Asymptotic Analysis of Data Deduplication with a Constant Number of Substitutions

Hao Lou, Farzad Farnoud (Hassanzadeh)
Electrical and Computer Engineering, University of Virginia, VA, USA. Email: {haolou,farzad}@virginia.edu

Abstract—**Data deduplication has gained attention in large-scale storage systems due to the explosive growth in digital data. Recently, the information-theoretic aspects of conventional deduplication algorithms have been studied and novel algorithms with better performance have been proposed. In this paper, we study the performances of variable-length deduplication and multi-chunk deduplication algorithms from the point of view of information theory. We consider a source model in which source strings are composed of repeated blocks with each data block containing a constant number of substitution edits. We show that over the proposed source model, the variable-length deduplication algorithm can achieve asymptotically arbitrarily large compression ratio and the multi-chunk deduplication algorithm is order optimal under mild conditions.**

I. Introduction

The task of reducing data storage costs is gaining increasing attention due to the explosive growth of the amount of digital data, especially redundant data [1]–[3]. Data deduplication was proposed to detect and remove repeats in the input data streams or files to save storage space. Compared with traditional data compression approaches, data deduplication is computationally more efficient, especially when dealing with large-scale data. It has been widely used in mass data storage systems, e.g., LBFS [4] and Venti [5]. A typical data deduplication system uses a chunking scheme to parse data streams into multiple data "chunks". Each chunk is put into the dictionary at the first occurrence, and the duplicates are replaced by pointers to the dictionary. In this paper, we aim to study the performance of two data deduplication algorithms from an information-theoretic point of view when repeated data segments are not necessarily exact copies.

The two data deduplication algorithms studied in this paper, variable-length deduplication (VLD) and multi-chunk deduplication (MCD), were proposed in [6]. The detailed descriptions of VLD and MCD are given in Section IV. Both algorithms use content-defined chunking (CDC) schemes. CDC uses a sliding-window technique on the content of data streams and determines a chunk breakpoint every time this sliding window meets some predefined conditions. The chunk lengths of CDC are therefore not fixed, and may, for example, have an exponential distribution [7]. CDC is widely used in practical deduplication systems, e.g., [3], [4], [8]–[15].

In CDC, a substantial fraction of the chunks can be of extremely small or large sizes. With large chunk sizes, duplicates in data tend to remain undetected, which leads to ineffective deduplication. On the other hand, chunks of small sizes introduce excessive metadata since the amount of metadata is proportional to the number of chunks. MCD can be regarded as a modification of VLD to address the problem caused by small chunk sizes.

Niesen presented an information-theoretic analysis of VLD and MCD in [6]. The performances of VLD and MCD were studied over a source model which produces data streams that are composed of blocks with each block being an exact copy of one of the source symbols, where the source symbols are pre-selected strings. It is often the case, however, that the copies of a block of data that is repeated many times are approximate, rather than exact. This may occur, for example, due to edits to the data, or in the case of genomic data[1], due to mutations. Thus, in this paper, we consider the problem of deduplication when the repeats are approximate. In particular, we allow blocks to be altered by random substitution edits. We study the expected length of the compressed strings produced by VLD and MCD for the source model that contains substitution edits. We show that if the blocks in data streams are mostly distinct after being altered by random substitution edits, then the expected length of the compressed strings produced by VLD is of greater order than the entropy. Meanwhile, compared with the length of the uncompressed strings, VLD can still achieve asymptotically arbitrarily large compression ratio. Further, under mild conditions, MCD is shown to be order optimal.

Data deduplication has been well studied from a practical perspective; see [17] for a comprehensive survey. However, its theoretical analysis from an information-theoretic point of view is limited, despite the suitability of such an approach. The first such analysis was presented by Niesen [6], as described above. The authors of the current paper studied VLD, as well as algorithms with fixed-length chunking schemes,[2] over a probabilistic source model in which edits are modeled as random substitution, where each bit may be flipped independently of others with a given probability [18]. While [18] extended the information-theoretic analysis of data deduplication to approximate repeats, the studied model has entropy linear in the length of the uncompressed string and the gain in compression is at best a constant factor. This makes compression less challenging and the distinction between the performance of compression methods less clear. In this paper, we assume each source block only contains a constant number of substitutions (randomly distributed) instead of iid bit flips, leading to the entropy being of smaller order than the length

[1] Repeats are common in genomic data. For example, a majority of the human genome consists of interspersed and tandem repeated sequences [16].

[2] Compared with CDC, fixed-length chunking partitions data stream into chunks of the same length, which can be chosen by taking the statistical properties of the source into account.

of the uncompressed string and thus high compression ratio can be achieved. Importantly, the current work also studies the MCD algorithm proposed by [6], which has not studied before in models with edits. The paper [19] also analyzed deduplication from an information-theoretic point of view but with a different source model and algorithm. The problem of deduplication under edits was considered also in [20], which focused on performing deduplication on two files. Many works are devoted to dealing with the problems arising from chunks that are too small or too large [13], [21], [22], but they did not provide information-theoretic analysis. In particular, [22] used a similar scheme as that in MCD that jointly encoding small chunks to avoid metadata overhead.

The rest of the paper is organized as follows. Notation and preliminaries are given in the next section. In Section III, we describe the studied information source model and bound its entropy. In Section IV, we formally describe the deduplication algorithms VLD and MCD that are analyzed in the rest of the paper. Bounds on the performance of algorithms are derived in Section V. Due to space limitation, some of the proofs are omitted or sketched.

II. Preliminary

In this paper, all logarithms are to base 2. We consider the binary alphabet $\{0,1\}$, denoted Σ. For a positive integer m, Σ^m denotes the set of all strings of length m over Σ. For strings $\boldsymbol{u}, \boldsymbol{v}$, the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted $\boldsymbol{uv}$. The length of $\boldsymbol{u}$ is denoted $|\boldsymbol{u}|$. A j-(sub)string is a (sub)string of length j. The cardinality of a set S is also denoted $|S|$. For an event $\mathcal{E}$, we define the indicator variable $\mathbb{1}_{\mathcal{E}}$ to be 1 if $\mathcal{E}$ is true, and 0 otherwise. A string is k-runlength-limited (RLL) if it does not contain k consecutive 0s, i.e., runs of 0s are all of lengths less than k.

III. Source model

The source model studied in this paper extends the one proposed in [6] by allowing random substitution edits. Let the source alphabet be denoted $\mathcal{X}$, with $|\mathcal{X}| = A$. The source alphabet $\mathcal{X}$ contains A strings over Σ, denoted $\mathsf{X}_1, \ldots, \mathsf{X}_A$. Fix a probability distribution $\mathbb{P}_l$ over positive integers with mean L. The A source symbols $\mathsf{X}_1, \ldots, \mathsf{X}_A$ are generated iid as follows. For each $1 \leq a \leq A$, X_a is chosen from Σ^{L_a} uniformly at random, where L_a is a positive integer drawn independently of other quantities from the distribution $\mathbb{P}_l$. To simplify some of the derivations, we adopt the same assumption as in [6] that $\mathbb{P}_l$ is concentrated around its mean, specifically, $\mathbb{P}_l(L/2 \leq l \leq 2L) = 1$. Note that here $\mathcal{X}$ is a multiset since source symbols might have duplicates.

After generating the source alphabet $\mathcal{X}$, we generate the source string $\boldsymbol{s}$ in the following way. Sample B times from $\mathcal{X}$ uniformly at random with replacement. Let the results be $\mathsf{X}_{J_1}, \mathsf{X}_{J_2}, \ldots, \mathsf{X}_{J_B}$ in order. For every X_{J_b}, we then flip t ($t \leq L/2$) symbols uniformly at random as a way to simulate edits and other changes to the data in a simple manner. The number of flipped symbols t will be referred to as the *substitution number*. The flipped version of X_{J_b} is denoted Y_b and referred to as a *source block*. The source string $\boldsymbol{s}$ is then constructed to be the concatenation of source blocks, i.e., $\boldsymbol{s} = Y_1 Y_2 \ldots Y_B$. The entropy of this source is denoted $H(\boldsymbol{s})$. Note that given $\boldsymbol{s}$, the boundary between Y_b and Y_{b+1} is not known to us.

We bound $H(\boldsymbol{s})$ in the next lemma. The proof is omitted due to space limitation.

Lemma 1. *The entropy of the above source model $H(\boldsymbol{s})$ satisfies*

$$B \log \binom{L/2}{t} \leq H(\boldsymbol{s}) \leq B \log\left(A \binom{2L}{t} \right) + (2L+1)A.$$

In this paper, we study the asymptotic regime in which $B, L, A \to \infty$ while substitution number t remains a constant. Unlike the case in which t is linear in BL [18], the entropy for constant t is sub-linear in the length of the uncompressed string. We are particularly interested in the regime where the source string uncertainty mainly results from substitution edits, i.e., the entropy $H(\boldsymbol{s})$ is dominated by the term $B \log \binom{L}{t}$. Therefore, we assume that asymptotically $\log A = O(\log L)$ and $AL = O(B \log L)$.

IV. Deduplication algorithms

The variable-length and multi-chunk deduplication algorithms were both studied in [6] and restated below.

In the *variable-length* deduplication algorithm, we fix an all-zero string of length M, 0^M, to be the marker. The source string $\boldsymbol{s}$ is then split into chunks by this marker. Specifically, the source string $\boldsymbol{s}$ is parsed as $\boldsymbol{s} = Z_1 \cdots Z_C$, where each Z_c (except for perhaps the last one) contains a single appearance of 0^M at the end. The encoding starts with representing the length of $\boldsymbol{s}$ by a prefix-free code. The chunks $\{Z_c\}_{c=1}^C$ are then processed sequentially. Starting with $c = 1$, if chunk Z_c appears for the first time, i.e., $Z_c \neq Z_i$ for all $i < c$, then it is encoded as the bit 1 followed by Z_c itself and is entered into the dictionary. Otherwise, when there already exists an entry in the dictionary storing the same string as Z_c, it will be encoded as the bit 0 followed by a pointer to that entry of the dictionary. The pointer is an index of the dictionary entries and thus can be encoded by $\log\left|T_{VL}^{c-1}\right| + 1$ bits, where T_{VL}^{c-1} is the dictionary right after Z_{c-1} is processed. The number of bits for variable-length deduplication to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{VL}(\boldsymbol{s})$.

In the *multi-chunk* deduplication algorithm, the source string $\boldsymbol{s}$ is again split into chunks by the marker 0^M, but with an additional requirement that chunk lengths are at least 2^{M-1}. We call the chunking process *multi-chunking*. With an abuse of notation, we still denote the chunks by $Z_1, \ldots, Z_C$. The encoding starts with a prefix-free code representing the length of $\boldsymbol{s}$. Chunks are encoded sequentially with a growing dictionary. Consider the chunk Z_c. We assume first that Z_c is new, i.e., it is different from any previously appeared chunk. Let V_c be the largest integer such that chunks $Z_c, Z_{c+1}, \ldots, Z_{c+V_c-1}$ are also new. These new chunks are bundled up and encoded as the bit 1, followed by an encoding of V_c using a prefix-free code for the positive integers, followed by the binary string $Z_c Z_{c+1} \cdots Z_{c+V_c-1}$. Moreover, $Z_c, \ldots, Z_{c+V_c-1}$ are entered into the dictionary in order. Note that each of them is identifiable because they end with the marker 0^M. On the other hand, assume Z_c is not new. Let $\tilde{c} < c$ be the smallest integer satisfying $Z_{\tilde{c}} = Z_c$. Consider the dictionary entry containing $Z_{\tilde{c}}$ and the list of subsequent entries. Let W_c be the largest integer such that the first W_c entries in this list are equal to $Z_c, Z_{c+1}, \ldots, Z_{c+W_c-1}$. Then the chunks Z_c through Z_{c+W_c-1} are bundled up and encoded together as the bit 0, followed by an encoding of W_c using a prefix-free code for the positive integers, followed by a pointer into the dictionary entry containing chunk $Z_{\tilde{c}}$. The expected number of bits for multi-chunk deduplication to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{MC}(\boldsymbol{s})$.

V. Performance analysis

In the following, we study the performance of variable-length and multi-chunk deduplication algorithms over the proposed source model.

A. Variable-length deduplication

We start with a lower bound on the expected length of the compressed strings produced by variable-length deduplication.

Theorem 2. *If $B \leq A\binom{L/2}{t}$, then the average length of the compressed strings produced by variable-length deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \Omega\left(\frac{BL^{\frac{1}{t+1}}}{\log L}\right).$$

Proof: In this proof, we lower bound $\mathcal{L}_{VL}(\boldsymbol{s})$ by the total length of the distinct chunks, denoted W, plus the number of chunks C since each chunk needs one bit indicating if it has appeared before. Clearly, C is greater than the number of non-overlapping marker strings in $\boldsymbol{s}$. Since each source block is a Bernoulli(1/2) process by itself, the expected number of non-overlapping marker strings in $\boldsymbol{s}$ is at least $\frac{BL}{M2^M}$. Hence,

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \mathbb{E}[W] + \frac{BL}{M2^M}. \tag{1}$$

We bound $\mathbb{E}[W]$ in the following.

For each source symbol X_a, we use n_a to denote the number of its descendants among the source blocks. Let $\mathcal{M}$ contain the information about $\{n_a\}_{a=1}^A$, the positions of substitutions in all source blocks, and the lengths of source symbols $\{L_a\}_{a=1}^A$.

We first bound the expected value of $\mathcal{L}_{VL}(\boldsymbol{s})$ conditioned on $\mathcal{M}$. Let $\ell = \min(2^{M-5}, L/4)$. Partition each X_a into segments of length ℓ, i.e., for each X_a, we write $\mathsf{X}_a = \boldsymbol{x}_{a,1}\boldsymbol{x}_{a,2}\cdots\boldsymbol{x}_{a,c_a}\boldsymbol{x}_{a,c_a+1}$, where $|\boldsymbol{x}_{a,1}| = \cdots = |\boldsymbol{x}_{a,c_a}| = \ell, c_{a+1} = \lceil L_a/\ell \rceil$. We consider the substrings of the descendants of X_a that correspond to $\boldsymbol{x}_{a,j}$, denoted $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{n_a}$ (see Figure 1). Each of $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{n_a}$ results from $\boldsymbol{x}_{a,j}$ through at most t substitutions. For each $1 \leq j \leq c_a$, we assume without loss of generality that $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{m_{a,j}}$ are distinct, where $m_{a,j}$ denotes the total number of distinct strings among $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{n_a}$. Note that $m_{a,j}$ is known given $\mathcal{M}$.

Consider the event $\mathcal{E}_1$ that any two $\ell/2$-substrings of the source alphabet are of Hamming distance at least $2t+1$ from each other. It can be shown by considering every pair of $\ell/2$-substrings and applying the union bound that

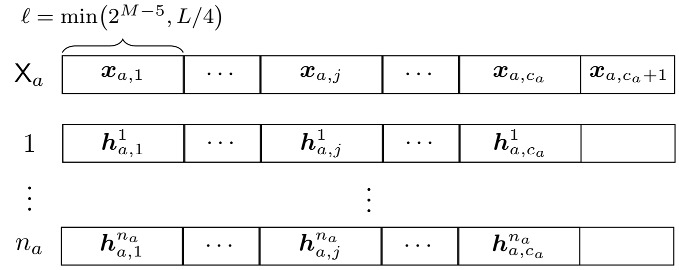


Figure 1. A partition of X_a and its n_a descendants into segments of length ℓ.

$$\Pr(\mathcal{E}_1) \geq 1 - (2AL)^2 \frac{(\ell/2)^{2t}}{2^{\ell/2}} \geq \frac{3}{4},$$

when $t/3 \leq \frac{\ell/2}{12\log(\ell/2)}$ and $\log(AL) \leq \ell/8 - 2$.

Assume $\mathcal{E}_1$ holds. Then different source alphabet $\ell/2$-substrings have different descendants. For instance, the only substrings that are possible to be the same as $\boldsymbol{h}_{a,j}^1$ are $\boldsymbol{h}_{a,j}^2, \ldots, \boldsymbol{h}_{a,j}^{n_a}$. Note that if we have defined $\mathcal{E}_1$ to be the event that any two ℓ-substrings of the source alphabet are of Hamming distance at least $2t{+}1$ from each other, then when $\mathcal{E}_1$ holds, $\boldsymbol{h}_{a,j}^1$ is still possible to be the same as some ℓ-substring that sits across boundaries of source blocks. Therefore, we can assume without loss of generality that $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{m_{a,j}}$ are the first time such strings appear. For any $\boldsymbol{h}_{a,j}^n$, $1 \leq n \leq m_{a,j}$, if $\boldsymbol{h}_{a,j}^n$ is M-RLL, then it is fully contained in some chunk, denoted Z. So Z must have not appeared before (since its substring $\boldsymbol{h}_{a,j}^n$ has not appeared before) and takes $|Z|$ bits to encode. Now that consider the set of distinct descendants of all ℓ-segments in the source alphabet, i.e.,

$$\mathcal{H} = \{\boldsymbol{h}_{a,j}^n : 1 \leq a \leq A, 1 \leq j \leq c_a, 1 \leq n \leq m_{a,j}\}.$$

Every M-RLL string in $\mathcal{H}$ is contained in a chunk that has not appeared before. To enter these chunks into the dictionary, it takes ℓ bits for each M-RLL string in $\mathcal{H}$ since strings in $\mathcal{H}$ do not overlap.

Since the source symbol X_a is a Ber(1/2) process, each $\boldsymbol{h}_{a,j}^n$ is M-RLL with probability at least $1 - 2^{M-5} \cdot 2^{-M} = 1 - 2^{-5}$. By Markov's inequality, with probability at least $3/4$, over $7/8$ of the strings in $\mathcal{H}$ are M-RLL.

Combining the two arguments, with probability at least $1/2$, there are $7|\mathcal{H}|/8$ distinct M-RLL substrings in $\mathcal{H}$, which contribute

$$\frac{7}{8}|\mathcal{H}|\ell = \frac{7}{8}\ell\sum_{a=1}^{A}\sum_{j=1}^{c_a} m_{a,j}$$

bits to the total length of distinct chunks W. It follows that when $\ell \geq 8(2 + \log(AL))$,

$$\mathbb{E}[W|\mathcal{M}] \geq \frac{7}{16}\ell\sum_{a=1}^{A}\sum_{j=1}^{c_a} m_{a,j},$$

and further,

$$\mathbb{E}[W]=\mathbb{E}[\mathbb{E}[W|\mathcal{M}]]\geq\frac{7}{16}\ell\sum_{a=1}^{A}\sum_{j=1}^{c_a}\mathbb{E}[m_{a,j}]. \tag{2}$$

Next, we compute the expected value of $m_{a,j}$. Note that $m_{a,j}$ is independent of the source alphabet. The probability of k substitutions occurring at a fixed set of positions in $\boldsymbol{x}_{a,j}$ is $\binom{L_a-\ell}{t-k}/\binom{L_a}{t}$. Hence,

$$\begin{aligned}\mathbb{E}[m_{a,j}]&=\sum_{k=0}^{t}\binom{\ell}{k}\cdot\left(1-\left(1-\frac{\binom{L_a-\ell}{t-k}}{A\binom{L_a}{t}}\right)^{B}\right)\\&\geq\frac{1}{2}\sum_{k=0}^{t}\binom{\ell}{k}\cdot\min\left(1,\frac{B\binom{L_a-\ell}{t-k}}{A\binom{L_a}{t}}\right)\\&\geq\frac{1}{2}\left(\binom{\ell}{0}\cdot\min\left(1,\frac{B\binom{L_a-\ell}{t-0}}{A\binom{L_a}{t}}\right)\right.\\&\qquad\left.+\binom{\ell}{t}\cdot\min\left(1,\frac{B\binom{L_a-\ell}{t-t}}{A\binom{L_a}{t}}\right)\right)\\&=\frac{1}{2}\left(1+\frac{B\binom{\ell}{t}}{A\binom{L_a}{t}}\right)\geq\frac{1}{2}\left(1+\frac{B\binom{\ell}{t}}{A\binom{2L}{t}}\right),\end{aligned} \tag{3}$$

where the second equality follows from

$$\frac{B\binom{L_a-\ell}{t}}{A\binom{L_a}{t}}\geq\frac{B}{A}\frac{\binom{L/4}{t}}{\binom{2L}{t}}=\frac{B}{A}\left(\frac{1}{8}\right)^{t}(1+o(1))\geq 1.$$

Combining (3), (2) and (1), $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ can be shown to be lower bounded by

$$\frac{7}{64}\left(AL+BL\left(\frac{\ell}{2L}\right)^{t}(1+o(1))\right)\mathbb{1}_{\ell\geq 8(2+\log(AL))}+\frac{BL}{M2^{M}}. \tag{4}$$

The desired result follows from minimizing (4) over M. ■

By the preceding theorem, if $\Omega\left(\frac{AL}{\log L}\right)\leq B\leq A\left(\binom{L/2}{t}\right)$, then $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ is greater than $H(\boldsymbol{s})$ by at least an order of $\frac{L^{\frac{1}{t+1}}}{\log^2 L}$.

In the following, we derive an upper bound on the performance of the variable-length deduplication algorithm.

Theorem 3. *The average length of the compressed strings produced by variable-length deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]\leq 2AL+\Theta\left(BL^{\frac{1}{2}}\log^{\frac{1}{2}}(BL)\right).$$

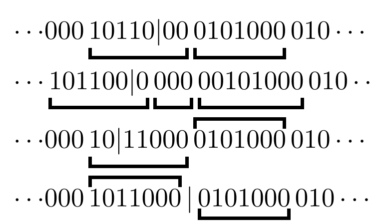
Proof: The variable-length deduplication partitions the source string $\boldsymbol{s}$ as a random number C of chunks, denoted $Z_1,\ldots,Z_C$. The length of $\boldsymbol{s}$ can be encoded in at most $2\log|\boldsymbol{s}|+1$ bits by Elias gamma coding [23]. Let T_{VL}^{c} denote the dictionary right after chunk Z_c is processed (T_{VL}^{0} denotes the initial empty dictionary). We first write

$$\begin{aligned}\mathcal{L}_{VL}(\boldsymbol{s})\leq\sum_{c=1}^{C}&\Big(\mathbb{1}_{Z_c\in T_{VL}^{c-1}}\left(1+\log|T_{VL}^{c-1}|+1\right)\\&+\mathbb{1}_{Z_c\notin T_{VL}^{c-1}}(1+|Z_c|)\Big)+2\log|\boldsymbol{s}|+1.\end{aligned} \tag{5}$$

We next consider a partition of $\boldsymbol{s}$ into a random number of "edit blocks". We first break $\boldsymbol{s}$ at all the boundaries of source blocks. Each source block Y_b is further split in the following way. For all $1\leq a\leq A$, we let the first descendant of X_a be $Y_{g(a)}$, i.e., $g(a)$ is the smallest index such that $J_{g(a)}=a$ (we define $g(a)$ only for source symbols that have at least one descendant). For any other descendant Y_b of X_a, we consider the mismatches between Y_b and $Y_{g(a)}$. Suppose Y_b differs from $Y_{g(a)}$ in positions $c_1,c_2,\ldots,c_m$, $0\leq c_m\leq 2t$. We break Y_b into c_m+1 segments at these positions. Specifically, for all $1\leq j\leq m$, we split between the (c_j-1)-th symbol and the c_j-th symbol. The first segment is set to be empty if $c_1=1$. These segments are referred to as edit blocks. As an example, if $c_1=2,c_2=5$ and $Y_b=01000101$, then the edit blocks are $0,100,0101$.

Thus, conditioned on the differences between each Y_b and its corresponding "first descendant" source block $Y_{g(J_b)}$, we can partition the source string $\boldsymbol{s}=Y_1Y_2\cdots Y_B$ into a random number K of edit blocks, denoted $D_1,\ldots,D_K$ (the boundaries of source blocks are also breakpoints). Note that each $Y_{g(a)}$ has no mismatch with itself, so they are partitioned as edit blocks by themselves, i.e., there exist $k_1,\ldots,k_A$ such that $D_{k_1}=Y_{g(1)},\ldots,D_{k_A}=Y_{g(a)}$.

We define a similar notion of interior chunks and boundary chunks as in [6, Theorem 3] but with respect to edit blocks. Consider chunks whose first symbols are in edit block D_k. Some of them are invariant of the neighboring source blocks and the first bit of D_k. In other words, by replacing D_{k-1}, D_{k+1} or the first bit of D_k by any other strings, the existence or content of these chunks do not change. They are referred to as "interior" chunks. We denote the set of indexes of interior chunks in D_k by $\mathcal{C}_k^{\circ}$. The chunks that are not interior chunks are referred to as "boundary" chunks. Their content depend on neighboring edit blocks D_{k-1}, D_{k+1} and the first bit of D_k, which corresponds to a mismatch between the source block containing D_k and its corresponding "first descendant" source block. We denote the set of indexes of all boundary chunks that start in D_k by $\partial\mathcal{C}_k$. We give examples in the following of boundary chunks (indicated by underbrackets) and interior chunks (indicated by overbrackets) when marker length $M=3$. Vertical bars indicate the boundaries of edit blocks. Different rows are independent examples.



We consider interior chunks and boundary chunks separately. By (5),

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \leq \mathbb{E}\left[\sum_{k=1}^{K}\left(\sum_{\substack{c\in\mathcal{C}_k^\circ\\ Z_c\notin T_{VL}^{c-1}}}|Z_c| + \sum_{\substack{c\in\partial\mathcal{C}_k\\ Z_c\notin T_{VL}^{c-1}}}|Z_c|\right)\right. \\ \left. + \sum_{Z_c\in T_{VL}^{c-1}}\left(1+\log\left|T_{VL}^{c-1}\right|\right) + C\right]. \quad (6)$$

Consider the interior chunks that appear for the first time. Consider the edit block D_k and the source block Y_b that contains D_k. If Y_b is not the first descendant of X_{J_b}, then D_k equals to the substring of $Y_{g(J_b)}$ at the same location with the first bit flipped. It follows from the definition of interior chunks that any interior chunk of D_k must have already appeared as a chunk in that substring of $Y_{g(J_b)}$. Thus, any interior chunk in $\boldsymbol{s}$ that has not appeared in the dictionary is a substring of one of $Y_{g(1)}, \ldots, Y_{g(a)}$. The total length of these chunks is hence less than the sum of lengths of $Y_{g(1)}, \ldots, Y_{g(a)}$. Hence,

$$\mathbb{E}\left[\sum_{k=1}^{K}\sum_{\substack{c\in\mathcal{C}_k^\circ\\ Z_c\notin T_{VL}^{c-1}}}|Z_c|\right] \leq 2AL. \quad (7)$$

Secondly, we upper bound the lengths of boundary chunks. We adopt a similar approach as [6]. We call an occurrence of 10^M *internal* to an edit block D if it starts in D but after its first (mismatch) bit. For edit block D, we use $\mathrm{head}(D)$ to denote the prefix of D which ends at the last zero of the first internal 10^M in D. We use $\mathrm{tail}(D)$ to denote the suffix of D which starts at the first zero of the rightmost 0^M in D. "Head" and "tail" are defined to be D itself if D does not contain corresponding patterns. It can be shown that the total length of boundary chunks is upper bounded by the total length of $\mathrm{head}(D_1), \ldots, \mathrm{head}(D_K), \mathrm{tail}(D_1), \ldots, \mathrm{tail}(D_K)$. Moreover, every D_k by itself is a Bernoulli(1/2) process. The expected number of bits forwards until the end of the first internal 10^M is $2^{M+1}+1$ and the expected number of bits backwards until the beginning of the rightmost 0^M is $2^{M+1}-2$ [24, Chapter 8]. It follows by noting $K \leq (2t+1)B$ that

$$\mathbb{E}\left[\sum_{k=1}^{K}\sum_{\substack{c\in\partial\mathcal{C}_k\\ Z_c\notin T_{VL}^{c-1}}}|Z_c|\right] \leq \mathbb{E}\left[\sum_{k=1}^{K}\left(2^{M+1}+1+2^{M+1}-2\right)\right] \\ \leq (2t+1)B\left(2^{M+2}-1\right). \quad (8)$$

Finally, we upper bound the remaining terms in (6). The number of chunks C is less than the number of occurrences of marker 0^M plus 1 (the last chunk may not contain any marker). The expected number of occurrences of 0^M inside source blocks is $\frac{BL}{2^M}$. It follows that $\mathbb{E}[C] \leq \frac{BL}{2^M} + 1 + B - 1$, where $B-1$ accounts for possible occurrences of 0^M across the boundaries of source blocks. Therefore,

$$\mathbb{E}\left[\sum_{Z_c\in T_{VL}^{c-1}}\left(1+\log\left|T_{VL}^{c-1}\right|\right) + C\right] \\ \leq \mathbb{E}[C(1+\log C) + C] \\ \leq B\left(1+\frac{L}{2^M}\right)(3+\log(BL)), \quad (9)$$

where the second inequality follows from $C \leq 2BL$.

Plugging (7), (8) and (9) in (6), it can be shown with some algebra that $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ is upper bounded by

$$2AL + B\left((2t+1)2^{M+2} + \frac{(3+\log(BL))L}{2^M}\right) \\ + \Theta(B\log(BL)).$$

The desired upper bound follows from choosing

$$M = \left\lceil \frac{1}{2}\log\left(\frac{3+\log(BL)L}{4(2t+1)}\right)\right\rceil.$$

■

Note that the expected length of the source string is BL. When $\log B = o(L)$, $BL^{\frac{1}{2}}\log^{\frac{1}{2}}(BL) = o(BL)$. Therefore, under this condition, the variable-length deduplication can achieve asymptotically arbitrarily large compression ratio.

By Theorem 2 and 3, the variable-length deduplication can achieve arbitrarily large compression ratio but may also spend number of bits larger than entropy by an arbitrarily large factor over the proposed source model.

B. Multi-chunk deduplication

In the following, we present an upper bound on the expected length of compressed strings produced by the multi-chunk algorithm.

Theorem 4. *The average length of the compressed strings by multi-chunk deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{MC}(\boldsymbol{s})] \leq \Theta(AL) + O(B\log(ABL)).$$

The preceding theorem can be proved in a similar fashion as Theorem 3, by partitioning source string into edit blocks and considering boundary and interior chunks but with respect to multi-chunking. The complete proof is omitted here due to space limitation.

By Theorem 4, when $\log B = O(\log L)$,

$$\frac{\mathbb{E}[\mathcal{L}_{MC}(\boldsymbol{s})]}{H(\boldsymbol{s})} \leq O(1).$$

Therefore, with the existence of substitutions, the multi-chunk algorithm can achieve a constant factor of optimal with respect to the entropy.

References

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east", *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[2] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication", *ACM Transactions on Storage*, vol. 7, no. 4, p. 14, 2012.

[3] A. El-Shimi, R. Kalach, A. Kumar, A. Ottean, J. Li, and S. Sengupta, "Primary data deduplicationâlarge scale study and system design", in *Presented as part of the 2012 USENIX Annual Technical Conference*, 2012, pp. 285–296.

[4] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system", in *ACM SIGOPS Operating Systems Review*, ACM, vol. 35, 2001, pp. 174–187.

[5] S. Quinlan and S. Dorward, "Venti: A new approach to archival storage", in *FAST*, vol. 2, 2002, pp. 89–101.

[6] U. Niesen, "An information-theoretic analysis of deduplication", *IEEE Transactions on Information Theory*, vol. 65, no. 9, pp. 5688–5704, Sep. 2019.

[7] W. Xia, H. Jiang, D. Feng, L. Tian, M. Fu, and Y. Zhou, "Ddelta: A deduplication-inspired fast delta compression approach", *Performance Evaluation*, vol. 79, pp. 258–272, 2014.

[8] D. Bobbarjung, C. Dubnicki, and S. Jagannathan, "Fingerdiff: Improved duplicate elimination in storage systems", *Proceedings of Mass Storage Systems and Technologies (MSST06)*, pp. 1–5, 2006.

[9] D. Teodosiu, N. Bjorner, Y. Gurevich, M. Manasse, and J. Porkka, "Optimizing file replication over limited-bandwidth networks using remote differential compression", 2006.

[10] B. Agarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese, "Endre: An end-system redundancy elimination service for enterprises", in *NSDI*, 2010, pp. 419–432.

[11] Y. Zhang, H. Jiang, D. Feng, W. Xia, M. Fu, F. Huang, and Y. Zhou, "Ae: An asymmetric extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication", in *2015 IEEE Conference on Computer Communications (INFOCOM)*, IEEE, 2015, pp. 1337–1345.

[12] C. Yu, C. Zhang, Y. Mao, and F. Li, "Leap-based content defined chunkingâtheory and implementation", in *2015 31st Symposium on Mass Storage Systems and Technologies (MSST)*, IEEE, 2015, pp. 1–12.

[13] E. Kruus, C. Ungureanu, and C. Dubnicki, "Bimodal content defined chunking for backup streams", in *Fast*, 2010, pp. 239–252.

[14] G. Lu, Y. Jin, and D. H. Du, "Frequency based chunking for data de-duplication", in *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE, 2010, pp. 287–296.

[15] B. Zhou and J. Wen, "Hysteresis re-chunking based metadata harnessing deduplication of disk images", in *2013 42nd International Conference on Parallel Processing*, IEEE, 2013, pp. 389–398.

[16] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, *et al.*, "Initial sequencing and analysis of the human genome", *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[17] W. Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu, Y. Zhang, and Y. Zhou, "A comprehensive study of the past, present, and future of data deduplication", *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1681–1710, 2016.

[18] H. Lou and F. Farnoud, "Data deduplication with random substitutions", in *2020 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2020, pp. 2377–2382.

[19] R. Vestergaard, Q. Zhang, and D. E. Lucani, "Generalized deduplication: Bounds, convergence, and asymptotic properties", *arXiv preprint arXiv:1901.02720*, 2019.

[20] L. Conde-Canencia, T. Condie, and L. Dolecek, "Data deduplication with edit errors", in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018, pp. 1–6.

[21] K. Eshghi and H. K. Tang, "A framework for analyzing and improving content-based chunking algorithms", *Hewlett-Packard Labs Technical Report TR*, vol. 30, no. 2005, 2005.

[22] D. R. Bobbarjung, S. Jagannathan, and C. Dubnicki, "Improving duplicate elimination in storage systems", *ACM Transactions on Storage (TOS)*, vol. 2, no. 4, pp. 424–448, 2006.

[23] P. Elias, "Universal codeword sets and representations of the integers", *IEEE transactions on information theory*, vol. 21, no. 2, pp. 194–203, 1975.

[24] R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*. Pearson Education India, 2013.