# Universal Compression of Large Alphabets with Constrained Compressors

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

***Abstract*—Over unknown, possibly large, alphabets, one approach for compressing sequences is to separately convey their symbols and patterns (sequences of integers representing orders in which the symbols appear). It has been shown that patterns generated by i.i.d. sources can be compressed with diminishing redundancy using compressors that know the number of occurrences of each integer symbol. Motivated by applications with resource restrictions, e.g., data deduplication, we study universal compression of patterns using compressors under constraints. A characterization of constrained compressors is given and general results for computing redundancies are derived. We also show that for patterns generated by i.i.d. sources over an alphabet of size $k$, the per-symbol average- and worst-case redundancies are at least $\Theta(\log(\min(k, n/\log n)))$ bits ($n$ is the sequence length), under the constraint that compressors only know the number of distinct integer symbols in the pattern. A simple sequential compressor satisfying this constraint is also analyzed and shown to achieve this redundancy in the first order term.**

## I. Introduction

Shannon's source coding theorem states that to compress a source $X$, we should represent each outcome $x$ with approximately $\log(1/p(x))$ bits. However, there are cases when the source distribution is unknown. A common approach is then to assume a class $\mathcal{P}$ of distributions, e.g., i.i.d. or Markov distributions, to which the source distribution belongs. A good compressor should have 'universality' over all possible sources in the family instead of just being entropy-approaching for a certain source.

A brief introduction to the universal compression framework is as follows. Let a source $X$ be distributed over a discrete support set $\mathcal{X}$ according to a distribution $p$. Every compressor of $X$ corresponds to a probability distribution $q$ over $\mathcal{X}$ where $x \in \mathcal{X}$ is represented by roughly $\log(1/q(x))$ bits. The extra number of bits required to represent $x$ when $q$ is used instead of $p$ is therefore

$$\log\frac{1}{q(x)} - \log\frac{1}{p(x)} = \log\frac{p(x)}{q(x)}.$$

The *worst-case* redundancy of $q$ with respect to $\mathcal{P}$ is defined as the largest number of extra bits used for any $x$ and any distribution $p$, i.e.,

$$\hat{R}(\mathcal{P}, q) = \sup_{p\in\mathcal{P}}\sup_{x\in\mathcal{X}}\log\frac{p(x)}{q(x)} = \sup_{x\in\mathcal{X}}\log\frac{\hat{p}(x)}{q(x)},$$

where $\hat{p}(x)$ denotes $\sup_{p\in\mathcal{P}} p(x)$, the maximum probability assigned to $x$ by any $p \in \mathcal{P}$. The *worst-case* redundancy of $\mathcal{P}$ is defined as

$$\hat{R}(\mathcal{P}) = \inf_q \hat{R}(\mathcal{P}, q) = \inf_q \sup_{x\in\mathcal{X}}\log\frac{\hat{p}(x)}{q(x)}, \tag{1}$$

the lowest number of extra bits in the worst case required by any compressor.

Similarly, one can define the *average-case* redundancy of $\mathcal{P}$ as

$$\bar{R}(\mathcal{P}) = \inf_q \sup_{p\in\mathcal{P}}\left(\sum_{x\in\mathcal{X}}\left(p(x)\log\frac{p(x)}{q(x)}\right)\right), \tag{2}$$

the lowest number of extra bits on average required by any compressor. Note that $\bar{R}$ and $\hat{R}$ are always nonnegative and $\hat{R}$ is an upper bound on $\bar{R}$.

Classic universal compression [1]–[5] considers encoding sequences generated by sources with a known alphabet. However, in applications like language modeling, alphabets can be very large or even unknown. To address this problem, [6] took the approach of describing the sequences in two separate parts: the set of symbols appearing in the sequence and the pattern they form. The pattern of a sequence is a sequence of integers representing the order in which the symbols appear. For example, the sequence "data" is represented by its symbols 'd', 'a', 't' and its pattern "1232". It was shown in [6] and subsequent works [7]–[11] that although the cost of encoding symbols is inevitable, alphabet-independent patterns can be efficiently compressed.

Another application of this encoding scheme is in data deduplication [12], [13]. Data deduplication is an efficient data reduction approach used in large-scale storage systems [14], [15]. As formalized in [16], a typical data deduplication algorithm [16] uses a chunking scheme to parse the data stream into a sequence of 'chunks'. The chunks are then sequentially processed by a dictionary-based compressor. The encoding of each chunk starts with an indicator of whether the chunk has appeared before. For a new chunk, what follows is simply the chunk itself. For a chunk that has previously appeared, its indicator is followed by a pointer to one of its previous appearances. Note that the specific chunking scheme varies in different systems, and is out of the scope of this paper. The above algorithm can be viewed as the symbol-pattern encoding with chunks being new unit 'symbols'. Unique chunks are stored in full, corresponding to the storage of symbols. Indicators and pointers are independent of the actual content of chunks, corresponding to the encoding of the pattern. Data deduplication has been well studied from a practical perspective; see [17] for a comprehensive survey. It was also studied from an information-theoretic point of view in [16], [18]–[20].

Due to the scale of data, encoding schemes in deduplication algorithms are designed to be of low complexity. As described above, every chunk is encoded by an indicator followed by a pointer if it has appeared before. To keep the encoding process time- and memory-efficient, the pointer is encoded by a number of bits equal to the log of the number of distinct chunks seen so far. In this approach, the only information needed to be kept in memory is the collection of chunks that have previously appeared and the computation is simple as all chunks appeared before are viewed as equally likely. The number of bits assigned by the compressor to a previously observed chunk is independent of its probability, which is clearly suboptimal. Pattern compressors that were shown in [6], [7] to have performance close to optimal need to store how many times integer symbols appear so that frequent integers get short representation. As a result, encoding schemes used in deduplication systems will lead to higher redundancies, as a cost to achieve low complexity.

Motivated by applications like data deduplication, we wish to gain a better understanding of the trade-off between complexity and redundancy. In this direction, we study compressors under low-complexity constraints in the framework of universal compression. Constrained compressors do not store or compute all relevant information about a sequence. Such estimators will assign the same probability to sets of sequences whose elements look identical to the compressor. As a result, we define constraints as partitions of the data space $\mathcal{X}$ into parts, where the elements in each part must be assigned the same probability. Alternatively, one can constrain the total space used by a compressor. The partition-based constraints studied here, however, appear more amenable to analysis, as well as more compatible with existing compressors in data deduplication, which are usually characterized by what type of information they use rather than their total space or computational complexity [16].

General results on the worst- and average-case redundancies with respect to the constrained compressors are derived. In particular, we consider universal compression of patterns generated by i.i.d. sources over an alphabet of size $k$ but with constrained compressors. We consider the constraint that compressors are only allowed to use the information about how many distinct symbols (integers) are there in the pattern sequence. (Patterns with the same number of distinct integers are assigned the same probability.) We compute the worst- and average-case redundancies for such compressors. It is shown that under this constraint, the per-symbol redundancies are at least a constant number of bits (diminishing redundancy can be achieved without any constraint). We also show that a simple dictionary-based encoding scheme satisfying this constraint achieves this lower bound up to the first-order term.

Due to space limitation, some of the proofs will be omitted or only sketched.

## II. Preliminaries and Notation

In this paper, we use $\log$ to denote the logarithm to base 2 and use $\ln$ to denote the natural log. We use $[n]$ to denote the set of positive integers $\{1, 2, \ldots, n\}$.

### A. Sequences and patterns

Let $x^n = x_1 x_2 \cdots x_n$ be a sequence of $n$ symbols. We use $|x^n|$ to denote the length of $x^n$ and $N(x^n)$ the number of distinct symbols in $x^n$. We define the index $\iota(x)$ of a symbol $x$ in a sequence $x^n$ to be one more than the number of distinct symbols preceding $x$'s first appearance in $x^n$. The *pattern* of $x^n$ is defined as the sequence of indexes, i.e.,

$$\Psi(x^n) = \iota(x_1)\iota(x_2)\cdots\iota(x_n).$$

As an example, in the sequence "*abacbbc*", $\iota(a) = 1, \iota(b) = 2, \iota(c) = 3$, and hence, $\Psi(abacbbc) = 1213223$. In the following, we use $\boldsymbol{\psi}$ to denote a generic pattern. Elements in patterns are referred to as *index integers*.

We consider a discrete alphabet $\mathcal{A}$ of size $k$. Let $\mathcal{A}^n$ denote the set of all sequences of length $n$ over $\mathcal{A}$ and let $\Psi(\mathcal{A}^n)$ denote the set of patterns of all sequences in $\mathcal{A}^n$, i.e,

$$\Psi(\mathcal{A}^n) = \{\Psi(x^n) : x^n \in \mathcal{A}^n\}.$$

It is clear that $\Psi(\mathcal{A}^n)$ is the same for any alphabet $\mathcal{A}$ of size $k$. It contains all patterns of length $n$ and at most $k$ distinct index integers. So we will write $\Psi^n_{\le k}$ instead of $\Psi(\mathcal{A}^n)$. For example, if $k = 2$ and $n = 3$, then

$$\Psi^3_{\le 2} = \{111, 112, 121, 122\}.$$

Let $\Psi^n_k$ denote the set of patterns of length $n$ and with exactly $k$ distinct index integers. It follows that $\Psi^n_{\le k} = \cup_{m=1}^k \Psi^n_m$.

For a pattern $\boldsymbol{\psi}$, the profile of $\boldsymbol{\psi}$ is a vector of length $|\boldsymbol{\psi}|$, defined as

$$\Phi(\boldsymbol{\psi}) = \big(\varphi_1, \varphi_2, \ldots, \varphi_{|\boldsymbol{\psi}|}\big),$$

where $\varphi_j$ is the number of index integers that appear $j$ times in $\boldsymbol{\psi}$. For example, in pattern 12131, one integer (namely, 1) appears 3 times and two integers (2 and 3) appear once. So $\Phi(12131) = (2, 0, 1, 0, 0)$.

Moreover, we define the innovation vector $\Lambda(\boldsymbol{\psi})$ of $\boldsymbol{\psi}$ to be the vector containing indexes of new symbols. Formally,

$$\Lambda(\boldsymbol{\psi}) = (\lambda_1, \lambda_2, \ldots, \lambda_{N(\boldsymbol{\psi})}),$$

where $\lambda_j$ is the index of the first occurrence of integer $j$. For example, in pattern 12131, integers 2 and 3 first appear in positions 2 and 4, respectively. So $\Lambda(12131) = (1, 2, 4)$. Note that we always have $\lambda_1 = 1$. We use $\Lambda^n_k$ to denote the set of innovation vectors of all patterns in $\Psi^n_k$, i.e.,

$$\Lambda^n_k = \{\Lambda(\boldsymbol{\psi}) : \boldsymbol{\psi} \in \Psi^n_k\},$$

and write $\Lambda^n_{\le k} = \cup_{m=1}^k \Lambda^n_m$.

### B. Universal compression of patterns over i.i.d. sources

The class of i.i.d. sources that generate length-$n$ sequences over $\mathcal{A}$ is denoted $\mathcal{I}^n_k$. Let $\Theta_k = \{(\theta_1, \theta_2, \ldots, \theta_k) : \sum_{i=1}^k \theta_i = 1, 0 \le \theta_i \le 1\}$. Each $p_{\boldsymbol{\theta}} \in \mathcal{I}^n_k$ is then parameterized by a vector $\boldsymbol{\theta} \in \Theta_k$.

Each $p_{\boldsymbol{\theta}}$ induces a distribution over $\Psi^n_{\le k}$ as

$$p_{\boldsymbol{\theta}}(\boldsymbol{\psi}) = \sum_{x^n : \Psi(x^n) = \boldsymbol{\psi}} p_{\boldsymbol{\theta}}(x^n).$$

For example, let $k = 2$, $n = 2$. For $\boldsymbol{\theta} = (0.4, 0.6)$, the induced pattern distribution is $p_{\boldsymbol{\theta}}(11) = 0.4^2 + 0.6^2 = 0.52$, $p_{\boldsymbol{\theta}}(12) = 2 \times 0.4 \times 0.6 = 0.48$. Note the dual use of $p_{\boldsymbol{\theta}}$: $p_{\boldsymbol{\theta}}(x^n)$ denotes the probability of sequence $x^n$ and $p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$ denotes the induced probability of pattern $\boldsymbol{\psi}$.

As mentioned, we are interested in universal compression of patterns generated by i.i.d. sources over alphabets of size $k$. Let $\mathcal{I}_{\Psi}^{n,k}$ denote the set of pattern distributions over $\Psi_{\leq k}^{n}$ induced by $\mathcal{I}_k^n$. From (1), the worst-case redundancy of $\mathcal{I}_{\Psi}^{n,k}$ equals

$$\hat{R}\left(\mathcal{I}_{\Psi}^{n,k}\right) = \inf_{q} \sup_{\boldsymbol{\psi}\in\Psi_{\leq k}^{n}} \log \frac{\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{q(\boldsymbol{\psi})},$$

where $\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi}) = \sup_{\boldsymbol{\theta}\in\Theta_k} p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$. From (2), the average-case redundancy of $\mathcal{I}_{\Psi}^{n,k}$ equals

$$\bar{R}\left(\mathcal{I}_{\Psi}^{n,k}\right) = \inf_{q} \sup_{\boldsymbol{\theta}\in\Theta_k} \left( \sum_{\boldsymbol{\psi}\in\Psi_{\leq k}^{n}} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{q(\boldsymbol{\psi})} \right).$$

In [7], it was shown that for an arbitrarily small $\epsilon > 0$ and $k \leq O\left(n^{(1-\epsilon)/3}\right)$, $\bar{R}\left(\mathcal{I}_{\Psi}^{n,k}\right) \geq \frac{k-1}{2} \log \frac{n^{1-\epsilon}}{k^3}(1+o(1))$. Other existing results mainly focus on the class of i.i.d. sources over arbitrarily large alphabet sizes, i.e., $\mathcal{I}_{\Psi}^{n} = \cup_{k=1}^{\infty} \mathcal{I}_{\Psi}^{n,k} = \mathcal{I}_{\Psi}^{n,n}$. It was shown in [6], [9] that both $\hat{R}(\mathcal{I}_{\Psi}^{n})$ and $\bar{R}(\mathcal{I}_{\Psi}^{n})$ are of order $n^{1/3}$ up to a logarithmic factor.

## III. Universality of Constrained Compressors

We consider constraints resulting from complexity restrictions. Resource-limited compressors are unable to fully process the data, which leads to some data inputs to be indistinguishable. With this intuition, we assume that every constraint $\mathcal{C}$ defines a partition of the support set $\mathcal{X}$ as $\mathcal{X} = \cup_{j=1}^{K} C_j$. Under $\mathcal{C}$, as elements in the same part are indistinguishable to the compressor, they are assigned the same probability. So the set of permitted compressors under $\mathcal{C}$ is

$$\mathcal{Q}_{\mathcal{C}} = \{q : q(x_1) = q(x_2) \text{ if } x_1 \sim x_2\},$$

where $\sim$ denotes the equivalence relation that $x_1$ and $x_2$ belong to the same partition set.

Similar to (1) and (2), for a class $\mathcal{P}$ of sources, we can define the worst- and average-case redundancies under $\mathcal{C}$ as

$$\hat{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) = \inf_{q\in\mathcal{Q}_{\mathcal{C}}} \sup_{x\in\mathcal{X}} \left( \log \frac{\hat{p}(x)}{q(x)} \right),$$

and

$$\bar{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) = \inf_{q\in\mathcal{Q}_{\mathcal{C}}} \sup_{p\in\mathcal{P}} \left( \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \right),$$

respectively.

For a generic distribution $p$ over $\mathcal{X}$, we use $\tilde{p}$ to denote the induced distribution by $p$ over the parts $\{C_j\}_{j=1}^{K}$,

$$\tilde{p}(j) = \sum_{x\in C_j} p(x), \quad j = 1, 2, \ldots, K.$$

For a family of distributions $\mathcal{P}$, we write $\tilde{\mathcal{P}} = \{\tilde{p} : p \in \mathcal{P}\}$.

Moreover, if we distribute $\tilde{p}(j)$ evenly among all $x \in C_j$, we get the flattened distribution of $p$, denoted $\bar{p}$,

$$\bar{p}(x) = \frac{\tilde{p}(j)}{|C_j|} = \frac{\sum_{x\in C_j} p(x)}{|C_j|}, \qquad \text{for } x \in C_j.$$

Next, we present lemmas for computing $\hat{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}})$ and $\bar{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}})$. The proofs are omitted due to space limitation.

**Lemma 1.** *The worst-case redundancy of $\mathcal{P}$ under constraint $\mathcal{C}$ satisfies*

$$\hat{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) = \log \left( \sum_{j=1}^{K} \left( |C_j| \cdot \sup_{x\in C_j} \hat{p}(x) \right) \right),$$

*where $\hat{p}(x) = \sup_{p\in\mathcal{P}} p(x)$.*

Lemma 1 can be proved by a similar argument to that of Shtarkov's sum [21]. The lowest redundancy in the worst case is achieved by assigning each $x$ probability proportional to the largest maximum probability in the same part. Note that when there is no constraint, i.e., the corresponding partition of $\mathcal{X}$ is $\mathcal{X} = \cup_x \{x\}$, $\hat{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}})$ is reduced to the normal case, i.e., $\log(\sum_x \hat{p}(x))$.

**Lemma 2.** *The average-case redundancy of $\mathcal{P}$ under constraint $\mathcal{C}$ satisfies*

$$L(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) \leq \bar{R}(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) \leq U(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}),$$

*where*

$$L(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) = \max \left( \sup_{p\in\mathcal{P}} (D(p||\bar{p})), \bar{R}\left(\tilde{\mathcal{P}}\right) \right),$$

$$U(\mathcal{P}, \mathcal{Q}_{\mathcal{C}}) = \sup_{p\in\mathcal{P}} (D(p||\bar{p})) + \bar{R}\left(\tilde{\mathcal{P}}\right).$$

The bounds on average-case redundancy under constraint is determined by $\sup_{p\in\mathcal{P}} D(p||\bar{p})$ and $\bar{R}\left(\tilde{\mathcal{P}}\right)$. The former is the maximum KL-divergence between a distribution $p$ and its flattened version $\bar{p}$ for all $p \in \mathcal{P}$, which can be viewed as a 'distance' between $\mathcal{P}$ and $\mathcal{Q}_{\mathcal{C}}$. The latter is the average-case redundancy of $\tilde{\mathcal{P}}$, the family of induced distributions over the parts. Note that when there is no constraint, $p, \bar{p}$ and $\tilde{p}$ are identical, so both upper and lower bounds are reduced to $\bar{R}\left(\tilde{\mathcal{P}}\right) = \bar{R}(\mathcal{P})$.

## IV. Lower Bounds on Pattern Redundancies

In this section, we consider two specific sets of constrained pattern compressors and present lower bounds on the worst- and average-case redundancies for encoding patterns generated by i.i.d. sources over an alphabet of size $k$.

The constraints are motivated by data deduplication. Recall that deduplication can be viewed as encoding the contents of chunks of data and their pattern separately. The pattern corresponding to the chunks is encoded sequentially, i.e., one index integer at a time. When the $i$-th integer is processed, the compressor knows the number of distinct index integers among the first $i-1$ positions (i.e., the number of distinct chunks among the first $i$ chunks), but not how many times each has appeared. We will start with the simpler block version of this constraint, denoted $\mathcal{C}_1$, which assigns the same probability to all patterns with the same number of distinct index integers. We then consider the sequential version, $\mathcal{C}_2$, where two patterns are assigned the same probability if their length-$i$ prefixes have the same number of distinct integers for all $i$. It is clear that $\mathcal{C}_1$ is more restrictive than $\mathcal{C}_2$ and any compressor that satisfies $\mathcal{C}_1$ also satisfies $\mathcal{C}_2$. We show later $\mathcal{C}_2$ is equivalent of encoding by innovation vectors. Note that although $\mathcal{C}_2$ is motivated by sequential algorithms, compressors satisfying $\mathcal{C}_2$ need not be sequential.

### *A. Pattern compressors under constraint $\mathcal{C}_1$*

The partition of $\Psi^n_{\le k}$ defined by $\mathcal{C}_1$ is $\Psi^n_{\le k} = \cup_{m=1}^k \Psi^n_m$, and the set of allowed compressors is

$$\mathcal{Q}_1 = \{q : q(\boldsymbol{\psi}_1) = q(\boldsymbol{\psi}_2) \text{ if } N(\boldsymbol{\psi}_1) = N(\boldsymbol{\psi}_2)\}.$$

**Theorem 3.** *As $n \to \infty$, $\hat{R}\Big(\mathcal{I}_\Psi^{n,k}, \mathcal{Q}_1\Big)$ is greater than*

$$\begin{cases} (n \log k - k \log n)(1 + o(1)), & \textit{for } k \le \frac{n}{\ln n}, \\ n(\log n - \log\log n)(1 + o(1)), & \textit{for } k > \frac{n}{\ln n}. \end{cases}$$

*Proof:* By Lemma 1,

$$\hat{R}\Big(\mathcal{I}_\Psi^{n,k}, \mathcal{Q}_1\Big) = \log\left(\sum_{m=1}^{k}\left(|\Psi^n_m| \cdot \sup_{\boldsymbol{\psi}\in\Psi^n_m} \hat{p}_\Psi(\boldsymbol{\psi})\right)\right), \quad (3)$$

where $\hat{p}_\Psi(\boldsymbol{\psi}) = \sup_{\boldsymbol{\theta}\in\Theta_k} p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$ is the maximum probability of pattern $\boldsymbol{\psi}$.

For a pattern $\boldsymbol{\psi}$ with profile $\Phi(\boldsymbol{\psi}) = (\varphi_1, \ldots, \varphi_n)$, it was pointed out in [6] that the maximum probability assigned by any i.i.d. distribution satisfies

$$\hat{p}_\Psi(\boldsymbol{\psi}) \ge \sum_{\mu=1}^{n} \varphi_\mu! \Big(\frac{\mu}{n}\Big)^{\mu\varphi_\mu}. \quad (4)$$

Consider any $m < n$. Let $\bar{\boldsymbol{\psi}}^m$ be any pattern sequence in $\Psi^n_m$ such that integer 1 appears $n - m + 1$ times and each of the integers $2, 3, \ldots, m$ appears only once.

We lower bound $\sup_{\boldsymbol{\psi}\in\Psi^n_m} \hat{p}_\Psi(\boldsymbol{\psi})$ by $\hat{p}_\Psi\big(\bar{\boldsymbol{\psi}}^m\big)$. The profile of $\bar{\boldsymbol{\psi}}^m$ equals $\Phi\big(\bar{\boldsymbol{\psi}}^m\big) = (\bar{\varphi}^m_1, \ldots, \bar{\varphi}^m_n)$ where $\bar{\varphi}^m_{n-m+1} = 1$, $\bar{\varphi}^m_1 = m - 1$, and $\bar{\varphi}^m_i = 0$ for all other values of $i$. By (4),

$$\begin{aligned} \hat{p}_\Psi\big(\bar{\boldsymbol{\psi}}^m\big) &\ge (m-1)! \frac{(n-m+1)^{n-m+1}}{n^n} \\ &= \Big(\frac{m}{n}\Big)^{m-1} \frac{\sqrt{2\pi m}}{e^m}\Big(1 - \frac{m-1}{n}\Big)^{n-m+1}, \end{aligned} \quad (5)$$

where the second inequality follows from Feller's bound on Stirling's approximation [22] that for any $m \ge 1$, $m! \ge \sqrt{2\pi m}\big(\frac{m}{e}\big)^m$.

To compute $|\Psi^n_m|$, we note that there is a one-to-one correspondence between $\Psi^n_m$ and the set of unordered $m$-partitions of $[n]$. The number of $m$-partitions of $[n]$ is known as the stirling number of the second kind and is lower bounded in [23] by

$$\frac{1}{2}(m^2 + m + 1)m^{n-m-1} - 1, \quad (6)$$

for $1 \le m \le n - 1$.

Plugging (5) and (6) into (3) gives

$$\begin{aligned} &\hat{R}\Big(\mathcal{I}_\Psi^{n,k}, \mathcal{Q}_1\Big) \ge \log\left(\sum_{m=1}^{\min(n-1,k)} |\Psi^n_m| \cdot \hat{p}_\Psi\big(\bar{\boldsymbol{\psi}}^m\big)\right) \\ &\ge \log\Bigg(\Big(\frac{1}{2}(m^2 + m + 1)m^{n-m-1} - 1\Big) \\ &\cdot \Big(\frac{m}{n}\Big)^{m-1} \frac{\sqrt{2\pi m}}{e^m}\Big(1 - \frac{m-1}{n}\Big)^{n-m+1}\Bigg)\Bigg|_{m=\min\left(\left\lfloor\frac{n}{\ln n}\right\rfloor, k\right)} \\ &= \begin{cases} (n \log k - k \log n)(1 + o(1)), & \text{for } k \le \frac{n}{\ln n} \\ n(\log n - \log\log n)(1 + o(1)), & \text{for } k > \frac{n}{\ln n}. \end{cases} \end{aligned}$$

■

**Theorem 4.** *Fix any $0 < \epsilon < 1$. As $n \to \infty$, $\bar{R}\Big(\mathcal{I}_\Psi^{n,k}, \mathcal{Q}_1\Big)$ is greater than*

$$\begin{cases} \Big(n \log k - (\log e)k(\ln n)^2\Big)(1 + o(1)), & \textit{for } k < \big(\frac{n}{\ln n}\big)^{1-\epsilon}, \\ (1-\epsilon)n(\log n - \log\log n)(1 + o(1)), & \textit{for } k \ge \big(\frac{n}{\ln n}\big)^{1-\epsilon}. \end{cases}$$

*Proof:* By Lemma 2, $\bar{R}\Big(\mathcal{I}_\Psi^{n,k}, \mathcal{Q}_1\Big)$ is lower bounded by[1]

$$\max\left(\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}), \bar{R}\Big(\tilde{\mathcal{I}}_\Psi^{n,k}\Big)\right) \ge \sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}).$$

Recall that $\bar{p}_{\boldsymbol{\theta}}$ is the flattened distribution of $p_{\boldsymbol{\theta}}$ with respect to the partition $\cup_{m=1}^k \Psi^n_m$ and $\tilde{\mathcal{I}}_\Psi^{n,k}$ is the set of distributions over $[k]$ induced by $\mathcal{I}_\Psi^{n,k}$.

We first find a lower bound on $\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$. Consider $J = \min\Big(k, \Big\lfloor\big(\frac{n}{\ln n}\big)^{1-\epsilon}\Big\rfloor\Big)$ for an $\epsilon \in (0,1)$ and vector $\boldsymbol{\theta}_J = (\theta_1, \theta_2, \ldots, \theta_k) \in \Theta_k$ where $\theta_j = \frac{\ln n}{n}$ for $j = 1, \ldots, J-1$, $\theta_J = 1 - (J-1)\frac{\ln n}{n}$ and $\theta_j = 0$ for $j > J$. We have $\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$ is bounded from below by $D(p_{\boldsymbol{\theta}_J}||\bar{p}_{\boldsymbol{\theta}_J})$, which can be shown to equal

$$-H_{\boldsymbol{\theta}_J}(\boldsymbol{\psi}) + \sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m)\log|\Psi^n_m| + \sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m)\log\frac{1}{p_{\boldsymbol{\theta}_J}(m)}, \quad (7)$$

where $H_{\boldsymbol{\theta}_J}(\boldsymbol{\psi})$ is the entropy of the pattern distribution parameterized by $\boldsymbol{\theta}_J$ and $p_{\boldsymbol{\theta}_J}(m) = \sum_{\boldsymbol{\psi}\in\Psi^n_m} p_{\boldsymbol{\theta}_J}(\boldsymbol{\psi}) = \Pr(\boldsymbol{\psi} \in \Psi^n_m | \boldsymbol{\theta}_J)$.

Since the distributions over patterns are induced by the i.i.d. distributions over sequences, $H_{\boldsymbol{\theta}_J}(\boldsymbol{\psi})$ is no larger than $H_{\boldsymbol{\theta}_J}(x^n)$, which equals

$$\begin{aligned} n\Bigg(&\Big(1 - (J-1)\frac{\ln n}{n}\Big)\log\frac{1}{1 - (J-1)\frac{\ln n}{n}} \\ &+ (J-1)\frac{\ln n}{n}\log\frac{n}{\ln n}\Bigg) < J \ln n \log\frac{ne}{\ln n}, \end{aligned} \quad (8)$$

where the inequality follows $(1-x)\log\frac{1}{1-x} < x\log(e)$ for all $0 < x < 1$.

[1] $\bar{R}\Big(\tilde{\mathcal{I}}_\Psi^{n,k}\Big)$ can be shown to be upper bounded by $\log k$, which can be seen later to be negligible. So it suffices to only consider $\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$.

For the second term in (7), we show that in the original sequence $x^n$, all of the $J$ symbols with positive probabilities will appear with high probability, i.e., $p_{\boldsymbol{\theta}_J}(\Psi(x^n) \in \Psi^n_J) \approx 1$, thus leading to a lower bound approximately $\log|\Psi^n_J|$. Rigorously, given $\boldsymbol{\theta}_J$, in the original sequence $x^n$, the probability that any symbol does not appear is less than or equal to $\left(1 - \frac{\ln n}{n}\right)^n \le \frac{1}{n}$. By the union bound, the probability that all $J$ symbols appear is greater than or equal to $1 - \frac{J}{n} \ge 1 - \left(\frac{n}{\ln n}\right)^{-\epsilon} \frac{1}{\ln n}$. So $p_{\boldsymbol{\theta}_J}(J) \ge 1 - \left(\frac{n}{\ln n}\right)^{-\epsilon} \frac{1}{\ln n}$ and

$$\begin{aligned}
&\sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m) \log|\Psi^n_m| \ge p_{\boldsymbol{\theta}_J}(J) \log|\Psi^n_J| \\
&\ge \left(1 - \left(\frac{n}{\ln n}\right)^{-\epsilon} \frac{1}{\ln n}\right) \log\left(\frac{1}{2}\left(J^2 + J + 1\right) J^{n-J-1} - 1\right) \\
&= (n - J + 1)(\log J)(1 + o(1)), \qquad (9)
\end{aligned}$$

where the inequality follows again from (6).

Combining (7), (8), (9) and trivially lower bounding the last term in (7) by 0 give $\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}) \ge D(p_{\boldsymbol{\theta}_J}||\bar{p}_{\boldsymbol{\theta}_J})$, which is further lower bounded by

$$\begin{aligned}
&(n - J + 1)(\log J)(1 + o(1)) - J \ln n \log \frac{ne}{\ln n} \\
&= \begin{cases} \left(n \log k - (\log e) k (\ln n)^2\right)(1 + o(1)), & \text{for } k < \left(\frac{n}{\ln n}\right)^{1-\epsilon}, \\ (1 - \epsilon) n (\log n - \log\log n)(1 + o(1)), & \text{for } k \ge \left(\frac{n}{\ln n}\right)^{1-\epsilon}. \end{cases}
\end{aligned}$$

■

Theorems 3 and 4 show that if compressors only know the number of distinct integers, the worst- and average-case redundancies are both greater than $\Theta\left(n \log\left(\min\left(k, \frac{n}{\log n}\right)\right)\right)$. Moreover, tightness of the lower bounds can be proved using similar arguments. The per-symbol redundancy thus goes to infinity as the alphabet size $k$ increases. On the other hand, from [9], the redundancies are upper bounded by $\tilde{\Theta}\left(n^{1/3}\right)$ for all $k \le n$ when there is no constraint, i.e., diminishing per-symbol redundancy can be achieved. This large discrepancy results from the fact that pattern probabilities are determined by the profiles, but under $\mathcal{C}_1$, little information about the profiles is available.

### B. Pattern compressors under constraint $\mathcal{C}_2$

We now consider the constraint $\mathcal{C}_2$, which requires compressors to encode patterns according to the number of distinct index integers in their prefixes. The number of distinct index integers is determined by the occurrences of new symbols. Therefore, the corresponding prefixes of two patterns have the same number of distinct integers if and only if the innovation vector of the two patterns are the same.

The partition of $\Psi^n_{\le k}$ defined by $\mathcal{C}_2$ is thus $\Psi^n_{\le k} = \cup_{\boldsymbol{\lambda}\in\Lambda^n_{\le k}} \Psi^n(\boldsymbol{\lambda})$, where $\Psi^n(\boldsymbol{\lambda}) = \{\boldsymbol{\psi} : \Lambda(\boldsymbol{\psi}) = \boldsymbol{\lambda}\}$. The set of allowed compressors is

$$\mathcal{Q}_2 = \{q : q(\boldsymbol{\psi}_1) = q(\boldsymbol{\psi}_2) \text{ if } \Lambda(\boldsymbol{\psi}_1) = \Lambda(\boldsymbol{\psi}_2)\}.$$

**Theorem 5.** *The worst-case redundancy of* $\mathcal{I}^{n,k}_{\Psi}$ *with respect to* $\mathcal{Q}_2$ *is the same as that with respect to* $\mathcal{Q}_1$, *i.e.,* $\hat{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_2\right) = \hat{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_1\right)$.

**Theorem 6.** *Fix any* $\epsilon, \delta \in (0, 1)$. *As* $n \to \infty$, $\bar{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_2\right)$ *is greater than*

$$\begin{cases} \left((1 - \delta) n \log k - \frac{\log e}{\delta} k (\ln n)^2\right)(1 + o(1)), \\ \qquad\qquad\qquad\qquad \textit{for } k < \left(\frac{n}{\ln n}\right)^{1-\epsilon}, \\ (1 - \delta - \epsilon) n (\log n - \log\log n)(1 + o(1)), \\ \qquad\qquad\qquad\qquad \textit{for } k \ge \left(\frac{n}{\ln n}\right)^{1-\epsilon}. \end{cases}$$

Proofs of Theorems 5 and 6 are omitted due to space limitation. It follows that although $\mathcal{C}_2$ allows compressors to acquire substantially more information compared with $\mathcal{C}_1$, the redundancies do not decrease.

## V. A Low-complexity Sequential Compressor

In this section, we analyze the dictionary-based pattern compressor used in data deduplication algorithms, e.g., the fixed-length and variable length algorithms in [16]. For a pattern $\boldsymbol{\psi} = \iota_1 \iota_2 \cdots \iota_n$, the compressor $q_d$ assigns probability sequentially as $q_d(\boldsymbol{\psi}) = \prod_{i=1}^n q_d\left(\iota_i | \iota_1^{i-1}\right)$, and

$$q_d\left(\iota_i | \iota_1^{i-1}\right) = \begin{cases} \frac{1}{|M_{i-1}|} \cdot \frac{1}{2} & \text{if } \iota_i \in M_{i-1}, \\ \frac{1}{2} & \text{if } \iota_i \notin M_{i-1}, \end{cases}$$

where $M_{i-1}$ is the set of all index integers in $\iota_1^{i-1}$, i.e., $M_{i-1} = \{\iota_1, \iota_2, \ldots, \iota_{i-1}\}$. It can be seen from definition that $q_d$ satisfies $\mathcal{C}_2$.

**Theorem 7.** *Let* $n \to \infty$. *For all patterns in* $\Psi^n_{\le k}$, *the compressor* $q_d$ *achieves redundancy* $\hat{R}\left(\mathcal{I}^{n,k}_{\Psi}, q_d\right)$ *at most*

$$\begin{cases} \left(n \log k - k \log n + n + \frac{(\log e) k^2}{n-k}\right)(1 + o(1)), \\ \qquad\qquad\qquad\qquad \textit{for } k < \frac{n}{\log n}, \\ \left(n \log \frac{n}{\log n} + \frac{(\log e) n}{(\log n)^2}\right)(1 + o(1)), \quad \textit{for } k \ge \frac{n}{\log n}. \end{cases}$$

Theorem 7 shows together with lower bounds derived in Section IV that under $\mathcal{C}_2$, the lowest redundancies achievable are of order $n \log\left(\min\left(k, \frac{n}{\log n}\right)\right)$.

## VI. Conclusion

Theorems 5, 6 and 7 show that the dictionary-based pattern compressor in deduplication algorithms has high pattern redundancy. Deduplication algorithms, although effective in practice, are far from optimal and the saving mainly results from removing duplicate chunks. Thus, finding constraints on pattern compressors that can achieve low redundancies while keeping time and memory costs affordable can benefit the performance of deduplication algorithms. This provides an intriguing direction for future work, which may benefit from Lemmas 1 and 2 (general ways for computing redundancies are derived).

Another direction of interest is determining families of distributions for which common constraints, such as $\mathcal{C}_2$, lead to low redundancy. Such families would represent suitable applications for existing deduplication algorithms.

## References

[1] W. Szpankowski and M. J. Weinberger, "Minimax pointwise redundancy for memoryless models over large alphabets", *IEEE transactions on information theory*, vol. 58, no. 7, pp. 4094–4104, 2012.

[2] G. I. Shamir, "On the mdl principle for iid sources with large alphabets", *IEEE transactions on information theory*, vol. 52, no. 5, pp. 1939–1955, 2006.

[3] A. Orlitsky and N. P. Santhanam, "Speaking of infinity", *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2215–2230, 2004.

[4] J. Rissanen, "Universal coding, information, prediction, and estimation", *IEEE Transactions on Information theory*, vol. 30, no. 4, pp. 629–636, 1984.

[5] J. Shtarkov, "Coding of discrete sources with unknown statistics", *Topics in information theory*, pp. 559–574, 1977.

[6] A. Orlitsky, N. P. Santhanam, and J. Zhang, "Universal compression of memoryless sources over unknown alphabets", *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1469–1481, 2004.

[7] G. I. Shamir, "Universal lossless compression with unknown alphabets–the average case", *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 4915–4944, 2006.

[8] J. Acharya, H. Das, and A. Orlitsky, "Tight bounds on profile redundancy and distinguishability", in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012.

[9] J. Acharya, H. Das, A. Jafarpour, A. Orlitsky, and A. T. Suresh, "Tight bounds for universal compression of large alphabets", in *2013 IEEE International Symposium on Information Theory*, IEEE, 2013, pp. 2875–2879.

[10] A. Garivier, "A lower-bound for the maximin redundancy in pattern coding", *Entropy*, vol. 11, no. 4, pp. 634–642, 2009.

[11] S. Boucheron, A. Garivier, and E. Gassiat, "Coding on countably infinite alphabets", *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 358–373, 2008.

[12] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezis, and P. Camble, "Sparse indexing: Large scale, inline deduplication using sampling and locality.", in *Fast*, vol. 9, 2009, pp. 111–123.

[13] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system.", in *Fast*, vol. 8, 2008, pp. 269–282.

[14] 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.

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

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

[17] W. Xia, H. Jiang, D. Feng, *et al.*, "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] ——, "Asymptotic analysis of data deduplication with a constant number of substitutions", in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 3296–3301.

[20] ——, "Data deduplication with random substitutions", *IEEE Transactions on Information Theory*, forthcoming.

[21] Y. M. Shtar'kov, "Universal sequential coding of single messages", *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[22] W. Feller, "An introduction to probability theory and its applications", *1957*,

[23] B. C. Rennie and A. J. Dobson, "On stirling numbers of the second kind", *Journal of Combinatorial Theory*, vol. 7, no. 2, pp. 116–121, 1969.