Interactive communication of balanced distributions and of correlated files*

Alon Orlitsky
AT&T Bell Laboratories

Abstract

(X, Y) is a pair of random variables distributed over a support set S. Person $P_{\mathcal{X}}$ knows X, Person $P_{\mathcal{Y}}$ knows Y, and both know S. Using a predetermined protocol, they exchange binary messages in order for $P_{\mathcal{Y}}$ to learn X. $P_{\mathcal{X}}$ may or may not learn Y. The m-message complexity, $\hat{C}_m$, is the number of information bits that must be transmitted (by both persons) in the worst case if only m messages are allowed. $\hat{C}_\infty$ is the number of bits required when there is no restriction on the number of messages exchanged.

We consider a natural class of random pairs. $\hat{\mu}$ is the maximum number of X values possible with a given Y value. $\hat{\eta}$ is the maximum number of Y values possible with a given X value. The random pair (X, Y) is *balanced* if $\hat{\mu} = \hat{\eta}$. The following hold for *all* balanced random pairs. One-way communication requires at most twice the minimum number of bits: $\hat{C}_1 \leq 2\hat{C}_\infty + 1$. This bound is almost tight: for every α, there is a balanced random pair for which $\hat{C}_1 \geq 2\hat{C}_\infty - 6 \geq \alpha$. Three-message communication is asymptotically optimum: $\hat{C}_3 \leq \hat{C}_\infty + 3\log\hat{C}_\infty + 11$. More importantly, the number of bits required is only negligibly larger than the number needed when $P_{\mathcal{X}}$ knows Y in advance: $\hat{C}_\infty \leq \hat{C}_3 \leq \log\hat{\mu} + 3\log\log\hat{\mu} + 11$.

We apply these results to the following *correlated files* problem. X and Y are binary strings (files) within a small edit distance from each other. $P_{\mathcal{X}}$ knows X while $P_{\mathcal{Y}}$ knows Y and wants to learn X. The results above imply efficient three-message protocols for conveying X to $P_{\mathcal{Y}}$. We provide efficient one-way protocols for certain restricted cases and discuss their possible generalizations.

*Appeared, SIAM Journal on Discrete Mathematics, 6:4, November 1993, pp. 548—564. This version was printed on March 21, 1996.

1 Introduction

The introduction is partitioned into three parts. Subsection 1.1 provides the necessary background. Subsection 1.2 describes the problem investigated in this paper and some of its possible applications. The last subsection reviews the results obtained.

1.1 Definitions and previous results

Consider two *communicators*: an *informant* $P_{\mathcal{X}}$ having a random variable X and a *recipient* $P_{\mathcal{Y}}$ having a random variable Y. The *random pair* (X, Y) is distributed according to a probability distribution that is known to both communicators. $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ want the recipient, $P_{\mathcal{Y}}$, to learn X with zero probability of error. The informant, $P_{\mathcal{X}}$, may or may not learn Y. How many bits must be transmitted?

This *interactive communication* problem can be viewed as a variation on communication complexity [1, 2]. The function computed by $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ is trivial: $f(X, Y) = X$, and the difficulty arises because the inputs, X and Y, are correlated.

To further specify the problem, we assume that the communicators alternate in transmitting *messages*: finite sequences of bits. Messages are transmitted over an error-free channel and are determined by an agreed-upon, deterministic, protocol. For every *input* — a possible value assignment for X and Y — the protocol determines a finite sequence of transmitted messages. The protocol is *m-message* if, for all inputs, the number of messages transmitted is at most m.

The *worst-case complexity* of a protocol is the number of bits it requires both communicators to transmit, maximized over all inputs. $\hat{C}_m$, the *m-message complexity*[1] of (X, Y), is the minimum complexity of an m-message protocol for (X, Y). For example, $\hat{C}_1$, the *one-way complexity* of (X, Y), is the number of bits required in the worst-case when $P_{\mathcal{Y}}$ cannot transmit to $P_{\mathcal{X}}$, and $\hat{C}_2$ is the number of bits required in the worst-case when at most two messages are permitted: $P_{\mathcal{Y}}$ transmits a message reflecting Y, then $P_{\mathcal{X}}$ responds with a message from which $P_{\mathcal{Y}}$ must infer X. Since empty messages are allowed, $\hat{C}_m$ is a non-increasing function of m bounded below by 0. We can therefore define $\hat{C}_\infty$, the *unbounded-message complexity of* (X, Y), to be the limit of $\hat{C}_m$ as $m \to \infty$. It is the *minimum* number of bits that must be transmitted for $P_{\mathcal{Y}}$ to know X, even if no restrictions are placed on the number

[1] The pair (X, Y) is implicit in the notation and will be implied by the context.

of messages exchanged. Clearly,

$$\hat{C}_1 \geq \hat{C}_2 \geq \hat{C}_3 \geq \cdots \geq \hat{C}_\infty \ .$$

The following example and results, taken from [3], relate these complexity measures. For others aspects of interactive communication, see [4, 5, 6].

Example 1 A league has t teams. $P_\mathcal{Y}$ knows two teams that played in a game, and $P_\mathcal{X}$ knows the team that won the game. They communicate in order for $P_\mathcal{Y}$ to learn the winning team.

If only one message is allowed, necessarily from $P_\mathcal{X}$ to $P_\mathcal{Y}$, it must be based solely on the winner (for that is all $P_\mathcal{X}$ knows). If the message transmitted when team i wins is the same as (or a prefix of) the message transmitted when team j wins, then in the event of a match between teams i and j, $P_\mathcal{Y}$ cannot tell who the winner is (or when the message ends). Therefore, there must be t different, prefix free, messages and at least one of them must be of length $\geq \lceil \log t \rceil$. This bound is clearly achievable, hence,

$$\hat{C}_1 = \lceil \log t \rceil \ .$$

If two messages are allowed, $P_\mathcal{Y}$ considers the binary representations of the two teams that played and transmits $\lceil \log \log t \rceil$ bits describing the location of the first bit where they differ. $P_\mathcal{X}$ responds by transmitting a single bit describing the bit value of the winning team in that location. Therefore, $\hat{C}_2 \leq \lceil \log \log t \rceil + 1$. It can be shown that for this example

$$\hat{C}_2 = \ldots = \hat{C}_\infty = \lceil \log \log t \rceil + 1 \ . \qquad \square$$

The example shows that for some random pairs, one-message complexity is exponentially higher than the minimum:

$$\hat{C}_1 = 2^{\hat{C}_\infty - 1} \ .$$

Yet two messages always suffice to reduce communication to almost the minimum: for all random pairs,

$$\hat{C}_2 \leq 4\hat{C}_\infty + 3 \ .$$

This contrasts with communication complexity where a succession of papers [7, 8, 9] showed that for every m there is a function whose m-message complexity is almost exponentially higher than its $(m+1)$-message complexity.

It is not known whether there is an m such that m-message is asymptotically optimum, namely, for all random pairs

$$\hat{C}_m \leq \hat{C}_\infty + o(\hat{C}_\infty) \ .$$

In this paper, we consider a natural class of random pairs for which stronger statements hold true.

1.2 Balanced pairs and correlated files

Let (X, Y) be a random pair. Its *support set* is the set S of possible inputs. The support set is of interest as it determines the m-message complexity $\hat{C}_m$ for all m. This evident property of worst-case complexities is formally proven in [3]. $P_{\mathcal{Y}}$'s *ambiguity* when he has the value y is

$$\mu(y) \stackrel{\text{def}}{=} |\{x : (x, y) \in S\}| \ , \tag{1}$$

the number of possible X values when $Y = y$. $P_{\mathcal{Y}}$'s *maximum ambiguity* is

$$\hat{\mu} \stackrel{\text{def}}{=} \max_y \{\mu(y)\}, \tag{2}$$

the maximum number of X values possible with any given Y value. $P_{\mathcal{X}}$'s ambiguity $\eta(x)$ when he has the value x, and his maximum ambiguity $\hat{\eta}$, are similarly defined. In the league problem of Example 1, $\hat{\mu} = 2$ as for every game known to $P_{\mathcal{Y}}$ there are two possible winners known to $P_{\mathcal{X}}$. Similarly, $\hat{\eta}$ in that case is $t - 1$, corresponding to the number of possible losing teams.

We consider the class of *balanced* random pairs: pairs satisfying

$$\hat{\mu} = \hat{\eta} \ .$$

Balanced pairs arise naturally whenever there is no distinction between the two communicators or when X and Y are known to be within some 'distance' from each other. For example[2]:

1. X and Y, inaccurate measurements of the same quantity, are integers within a bounded absolute difference from each other.

2. X and Y, obtained from a noisy binary transmission or from a faulty memory, are n-bit strings within a bounded Hamming distance from each other.

[2]The pairs below are in fact *symmetric*: $(x, y) \in S$ if and only if $(y, x) \in S$. Clearly, every symmetric pair is also balanced. Our results hold for all balanced pairs, hence are presented that way.

Of these and other examples of balanced pairs, the following *correlated-files*, or *edit-distance problem* shows most promise of being practically useful. The *edit distance* between two binary strings X and Y is the minimum number of deletions and insertions to X needed to derive Y. For example, the edit distance between the empty string and any n-bit string is n, and the edit distance between 01010 and 10101 is two. In the correlated-files problem, X and Y are binary strings within a small edit distance from each other. $P_{\mathcal{X}}$ knows X while $P_{\mathcal{Y}}$ knows Y and wants to learn X.

This problem can arise in many situations: (1) $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ write a joint book and each updates his version individually. (2) X is the new digital image taken by a satellite $P_{\mathcal{X}}$ and Y is the previous frame, known to $P_{\mathcal{Y}}$ (successive images are likely to be within a small edit distance). (3) X and Y are different versions of the same program or file. (4) X and Y were received from the same binary transmission with erroneous insertions, deletions, and reversal of bits.

In all those cases, the edit distance between X and Y is much smaller than the number of bits in each. We are looking for a way to communicate X to $P_{\mathcal{Y}}$ without transmitting all of it. Of course, in cases (1) and (2), if $P_{\mathcal{X}}$ keeps the original versions of the file (or image), he can efficiently transmit the locations of the insertions/deletions. But in cases (3) and (4), there is no such reference sequence. Surprisingly, there is almost no difference between the number of bits required in the two cases. We show that even when $P_{\mathcal{X}}$ knows only X (as we assume), X can be communicated to $P_{\mathcal{Y}}$ using only negligibly more bits than the number needed if $P_{\mathcal{X}}$ knew Y in advance.

The protocols achieving this near-minimum number of bits are interactive and require $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ to exchange three messages. For general one-way protocols we can only prove that they require twice as many bits as needed if $P_{\mathcal{X}}$ knows Y. However, efficient one-way protocols for this problem would have additional applications: (1) Simultaneous updates of many *different* files (without the need to respond individually to each one). (2) Transmission of images to many recipients, each with a different (possibly erroneous) prior image. (3) Efficient backup of files without keeping track of individual edit operations. (4) Compression of long related sequences without keeping a reference sequence.

For these reasons we consider the possibility of efficient one-way protocols for the correlated-files problem. Unfortunately, we can prove only partial results and the main question remains unsolved.

1.3 Results

For general distributions, one-way communication may require exponentially more bits than the minimum necessary (e.g., $\hat{C}_1 = \lceil \log t \rceil$ vs. $\hat{C}_\infty = \lceil \log \log t \rceil + 1$ for the league problem with t teams). Yet, Corollary 1 in Section 2 shows that for all balanced random pairs, one-way communication requires at most twice the minimum necessary:

$$\hat{C}_1 \leq 2\hat{C}_\infty + 1 \ .$$

This bound is almost tight. Lemma 2 shows that for all $\alpha \geq 0$ there is a balanced random pair such that

$$\hat{C}_1 \geq 2\hat{C}_\infty - 6 \geq \alpha \ .$$

In Section 3 we show that three-messages communication is asymptotically optimum. More importantly, we prove that although the informant, $P_\mathcal{X}$, does not know Y, the number of bits needed to convey X to $P_\mathcal{Y}$ is only negligibly larger than would be required if $P_\mathcal{X}$ knew Y in advance.

Specifically, Equations (1) and (2) defined $P_\mathcal{Y}$'s ambiguity, $\mu(y)$, when he has the value y and his maximum ambiguity $\hat{\mu}$. Clearly, at least $\lceil \log \mu(y) \rceil$ bits must be transmitted in the worst case when $P_\mathcal{Y}$'s value is y. Hence,

$$\hat{C}_\infty \geq \lceil \log \hat{\mu} \rceil \ . \tag{3}$$

Had $P_\mathcal{X}$ known Y in advance, this bound would be tight:

$$\hat{C}^* = \lceil \log \hat{\mu} \rceil \ ,$$

where $\hat{C}^*$ denotes the number of bits needed *if* $P_\mathcal{X}$ knows Y in advance (note that in this case interaction cannot help, hence we omitted the subscript representing the number of messages). However, $P_\mathcal{X}$ does not know Y, hence $\lceil \log \hat{\mu} \rceil$ bits cannot be always achieved. In the league problem, for example, the maximum ambiguity, $\hat{\mu}$, is two, and if $P_\mathcal{X}$ knew Y (the game) he would need to transmit only one bit (say, whether the winning team is lexicographically first). Yet, $P_\mathcal{X}$ does not know the game and we saw that many more than $\lceil \log \hat{\mu} \rceil$ bits must be transmitted: $\hat{C}_\infty = \lceil \log \log t \rceil + 1$ bits.

However, Section 3 shows that *for all balanced pairs, there is almost no increase in communication due to* $P_\mathcal{X}$ *not knowing* Y*:*

$$\hat{C}_3 \leq \log \hat{\mu} + 3 \log \log \hat{\mu} + 11 \ . \tag{4}$$

Furthermore, for general distributions it is not known whether there is an m such that m messages are asymptotically optimum. As a corollary of the last result, we get that for all balanced distributions, three messages are asymptotically optimum:

$$\hat{C}_3 \leq \hat{C}_\infty + 3\log\hat{C}_\infty + 11 \ .$$

The bound of Inequality 4 is refined in Section 4 for balanced distributions where $P_\mathcal{Y}$'s ambiguity, $\mu(y)$, varies widely with y. We show (see Theorem 2 for the precise formulation) that for all random pairs, there is a four-message protocol that, for all y with a given $\mu(y)$, requires an average of at most $\log\mu(y) + 4\log\log\mu(y)$ transmitted bits.

In Subsection 5.1 we apply these results to the correlated-files (edit-distance) problem. We derive three-message protocols that are bitwise-efficient both asymptotically and for moderately-sized files. We show that similar protocols can be constructed also for more realistic edit models (e.g., where an edit operation can move a segment).

For one-way communication, the general results imply only protocols that require at most twice the number of bits necessary when $P_\mathcal{X}$ knows Y in advance. In view of their significance, we consider asymptotically-optimal, one-way protocols for the correlated files problem in Subsection 5.2. Unfortunately, we cannot prove that they generally exist. Instead, we: (1) Reduce the problem of conveying edited files to that of conveying files with insertions alone; this simplifies the task of designing protocols and analyzing them. (2) Describe an optimal one-way protocol for a single insertion / deletion, showing that for that case $\hat{C}_1 = \ldots = \hat{C}_\infty = \lceil\log(n+2)\rceil$ where n is the length of the string Y. (3) Analyze a possible reduction of insertion protocols to protocols identifying inversions — a problem for which efficient protocols exist (cf. Example 4). We then use this approach to construct efficient protocols for a very restricted case of insertions and deletions.

2 One-way complexity is at most twice the minimum

For general pairs, one-way communication may require exponentially more bits than the minimum necessary (e.g., $\lceil\log t\rceil$ vs. $\lceil\log\log t\rceil + 1$ for the league problem). However, for all balanced pairs one message requires at most twice the minimum number of bits. This will become apparent once we delineate some preliminary definitions and results.

Let (X,Y) be a random pair with support set S. The *support set of* X is the set

$$S_\mathcal{X} \stackrel{\text{def}}{=} \{x : (x,y) \in S \text{ for some } y\}$$

of possible values of X. The *support set* $S_{\mathcal{Y}}$ *of* Y is similarly defined. Instrumental in determining $\hat{C}_1$ is G, the *characteristic hypergraph* of (X, Y). Its vertex set is $S_{\mathcal{X}}$ and for every $y \in S_{\mathcal{Y}}$, it contains the hyperedge:

$$E(y) \stackrel{\text{def}}{=} \{x : (x, y) \in S\} \ . \tag{5}$$

The characteristic hypergraph is equivalent to a graph defined by Witsenhausen [10] who considered the one-message version of this problem. For the league problem with t teams, for example, G has t vertices, one corresponding to each (team) value of X. It has $\binom{t}{2}$ edges, one corresponding to each possible (game) value of Y. Each edge contains two vertices (the possible winning teams in the game). In other words, G is K_t, the complete graph on t vertices.

A coloring of a hypergraph G is an assignment of a color to every vertex of G such that no two vertices sharing a hyperedge are assigned the same color. The chromatic number $\chi(G)$ of G is the minimum number of colors required to color G. It can be shown that:

Result 1 (Lemma 2 in [3]) For all random pairs,

$$\hat{C}_1 = \lceil \log \mathcal{G} \rceil \ . \qquad \square$$

This proves, again, that for the league problem with t teams $\hat{C}_1 = \lceil \log \chi(K_t) \rceil = \lceil \log t \rceil$, as was shown in Example 1.

Lemma 1 For all random pairs,

$$\hat{C}_1 \leq \log \hat{\mu} + \log \hat{\eta} + 1 \ .$$

Proof: According to Result 1, the one-way complexity of (X, Y) is determined by the chromatic number of the characteristic hypergraph:

$$\hat{C}_1 = \lceil \log \mathcal{G} \rceil \ .$$

Each vertex in G belongs to at most $\hat{\eta}$ edges, and each edge contains at most $\hat{\mu}$ vertices. Hence,

$$\mathcal{G} \leq \hat{\eta} \cdot (\hat{\mu} - 1) + 1 \leq \hat{\mu} \cdot \hat{\eta} \ . \qquad \square$$

Corollary 1 For all balanced random pairs,

$$\hat{C}_1 \leq 2 \log \hat{\mu} + 1 \leq 2\hat{C}_\infty + 1 \ .$$

Proof: Immediate from the previous lemma and Inequality (3). $\square$

The next lemma proves this bound almost tight. Example 2 in Section 3 provides a slightly weaker result for another, more elementary, family of balanced random pairs.

Lemma 2 [With Jeff Kahn] For every $\alpha \geq 0$, there is a balanced random pair satisfying

$$\hat{C}_1 \geq 2\hat{C}_\infty - 6 \geq \alpha \ .$$

Proof: Let n be a prime power. Then there is a projective plane π of order n. π consists of n^2+n+1 points and the same number of lines. Every line is incident upon $n+1$ points and every point is on $n+1$ lines. Every two points are connected by exactly one line, and every two lines intersect at exactly one point.

$P_\mathcal{Y}$ knows a line Y of π and $P_\mathcal{X}$ knows a point X on Y. The pair (X, Y) is balanced as

$$\hat{\mu} = \hat{\eta} = n+1 \ .$$

Since every two points are connected by a line, in a one-message protocol, each of the n^2+n+1 points of π must be assigned a different message, therefore

$$\hat{C}_1 = \lceil \log(n^2+n+1) \rceil \ .$$

We now construct a two-message protocol showing that

$$\hat{C}_2 \leq \lceil \log(n+1) \rceil + 2 \ .$$

Let p_1, p_2, and p_3 be distinct points of π, not all on a single line. For each of the three points, $P_\mathcal{X}$ and $P_\mathcal{Y}$ agree in advance on a $\lceil \log(n+1) \rceil$-bit encoding of the $n+1$ lines incident upon it.

When $P_\mathcal{Y}$ is given a line Y and $P_\mathcal{X}$ is given a point X on Y, $P_\mathcal{Y}$ transmits the index i of a point among p_1, p_2, and p_3 that is not on Y (there must be one by the choice of the three points). $P_\mathcal{X}$ responds with the encoding of the (unique) line incident upon p_i and connecting it to X.

$P_\mathcal{Y}$ now knows two lines, both incident upon X. As every two lines intersect at a unique point, $P_\mathcal{Y}$ can determine X. $\square$

We have proven that a single message may require up to twice the minimum number of bits. The next section shows that three messages require at most negligibly more bits than the minimum necessary.

3 Three-message communication is asymptotically optimum

In Subsection 1.3 we noted that for general distributions: (1) it is not known whether there is an m such that m-message communication is asymptotically optimum; (2) there may be a large discrepancy between the number $\lceil \log \hat{\mu} \rceil$ of bits needed when $P_{\mathcal{X}}$ knows Y in advance and $\hat{C}_\infty$, that required when $P_{\mathcal{X}}$ does not know Y.

Balanced pairs are different. We show that three messages require only negligibly more than $\lceil \log \hat{\mu} \rceil$ bits. In particular, three-message commuunication is asymptotically optimum.

First we describe a two-message protocol that improves on Lemma 1 for random pairs with $\hat{\mu} \ll \hat{\eta}$. Recall that the support set, S, of a random pair (X, Y) is the set of all possible inputs, that the support set of X is the set $S_{\mathcal{X}}$ of all possible values of X, and that $S_{\mathcal{Y}}$ is similarly defined. $P_{\mathcal{Y}}$'s *ambiguity set* when his random variable attains the value $y \in S_{\mathcal{Y}}$ is the set $E(y)$, defined in Equation (5), of possible X values in that case. Denote the collection of $P_{\mathcal{Y}}$'s ambiguity sets by

$$\mathcal{E} \stackrel{\text{def}}{=} \{E(y) : y \in S_{\mathcal{Y}}\} \ .$$

A collection of functions, each defined over $S_{\mathcal{X}}$, *perfectly hashes* $\mathcal{E}$ if for every $y \in S_{\mathcal{Y}}$ there is a function in the collection that is one-to-one over (or *hashes*) $E(y)$. For various results on, and applications of perfect-hash functions see [11, 12, 13, 14].

Let b be an integer and let $\mathcal{F}$ be a collection of functions from $S_{\mathcal{X}}$ to $\{1, \ldots, b\}$ that perfectly hashes $\mathcal{E}$. We show that

$$\hat{C}_2 \leq \lceil \log |\mathcal{F}| \rceil + \lceil \log b \rceil \ . \tag{6}$$

$P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ agree in advance on a $\lceil \log |\mathcal{F}| \rceil$-bit encoding of the functions in $\mathcal{F}$ and on a $\lceil \log b \rceil$-bit encoding of $\{1, \ldots, b\}$. When $P_{\mathcal{X}}$ is given X and $P_{\mathcal{Y}}$ is given Y, they execute the following protocol. $P_{\mathcal{Y}}$ finds a function $f_Y \in \mathcal{F}$ that perfectly hashes $E(Y)$. Using $\lceil \log |\mathcal{F}| \rceil$ bits, he transmits the encoding of f_Y to $P_{\mathcal{X}}$. Now $P_{\mathcal{X}}$ knows X and f_Y. He computes $f_Y(X)$ and transmits it to $P_{\mathcal{Y}}$ using $\lceil \log b \rceil$ bits. Since f_Y is one-to-one over $E(Y)$, $P_{\mathcal{Y}}$ can recover X from $f_Y(X)$.

Lemma 3 For all nontrivial[3] (X, Y) pairs,

$$\hat{C}_2 \leq 2 \log \hat{\mu} + \log \log \max\{\hat{\mu}, \hat{\eta}\} + 4 \ .$$

[3]A pair (X, Y) is *trivial* if $\hat{\mu} = 1$. For trivial pairs, $\hat{C}_1 = \ldots = \hat{C}_\infty = 0$.

Proof: In view of the above, we show the existence of a small collection $\mathcal{F}$ of functions with a small range $\{1,\ldots,b\}$ that perfectly hashes $\mathcal{E}$. Let $b \stackrel{\text{def}}{=} \hat{\mu}^2$. Pick a random function $F : S_{\mathcal{X}} \to \{1,\ldots,b\}$ by assigning, uniformly at random, a value in $\{1,\ldots,b\}$ to each element of $S_{\mathcal{X}}$. For all $y \in S_{\mathcal{Y}}$, the probability that $E(y)$ is hashed by F is:

$$\frac{b^{\underline{\mu(y)}}}{b^{\mu(y)}} \geq \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}} \geq \left(1 - \frac{1}{\hat{\mu}}\right)^{\hat{\mu}} \geq \frac{1}{4} \ .$$

where $x^{\underline{i}} \stackrel{\text{def}}{=} x \cdot (x-1) \cdot \ldots \cdot (x-i+1)$ denotes the ith falling power of x.

Independently pick m such random functions. Let $B(y)$ be the "bad" event that $E(y)$ is not hashed by any of these functions. For each y,

$$\Pr(B(y)) \leq \left(\frac{3}{4}\right)^m \ .$$

Furthermore, each $E(y)$ intersects $E(y')$ for at most $\hat{\mu} \cdot (\hat{\eta} - 1)$ values of $y' \in S_{\mathcal{Y}}$. Therefore, the event $B(y)$ is independent of the events $B(y')$ for all but at most $\hat{\mu} \cdot (\hat{\eta} - 1)$ values of $y' \in S_{\mathcal{Y}}$. By the Lovász local lemma (e.g., Chapter 8 in [15]), if

$$4 \cdot \left(\frac{3}{4}\right)^m \hat{\mu} \cdot (\hat{\eta} - 1) < 1 \ , \tag{7}$$

then $\Pr\left(\bigcup_{y \in S_{\mathcal{Y}}} B(y)\right) < 1$, namely, there is a collection of m functions that perfectly hashes $\mathcal{E}$. To conclude, we note that Inequality (7) holds for

$$m = \left\lceil \frac{\log \hat{\mu} + \log \hat{\eta} + 2}{\log 4/3} \right\rceil \leq 8 \log \max\{\hat{\mu}, \hat{\eta}\} \ . \qquad \square$$

If $\log \hat{\mu}$ is larger than $2 \log \log \max\{\hat{\mu}, \hat{\eta}\} + 7$, we can reduce the total number of transmitted bits. First we use a $\log \hat{\mu}$-bit message to convert the pair (X, Y) into a pair (X', Y) with $\hat{\eta}' = \hat{\eta}$ and $\hat{\mu}' \leq \log \max\{\hat{\mu}, \hat{\eta}\} + 4$ and then apply the lemma.[4]

Theorem 1 For all nontrivial random pairs,

$$\hat{C}_3 \leq \log \hat{\mu} + 3 \log \log \max\{\hat{\mu}, \hat{\eta}\} + 11.$$

Proof: Pick a random function $F : S_{\mathcal{X}} \to \{1,\ldots,\hat{\mu}\}$ by assigning, uniformly at random, a value in $\{1,\ldots,\hat{\mu}\}$ to each element of $S_{\mathcal{X}}$. For an integer k, let $B(y)$ be the "bad" event that F assigns some value in $\{1,\ldots,\hat{\mu}\}$ to more than k elements in $E(y)$. For all y,

$$\Pr(B(y)) \leq \hat{\mu} \cdot \binom{\mu(y)}{k} \cdot \frac{1}{\hat{\mu}^k} \leq \frac{\hat{\mu}}{k!} \ .$$

[4] We use $\hat{\eta}'$ and $\hat{\mu}'$ to denote $P_{\mathcal{X}}$'s and $P_{\mathcal{Y}}$'s maximum ambiguities for the pair (X', Y).

Furthermore, the event $B(y)$ is independent of the events $B(y')$ for all but at most $\hat{\mu}\cdot(\hat{\eta}-1)$ values of $y' \in S_{\mathcal{Y}}$. By the Lovász local lemma, if

$$\frac{4\cdot\hat{\mu}\cdot(\hat{\eta}-1)\cdot\hat{\mu}}{k!} < 1 \ , \tag{8}$$

then there is a function $f : S_{\mathcal{X}} \to \{1,\ldots,\hat{\mu}\}$ that is not "bad" for any y, namely, for all $y \in S_{\mathcal{Y}}$ and all $i \in \{1,\ldots,\hat{\mu}\}$,

$$|\{x \in E(y) : f(x) = i\}| \leq k \ .$$

We are interested in a small k satisfying Inequality (8). Using the inequality $k! \geq \left(\frac{k}{e}\right)^k$, it suffices to require

$$2\log\hat{\mu} + \log\hat{\eta} + 2 \leq k\log\left(\frac{k}{e}\right) \ . \tag{9}$$

It is easy to verify that

$$k = \lceil \log\max\{\hat{\mu},\hat{\eta}\} + 4 \rceil$$

satisfies Inequality (9) for sufficiently large $\max\{\hat{\mu},\hat{\eta}\}$'s and satisfies Inequality (8) directly for smaller $\max\{\hat{\mu},\hat{\eta}\}$'s.

Therefore, if $P_{\mathcal{X}}$'s first message consists of $\lceil\log\hat{\mu}\rceil$ bits describing $f(X)$, then $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ can restrict their attention to a random pair (X',Y) where X' is the random variable X restricted to the domain $\{x : f(x) = f(X)\}$. By choice of f we have $\hat{\mu}' \leq \lceil\log\max\{\hat{\mu},\hat{\eta}\}+4\rceil$ and $\hat{\eta}' \leq \hat{\eta}$. Thereafter, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ can use the two-message protocol of the previous lemma to convey X' to $P_{\mathcal{Y}}$. The total number of bits transmitted is at most

$$\begin{aligned}&\lceil\log\hat{\mu}\rceil + 2\log\lceil\log\max\{\hat{\mu},\hat{\eta}\}+4\rceil + \log\log(\max\{\hat{\mu},\hat{\eta}\}+3) + 4 \\ &\qquad\qquad \leq \log\hat{\mu} + 3\log\log\max\{\hat{\mu},\hat{\eta}\} + 11 \ .\end{aligned}$$

With more care, the constant 11 can be reduced. □

For balanced pairs $\hat{\mu} = \hat{\eta}$ and we readily obtain:

Corollary 2 For all balanced random pairs,

$$\hat{C}_3 \leq \log\hat{\mu} + 3\log\log\hat{\mu} + 11 \ . \qquad \square$$

The techniques used to prove Lemma 3 and Theorem 1 work extremely efficiently in the following example.

Example 2 Consider an n by n chessboard with two rooks in mutually capturing positions (i.e., in the same row, or the same column, or both (same position)). $P_{\mathcal{X}}$ knows the position X of one rook. $P_{\mathcal{Y}}$ knows the position Y of the other rook and wants to learn X. Formally,

$$S \stackrel{\text{def}}{=} \{((x_1, x_2), (y_1, y_2)) : 0 \leq x_1, x_2, y_1, y_2 \leq n-1 \text{ and either } x_1 = y_1 \text{ or } x_2 = y_2\} \ .$$

Consider one-way complexity first. If $P_{\mathcal{X}}$ assigns the same message to two different rook positions, (x_1, x_2) and (x_1', x_2'), then in the case that $P_{\mathcal{Y}}$'s rook position is (x_1, x_2') (or (x_1', x_2)), he will not know X. Hence

$$\hat{C}_1 = \lceil 2 \log n \rceil \ .$$

For every rook position Y known to $P_{\mathcal{Y}}$ there are $2n-1$ possible X positions, hence

$$\hat{\mu} = 2n - 1.$$

Therefore,

$$\hat{C}_\infty \geq \lceil \log(2n-1) \rceil,$$

and this many bits are needed even if $P_{\mathcal{X}}$ knew Y in advance.

A simple three-message protocol based on the league-problem shows that

$$\hat{C}_\infty \leq \hat{C}_3 \leq \lceil \log n \rceil + \lceil \log \log n \rceil + 1.$$

Using $\lceil \log n \rceil$ bits, $P_{\mathcal{X}}$ transmits the diagonal $d = (x_1 + x_2) \bmod n$ that his rook is on. Now, $P_{\mathcal{Y}}$ knows that (x_1, x_2) is either $(y_1, (d - y_1)_n)$ or $((d - y_2)_n, y_2)$. They use the two-message league-problem protocol to determine (x_1, x_2) while exchanging $\lceil \log \log n \rceil + 1$ bits. Thus, for this example,

$$\hat{C}_3 \leq \log \hat{\mu} + \log \log \hat{\mu} \leq \hat{C}_\infty + \log \hat{C}_\infty \ ,$$

providing another example of a balanced random pair where $\hat{C}_1 \approx 2\hat{C}_\infty$. □

We do not know of a two-message protocol for this example whose worst-case complexity is $\log n + o(\log n)$ bits.

4 Distributions with varying ambiguities

For some balanced pairs, $\mu(y)$ varies widely with y. For these pairs, the bound provided by Corollary 2, may not be very useful. We describe an efficient refinement for such cases.

Let $\ell_\phi(x,y)$ denote the number of bits transmitted under the protocol ϕ for the input (x,y). Consider a (not necessarily balanced) random pair (X,Y), possibly with widely varying $\mu(y)$. For every y there are $\mu(y)$ possible x values. Although $\ell_\phi(x,y)$ may be low for some x's, when averaged over the x's, it must be at least $\log\mu(y)$. Namely, for all y,

$$\frac{1}{\mu(y)}\sum_{x:(x,y)\in S}\ell_\phi(x,y)\geq\log\mu(y)\ .$$

Note that this holds true even if $P_{\mathcal{X}}$ knows that $Y=y$ in advance. For $\mu\in\mathcal{Z}$, let

$$S_{\mathcal{Y}}(\mu)\stackrel{\rm def}{=}\{y:\mu(y)=\mu\}$$

be the set of Y values with ambiguity μ. Clearly, for all feasible[5] μ,

$$\frac{1}{|S_{\mathcal{Y}}(\mu)|}\sum_{y\in S_{\mathcal{Y}}(\mu)}\frac{1}{\mu}\sum_{x:(x,y)\in S}\ell_\phi(x,y)\geq\log\mu\ .$$

The next theorem says that this average lower bound can be almost met. Its proof resembles that of Theorem 3 in [6], hence omitted.

Theorem 2 Let (X,Y) be a random pair. There is a four-message protocol ϕ such that for all feasible μ,

$$\frac{1}{|S_{\mathcal{Y}}(\mu)|}\sum_{y\in S_{\mathcal{Y}}(\mu)}\frac{1}{\mu}\sum_{x:(x,y)\in S}\ell_\phi(x,y)\leq\log\mu+4\log\log\mu\ .\qquad\square$$

Two, progressively stronger, statements are false in general.

1. For every (balanced) (X,Y) pair, there is a protocol ϕ such that for all y
$$\frac{1}{\mu(y)}\sum_{x:(x,y)\in S}\ell_\phi(x,y)\leq\log\mu(y)+o(\log\mu(y))\ .$$

2. For every (balanced) (X,Y) pair, there is a protocol ϕ such that for all y
$$\max\{\ell_\phi(x,y):(x,y)\in S\}\leq\log\mu(y)+o(\log\mu(y))\ .$$

To see some cases where these statements are false, consider a generalization of the league problem (Example 1) with t teams, μ of them playing in every game. (Again, $P_{\mathcal{Y}}$ knows the game and wants to learn the winner, known to $P_{\mathcal{X}}$.) If statement (1) were true, there would have been a protocol for that problem with worst-case complexity of at most $\mu\log\mu+o(\mu\log\mu)$. Results in [3] imply that every protocol for that problem must have worst-case complexity of at least $\log\log t+1$ bits. For $t\gg\mu$, these two conclusions are contradictory. An even larger discrepancy is derived when statement (2) is assumed to hold. Note that this problem is not balanced, but can be balanced easily.

[5] μ is *feasible* if $S_{\mathcal{Y}}(\mu)$ is nonempty.

5 Correlated files

Subsection 1.2 described the correlated-files problem and some of its potential applications. Essentially, X and Y are binary strings within edit-distance of at most α from each other, $P_{\mathcal{X}}$ knows X while $P_{\mathcal{Y}}$ knows Y and wants to learn X. When we want to emphasize the maximum distance between X and Y, we refer to the problem as the *α-edits problem.*

In the next subsection we apply results proven in the previous sections to obtain efficient interactive protocols for this problem. In Subsection 5.2 we discuss one-way protocols.

5.1 Interactive communication

First, we evaluate $\mu(y)$ — $P_{\mathcal{Y}}$'s ambiguity when his string is y. $P_{\mathcal{X}}$'s string, x, can be obtained from y by i insertions and d deletions where $i + d \leq \alpha$. In the special case where x can be obtained from y by insertions alone, we say that x is a *superstring* of y (and that y is a *substring* of x). The number of $(n-\alpha)$-bit substrings of an n-bit string y depends on y. For example 010 has three 2-bit substrings (01, 00, 10) whereas 000 has only one. Yet it is known that the number of $(n+\alpha)$-bit superstrings of y is the same for every n-bit string y:

Lemma 4 The number of $(n+\alpha)$-bit superstings of an n-bit string is $\sum_{i=0}^{\alpha}\binom{n+\alpha}{i}$. □

We use the lemma to estimate $P_{\mathcal{Y}}$'s ambiguity in the α-edits problem. Let $|y|$ denote the number of bits in a string y, then:

$$\binom{|y|+\alpha}{\alpha} \leq \sum_{k=0}^{\alpha}\sum_{i=0}^{k}\binom{|y|+k}{i} \leq \mu(y) \leq \sum_{k=0}^{\alpha} k \cdot \sum_{i=0}^{k}\binom{|y|+k}{i} \leq \alpha^3\binom{|y|+\alpha}{\alpha} . \tag{10}$$

Therefore, the number of bits $P_{\mathcal{X}}$ must transmit *if he knows Y in advance* is bounded by:

$$\log\binom{|y|+\alpha}{\alpha} \leq \lceil \log \mu(y) \rceil \leq \log\binom{|y|+\alpha}{\alpha} + 3\log\alpha . \tag{11}$$

In cases of interest, $|y|$ is (much) larger than α, hence this bound is tight. A simple protocol almost achieves this number: for each edit operation needed to convert y to x, $P_{\mathcal{X}}$ describes its nature (delete, insert 0, or insert 1) and location. The total number of bits transmitted under this protocol is at most $\alpha(\lceil\log(|Y|+\alpha)\rceil + 2)$. Note that the protocol assumes that $P_{\mathcal{X}}$ knows Y in advance.

The maximum ambiguity of the α-edits problem is infinite: there is no upper bound on the length of y, hence on $\mu(y)$. Therefore, all worst-case complexities are infinite too. For a more sensitive measure, let ϕ be a protocol for a random pair (X,Y) and define $l_\phi(x,y)$ to be the number of bits exchanged under ϕ for the input (x,y). For every $y \in S_{\mathcal{Y}}$, let

$$\hat{l}_\phi(y) \stackrel{\text{def}}{=} \max\{l_\phi(x,y) : (x,y) \in S\}$$

be the number of bits transmitted under ϕ in the worst case when $P_{\mathcal{Y}}$'s value is y. Clearly,

$$\hat{l}_\phi(y) \geq \lceil \log \mu(y) \rceil$$

and equality holds *if $P_{\mathcal{X}}$ knows Y in advance.* ϕ is *asymptotically efficient for every y*, or *efficient* for short, if for all $y \in S_{\mathcal{Y}}$,

$$\hat{l}_\phi(y) \leq \log \mu(y) + o(\log \mu(y)) \ .$$

Namely, if ϕ requires only marginally more bits than needed when $P_{\mathcal{X}}$ knows y in advance.

The discussion at the end of the last section implies that many (X,Y) pairs, even balanced ones, do not have efficient protocols: $\hat{l}_\phi(y)$ may have to be much larger than $\log \mu(y)$ for some y's. However, the next corollary shows that the correlated-files problem has an efficient three-message protocol (one-way protocols will be discussed in the next subsection).

Corollary 3 For every α, the α-edits problem has an efficient three-message protocol ϕ satisfying:

$$\hat{l}_\phi(y) \leq \log \mu(y) + 3 \log\log \mu(y) + \log \alpha + 13.$$

Proof: The lengths $|X|$ and $|Y|$ of the strings X and Y, are within α from each other. Hence, if $P_{\mathcal{X}}$ transmits $|X| \bmod (2\alpha+1)$, then $P_{\mathcal{Y}}$ can infer $|X|$. Thereafter, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ can use the three-message protocol of Corollary 2 for strings of length at most $|X|+\alpha$. $\square$

The next, numerical, example shows that the protocol implied by the Corollary is efficient not only asymptotically, but also for moderately-sized files.

Example 3 Suppose that Y is a 2^{23}-bit file (roughly a million characters) and that X can be derived from Y via 8,000 insertions and deletions of bits (say a thousand character changes). If $P_{\mathcal{X}}$ knew Y in advance, he would need to transmit $\lceil \log \mu(Y) \rceil$ bits in the worst case, which Inequality (11) evaluates as:

$$91,800 \leq \log \binom{2^{23}+8000}{8000} \leq \lceil \log \mu(Y) \rceil \leq \log \binom{2^{23}+8000}{8000} + 3\log(8000) + 1 \leq 91,845 \ .$$

The corollary shows that even when $P_{\mathcal{X}}$ does not know Y in advance, using three messages, he can convey X to $P_{\mathcal{Y}}$ using at most

$$\log \mu(Y) + 3 \log\log \mu(Y) + \log(8000) + 13 \leq \log \mu(Y) + 75 \text{ bits.} \qquad \square$$

Generalized edit distance

We can assume a more general setting than before. For example, in addition to inserting and deleting bits, a single edit operation (a mouse, or a key click) may move or complement contiguous segments of arbitrary length. Still, the number of operations needed to derive a string from another can be easily seen to be a 'distance' between the two.

Assume that X and Y are derived from a string Z using, respectively, α_1 and α_2 edit operations. As all the above operations are reversible, X can be derived from Y using at most $\alpha = \alpha_1 + \alpha_2$ edit operations. Every edit operation is characterized by a location in the string and a small number of bits, say 4, identifying the operation (delete, insert 0, insert 1, move, complement, etc.). If Y is of length $n \gg \alpha$, then the number of possible X sequences within α edit operations from Y is at most $2^{\alpha(\log(n+\alpha)+4)}$. This bound is not far from the actual number as *for every* Y there are at least $\binom{n+\alpha}{\alpha}$ possible X string derived by α insertions to Y. We can therefore apply Corollary 2 to derive a near-optimum protocol for this problem.

The theorem still holds if we allow the editor to delete arbitrary long strings. However, now, $\mu(Y)$ can be arbitrarily larger than the number of bits in Y (say Y was derived by erasing all of X) and so all complexity measures are unbounded. Note that in this case, even if $P_{\mathcal{X}}$ knew Y in advance, he would still need to transmit arbitrarily many (in the length of Y) bits to describe X.

5.2 One-way communication

Subsection 1.2 mentioned several reasons rendering one-way protocols more useful than interactive ones. However, the general results proved in Section 2 only imply one-way protocols requiring at most twice the number of bits needed when $P_{\mathcal{X}}$ knows Y in advance. We therefore investigate one-way protocols for correlated files in more detail.

A one-way protocol for the α-edits problem is a function $\phi : \{0,1\}^* \to \{0,1\}^*$ such that if x and x' are distinct strings obtained from the same y via at most α edits (insertions and/or deletions), then neither $\phi(x)$ nor $\phi(x')$ is a prefix[6] of the other. This guarantees that $P_{\mathcal{Y}}$ can tell when $P_{\mathcal{X}}$'s message ends and then deduce the value of X.

A special case of the α-edits problem is the *α-insertions problem* where X is obtained from Y by *exactly* α insertions. A one-way protocol for the α-insertions problem, or an

[6]The string 011 is a *prefix* of 011 and of 01101, hence if a string is not a prefix of another, they cannot, in particular, be equal.

α-*insertions protocol*, is therefore a mapping $\phi : \{0{,}1\}^* \rightarrow \{0{,}1\}^*$ such that if x and x' are distinct strings obtained from the same y by exactly α insertions, then neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other. To simplify the statement of the next lemma, we place an additional demand on the behavior of the protocol for strings of length less than α. We require that if $\beta < \alpha$ then $\phi(x)$ is not a prefix of $\phi(x')$ for all distinct $x, x' \in \{0{,}1\}^\beta$. Intuitively, this corresponds to allowing $P_{\mathcal{Y}}$ to have a string of "negative length" $\beta - \alpha$. As all β-bit strings can be obtained by α insertions to this string, the protocol must distinguish between them.

Lemma 5 Let α be a nonnegative integer. Any α-insertions protocol is also a β-insertions protocol for all $0 \leq \beta \leq \alpha$.

Proof: By definition. Let ϕ be an α-insertions protocol. If $|x| = |x'| < \beta$ then neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other by the additional requirement above. If x and x' can be obtained from a string y by β insertions, let y' be a string derived from y by deleting any $\alpha - \beta$ bits. Then x and x' are obtained from y' by α insertions and again neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other. $\square$

Conceptually, insertion protocols are simpler to construct than edit protocols. They need to handle only insertions and only a fixed number of them (unlike insertions, different number of edits may result in strings of the same length). The α-insertions problem is also more appealing analytically. Whereas Inequality (10) only bounds $\hat{\mu}$ for the α-edits problem, Lemma 4 says that for the α-insertions problem, all strings y have $\mu(y) = \sum_{i=0}^{\alpha} \binom{|y|+\alpha}{i}$.

The next Theorem shows that, in a sense, the two problems are equally difficult. It reduces the problem of constructing a one-way protocol that handles insertions and deletions to that of constructing a one-way protocol that handles only insertions.

Theorem 3 If the α-insertions problem has an efficient one-message protocol, then so does the α-edits problem.

Proof: Let ϕ be a one-message α-insertions protocol. Define

$$\psi(x) = (|x| \bmod (2\alpha + 1), \phi(x))$$

to be the protocol that, for every string x, transmits the binary representation of the length of x mod $(2\alpha + 1)$, followed by $\phi(x)$. We show that if x and x' are distinct strings that can be derived from a string y by at most α edits, then $\psi(x)$ is not a prefix of $\psi(x')$.

If $|x| \neq |x'|$ then $|x| \neq |x'| \bmod (2\alpha + 1)$, hence $\psi(x)$ and $\psi(x')$ have different beginnings. If $|x| = |x'|$ then there are i, i', d, d' such that:

1. $i + d \leq \alpha$ and x can be derived from y by i insertions and d deletions

2. $i' + d' \leq \alpha$ and x' can be derived from y by i' insertions and d' deletions

3. $i - d = i' - d'$ (because $|x| = |x'|$).

From (1) and (2), there is a sequence of d insertions and i deletions taking x into y and a sequence of i' insertions and d' deletions taking y into x'.

We can change the order of a sequence of insertions and deletions applied to a string without affecting the resulting string. We must only ensure that if in the original sequence of insertions and deletions, a bit is inserted and then deleted and this order is reversed in the new sequence, then in the new sequence we neither delete this (yet nonexistent) bit nor insert it later on. Therefore, there is $\beta \leq i + d'$ such that a sequence of β deletions followed by a sequence of β insertions takes x to x'. Let y be the sequence obtained from x after the β deletions. Both x and x' can be obtained from y by β insertions. From (3), $i + d' = i' + d$, hence $\beta \leq i + d' = \frac{1}{2}(i + d' + i' + d) \leq \alpha$. By Lemma 5, $\phi(x)$ is not a prefix of $\phi(x')$.

Regarding efficiency, ψ transmits only $\lceil \log(2\alpha + 1) \rceil = o(\log \binom{|y|+\alpha}{\alpha})$ bits more than ϕ. The maximum ambiguity of the α-edits problem is higher than that of the α-insertions problem. Hence, if ϕ is efficient, so is ψ. $\square$

In view of the theorem we consider only insertions protocols. First, we discuss their efficiency. Let ϕ be an α-insertions protocol. Define

$$\ell_\phi(n) \stackrel{\text{def}}{=} \max\{|\phi(x)| : x \in \{0,1\}^{n+\alpha}\} \ .$$

to be the maximum number of bits $P_{\mathcal{X}}$ transmits according to ϕ when y is n-bit long. Lemma 4 said that every n-bit string has $\sum_{i=0}^{\alpha} \binom{n+\alpha}{i}$ superstrings of length $n + \alpha$. Hence,

$$\ell_\phi(n) \geq \log \sum_{i=0}^{\alpha} \binom{n+\alpha}{i}$$

and ϕ is efficient if

$$\lim_{n\to\infty} \frac{\ell_\phi(n)}{\log \sum_{i=0}^{\alpha} \binom{n+\alpha}{i}} = 1 \ .$$

Namely, if the number of bits transmitted for every y is asymptotically the same as that needed when $P_{\mathcal{X}}$ knows Y in advance.

We now consider efficient insertion protocols for some restricted cases. We begin with the most elementary of these protocols: one that identifies a single insertion (and, by arguments similar to the ones used to prove Theorem 3, a single deletion too).

Suppose that Y is an n-bit string and that X is an $(n+1)$-bit string obtained from Y by a single insertion. According to Lemma 4, for every possible value of Y, there are $n+2$ possible X's. Therefore, even if $P_{\mathcal{X}}$ knew Y in advance, he would have to transmit $\lceil\log(\hat{\mu})\rceil = \lceil\log(n+2)\rceil$ bits in the worst case. We describe a protocol that transmits exactly that many bits and does not require $P_{\mathcal{X}}$ to know Y in advance.

Lemma 6 Let $X \in \{0,1\}^{n+1}$ be a superstring of $Y \in \{0,1\}^n$. Then

$$\hat{C}_1 = \ldots = \hat{C}_\infty = \lceil\log\hat{\mu}\rceil = \lceil\log(n+2)\rceil.$$

Proof: View a string z as an integer sequence $z_1,\ldots,z_{|z|}$. Using $\lceil\log(n+2)\rceil$ bits, $P_{\mathcal{X}}$ transmits:

$$\phi(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n+1} ix_i \bmod (n+2)$$

(here $\phi(x)$ is a number rather than its binary representation). To verify that $P_{\mathcal{Y}}$ can deduce X from $\phi(X)$, we show that ϕ maps all superstrings of a given string into different numbers. For example, if Y is 010 then there are $3+2=5$ possible X sequences, and ϕ maps each one into a different integer mod 5. $0100 \to 2,\ 0010 \to 3,\ 1010 \to 4,\ 0110 \to 0,\ 0101 \to 1$.

In the general case, assume that y has k 'ones' (hence $n-k$ 'zeros') and consider the effect of a single insertion on the value of the function $f : \{0,1\}^* \to \mathcal{Z}^+$ defined by $f(z) = \sum_{i=1}^{|z|} iz_i$.

Assume first that x is derived from y by inserting a 'zero' between the ith and $(i+1)$st *rightmost* 'one's in y where $0 \le i \le k$. Each of the i 'one's to the right of the inserted bit is shifted one position to the right and so $f(x) = f(y) + i$.

Next, assume that x is derived from y by inserting a 'one' between the ith and $(i+1)$st *leftmost* 'zero's in y, where $0 \le i \le n-k$, and assume further that the 'one' was inserted between the jth and $(j+1)$st leftmost bits of y. It is easy to verify that, in that case, $f(x) = f(y) + (j+1) + ((n-j) - (n-k-i)) = f(y) + k + 1 + i$.

Therefore, if x is derived from y by inserting a 'zero' $f(y)$ increases by an integer ranging from 0 to k, while if x is derived from y by inserting a 'one' $f(y)$ increases by an integer ranging from $k+1$ to $n+1$. As $\phi(x)$ is $f(x) \bmod (n+2)$ it too is different for every supersequence x of y.

This one-insertion protocol is related to an insertion/deletion code of Varshamov and Tenengolt's [16]. This code consists of all n-bit codewords x with a fixed $\sum_{i=1}^{n} ix_i \bmod (n+2)$. We note however that an efficient code does not imply an efficient one-way protocol. An efficient protocol corresponds to a collection of disjoint efficient codes that cover $\{0,1\}^*$. $\square$

To derive a protocol for identifying more than a single insertion one could try to generalize the protocol above. In the sequel we suggest an alternative approach that may prove more tractable. It attempts to reduce the problem to one of exchanging strings within small Hamming distance from each other. We show that this approach works in a restricted case of insertions and deletions.

The *Hamming distance* $d_H(x, y)$ between two equal-length strings x and y is the number of bit locations where they differ. In the *Hamming-distance problem*, X and Y are n-bit sequences within a small Hamming-distance from each other. $P_{\mathcal{X}}$ knows X while $P_{\mathcal{Y}}$ knows Y and wants to learn X. Witsenhausen and Wyner [17] considered the problem in the context of video compression. The (independent, but essentially the same) solution described here is taken from [18] and was done with A. El Gamal and Y. Brandman, extending Example 5 in [4].

Example 4 $P_{\mathcal{X}}$ has an n-bit sequence X and $P_{\mathcal{Y}}$ has an n-bit sequence Y. The Hamming distance between X and Y is at most t. How many bits must be transmitted in the worst case for $P_{\mathcal{Y}}$ to learn X?

$P_{\mathcal{Y}}$'s ambiguity is the same for every $y \in \{0,1\}^n$:

$$\mu(y) = \sum_{i=1}^{t} \binom{n}{i} .$$

Therefore,

$$\hat{C}_1 \geq \hat{C}_\infty \geq \left\lceil \log \sum_{i=1}^{t} \binom{n}{i} \right\rceil \geq t(\log n - \log t) .$$

Using error-correcting codes, we show that

$$\hat{C}_1 \leq t \lceil \log(n+1) \rceil . \tag{12}$$

We are mostly interested in cases where $t \ll n$. For these cases, the two bounds are quite close. We assume that the reader is familiar with basic results concerning linear error-correcting codes. The protocol is based on an (n, k) linear t-error correcting code C (we will determine k later). $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ agree a priori on a parity-check matrix H for C. When $P_{\mathcal{X}}$ is given X and $P_{\mathcal{Y}}$ is given Y such that $d_H(X, Y) \leq t$, they execute the following protocol.

$P_{\mathcal{X}}$ transmits XH^T, the $(n-k)$-bit syndrome of X. $P_{\mathcal{Y}}$ computes $XH^T - YH^T$, the syndrome of $(X - Y)$. He finds an n-bit sequence Z with Hamming weight $\leq t$ such that $ZH^T = (X - Y)H^T$ and decides that X is $Z + Y$.

To prove that $P_{\mathcal{Y}}$ always determines X correctly, we now show that $Z = (X - Y)$. Namely,
(1) $(X - Y)$ has Hamming weight $\leq t$.
(2) No other n-bit sequence with Hamming weight $\leq t$ has the syndrome $(X - Y)H^T$.

(1) holds as $d_H(X, Y) \leq t$. To prove (2), notice that if an n-bit sequence s has Hamming weight $\leq t$ and syndrome $(X - Y)H^T$ then $(X - Y) - s$ has Hamming weight $\leq 2t$ and syndrome 0; that is, $(X - Y) - s$ is a codeword in C with Hamming weight $\leq 2t$. Since C is t-error correcting, it has minimum distance $\geq 2t + 1$, hence $s = X - Y$.

The number of bits transmitted under this protocol is $n - k$. To prove a low one-message complexity we need a t-error correcting code with low redundancy. The BCH bound (e.g., [19], or Theorem 9.2 in [20]) guarantees that if $n + 1$ is a power of two, then there is a linear t-error correcting code with $n - k \leq t \log(n + 1)$ implying Inequality (12).

□

To reduce the edit-distance problem to the Hamming-distance problem we need two mappings:

1. $\xi : \{0,1\}^n \to \{0,1\}^m$

2. $\zeta : \{0,1\}^{n+\alpha} \to \{0,1\}^m$

(the value of m will be discussed later) such that for all $y \in \{0,1\}^n$ and for all $x \neq x' \in \{0,1\}^{n+\alpha}$ that are superstrings of y:

1. $\zeta(x) \neq \zeta(x')$

2. $d_H(\zeta(x), \xi(y)) \leq \alpha$.

If ξ and ζ exist, we can construct a one-message protocol for identifying α insertions as follows. $P_{\mathcal{X}}$ computes $\zeta(x)$ and $P_{\mathcal{Y}}$ computes $\xi(y)$. Condition (2) above guarantees that the Hamming distance between $\zeta(x)$ and $\xi(y)$ is at most α. By the last example, $P_{\mathcal{X}}$ can convey $\zeta(x)$ to $P_{\mathcal{Y}}$ while transmitting $\alpha\lceil\log m\rceil$ bits. Condition (1) above ensures that x is the only superstring of y mapped by ζ into $\zeta(x)$, hence $P_{\mathcal{Y}}$ can determine x.

The number of bits transmitted by the protocol is $\alpha\lceil\log m\rceil$ whereas the number of bits needed if $P_{\mathcal{X}}$ knows Y in advance is $\log\mu(y) = \log\left(\sum_{i=0}^{\alpha}\binom{n+\alpha}{i}\right)$. For the protocol to be efficient, m must grow as $(n + \alpha)^{1+o(1)}$ (say, $m = (n + \alpha)\log(n + \alpha)$).

The smallest value m can attain is $n + \alpha$: the number of $(n + \alpha)$-bit superstrings of y is $\sum_{i=0}^{\alpha}\binom{n+\alpha}{i}$ and according to conditions above, this number must be smaller than $\sum_{i=0}^{\alpha}\binom{m}{i}$, the number of $(n+\alpha)$-bit strings within Hamming distance α from the m-bit string $\xi(y)$. Hence, at least in terms of 'first-order' cardinalities, $m = n+\alpha$ may be possible. So far, ζ and ξ with $m = n+\alpha$ are known only in very special cases: all n and $\alpha = 1$; all α and $n = 1$; and, due to J. Reeds, $n = \alpha = 2$.

We do not know whether ξ and ζ exist in general (even with $m > n+\alpha$). However, following is a construction for a restricted case where x is obtained from y by α edits that *do not introduce new runs or destroy existing ones.* This restriction severely limits the sequences that can be obtained from y. For example, the string 0 cannot be transformed into the string 1 without destroying a run, and 010000 can be transformed into 000010 by four insertions and deletions, but six are needed if no runs are to be eliminated or introduced.

We demonstrate ξ and ζ for this *restricted-edit problem* by converting the strings x and y known to $P_\mathcal{X}$ and $P_\mathcal{Y}$ into integer sequences within L_1 distance of at most α from each other. In turn, this L_1-*distance problem* can be reduced via a Gray code to the Hamming-distance problem thus exhibiting ξ and ζ.

Represent a string z with r runs[7] as an r-element integer sequence whose ith element is the length of the ith run in z. For example, 010000 is represented as $1, 1, 4$. If x can be obtained from y by insertions and deletions that do not introduce or destroy runs, then x and y have the same number of runs and both start with the same bit. Therefore, given y, there is a one-to-one correspondence between x and its representation above. It follows that the restricted-edit problem is almost equivalent to the following L_1-distance problem.

Example 5 let r and α be positive integers and let $X_1, \ldots, X_r$ and $Y_1, \ldots, Y_r$ be integers such that

$$\sum_{i=1}^{r} |X_i - Y_i| \leq \alpha.$$

$P_\mathcal{X}$ knows $X = (X_1, \ldots, X_r)$ while $P_\mathcal{Y}$ knows $Y = (Y_1, \ldots, Y_r)$ and wants to learn X. How many bits must be transmitted in the worst case?

It is easy to verify that for all $y = (y_1, \ldots, y_r)$,

$$\mu(y) = 1 + \sum_{i=1}^{\alpha} \sum_{j=1}^{\min\{i,r\}} 2^j \binom{r}{j} \binom{i-1}{j-1} .$$

Using Gray- and error-correcting codes, we show that

$$\hat{C}_1 \leq \alpha \lceil \log(1 + r\log(2\alpha+1)) \rceil .$$

[7] A *run* in a string is a contiguous sequence of 'zero's or 'one's.

The bound is asymptotically tight and the implied protocol is efficient whenever $\alpha/r \to 0$.

Let

$$\beta \stackrel{\text{def}}{=} \lceil \log(2\alpha+1) \rceil \ .$$

A β-bit Gray code is a sequence $g_0,\ldots,g_{2^\beta-1}$ of β-bit strings such that for[8] $i = 0,\ldots,2^\beta-1$:

$$d_H(g_i, g_{i+1}) = 1 \ .$$

Simple constructions of Gray codes are known for every positive β. Map the sequence $x_1,\ldots,x_r$ into the $(r\beta)$-bit string $g_{x_1} g_{x_2} \cdots g_{x_r}$ by concatenating the corresponding Gray-code strings (again, indices are interpreted modulo 2^β). Map $y_1,\ldots,y_r$ into $g_{y_1} g_{y_2} \cdots g_{y_r}$ similarly. Simple properties of Gray codes guarantee that

$$d_H(g_{x_i}, g_{y_i}) \le |x_i - y_i| \ .$$

Hence,

$$d_H(g_{x_1} g_{x_2} \cdots g_{x_r}\ ,\ g_{y_1} g_{y_2} \cdots g_{y_r}) \le sum_{i=1}^{r} |x_i - y_i| \le \alpha \ .$$

Example 4 implies that $P_{\mathcal{X}}$ can transmit $\alpha\lceil\log(r\beta)\rceil$ bits enabling $P_{\mathcal{Y}}$ to learn $g_{x_1} g_{x_2} \cdots g_{x_r}$. By choice of β, two integers that are mapped into the same Gray sequence must be at least $2\alpha + 1$ apart, hence, $P_{\mathcal{X}}$ can infer $x_1,\ldots,x_r$. □

We apply the example to prove

Corollary 4 The α-restricted-edits problem has an efficient one-way protocol.

Proof: The length of X can be transmitted using $\lceil\log(2\alpha+1)\rceil$ bits. Thereafter, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ can use the protocol given in the example. The total number of bits, transmitted for X is:

$$\alpha\lceil\log(|X|\log(2\alpha+1))\rceil + \lceil\log(2\alpha+1)\rceil = \alpha\log|X| + o(\alpha\log|X|) \ .$$

For fixed α,

$$log(\mu(y)) = \alpha\log|X| - o(\alpha\log|X|) \ ,$$

hence the protocol is efficient. Note that by invoking the example, we implicitly used the fact that the restricted-edits problem has ξ and ζ with range-dimension $m = |X|\lceil\log(2\alpha+1)\rceil$. □

[8] In this equation and below, Gray-sequence indices are taken modulo 2^β, hence g_{2^β} stands for g_0.

6 Open Problem

The main open problem suggested by this paper is the construction of efficient one-way insertion protocols. As shown in Theorem 3, such protocols imply efficient one-way protocols for the edit-distance problem, hence may have practical applications. Two possible directions towards such a construction are: (1) extension of the single-insertion protocol of Lemma 6; (2) construction of the functions ξ and ζ discussed in the last section for additional values of n and α.

Acknowledgment

I thank Jeff Kahn for his help in formulating and proving Lemma 2.

References

[1] H. Abelson. Lower bounds on information transfer in distributed computations. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, 1978.

[2] A.C. Yao. Some complexity questions related to distributive computing. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 209–213, 1979.

[3] A. Orlitsky. Worst-case interactive communication I: Two messages are almost optimal. *IEEE Transactions on Information Theory*, 36(5):1111–1126, September 1990.

[4] A. El Gamal and A. Orlitsky. Interactive data compression. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, pages 100–108, 1984.

[5] A. Orlitsky. Worst-case interactive communication II: Two messages are not optimal. *IEEE Transactions on Information Theory*, 37(4):995–1005, July 1991.

[6] A. Orlitsky. Average-case interactive communication. *IEEE Transactions on Information Theory*, 38(4):1534–1547, July 1992.

[7] C.H. Papadimitriou and M. Sipser. Communication complexity. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 196–200, 1982.

[8] P. Duris, Z. Galil, and G. Schnitger. Lower bounds on communication complexity. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 81–91, 1984.

[9] N. Nisan and A. Wigderson. Rounds in communication complexity revisited. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, 1991.

[10] H. Witsenhausen. The zero-error side information problem and chromatic numbers. *IEEE Transactions on Information Theory*, 22(5):592–593, September 1976.

[11] M.L. Fredman, J. Komlós, and E. Szemeredi. Storing a sparse table with $o(1)$ worst case access time. *Journal of the Association for Computing Machinery*, 31:538–544, 1984.

[12] M.L. Fredman and J. Komlós. On the size of separating systems and families of perfect hash functions. *SIAM Journal on Algebraic and Discrete Methods*, 5(1):61–68, 1984.

[13] J. Friedman. Constructing $o(n \log n)$ size monotone formula for the kth threshold function on n variables. *SIAM J Computing*, 15:641–654, 1986.

[14] J. Körner. Fredman-Komlós bounds and information theory. *SIAM. J. Alg. Disc. Meth.*, 7:560–570, 1986.

[15] J. Spencer. *Ten Lectures on the Probabilistic Method.* SIAM, 1987.

[16] V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information transmission*, 1(1):8–17, 1965.

[17] H. Witsenhausen and A.D. Wyner. Interframe coder for video signals, 1980. United States Patent number 4,191,970.

[18] A. Orlitsky. *Communication Issues in Distributed Communication.* PhD thesis, Stanford University, 1986.

[19] R.E. Blahut. *Theory and Practice of Error Control Codes.* Eddison and Wesley, 1984.

[20] W.W. Peterson and E.J. Weldon. *Error-Correcting Codes.* MIT Press, second edition, 1972.