists j such that $|n_{j,0} - n_{j,1}| \leqslant \tau_1$, so all symbols y_i are decoded according to the majority algorithm with threshold τ_2. We first notice that y_j =?. First we show that there does not exist a different index i such that the i-th symbol y_i is in error. Assume in the contrary that y_i is erroneous, then for the j-th and i-th symbols we get that $e_j \geqslant \frac{N-\tau_1}{2}$ and $e_i > \frac{N+\tau_2}{2}$. Since all channel outputs are different, the number of channel outputs where both the j-th and the i-th bit are in error is at most $Y = \sum_{i=0}^{t-2}\binom{n-2}{i}$. Now we get that the number of channel outputs has to be greater than

$$\frac{N-\tau_1}{2} + \frac{N+\tau_2}{2} - Y = N + \frac{\tau_2-\tau_1}{2} - Y = N,$$

which is a contradiction. Therefore, the vector $\boldsymbol{y}$ cannot have an error.

Now we show that in this case the number of erasures is at most $8t$. The total number of errors in the N channels is no greater than tN. If the i-th symbol of $\boldsymbol{y}$ is decoded with a ?, then there are at least $\frac{N-\tau_2}{2}$ errors of this symbol. Hence, the total number of erasures in $\boldsymbol{y}$ is at most

$$\frac{tN}{\frac{N-\tau_2}{2}} = 2t\frac{N}{N-\tau_2} \leqslant 8t,$$

since for n large enough $\tau_2 < 3N/4$.

Next, we assume that for all i, $|n_{i,0} - n_{i,1}| > \tau_1$, so every symbol y_i is decoded according to the majority algorithm with threshold τ_1 and note that there are no erasures in this case. We show that there exists at most one error. Assume to the contrary that there are two errors, then the number of channels has to be greater than

$$\frac{N+\tau_1}{2} + \frac{N+\tau_1}{2} - Y = N,$$

which is again a contradiction. This proves the correctness of the last step of the algorithm and that the decoded word $\hat{\boldsymbol{c}}$ equals $\boldsymbol{c}$. ■

While Algorithm 20 provides an efficient decoder for codes with minimum Hamming distance three, it is not clear how to extend it in order to support larger distances. We believe that similar approach of majority decoders with thresholds can work, but this part is left for future work.

VII. Conclusion

This paper proposed a model of an associative memory: Two words are associated if their Hamming distance is no greater than some prescribed value t. Our main goal was to study the maximum size of the associative memory output as a function of the number of input words and their minimum distance. We observed that this problem is a generalization of the sequences reconstruction problem that was proposed by Levenshtein. Finally, we presented decoding algorithms for the sequences reconstruction problem.

Appendix A
Proof of Corollary 3

In this section we give the proof of Corollary 3

Proof: According to (6) 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,

$$\begin{aligned}
&N_t(2,d)\\
&= \sum_{k=0}^{\min\{d,t\}}\left(\binom{d-1}{k-1}+\binom{d-1}{k}\right)\sum_{i=0}^{t-\max\{k,d-k\}}\binom{n-d}{i}\\
&= \sum_{k=0}^{\min\{d,t\}-1}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k+1,d-1-k\}}\binom{n-d}{i}\\
&\quad+\sum_{k=0}^{\min\{d,t\}-1}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k,d-k\}}\binom{n-d}{i}+\binom{d-1}{\min\{d,t\}}\\
&= \sum_{k=0}^{\min\{d,t\}-1}\binom{d-1}{k}\left(\sum_{i=0}^{t-\max\{k+1,d-1-k\}}\binom{n-d}{i}\right.\\
&\quad\left.+\sum_{i=0}^{t-\max\{k,d-k\}}\binom{n-d}{i}\right)+\binom{d-1}{\min\{d,t\}}\\
&= \sum_{k=0}^{\frac{d}{2}-1}\binom{d-1}{k}\left(\sum_{i=0}^{t-(d-1-k)}\binom{n-d}{i}+\sum_{i=0}^{t-(d-k)}\binom{n-d}{i}\right)\\
&\quad+\sum_{k=\frac{d}{2}}^{\min\{d,t\}-1}\binom{d-1}{k}\left(\sum_{i=0}^{t-(k+1)}\binom{n-d}{i}+\sum_{i=0}^{t-k}\binom{n-d}{i}\right)\\
&\quad+\binom{d-1}{\min\{d,t\}}
\end{aligned}$$

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

$$\begin{aligned}N_t(2,d)&=\sum_{k=0}^{\frac{d}{2}-1}\binom{d-1}{k}\sum_{i=0}^{t-(d-1-k)}\binom{n-(d-1)}{i}\\&\quad+\sum_{k=\frac{d}{2}}^{\min\{d,t\}-1}\binom{d-1}{k}\sum_{i=0}^{t-k}\binom{n-(d-1)}{i}+\binom{d-1}{\min\{d,t\}}\\&=\sum_{k=0}^{\min\{d,t\}-1}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k,d-1-k\}}\binom{n-(d-1)}{i}\\&\quad+\binom{d-1}{\min\{d,t\}}.\end{aligned}$$

If $t\geqslant d$ then $\min\{d,t\}-1=\min\{d-1,t\}$ and $\binom{d-1}{\min\{d,t\}}=0$ so we get

$$\begin{aligned}N_d(2,d)&=\sum_{k=0}^{\min\{d-1,t\}}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k,d-1-k\}}\binom{n-(d-1)}{i}\\&=N_d(2,d-1).\end{aligned}$$

Otherwise, $t\leqslant d-1$, so $\min\{d,t\}=\min\{d-1,t\}=t$ and we get

$$\begin{aligned}N_t(2,d)&=\sum_{k=0}^{\min\{d-1,t\}-1}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k,d-1-k\}}\binom{n-(d-1)}{i}\\&\quad+\binom{d-1}{\min\{d-1,t\}}\\&=\sum_{k=0}^{\min\{d-1,t\}}\binom{d-1}{k}\sum_{i=0}^{t-\max\{k,d-1-k\}}\binom{n-(d-1)}{i}\\&=N_t(2,d-1).\end{aligned}$$

■

Appendix B
Proof of Lemma 5

In this section we give the proof of Lemma 5.

Proof: As in the proof of Lemma 4, we find the cardinality of the set

$$S_t(\{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\})=\{\boldsymbol{u}\in\{0,1\}^n: d_H(\boldsymbol{u},\boldsymbol{x}),d_H(\boldsymbol{u},\boldsymbol{y}),d_H(\boldsymbol{u},\boldsymbol{z})\leqslant t\}.$$

We define the following eight sets:

$$S_{b_1,b_2,b_3}=\{i: y_i=x_i+b_1, z_i=x_i+b_2, u_i=x_i+b_3\},$$

for $b_1,b_2,b_3\in\{0,1\}$.

Since $d_H(\boldsymbol{x},\boldsymbol{y})=d_H(\boldsymbol{x},\boldsymbol{z})=d$ and $d_H(\boldsymbol{y},\boldsymbol{z})=d+1$ we get the following equalities

$$\begin{aligned}|S_{1,0,0}|+|S_{1,0,1}|+|S_{1,1,0}|+|S_{1,1,1}|&=d,\\|S_{0,1,0}|+|S_{0,1,1}|+|S_{1,1,0}|+|S_{1,1,1}|&=d,\\|S_{0,1,0}|+|S_{0,1,1}|+|S_{1,0,0}|+|S_{1,0,1}|&=d+1,\end{aligned}$$

so we derive that

$$\begin{aligned}|S_{0,1,0}|+|S_{0,1,1}|=|S_{1,0,0}|+|S_{1,0,1}|&=\frac{d+1}{2},\\|S_{1,1,0}|+|S_{1,1,1}|&=\frac{d-1}{2},\\|S_{0,0,0}|+|S_{0,0,1}|&=n-\frac{3d+1}{2}.\end{aligned}$$

Let us define

$$|S_{0,0,1}|=i_1,\quad |S_{0,1,0}|=i_2,\quad |S_{1,0,0}|=i_3,\quad |S_{1,1,0}|=i_4.$$

We also have the following inequalities:

$$\begin{aligned}d_H(\boldsymbol{u},\boldsymbol{x})&=i_1+\frac{d+1}{2}-i_2+\frac{d+1}{2}-i_3+\frac{d-1}{2}-i_4\leqslant t,\\d_H(\boldsymbol{u},\boldsymbol{y})&=i_1+\frac{d+1}{2}-i_2+i_3+i_4\leqslant t,\\d_H(\boldsymbol{u},\boldsymbol{z})&=i_1+i_2+\frac{d+1}{2}-i_3+i_4\leqslant t,\end{aligned}$$

where $0\leqslant i_1\leqslant n-\frac{3d+1}{2}, 0\leqslant i_2,i_3\leqslant\frac{d+1}{2}, 0\leqslant i_4\leqslant\frac{d-1}{2}$. Finally, the size of the intersection is the number of solutions to the above inequalities including the choices of the coordinates we can choose for every choice of i_1,i_2,i_3,i_4. That is,

$$\sum_{i_1,i_2,i_3,i_4}\binom{n-\frac{3d+1}{2}}{i_1}\binom{\frac{d+1}{2}}{i_2}\binom{\frac{d+1}{2}}{i_3}\binom{\frac{d-1}{2}}{i_4},$$

where i_1,i_2,i_3,i_4 satisfy the following constraints.

1) $0\leqslant i_1\leqslant t-\frac{d+1}{2}$,
2) $i_1+\frac{d-1}{2}-t\leqslant i_4\leqslant t-\frac{d+1}{2}-i_1$,
3) $d+1-t+i_1\leqslant i_3\leqslant t-(i_1+i_4)$,
4) $\max\{i_1-i_3-i_4+\frac{3d+1}{2}-t, i_1+i_3+i_4+\frac{d+1}{2}-t\}\leqslant i_2\leqslant t-(i_1+i_4+\frac{d+1}{2}-i_3)$.

■

Acknowledgement

The authors thank Tuvi Etzion for helpful discussions on anticodes. They also thank the three anonymous reviewers and the associate editor Joerg Kliewer for their valuable comments and suggestions, which have contributed for the clarity of the paper and its presentation.

References

[1] R. Ahlswede, H. K. Aydinian, and L. H. Khachatrian, "On perfect codes and related concepts," *Designs, Codes Cryptogr.*, vol. 22, no. 3, pp. 221–237, 2001.

[2] R. Ahlswede and L. H. Khachatrian, "The diametric theorem in hamming spaces—Optimal anticodes," *Adv. Appl. Math.*, vol. 20, no. 4, pp. 429–449, 1998.

[3] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proc. 15th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2004, pp. 903–911.

[4] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 8011–8020, Dec. 2011.
[5] Y. Cassuto and S. Litsyn, "Symbol-pair codes: Algebraic constructions and asymptotic bounds," in *Proc. Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul./Aug. 2011, pp. 2348–2352.
[6] R. Gabrys and E. Yaakobi, "Sequence reconstruction over the deletion channel," in *Proc. Int. Symp. Inf. Theory*, Jul. 2016, pp. 1596–1600.
[7] V. Gripon and M. Rabbat, "Maximum likelihood associative memories," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2013, pp. 1–5.
[8] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," in *Proc. 19th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2008, pp. 389–398.
[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.
[10] T. Jiang and A. Vardy, "Asymptotic improvement of the Gilbert-Varshamov bound on the size of binary codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1655–1664, Aug. 2004.
[11] V. Junnila and T. Laihonen, "Codes for information retrieval with small uncertainty," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 976–985, Feb. 2014.
[12] V. Junnila and T. Laihonen, "Information retrieval with unambiguous output," *Inf. Comput.*, vol. 242, pp. 354–368, Jun. 2015.
[13] V. Junnila and T. Laihonen, "Information retrieval with varying number of input clues," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 625–638, Feb. 2016.
[14] S. Kannan and A. McGregor, "More on reconstructing strings from random traces: Insertions and deletions," in *Proc. IEEE Int. Sym. Inf. Theory*, Sep. 2005, pp. 297–301.
[15] D. J. Kleitman, "On a combinatorial conjecture of Erdös," *J. Combinat. Theory*, vol. 1, no. 2, pp. 209–214, 1966.
[16] E. Konstantinova, "On reconstruction of signed permutations distorted by reversal errors," *Discrete Math.*, vol. 308, pp. 974–984, Mar. 2008.
[17] E. Konstantinova, "Reconstruction of permutations distorted by single reversal errors," *Discrete Appl. Math.*, vol. 155, no. 18, pp. 2426–2434, 2007.
[18] E. Konstantinova, V. I. Levenshtein, and J. Siemons. (Feb. 2007). "Reconstruction of permutations distorted by single transposition errors." [Online]. Available: https://arxiv.org/abs/math/0702191
[19] T. Laihonen, "Information retrieval and the average number of input clues," *Adv. Math. Commun.*, vol. 11, no. 1, pp. 203–223, Feb. 2017.
[20] T. Laihonen and T. Lehtilä, "Improved codes for list decoding in the Levenshtein's channel and information retrieval," in *Proc. Int. Symp. Inf. Theory*, Aachen, Germany, Jun. 2017, pp. 2643–2647.
[21] V. I. Levenshtein, "Reconstructing objects from a minimal number of distorted patterns," (in Russian), *Dokl. Acad. Nauk*, vol. 354, pp. 593–596, 1997.
[22] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.
[23] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *J. Combinat. Theory A*, vol. 93, no. 2, pp. 310–332, 2001.
[24] V. I. 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, 2008.
[25] V. I. Levenshtein and J. Siemons, "Error graphs and the reconstruction of elements in groups," *J. Combinat. Theory, A*, vol. 116, no. 4, pp. 795–815, 2009.
[26] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans. Inf. Theory*, vol. 33, no. 4, pp. 461–482, Jul. 1987.
[27] S. Motahari, G. Bresler, and D. Tse. (2012). "Information theory of DNA shotgun sequencing." [Online]. Available: https://arxiv.org/abs/1203.6233v4
[28] S. Motahari, G. Bresler, and D. Tse, "Information theory for DNA sequencing: Part I: A basic model," in *Proc. Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2741–2745.
[29] F. Sala, R. Gabrys, C. Schoeny, and L. Dolecek, "Three novel combinatorial theorems for the insertion/deletion channel," in *Proc. Int. Symp. Inf. Theory*, Hong Kong, Jun. 2015, pp. 2702–2706.
[30] K. Viswanathan and R. Swaminathan, "Improved string reconstruction over insertion-deletion channels," in *Proc. 19th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2008, pp. 399–408.
[31] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 106–110.
[32] E. Yaakobi, J. Bruck, and P. H. Siegel, "Decoding of cyclic codes over symbol-pair read channels," in *Proc. Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2901–2905.
[33] E. Yaakobi, M. Schwartz, M. Langberg, and J. Bruck, "Sequence reconstruction for Grassmann graphs and permutations," in *Proc. Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 874–878.

Eitan Yaakobi (S'07–M'12–SM'17) is an Assistant Professor at the Computer Science Department at the Technion — Israel Institute of Technology. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the department of Electrical Engineering at the California Institute of Technology. His research interests include information and coding theory with applications to non-volatile memories, associative memories, data storage and retrieval, and voting theory. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010–2011.

Jehoshua Bruck (S'86–M'89–SM'93–F'01) is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and systems, coding for memories, and the analysis and design of biological information systems. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1989.

His industrial and entrepreneurial experiences include working with IBM Research where he participated in the design and implementation of the first IBM parallel computer; cofounding and serving as Chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that created the first virtualization solution for Network Attached Storage; as well as cofounding and serving as Chairman of XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system.

He is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award, and the IBM Outstanding Technical Achievement Award.