Average-case interactive communication*

Alon Orlitsky
AT&T Bell Laboratories

March 22, 1996

Abstract

X and Y are random variables. Person $P_{\mathcal{X}}$ knows X, Person $P_{\mathcal{Y}}$ knows Y, and both know the joint probability distribution of the pair (X,Y). Using a predetermined protocol, they communicate over a binary, error-free, channel in order for $P_{\mathcal{Y}}$ to learn X. $P_{\mathcal{X}}$ may or may not learn Y. How many information bits must be transmitted (by both persons) on the average?

At least $H(X|Y)$ bits must be transmitted and $H(X)+1$ bits always suffice[1]. If the support set of (X,Y) is a Cartesian product of two sets, then $H(X)$ bits must be transmitted. If the random pair (X,Y) is uniformly distributed over its support set, then $H(X|Y)+3\log(H(X|Y)+1)+17$ bits suffice. Furthermore, this number of bits is achieved when $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ exchange four messages (sequences of binary bits).

The last two results show that when the *arithmetic* average number of bits is considered: (1) there is no asymptotic advantage to $P_{\mathcal{X}}$ knowing Y in advance; (2) four messages are asymptotically optimum. By contrast, for the *worst-case* number of bits: (1) communication can be significantly reduced if $P_{\mathcal{X}}$ knows Y in advance; (2) it is not known whether a constant number of messages is asymptotically optimum.

These results extend the work of El Gamal and Orlitsky [1] on the number of bits that must be transmitted when $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ wish to exchange X and Y. They have several implications for a problem of Slepian and Wolf [2] where $P_{\mathcal{X}}$ knows a random variable X, $P_{\mathcal{Y}}$ knows a random variable Y, and each communicates with a third person P who wants to learn both X and Y.

*To appear in the IEEE Transactions on Information Theory, September 1992. This version printed, March 22, 1996.

[1] $H(X)$ is the entropy of X and $H(X|Y)$ is the conditional entropy of X given Y. Entropies are binary.

1 Introduction

1.1 The problem

Consider two *communicators*: an *informant* $P_{\mathcal{X}}$ having a random variable X and a *recipient* $P_{\mathcal{Y}}$ having a, possibly dependent, random variable Y. Both communicators want the recipient, $P_{\mathcal{Y}}$, to learn X with no probability of error, whereas the informant, $P_{\mathcal{X}}$, may or may not learn Y. To that end they communicate over an error-free channel. How many information bits must be transmitted on the average?

This problem is a variation on a scenario considered by El Gamal and Orlitsky [1] where both communicators, not just $P_{\mathcal{Y}}$, want to learn the other's random variable. As elaborated on below, some of our results either follow from or extend results therein.

We assume that the communicators alternate in transmitting *messages*: finite sequences of bits. Messages are determined by an agreed-upon, deterministic, protocol. A formal definition of protocols for the current model is given in [3]. Essentially, a *protocol for* (X,Y) (i.e., a protocol for transmitting X to a person who knows Y) guarantees that the following properties hold. (1) Separate transmissions: each message is based on the random variable known to its transmitter and on previous messages. (2) Implicit termination: when one communicator transmits a message, the other knows when it ends, and when the last message ends, both communicators know that communication has ended. (3) Correct decision: when communication ends, the recipient, $P_{\mathcal{Y}}$, knows X.

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 *average complexity* of the protocol is the expected number of bits it requires both communicators to transmit (expectation is taken over all inputs). $\bar{C}_m(X|Y)$, the *m-message average complexity* of (X,Y), is the minimum average complexity of an m-message protocol for (X,Y). It is the minimum average number of bits transmitted by both communicators using a protocol that never exchanges more than m messages. Since empty messages are allowed, $\bar{C}_m(X|Y)$ is a decreasing function of m bounded below by 0. We can therefore define $\bar{C}_\infty(X|Y)$, the *unbounded-message complexity of* (X,Y), to be the limit of $\bar{C}_m(X|Y)$ as $m \to \infty$. It is the *minimum* number of bits that must be transmitted on the average for $P_{\mathcal{Y}}$ to know X, even if no restrictions are placed on the number of messages exchanged. Note that these complexity measures are not asymptotic — the minimum is always achieved by some finite number of messages and the subscript ∞

is used for notational convenience. In summary, for all (X, Y) pairs,

$$\bar{C}_1(X|Y) \geq \bar{C}_2(X|Y) \geq \bar{C}_3(X|Y) \geq \cdots \geq \bar{C}_\infty(X|Y) \ .$$

A precise definition of complexities is given in Subsection 1.5. Here, we demonstrate them with the following example.

Example 1 A league has t teams named $1, \ldots, t$. Every week two random teams play each other. The outcome of the match is announced over the radio. All announcements have the same format: "The match between team I and team J was won by team K" where $1 \leq I < J \leq t$ and K is either I or J. The distribution of games and winners is uniform:

$$\Pr(k, (i, j)) = \begin{cases} \frac{1}{n(n-1)} & \text{if } 1 \leq i < j \leq t \text{ and } k \in \{i, j\}, \\ 0 & \text{otherwise.} \end{cases}$$

One day, while $P_{\mathcal{Y}}$ listens to a match announcement, $P_{\mathcal{X}}$ grabs the radio from him. Consequently, $P_{\mathcal{Y}}$ hears the first part of the announcement: "The match between team I and team J" and $P_{\mathcal{X}}$ hears the second part: "was won by team K."

$P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ agree that $P_{\mathcal{Y}}$ should know the winner. They are looking for a protocol that, on the average, requires few transmitted bits.

If no interaction is allowed, $P_{\mathcal{X}}$ has to send a single message enabling $P_{\mathcal{Y}}$ to uniquely determine the winner. This message is based solely on the winner (for that is all $P_{\mathcal{X}}$ knows). Let $\sigma(i)$ be the message sent by $P_{\mathcal{X}}$ when he hears "was won by team i." If the messages $\sigma(i)$ and $\sigma(j)$ are the same for $i \neq j$, then in the event of a match between teams i and j, $P_{\mathcal{Y}}$ cannot tell who the winner is. Also, if $\sigma(i)$ is a prefix of $\sigma(j)$ for $i \neq j$ then in the event of a match between i and j $P_{\mathcal{Y}}$ does not know when the message ends. Therefore, the messages $\sigma(1), \ldots, \sigma(t)$ must all be different and none can be a proper prefix of another. Hence the average-case one-way complexity satisfies

$$H(K) \leq \bar{C}_1(K|I, J) \leq H(K) + 1 \ .$$

where the upper bound derives from the one-way protocol where $P_{\mathcal{X}}$ transmits the Huffman code representation of K. In our case, K is uniformly distributed over $\{1, \ldots, t\}$, hence

$$\log t \leq \bar{C}_1(K|I, J) \leq \log t + 1 \ .$$

A two-message protocol that requires only $\lceil \log \log t \rceil + 1$ bits in the worst case was described in [3]. $P_{\mathcal{Y}}$ considers the binary representations $I_1, \ldots, I_{\lceil \log t \rceil}$ and $J_1, \ldots, J_{\lceil \log t \rceil}$ of

I and J. Since $I \neq J$, there must be a first bit location L where the binary representations differ: $I_L \neq J_L$. Using $\lceil \log \log t \rceil$ bits, $P_{\mathcal{Y}}$ transmits L. $P_{\mathcal{X}}$ responds by transmitting K_L – the L'th bit in the binary representation of the winning team K. The total number of bits exchanged is $\lceil \log \log t \rceil + 1$. It was also shown that no other protocol requires fewer bits in the worst case.

M. Costa [4] used the following protocol to reduce the average number of bits transmitted by $P_{\mathcal{Y}}$. Sequentially over ℓ, $P_{\mathcal{Y}}$ considers the ℓth bit of I and J. He transmits 0 if $I_\ell = J_\ell$ and transmits 1 if $I_\ell \neq J_\ell$, stopping after the first transmitted 1. Assume, for simplicity, that t is a power of two. For $\ell = 1, \ldots, \log t$, the probability that the first bit location where I and J differ is ℓ is:

$$\Pr(L = \ell) = \frac{t}{t-1} \cdot \frac{1}{2^\ell} \;.$$

Therefore, the expected number of bits transmitted by $P_{\mathcal{Y}}$ is

$$\frac{t}{t-1} \sum_{l=1}^{\log t} \frac{\ell}{2^\ell} = \frac{t}{t-1} \cdot \frac{2t - \log t - 2}{t} = 2 - \frac{\log t}{t-1}.$$

Including the bit transmitted by $P_{\mathcal{X}}$, we obtain

$$\bar{C}_2(K|I,J) \leq 3 - \frac{\log t}{t-1} \;. \qquad \Box$$

Several other examples are described in [1]. They are concerned with communicators $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ who want to learn each other's random variable, but can be easily modified to our case where only $P_{\mathcal{Y}}$ wants to learn X. We note that in all the examples in [1] a single message is optimum: it achieves the minimum number of bits. Yet in Example 1 above, a single message requires $\lceil \log t \rceil$ bits on the average while two messages require at most three bits.

1.2 Results

In Section 2, we prove that for all (X, Y) pairs,

$$H(X|Y) \leq \bar{C}_\infty(X|Y) \leq \ldots \leq \bar{C}_1(X|Y) \leq H(X) + 1 \tag{1}$$

where $H(X)$ is the binary entropy of X and $H(X|Y)$ is the conditional binary entropy of X given Y. These bounds are not tight as $H(X|Y)$ can be much smaller than $H(X)$. However, Sections 3 and 4 show that, in the following sense, they are the tightest expressible in terms of Shannon entropies.

The *support set* of (X, Y) is

$$S_{X,Y} \stackrel{\text{def}}{=} \{(x, y) : p(x, y) > 0\} \ ,$$

the set of all possible inputs. The support set is a *Cartesian product* if $S_{X,Y} = A \times B$ for some sets A and B. In Section 3 we show that if $S_{X,Y}$ is a Cartesian product then the upper bound in (1) is essentially tight:

$$H(X) \leq \bar{C}_\infty(X|Y) \leq \ldots \leq \bar{C}_1(X|Y) \leq H(X) + 1 \ . \tag{2}$$

The pair (X, Y) is *uniformly distributed* (or *uniform*) if all possible values of (X, Y) are equally likely:

$$p(x, y) = \begin{cases} \frac{1}{|S_{X,Y}|} & \text{if } (x, y) \in S_{X,Y}, \\ 0 & \text{if } (x, y) \notin S_{X,Y}. \end{cases}$$

In Section 4 we show that whenever (X, Y) is uniformly distributed, the lower bound of Inequality (1) can almost be achieved:

$$H(X|Y) \leq \bar{C}_\infty(X|Y) \leq \ldots \leq \bar{C}_4(X|Y) \leq H(X|Y) + 3\log\left(H(X|Y) + 1\right) + 17. \tag{3}$$

When (X, Y) is uniformly distributed, the average number of bits transmitted is simply the number of bits transmitted for an input (x, y), *arithmetically* averaged over all inputs in $S_{X,Y}$. The last inequality shows that whenever this arithmetic average number of bits is considered:

1. No asymptotic[2] advantage is gained when $P_\mathcal{X}$ knows Y in advance: roughly the same number of bits is required whether or not $P_\mathcal{X}$ knows Y before communication begins.

2. While Example 1 showed that one message may require arbitrarily more bits than the minimum necessary, four messages require only negligibly more bits than the minimum.

We note that, with more complicated proofs, the constants 3 and 17 in Inequality (3) can be reduced. The following example illustrates Inequalities (2) and (3) in two simple cases.

Example 2 (Modified from [1] to fit our model.) Consider two random pairs.

1. (X_1, Y_1) is distributed over $\{1, \ldots, n\} \times \{1, \ldots, n\}$ according to

$$p_1(x, y) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{n} & \text{if } x = y, \\ 0 & \text{if } x \neq y. \end{cases}$$

[2]We remark, again, that the results hold for all random pairs. Only the conclusions are "asymptotic" because then $3\log(H(X|Y) + 1) + 17$ is negligible with respect to $H(X|Y)$.

2. Let $\epsilon > 0$ be arbitrarily small. (X_2, Y_2) is distributed over $\{1,\ldots,n\} \times \{1,\ldots,n\}$ according to

$$p_2(x,y) \stackrel{\text{def}}{=} \begin{cases} \frac{1-\epsilon}{n} & \text{if } x = y, \\ \frac{\epsilon}{n(n-1)} & \text{if } x \neq y. \end{cases}$$

The probability distributions underlying the two random pairs are illustrated in Figure 1. Being so close to each other, the random pairs have similar entropies:

$$H(X_2) \approx H(X_1) = \log n \ ,$$

and

$$H(X_2|Y_2) \approx H(X_1|Y_1) = 0 \ .$$

Their m-message complexities, however, are quite different. The pair (X_1, Y_1) is trivially uniformly distributed. Hence Inequality (3) implies[3] that the average complexities are close to the conditional entropy $H(X_1|Y_1) = 0$. Indeed, if $P_{\mathcal{Y}}$ knows Y_1, he also knows X_1, hence

$$\bar{C}_\infty(X_1|Y_1) = \ldots = \bar{C}_1(X_1|Y_1) = 0 \ .$$

On the other hand, S_{X_2,Y_2} is a Cartesian product, hence Inequality (2) implies that all average complexities are about $H(X_2)$:

$$\log n \leq \bar{C}_\infty(X_2|Y_2) \leq \ldots \leq \bar{C}_1(X_2|Y_2) \leq \log n + 1 \ . \qquad \Box$$

1.3 Related results

El Gamal and Orlitsky [1] consider the average number of transmitted bits required when *both* communicators want to know the other's random variable. It is shown that $H(X|Y) + H(Y|X)$ transmitted bits are always needed and that $H(X,Y) + 2$ bits always suffice. If $S_{X,Y}$ is a Cartesian product then $H(X,Y)$ bits are needed, while if (X,Y) is uniform over its support set, then $H(X|Y) + H(Y|X) + 3.1 \log\log |S_{X,Y}| + c$ bits always suffice.

Both bounds expressed in Inequality (2) and the lower bound in Inequality (3) follow from easy modifications of the corresponding results in [1]. But the upper bound of Inequality (3) does not. The new upper bound shows that the exchange of X and Y can be performed in two stages with no asymptotic loss of efficiency. First, $P_{\mathcal{X}}$ conveys X to $P_{\mathcal{Y}}$, then $P_{\mathcal{Y}}$ conveys Y to $P_{\mathcal{X}}$. More importantly, the new bound reduces the number of transmitted bits even

[3]This is an extremely degenerate example of a uniform pair, hence does not capture the essence of the upper bound.

after breaking the problem into these two stages. The extraneous term in [1] is proportional to $\log\log|S_{X,Y}|$ bits which can be arbitrarily higher than $H(X|Y)+H(Y|X)$, the minimum number necessary (e.g., $\log\log n$ bits versus none for the pair (X_1,Y_1) in Example 2). In the new upper bound the extraneous term is proportional to $\log\min\{H(X|Y),H(Y|X)\}$ bits which is negligible in comparison with $H(X|Y)+H(Y|X)$.

Additionally, the results presented here, apply even when only one of the random variables X and Y needs to be conveyed. This allows for an application of our results to the single-event analog of a problem by Slepian and Wolf [2] that is described in the next subsection and analyzed in Section 5.

Another aspect of the problem is the number $\hat{C}_m(X|Y)$ of bits that must be transmitted in the *worst-case* for $P_{\mathcal{Y}}$ to learn X when only m messages are allowed. As shown in [3, 5]:

1. A single message may require exponentially more bits than the minimum number needed: for all (X,Y) pairs, $\hat{C}_1(X|Y)\le 2^{\hat{C}_\infty(X|Y)-1}$ with equality for some pairs.

2. With just two messages, the number of bits required is at most four times the minimum: for all (X,Y) pairs, $\hat{C}_2(X|Y)\le 4\hat{C}_\infty(X|Y)+3$.

3. Two messages are not always optimum: for some (X,Y) pairs, $\hat{C}_2(X|Y)\ge 2\hat{C}_\infty(X|Y)$.

4. For some (X,Y) pairs, the number of bits required when $P_{\mathcal{X}}$ knows Y in advance is logarithmically smaller than that $\hat{C}_\infty(X|Y)$. In the worst-case analog of Example 1, for instance, $\lceil\log\log t\rceil+1$ bits are required when (as we assume) $P_{\mathcal{X}}$ does not know the losing team in advance, whereas only one bit is needed if he does.

These results sharply contrast average- and worst-case complexities in at least two respects:

1. For average-case complexity, there is almost no difference between the number of bits required when $P_{\mathcal{X}}$ knows Y in advance and when he does not. For worst-case complexity there can be a logarithmic reduction in the number of bits needed if $P_{\mathcal{X}}$ knows Y before communication begins.

2. For average-case complexity, four messages are asymptotically optimum. For worst-case complexity, it is not known whether a constant number of messages are asymptotically optimum. Namely, whether there is an m such that for all (X,Y) pairs,

$$\hat{C}_m(X|Y)\le\hat{C}_\infty(X|Y)+o(\hat{C}_\infty(X|Y))\ .$$

We note that in the special case where the support set is *balanced*,[4] it has been recently shown [6] that three messages are asymptotically optimum for worst-case complexity.

A large discrepancy between the number of bits required with m- and $(m+1)$ messages occurs in worst-case communication complexity. There, see [7] for precise definitions, $P_{\mathcal{X}}$ knows X while $P_{\mathcal{Y}}$ knows Y and wants to learn the value of a predetermined boolean function $f(X,Y)$. A succession of papers [8, 9, 10] has shown that for every number m of messages, there is a boolean function f whose m-message worst-case complexity is exponentially higher than its $(m+1)$-message worst-case complexity. Furthermore, m-message complexity is never more than exponentially higher than $(m+1)$-message complexity. By contrast, in average-case interactive communication there may be an unbounded discrepancy between $\bar{C}_1(X|Y)$ and $\bar{C}_2(X|Y)$ (e.g., $\log t$ versus three bits in the league problem). Yet four messages are asymptotically optimum hence m messages, for any m, are at most negligibly better than four. Although the corresponding result is not known to hold for worst-case interactive communication, it can still be distinguished from communication complexity: as stated above, two messages require at most four times the minimum communication.

1.4 The Slepian-Wolf problem

Consider a variation on the model discussed so far. As before, $P_{\mathcal{X}}$ knows a random variable X and $P_{\mathcal{Y}}$ knows a random variable Y. Now, however, there is a third person P who wants to learn both X and Y. $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ may or may not learn each other's random variable.

Communication is conducted over two error-free channels. As illustrated in Figure 2, one channel connects $P_{\mathcal{X}}$ and P, the other connects $P_{\mathcal{Y}}$ and P. $P_{\mathcal{X}}$ and P can communicate back and forth over the channel connecting them, as can $P_{\mathcal{Y}}$ and P, but $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ cannot communicate directly (although they can do so via P). Each bit, transmitted over either channel, in either direction, is counted; hence a bit transmitted from $P_{\mathcal{X}}$ to $P_{\mathcal{Y}}$ via P is counted twice.

The three communicators agree on a protocol that, for every input, determines a sequence of exchanged messages, hence ascribes an expected number of bits transmitted (in both directions) over each of the two channels. To save space, we skip the formal definition of a protocol.

[4]That is, the maximum number of X values possible with a given Y value is about the same as the maximum number of Y values possible with a given X value.

$$
\begin{array}{c|c|c|c|c|c|}
 & 1 & 2 & 3 & \cdot & \text{n} \\
\hline
1 & \frac{1}{n} & 0 & 0 & \cdot & 0 \\
\hline
2 & 0 & \frac{1}{n} & 0 & \cdot & 0 \\
\hline
3 & 0 & 0 & \frac{1}{n} & \cdot & 0 \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
n & 0 & 0 & 0 & \cdot & \frac{1}{n} \\
\hline
\end{array}
\qquad\qquad
\begin{array}{c|c|c|c|c|c|}
 & 1 & 2 & 3 & \cdot & \text{n} \\
\hline
1 & \frac{1-\epsilon}{n} & \delta & \delta & \cdot & \delta \\
\hline
2 & \delta & \frac{1-\epsilon}{n} & \delta & \cdot & \delta \\
\hline
3 & \delta & \delta & \frac{1-\epsilon}{n} & \cdot & \delta \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
n & \delta & \delta & \delta & \cdot & \frac{1-\epsilon}{n} \\
\hline
\end{array}
$$

$$p_1 \qquad\qquad\qquad\qquad p_2$$

Figure 1: Similar random pairs with different complexities; $\delta \stackrel{\text{def}}{=} \frac{\epsilon}{n(n-1)}$.

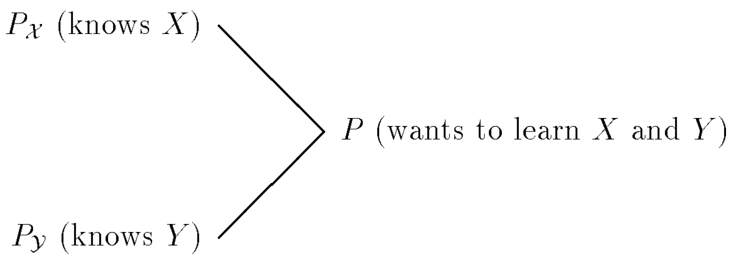


Figure 2: Set-up for the Slepian-Wolf problem

A pair $(C^{\mathcal{X}}, C^{\mathcal{Y}})$ of real numbers is *achievable* for (X, Y) if there is a protocol for (X, Y) such that the expected number of bits transmitted over the channel connecting $P_{\mathcal{X}}$ and P is at most $C^{\mathcal{X}}$ and the expected number of bits transmitted over the channel connecting $P_{\mathcal{Y}}$ and P is at most $C^{\mathcal{Y}}$. The set of all achievable $(C^{\mathcal{X}}, C^{\mathcal{Y}})$ pairs is the *achievable region*. Given the probability distribution of (X, Y), the problem is to determine the achievable region.

In Section 5 we consider the achievable regions in three cases.

General pairs. Corollary 3 shows that for all (X, Y) pairs, the achievable region contains the double-dashed area in Figure 3a and is roughly[5] contained in the single-dashed area.

Cartesian-product pairs. Corollary 4 shows that if the support set of (X, Y) is a Cartesian product then the achievable region is roughly the shaded area of Figure 3b.

Uniform pairs. Corollary 5 shows that if (X, Y) is uniform over its support set then the achievable region is roughly the shaded area of Figure 3c.

This last region is the familiar achievable region of the Slepian-Wolf problem investigated in [2]. It shows that the Slepian-Wolf region can be achieved even if: (1) there is only one pair of random variables (rather than a sequence of independent, identically distributed pairs) and (2) no-errors are allowed. All we assume is that: (1) the pair is uniformly distributed over its support set and (2) interaction (transmissions from P to $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$) is permitted.

Section 5 also describes the Slepian-Wolf problem and discusses some implications of our results for it. It is shown, for example, that if, in the original Slepian-Wolf problem, the support set is a Cartesian product and if no errors are allowed, then, even with interaction, the achievable region is that of Figure 5b (see Section 5), rather that of Figure 5a. Therefore, allowing errors increases the achievable region. This contrasts with transmission of a single random sequence ($P_{\mathcal{X}}$ knows X, but $P_{\mathcal{Y}}$ does not know Y) where the achievable region is the same with and without errors. On the other hand, if errors are allowed, our results imply that the set of jointly-typical sequences can be communicated without any errors and with no asymptotic loss in efficiency.

[5]The precise regions here and below are described in the cited Corollaries.

1.5 Complexity measures

The support set of a random pair (X, Y) with underlying probability distribution $p(x, y)$ was defined earlier as the set

$$S_{X,Y} = \{(x, y) : p(x, y) > 0\}$$

of ordered pairs occurring with positive probability. An *input* is an element of $S_{X,Y}$ viewed as value assignments to X and Y. For every input (x, y), the protocol used, ϕ, determines a finite sequence $\sigma_1(x, y), \ldots, \sigma_{m_\phi(x,y)}(x, y)$ of transmitted messages.[6] Since $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ alternate in transmitting these messages, all even numbered messages in this sequence are transmitted by one communicator and all odd numbered messages are transmitted by the other. $m_\phi(x, y)$ is the number of messages exchanged for (x, y). The protocol ϕ is *m-message* if $m_\phi(x, y) \leq m$ for all $(x, y) \in S_{X,Y}$. The *length*, $|\sigma|$, of a message σ is the number of bits it contains. The *transmission length* of the input (x, y) is

$$l_\phi(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{m_\phi(x,y)} |\sigma_i(x, y)| \ ,$$

the number of bits transmitted by both communicators when $X = x$, $Y = y$, and the protocol used is ϕ.

The *average complexity of* ϕ is the expectation, over all inputs, of the number of bits transmitted during the execution of ϕ:

$$\bar{L}_\phi \stackrel{\text{def}}{=} \sum_{(x,y) \in S_{X,Y}} p(x, y) \cdot l_\phi(x, y) \ .$$

The *m-message average complexity of* (X, Y) is

$$\bar{C}_m(X|Y) \stackrel{\text{def}}{=} \inf\{\bar{L}_\phi : \phi \text{ is an } m\text{-message protocol for } (X, Y)\} \ ,$$

the number of bits that the two communicators must transmit on the average if restricted to protocols that require at most m messages. Since empty messages are allowed, $\bar{C}_m(X|Y)$ is a decreasing function of m bounded below by 0. We can therefore define $\bar{C}_\infty(X|Y)$, the *unbounded-message average complexity of* (X, Y), to be the limit of $\bar{C}_m(X|Y)$ as $m \to \infty$. It is the *minimum* number of bits that must be transmitted on the average for $P_{\mathcal{Y}}$ to know X. A protocol for (X, Y) whose average complexity is $\bar{C}_\infty(X|Y)$ is an *optimal* protocol for (X, Y).

[6] For notational simplicity, the protocol ϕ is implicit in $\sigma_i(x, y)$. Also, we do not assume that the same number of messages is transmitted for all inputs or that the same communicator transmits the first message for all inputs.

2 General pairs

We prove that for all (X, Y) pairs,

$$H(X|Y) \leq \bar{C}_\infty(X|Y) \leq \bar{C}_1(X|Y) \leq H(X) + 1 \ . \tag{4}$$

These bounds are not tight as $H(X|Y)$ can be much smaller than $H(X)$. However, Sections 3 and 4 show that for some (X, Y) pairs the upper bound is tight, for others the lower bound can be achieved. In that sense, the bounds of Inequality (4) are the tightest expressible in terms of Shannon entropies.

The upper bound is achieved by a Huffman code based on the marginal probabilities of X. The lower bound is intuitive too. Even if $P_\mathcal{X}$ knows Y, the expected number of bits he must transmit when $Y = y$ is at least $H(X|Y = y)$. Therefore, the expectation, over all inputs, of the number of transmitted bits, is at least $H(X|Y)$. This, however, is only a heuristic argument. When $P_\mathcal{X}$ and $P_\mathcal{Y}$ communicate, they alternate in transmitting messages, therefore the sequence of bits transmitted by $P_\mathcal{X}$ is parsed into messages. It is conceivable that the "commas" separating the messages can be used to mimic a ternary alphabet, thus decrease the number of transmitted bits (we count only "0"s and "1"s, not commas). To turn the above argument into a proof, we need to show that $P_\mathcal{X}$ cannot use the parsing to further encode his information. This is done in the next lemma. Lemma 2 incorporates this result in the argument above.

Let ϕ be a protocol for (X, Y). In Subsection 1.5 we defined $\sigma_1(x, y), \ldots, \sigma_{m_\phi(x,y)}(x, y)$ to be the sequence of messages transmitted by $P_\mathcal{X}$ and $P_\mathcal{Y}$ for the input (x, y). Let $\sigma_\phi^\mathcal{X}(x, y)$ denote the concatenation of all messages transmitted by $P_\mathcal{X}$ for that input.

Lemma 1 Let ϕ be a protocol for (X, Y). If $(x, y), (x', y) \in S_{X,Y}$ and $x \neq x'$, then neither $\sigma_\phi^\mathcal{X}(x, y)$ nor $\sigma_\phi^\mathcal{X}(x', y)$ is a prefix of the other (in particular, they are not equal).

Proof: We show that if one of $\sigma_\phi^\mathcal{X}(x, y)$ and $\sigma_\phi^\mathcal{X}(x', y)$ is a prefix of the other then

1) $m_\phi(x, y) = m_\phi(x', y)$,

2) for $j = 1, \ldots, m_\phi(x, y)$, $\sigma_j(x, y) = \sigma_j(x', y)$.

Therefore, the correct-decision property implies that $x = x'$.

To prove (2) we show by induction on $i \leq m_\phi(x, y)$ that $\sigma_j(x, y) = \sigma_j(x', y)$ for $j = 1, \ldots, i$. The induction hypothesis holds for $i = 0$. Assume it holds for $i - 1$. If $\sigma_i(x, y)$ is transmitted by $P_\mathcal{Y}$ then, by the separate-transmissions property, he also transmits $\sigma_i(x', y)$, and $\sigma_i(x, y) = \sigma_i(x', y)$. If $\sigma_i(x, y)$ is transmitted by $P_\mathcal{X}$ then, since one of $\sigma_\phi^\mathcal{X}(x, y)$ and

$\sigma_\phi^{\mathcal{X}}(x',y)$ is a prefix of the other and the previous messages are identical, one of $\sigma_i(x,y)$ and $\sigma_i(x',y)$ is a prefix of the other. By the implicit-termination property, $\sigma_i(x,y) = \sigma_i(x',y)$.

Now, (1) follows from the implicit-termination property as we have just shown that $\sigma_j(x,y) = \sigma_j(x',y)$ for $j = 1,\ldots,m_\phi(x,y)$. □

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

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

the set of possible values of Y. $P_{\mathcal{Y}}$'s *ambiguity set* when his random variable attains the value $y \in S_{\mathcal{Y}}$ is:

$$S_{X|Y}(y) = \{x : (x,y) \in S_{X,Y}\}\ ,$$

the set of possible X values when $Y = y$. Lemma 1 implies that for every $y \in S_{\mathcal{Y}}$, the multiset[7] $\{\sigma_\phi^{\mathcal{X}}(x,y) : x \in S_{X|Y}(y)\}$ is prefix free. Namely, it contains $|S_{X|Y}(y)|$ distinct elements, none of which is a prefix of another.

Lemma 2 For all (X,Y) pairs,

$$H(X|Y) \le \bar{C}_\infty(X|Y) \le \bar{C}_1(X|Y) \le H(X) + 1\ .$$

Proof: We mentioned that the upper bound is achieved by an Huffman code. We prove the lower bound. Let ϕ be a protocol for (X,Y). The preceding discussion and the data-compression theorem imply that for all $y \in S_{\mathcal{Y}}$,

$$\sum_{x \in S_{X|Y}(y)} p(x|y) \cdot l_\phi^{\mathcal{X}}(x,y) \ge H(X|Y=y)$$

where $l_\phi^{\mathcal{X}}(x,y) \stackrel{\text{def}}{=} |\sigma_\phi^{\mathcal{X}}(x,y)|$ denotes the number of bits transmitted by $P_{\mathcal{X}}$ for the input (x,y). The average number of bits transmitted by $P_{\mathcal{X}}$ under ϕ is:

$$\begin{aligned}
\sum_{(x,y)\in S_{X,Y}} p(x,y) \cdot l_\phi^{\mathcal{X}}(x,y) &= \sum_{y\in S_{\mathcal{Y}}} p(y) \sum_{x\in S_{X|Y}(y)} p(x|y) \cdot l_\phi^{\mathcal{X}}(x,y) \\
&\ge \sum_{y\in S_{\mathcal{Y}}} p(y) H(X|Y=y) \\
&= H(X|Y)\ .
\end{aligned}$$

□

Note that we proved a stronger version of the lower bound, than claimed in the lemma. We showed that $P_{\mathcal{X}}$ alone must transmit an average of at least $H(X|Y)$ bits, regardless of $P_{\mathcal{Y}}$'s transmissions.

[7] A *multiset* allows for multiplicity of elements, e.g., $\{0,1,1\}$.

3 Pairs with Cartesian-product support sets

A support set $S_{X,Y}$ is a *Cartesian product* if for some sets $A \subseteq S_{\mathcal{X}}$ and $B \subseteq S_{\mathcal{Y}}$,

$$S_{X,Y} = A \times B \ .$$

Figure 4 illustrates one such set. We show that for all (X,Y) pairs with Cartesian-product support sets, the upper bound of Inequality (1) is essentially tight:

$$H(X) \leq \bar{C}_\infty(X|Y) \leq \bar{C}_1(X|Y) \leq H(X) + 1 \ .$$

Cartesian-product support sets are important mainly because for many (X,Y) pairs all inputs $(x,y) \in S_{X,Y}$ have positive probability, i.e., $S_{X,Y} = S_{\mathcal{X}} \times S_{\mathcal{Y}}$. In all these cases, at least $H(X)$ bits must be transmitted on the average. The same as the number of bits required when $P_{\mathcal{Y}}$ has no information at all, and when no interaction is allowed. In other words, when $S_{X,Y}$ is a Cartesian product, neither $P_{\mathcal{Y}}$'s knowledge of Y, nor interaction, help reduce communication.

Lemma 3 Let ϕ be a protocol for (X,Y). If $(x',y),(x,y),(x,y') \in S_{X,Y}$ and $x \neq x'$, then neither $\sigma_\phi(x',y)$ nor $\sigma_\phi(x,y')$ is a prefix of the other (in particular, they are not equal).
Proof: The proof is analogous to that of Lemma 1, hence omitted. □

Just as $S_{\mathcal{Y}}$ denotes the set of possible values of Y and $S_{X|Y}(y)$ is $P_{\mathcal{Y}}$'s ambiguity set when he knows the value $y \in S_{\mathcal{Y}}$, we define

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

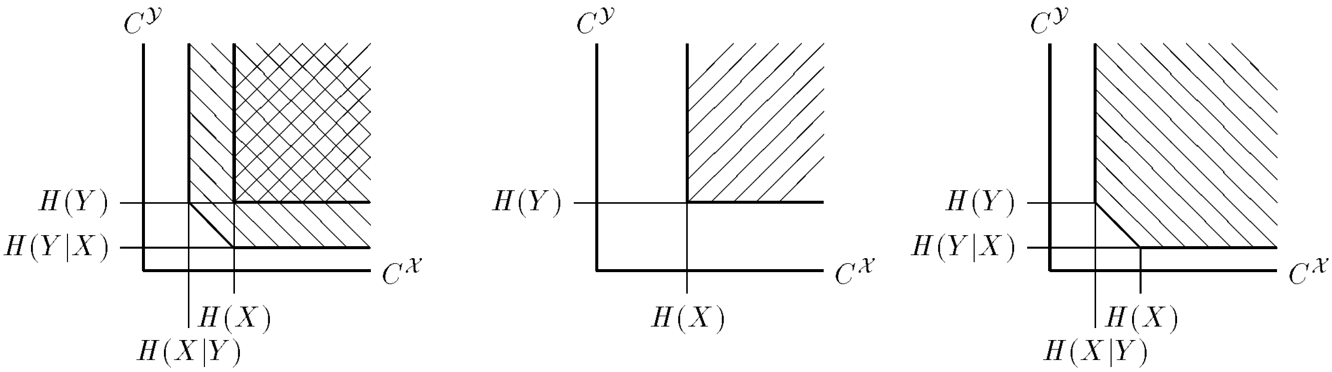
to be the set of possible values of X and let $P_{\mathcal{X}}$'s *ambiguity set* when $X = x$ be

$$S_{Y|X}(x) \stackrel{\text{def}}{=} \{y : (x,y) \in S_{X,Y}\} \ .$$

Corollary 1 If the support set of (X,Y) is a Cartesian product then

$$\bar{C}_\infty(X|Y) \geq H(X) \ .$$

Proof: Let ϕ be a protocol for (X,Y). Recall that $l_\phi(x,y)$ denotes the number of bits transmitted under ϕ for the input (x,y) and that $\bar{L}_\phi$ is the number of transmitted bits, averaged over all inputs. We show that $\bar{L}_\phi \geq H(X)$.



(a) general pairs (b) Cartesian product (c) uniform pairs

Figure 3: Achievable regions for the single-event Slepian-Wolf problem

	1	2	3	4	5	6	
1	0	0	0	0	0	0	
2	$\frac{1}{3}$	0	0	$\frac{1}{8}$	0	$\frac{1}{12}$	$S_{\mathcal{Y}}$
3	0	0	0	0	0	0	
4	$\frac{1}{6}$	0	0	$\frac{1}{24}$	0	$\frac{1}{4}$	

$S_{\mathcal{X}}$

Figure 4: A probability distribution with a Cartesian-product support set.

Since $S_{X,Y}$ is a Cartesian product, $S_{X,Y} = A \times B$ for some sets $A \subseteq S_{\mathcal{X}}$ and $B \subseteq S_{\mathcal{Y}}$. For every $x \in A$, let

$$l_\phi(x) \stackrel{\text{def}}{=} \min\{l_\phi(x,y) : y \in S_{Y|X}(x)\}$$

be the minimum number of bits transmitted by both $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ when $X = x$. Since $S_{X,Y}$ is a Cartesian product, Lemma 3 implies that for all pairs $(x_1, y_1), (x_2, y_2) \in S_{X,Y}$ such that $x_1 \neq x_2$, neither $\sigma_\phi(x_1, y_1)$ nor $\sigma_\phi(x_2, y_2)$ is a prefix of the other. Hence, the $l_\phi(x)$'s are the lengths of $|A|$ strings, none of which is a prefix of the other. By the data-compression theorem:

$$\sum_{x \in A} p(x) l_\phi(x) \geq H(X) \ .$$

Therefore,

$$\begin{aligned}
\bar{L}_\phi &= \sum_{(x,y) \in A \times B} p(x,y) l_\phi(x,y) \\
&= \sum_{x \in A} p(x) \sum_{y \in S_{Y|X}(x)} p(y|x) l_\phi(x,y) \\
&\geq \sum_{x \in A} p(x) l_\phi(x) \\
&\geq H(X) \ . \qquad \Box
\end{aligned}$$

Of course, we cannot prove, as we did in the last section, that this lower bound holds even for the average number of bits transmitted by $P_{\mathcal{X}}$ alone. There is always a protocol in which $P_{\mathcal{X}}$ transmits at most $H(X|Y) + 1$ bits on the average.

4 Uniformly-distributed pairs

In the last section we proved that if the support set of (X, Y) is a Cartesian product, the upper bound of Inequality (1) is tight: $H(X) \leq \bar{C}_\infty(X|Y) \leq \bar{C}_1(X|Y) \leq H(X) + 1$. We now show that if (X, Y) is uniformly distributed, the lower bound of Inequality (1) can be almost achieved:

$$H(X|Y) \leq \bar{C}_\infty(X|Y) \leq \bar{C}_4(X|Y) \leq H(X|Y) + 3\log\left(H(X|Y) + 1\right) + 15.5 \ .$$

As we mentioned in the introduction, this shows that

1. $P_{\mathcal{X}}$ can communicate X to $P_{\mathcal{Y}}$ using roughly the number of bits needed when $P_{\mathcal{X}}$ knows Y in advance.

2. Four messages are asymptotically optimal for average-case complexity. The number $\bar{C}_1(X|Y)$ of bits needed when only one message is allowed may be arbitrarily larger then the minimum number necessary, $\bar{C}_\infty(X|Y)$, yet with four message only negligibly more than $\bar{C}_\infty(X|Y)$ bits are needed. We remark that the corresponding result is not known to hold for worst-case complexity.

We need a few definitions. Let Z be a random variable distributed over a support set $S_\mathcal{Z}$ according to a probability distribution p. A *(binary, prefix-free) encoding* of Z is a mapping ϕ from $S_\mathcal{Z}$ to $\{0,1\}^*$ such that for all $z_1, z_2 \in S_\mathcal{Z}$, the *codeword*, $\phi(z_1)$, of z_1 is not a prefix of $\phi(z_2)$ – the codeword of z_2. For $z \in S_\mathcal{X}$, let $\ell_\phi(z)$ be the length of $\phi(z)$. The *expected length* of ϕ is

$$\overline{\ell}_\phi(Z) \stackrel{\text{def}}{=} \sum_{z \in S_\mathcal{Z}} p(z)\ell_\phi(z) \ .$$

The *minimum encoding length* of Z is

$$\mathcal{L}(Z) \stackrel{\text{def}}{=} \min\{\overline{\ell}_\phi(Z) : \phi \text{ is an encoding of } Z\} \ .$$

It is well known that

$$H(Z) \leq \mathcal{L}(Z) \leq H(Z) + 1 \ . \tag{5}$$

Recall that the support set, $S_{X,Y}$, of a random pair (X,Y) is the set of all possible inputs. (X,Y) is uniformly distributed if all elements in $S_{X,Y}$ are equally likely. The support set of X is the set $S_\mathcal{X}$ of all possible values of X, $S_\mathcal{Y}$ was similarly defined. $P_\mathcal{Y}$'s ambiguity set when his random variable attains the value $y \in S_\mathcal{Y}$ is the set $S_{X|Y}(y)$ of possible X values in that case. Denote the collection of $P_\mathcal{Y}$'s ambiguity sets by

$$\mathcal{S}_{X|Y} \stackrel{\text{def}}{=} \{S_{X|Y}(y) : y \in S_\mathcal{Y}\} \ .$$

A collection of functions, each defined over $S_\mathcal{X}$, *perfectly hashes* $\mathcal{S}_{X|Y}$ if for every $y \in S_\mathcal{Y}$ there is a function in the collection that is one-to-one over (or *hashes*) $S_{X|Y}(y)$. Perfect hash collections are discussed in [11, 12, 13, 14] and others.

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 $\mathcal{S}_{X|Y}$. Assume also that the mapping hash(y) assigns, for each $y \in S_\mathcal{Y}$, a function in $\mathcal{F}$ that hashes $S_{X|Y}(y)$. Then the random variable hash(Y) denotes a function in $\mathcal{F}$ that hashes $P_\mathcal{Y}$'s ambiguity set $S_{X|Y}(Y)$. We now show that

$$\bar{C}_2(X|Y) \leq \mathcal{L}(\text{hash}(Y)) + \lceil \log b \rceil \ . \tag{6}$$

$P_\mathcal{X}$ and $P_\mathcal{Y}$ agree in advance on the collection $\mathcal{F}$, on the mapping $\text{hash}(y)$, and on an encoding of $\text{hash}(Y)$ whose expected length is $\mathcal{L}(\text{hash}(Y))$. When $P_\mathcal{X}$ is given X and $P_\mathcal{Y}$ is given Y, they execute the following two-message protocol. $P_\mathcal{Y}$ transmits the encoding of $\text{hash}(Y)$. Now $P_\mathcal{X}$ knows X and $f_{\text{hash}(Y)}$. He computes and transmits the binary representation of $f_{\text{hash}(Y)}(X)$. Since $f_{\text{hash}(Y)}$ is one-to-one over $S_{X|Y}(Y)$, $P_\mathcal{Y}$ can recover X. The number of bits transmitted by $P_\mathcal{X}$ is always at most $\lceil \log b \rceil$. The expected number of bits transmitted by $P_\mathcal{Y}$ is $\mathcal{L}(\text{hash}(Y))$.

To upper bound $\bar{C}_2(X|Y)$, we therefore demonstrate a collection $\mathcal{F}$ that perfectly hashes $\mathcal{S}_{X|Y}$, and a mapping $\text{hash}(y)$. By construction, each function in $\mathcal{F}$ has a small range while Lemmas 4 and 5 show that $\mathcal{L}(\text{hash}(Y))$ is low. Theorem 1 combines this result with Inequality (6).

One way to guarantee that $\mathcal{L}(\text{hash}(Y))$ is low, is to ensure that $\text{hash}(Y)$ assumes certain values with high probability. The next lemma shows that for any collection of subsets, there is a function that hashes a relatively large fraction of these subsets.

Lemma 4 Let $\mathcal{S}$ be a finite collection of finite sets and let Pr be a probability measure over $\mathcal{S}$, namely, $\Pr(S) \geq 0$ for every $S \in \mathcal{S}$ and $\sum_{S\in\mathcal{S}} \Pr(S) = 1$.[8]

Define $\hat{\mu} \stackrel{\text{def}}{=} \max\{|S| : S \in \mathcal{S}\}$ to be the size of the largest set in $\mathcal{S}$. Then, for every $b \geq \hat{\mu}$ there is a function $f : \bigcup_{S\in\mathcal{S}} S \to \{1,\ldots,b\}$ that perfectly hashes a subcollection of $\mathcal{S}$ whose probability is at least $\frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}}$, that is,

$$\sum_{S\in\mathcal{S}} \Pr(S) 1_{f \text{ hashes } S} \geq \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}}$$

where $b^{\underline{\hat{\mu}}} \stackrel{\text{def}}{=} b \cdot (b-1) \cdot \ldots (b - \hat{\mu} + 1)$ is the $\hat{\mu}$th falling power of b and

$$1_{f \text{ hashes } S} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } f \text{ hashes } S \\ 0 & \text{otherwise.} \end{cases}$$

is an indicator function.

Proof: Let $\mathcal{F}$ be the set of all functions from $\bigcup_{S\in\mathcal{S}} S$ to $\{1,\ldots,b\}$. Generate a random function F in $\mathcal{F}$ by assigning to every element in $\bigcup_{S\in\mathcal{S}} S$ an element of $\{1,\ldots,b\}$ uniformly and independently.

[8] In this lemma we do not assume that Pr results from an underlying distribution on the elements of the sets $S \in \mathcal{S}$. For example, a larger set can have smaller probability.

For every given set $S \in \mathcal{S}$, the probability that F hashes S is

$$\sum_{f\in\mathcal{F}} \Pr(f) 1_{f \text{ hashes } S} = \frac{b^{\underline{|S|}}}{b^{|S|}} \geq \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}} .$$

Therefore,

$$\begin{aligned}\sum_{f\in\mathcal{F}} \Pr(f) \sum_{S\in\mathcal{S}} \Pr(S) 1_{f \text{ hashes } S} &= \sum_{S\in\mathcal{S}} \Pr(S) \sum_{f\in\mathcal{F}} \Pr(f) 1_{f \text{ hashes } S} \\ &\geq \sum_{S\in\mathcal{S}} \Pr(S) \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}} \\ &= \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}} .\end{aligned}$$

Hence the required inequality holds for at least one function $f \in \mathcal{F}$. □

Using this result, we construct a sequence $f_1, , \ldots, f_k$ of functions that perfectly hashes $\mathcal{S}_{X|Y}$. The probability that f_i is the first function to hash $S_{X|Y}(Y)$ rapidly decreases with i.

$P_{\mathcal{Y}}$'s *ambiguity* when his random variable attains the value $y \in S_{\mathcal{Y}}$ is

$$\mu_{X|Y}(y) \stackrel{\text{def}}{=} |S_{X|Y}(y)| ,$$

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

$$\hat{\mu}_{X|Y} \stackrel{\text{def}}{=} \sup\{\mu_{X|Y}(y) : y \in S_{\mathcal{Y}}\},$$

the maximum number of X values possible with any given Y value. $P_{\mathcal{X}}$'s *ambiguity*, $\mu_{Y|X}(x)$, when $X = x$ and his *maximum ambiguity*, $\hat{\mu}_{Y|X}$, are defined symmetrically. In the sequel we frequently use the following abbreviation:

$$\beta \stackrel{\text{def}}{=} \frac{b^{\underline{\hat{\mu}_{X|Y}}}}{b^{\hat{\mu}_{X|Y}}} , \tag{7}$$

where $b \geq \hat{\mu}_{X|Y}$ is an integer. Let $\mathcal{F}$ be the set of functions from $S_{\mathcal{X}}$ to $\{1, \ldots, b\}$. For each function $f \in \mathcal{F}$ let

$$\alpha_1(f) \stackrel{\text{def}}{=} \Pr\Big(f \text{ hashes } S_{X|Y}(Y)\Big)$$

be the probability that f hashes $P_{\mathcal{Y}}$'s ambiguity set. Let f_1 be a function in $\mathcal{F}$ that maximizes α_1. The last lemma implies that $\alpha_1(f_1) \geq \beta$. Recursively, if $f_1, \ldots, f_{i-1}$ do not hash all ambiguity sets, let

$$\alpha_i(f) \stackrel{\text{def}}{=} \Pr\Big(f \text{ hashes } S_{X|Y}(Y) \mid \text{none of } f_1, \ldots, f_{i-1} \text{ hashes } S_{X|Y}(Y)\Big)$$

be the probability that $P_{\mathcal{Y}}$'s ambiguity set is hashed by the function $f \in \mathcal{F}$, given that it is not hashed by any of $f_1, \ldots, f_{i-1}$. Let f_i be a function in $\mathcal{F}$ that maximizes α_i. Again, the lemma guarantees that $\alpha_i(f_i) \geq \beta$.

Every time a function f_i is determined, it hashes at least one additional ambiguity set. Since $S_{X,Y}$ is finite, there are finitely many ambiguity sets, hence the process stops after some finite number, say k, of steps.

Once the sequence $f_1, \ldots, f_k$ has been defined, let hash(y) be the first function in $f_1, \ldots, f_k$ that hashes $S_{X|Y}(y)$. Lemma 5 uses a sequence of three claims to show that $\mathcal{L}(\text{hash}(Y))$ is low.

We write $a_1, \ldots, a_k$ to denote the k element sequence whose ith element is a_i; we use the notation $a_1, a_2, , \ldots,$ if the sequence can be either finite or infinite. A sequence $q_1, q_2, \ldots$ *majorizes* a sequence $p_1, p_2, \ldots$ if for all $j \geq 1$

$$\sum_{i=1}^{j} q_i \geq \sum_{i=1}^{j} p_i$$

where 0's are appended to the tails of finite sequences.

For $i \in \{1, \ldots, k\}$ let

$$q_i \stackrel{\text{def}}{=} \Pr(\text{hash}(Y) = f_i)) \tag{8}$$

be the probability that f_i is the first function that hashes $P_{\mathcal{Y}}$'s ambiguity set. And for $i \in \mathcal{Z}^+$, let

$$p_i = \beta(1-\beta)^{i-1} \tag{9}$$

be the probability that a geometric random variable with probability β of success attains the value i.

The first claim in the proof of Lemma 5 implies that $q_1, \ldots, q_k$ majorizes $p_1, p_2, \ldots$. The second claim shows that if $q_1, \ldots, q_k$ majorizes $p_1, p_2, \ldots$, then $\mathcal{L}(q_1, \ldots, q_k) \leq \mathcal{L}(p_1, p_2, \ldots)$. (The minimum encoding length of a distribution is defined as that of a random variable). The third claim combines Inequality (5) with the entropy of the geometric distribution to show that

Lemma 5 For all (X, Y) pairs

$$\mathcal{L}(\text{hash}(Y)) \leq \log\frac{1}{\beta} + \log e + 1$$

where β was defined in Equation (7).

Proof: We first show that $q_1, \ldots, q_k$, defined in Equation (8), majorizes $p_1, p_2, \ldots$, defined in Equation (9).

Claim 1 Let $p_1, p_2, \ldots$ and $q_1, q_2, \ldots$ be probability distributions such that

$$\frac{q_i}{1 - q_1 - \ldots - q_{i-1}} \geq \frac{p_i}{1 - p_1 - \ldots - p_{i-1}}$$

for all $i \geq 1$. Then $q_1, q_2, \ldots$ majorizes $p_1, p_2, \ldots$.

Proof: An induction on i shows that

$$p_{i+1} + p_{i+2} + \ldots = \prod_{j=1}^{i} \left(1 - \frac{p_j}{1 - p_1 - \ldots - p_{j-1}}\right)$$

and the corresponding equality holds for $q_{i+1} + q_{i+2} + \ldots$, hence, for all $i \geq 1$,

$$q_{i+1} + q_{i+2} + \ldots \leq p_{i+1} + p_{i+2} + \ldots \qquad \square$$

The discussion preceding this lemma showed

$$\frac{q_i}{1 - q_1 - \ldots - q_{i-1}} \geq \beta = \frac{p_i}{1 - p_1 - \ldots - p_{i-1}} \ ,$$

hence $q_1, \ldots, q_k$ majorizes $p_1, p_2, \ldots$. We use this fact to upper bound the minimum encoding length of hash(Y).

Claim 2 Let $p_1, p_2, \ldots$ be a nonincreasing probability distribution ($p_i \geq p_{i+1} \geq 0$ and $\sum p_i = 1$). If the probability distribution $q_1, \ldots, q_k$ majorizes $p_1, p_2, \ldots$ then

$$\mathcal{L}(q_1, \ldots, q_k) \leq \mathcal{L}(p_1, p_2, \ldots) \ .$$

Proof: By majorization, $\sum_{i=1}^{k} p_i \leq 1$, hence p_i is positive for all $i \in \{1, \ldots, k\}$. For i in this range, let ℓ_i be the length of the codeword corresponding to p_i in an encoding achieving $\mathcal{L}(p_1, p_2, \ldots)$. Since $p_i \leq p_{i-1}$, we must have $\ell_i \geq \ell_{i-1}$. Therefore,

$$\begin{aligned}
\mathcal{L}(q_1, \ldots, q_k) &\leq \sum_{i=1}^{k} q_i \ell_i \\
&= \sum_{i=1}^{k} (\sum_{j=i}^{k} q_j)(\ell_i - \ell_{i-1}) \\
&\leq \sum_{i=1}^{k} (\sum_{j=i}^{k} p_j)(\ell_i - \ell_{i-1}) \\
&= \sum_{i=1}^{k} p_i \ell_i \\
&= \mathcal{L}(p_1, p_2, \ldots) \ . \qquad \square
\end{aligned}$$

The two claims show that the minimum encoding length of hash(Y) is upper bounded by that of a geometric random variable with probability β of success. The next claim, given for completeness, estimates this length.

Claim 3 As defined in Equation (9), let $p_1, p_2, \ldots$ be the probability distribution of a geometric random variable with probability β of success. Then

$$\mathcal{L}(p_1, p_2, \ldots) \le \log \frac{1}{\beta} + \log e + 1 \ .$$

Proof: We combine Inequality (5) and the following standard calculation.

$$\begin{aligned}
H(p_1, p_2, \ldots) &= -\sum_{i=1}^{\infty} \beta(1-\beta)^{i-1} \log(\beta(1-\beta)^{i-1}) \\
&= -\beta \log \beta \sum_{i=1}^{\infty} (1-\beta)^{i-1} - \beta \log(1-\beta) \sum_{i=1}^{\infty} (i-1)(1-\beta)^{i-1} \\
&= \log \frac{1}{\beta} + \frac{1-\beta}{\beta} \log \frac{1}{(1-\beta)} \\
&\le \log \frac{1}{\beta} + \log e
\end{aligned}$$

where we used the (natural logarithm) inequality:

$$\frac{1-\beta}{\beta} \ln \frac{1}{1-\beta} = \frac{1-\beta}{\beta} \ln \left(1 + \frac{\beta}{1-\beta}\right) \le \frac{1-\beta}{\beta} \cdot \frac{\beta}{1-\beta} = 1 \ . \qquad \square$$

The lemma follows. $\square$

We can now derive our first theorem:

Theorem 1 For all (X, Y) pairs,

$$\bar{C}_2(X|Y) \le 2 \log \hat{\mu}_{X|Y} + 5.5 \ .$$

Proof: Let $b \ge \hat{\mu}_{X|Y}$ be an integer. Combining Inequality (6) and Lemma 5, and abbreviating $\hat{\mu}_{X|Y}$ as $\hat{\mu}$, we obtain

$$\begin{aligned}
\bar{C}_2(X|Y) &\le \log \binom{b^{\hat{\mu}}}{b^{\underline{\hat{\mu}}}} + \lceil \log b \rceil + \log e + 1 \\
&\le \log \frac{b^{\hat{\mu}}}{(b-\hat{\mu})^{\hat{\mu}}} + \log b + \log e + 2 \\
&= \hat{\mu} \log \frac{b}{b-\hat{\mu}} + \log b + \log e + 2 \ .
\end{aligned}$$

Choosing $b = \hat{\mu}^2 + \hat{\mu}$, we derive:

$$\begin{aligned}
\bar{C}_2(X|Y) &\leq \hat{\mu} \log\left(1 + \frac{1}{\hat{\mu}}\right) + \log(\hat{\mu}^2 + \hat{\mu}) + \log e + 2 \\
&\leq 2\log \hat{\mu} + 2\log e + \log\left(1 + \frac{1}{\hat{\mu}}\right) + 2 \\
&\leq 2\log \hat{\mu} + 5.5\ . \qquad \square
\end{aligned}$$

The theorem can be used to prove that for all uniform (X, Y) pairs

$$\bar{C}_2(X|Y) \leq 2H(X|Y) + 2\log\left(H(X|Y) + 1\right) + 7.5\ . \tag{10}$$

This result is of independent interest. One message may require arbitrarily more bits than the minimum necessary (e.g. $\bar{C}_1(X|Y) = \log t$ while $\bar{C}_2(X|Y) \leq 3$ in the League problem of Example 1). Yet Inequality (10) shows that two messages require at most twice the minimum number of bits. We proceed to show that by adding two more messages, communication can be reduced roughly by half, thereby asymptotically achieving the minimum possible.

Let k be an integer; a function f is *k-smooth* over a subset S of its domain if for every r in its range,

$$|S \cap f^{-1}(r)| \leq k.$$

Namely, if f does not assign the same value to more than k elements of S. The function f is *k-smooth* over a collection of subsets if it is k-smooth over each subset.

Lemma 6 Let $\hat{\mu}$ be an integer ≥ 3, let $\mathcal{S}$ be a finite collection of finite sets, each of size $\leq \hat{\mu}$, and let Pr be a probability measure over $\mathcal{S}$. Then, there is a function $f : \bigcup_{S \in \mathcal{S}} S \to \{1, \ldots, \hat{\mu}\}$ that is $\frac{2.35 \log \hat{\mu}}{\log\log \hat{\mu}}$-smooth over a subcollection of $\mathcal{S}$ whose probability is at least $\frac{1}{2}$. Namely, if $S \in \mathcal{S}$ is selected at random according to Pr, then

$$\Pr\left(f \text{ is } \frac{4\log\hat{\mu}}{\log\log\hat{\mu}}\text{-smooth over } S\right) \geq \frac{1}{2}\ .$$

Proof: Generate a random function F from $\bigcup_{S \in \mathcal{S}} S$ to $\{1, \ldots, \hat{\mu}\}$ by assigning to every element in $\bigcup_{S \in \mathcal{S}} S$ an element of $\{1, \ldots, \hat{\mu}\}$ uniformly and independently. For every set $S \in \mathcal{S}$, every $i \in \{1, \ldots, \hat{\mu}\}$, and every integer k,

$$\Pr\left(|S \cap F^{-1}(i)| \geq k\right) \leq \binom{\hat{\mu}}{k} \frac{1}{\hat{\mu}^k} \leq \frac{1}{k!}\ .$$

Note that Pr now represents the joint (product) probability of S and F. Using the union bound,

$$\Pr(F \text{ is } \textit{not } k\text{-smooth over } S) \leq \frac{\hat{\mu}}{k!} \leq \hat{\mu}\frac{e^k}{k^k}\ .$$

For $k \geq \frac{2.35 \log \hat{\mu}}{\log \log \hat{\mu}}$, this probability is at most $\frac{1}{2}$ (for small $\hat{\mu}$ the last inequality should be verified directly). The lemma now follows from an argument similar to the one used to prove Lemma 4. $\square$

As in the discussion following Lemma 4, we can now assign to every y a function smooth(y) that is $\frac{4 \log \hat{\mu}_{X|Y}}{\log \log \hat{\mu}_{X|Y}}$-smooth over $S_{X|Y}(y)$. The arguments used in Lemma 5 then show that the probability distribution underlying smooth(Y) majorizes the geometric distribution with probability $\frac{1}{2}$ of success, hence:

Lemma 7 For all (X, Y) pairs with $\hat{\mu}_{X|Y} \geq 3$,

$$\mathcal{L}(\text{smooth}(Y)) \leq 2. \qquad \square$$

Theorem 2 For all (X, Y) pairs with $\hat{\mu}_{X|Y} \geq 3$

$$\bar{C}_4(X|Y) \leq \log \hat{\mu}_{X|Y} + 2 \log \log \hat{\mu}_{X|Y} - 2 \log \log \log \hat{\mu}_{X|Y} + 13 \ .$$

Proof: $P_{\mathcal{Y}}$ transmits the encoding of the function $f = \text{smooth}(Y)$ using at most 2 bits on the average. Then $P_{\mathcal{X}}$ transmits $\lceil \log \hat{\mu}_{X|Y} \rceil$ bits describing $f(X)$. Now $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ concentrate on a random pair whose maximum ambiguity is at most $\lceil \frac{2.35 \log \hat{\mu}_{X|Y}}{\log \log \hat{\mu}_{X|Y}} \rceil$. Theorem 1 says that X can be conveyed to $P_{\mathcal{Y}}$ using an expected number of bits of at most

$$2 \log \left\lceil \frac{2.35 \log \hat{\mu}_{X|Y}}{\log \log \hat{\mu}_{X|Y}} \right\rceil + 5.5 \leq 2 \log \log \hat{\mu}_{X|Y} - 2 \log \log \log \hat{\mu}_{X|Y} + 10 \ . \qquad \square$$

For the first time, we now use the uniformity of (X, Y). It implies that

$$\log \mu_{X|Y}(y) = H(X|y)$$

for all $y \in S_{\mathcal{Y}}$. Hence, if $P_{\mathcal{Y}}$'s ambiguity, $\mu_{X|Y}(y)$, is the same for all $y \in S_{\mathcal{Y}}$, then

$$\log \hat{\mu}_{X|Y} = H(X|Y)$$

and the theorem becomes:

$$\bar{C}_4(X|Y) \leq H(X|Y) + 2 \log H(X|Y) - 2 \log \log H(X|Y) + 13 \ .$$

This bound applies when $\hat{\mu}_{X|Y} \geq 3$. For $\hat{\mu}_{X|Y} \leq 2$, Theorem 1 implies

$$\bar{C}_4(X|Y) \leq \bar{C}_2(X|Y) \leq 2 \log \hat{\mu}_{X|Y} + 5.5 \leq 7.5.$$

If $\mu_{X|Y}(y)$ varies with y, then $\log \hat{\mu}_{X|Y}$ can be much larger then $H(X|Y)$. In that case we prefix the above protocol with a stage that identifies $\mu_{X|Y}(y)$ to within a factor of two.

Theorem 3 For all *uniform* (X, Y) pairs

$$\bar{C}_4(X|Y) \leq H(X|Y) + 3\log\left(H(X|Y)+1\right) + 17 \; .$$

Proof: First we describe a prefix-free encoding of the non-negative integers where the codeword of 0 is 2-bits long and the codeword of any other i is at most $2\lceil\log(i+1)\rceil$-bits long. Consider the $\lceil\log(i+1)\rceil$-bit binary representation of i. Encode every 0 (or 1) in this representation, except the last, as 00 (or 01). Encode the last 0 (or 1) as 10 (or 11). For example, 0, 1, and 2, are encoded as 10, 11, and 0110, respectively.

This construction can be used to derive a prefix-free encoding of the integers where the codewords of 0 and 1 are 3-bits long and any other i is encoded into $\lceil\log(i+1)\rceil + 2\lceil\log\log(i+1)\rceil$ bits. Given i, we first encode the integer $\lceil\log(i+1)\rceil - 1$ as above. Then we provide the $\lceil\log(i+1)\rceil$ bits in the binary representation of i. More sophisticated constructions exist (e.g., [15]) but they will only marginally improve our results.

The protocol proceeds as follows. First, $P_\mathcal{Y}$ uses the encoding above to convey $\lceil\log \mu_{X|Y}(y)\rceil$. Then the communicators use the protocol of Theorem 2 to convey X to $P_\mathcal{Y}$ as if the maximum ambiguity was $2^{\lceil\log\mu_{X|Y}(y)\rceil}$. The total number of bits transmitted is at most

$$\begin{aligned}
\bar{L}_\phi &\leq \sum_{(x,y)\in S_{X,Y}} p(x,y)\Big(\log\mu_{X|Y}(y) + 3\log\Big(1+\log\mu_{X|Y}(y)\Big) + 17\Big) \\
&= \sum_{y\in S_Y} p(y)\Big(\log\mu_{X|Y}(y) + 3\log\Big(1+\log\mu_{X|Y}(y)\Big) + 17\Big) \\
&= \sum_{y\in S_Y} p(y)(H(X|y) + 3\log\left(H(X|y)+1\right) + 17) \\
&\leq \sum_{y\in S_Y} p(y)H(X|y) + 3\log\sum_{y\in S_Y} p(y)(H(X|y)+1) + 17 \\
&= H(X|Y) + 3\log\left(H(X|Y)+1\right) + 17
\end{aligned}$$

where the second inequality uses the convexity of the logarithm. Note that the first two phases of the protocol consist of bits transmitted by $P_\mathcal{Y}$ and are combined into one message.

$\square$

We relate the number of bits required by a four-message protocol to the optimal number.

Corollary 2 For all *uniform* (X, Y) pairs

$$\bar{C}_4(X|Y) \leq \bar{C}_\infty(X|Y) + 3\log\Big(\bar{C}_\infty(X|Y)+1\Big) + 17 \; . \qquad \square$$

Remarks:

1. The results proven in this section apply when the pair (X, Y) is uniform only for each given value of Y, that is,

$$p(x|y) = \begin{cases} \frac{1}{|S_{X|Y}(y)|} & \text{if } x \in S_{X|Y}(y), \\ 0 & \text{if } x \notin S_{X|Y}(y). \end{cases}$$

2. Except for the last theorem and corollary, all other results in this section apply to all random pairs, not only uniform ones.

5 A single-event analog of the Slepian-Wolf theorem

We now consider the modified problem where $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ want to convey X and Y to a third person P. The introduction describes the model in more detail and defines the *region* of *achievable pairs*. We first establish the achievable regions for arbitrary, uniform, and Cartesian-product random pairs. Then we describe the original Slepian-Wolf problem and note some of the ways our results apply to it.

The first result is an analog of results discussed in Section 2.

Corollary 3 For all (X, Y) pairs,

1. Every pair $(C^{\mathcal{X}}, C^{\mathcal{Y}})$ satisfying $C^{\mathcal{X}} \geq H(X) + 1$ and $C^{\mathcal{Y}} \geq H(Y) + 1$ is achievable.

2. All achievable pairs satisfy: $C^{\mathcal{X}} \geq H(X|Y)$, $C^{\mathcal{Y}} \geq H(Y|X)$, and $C^{\mathcal{X}} + C^{\mathcal{Y}} \geq H(X, Y)$.

Therefore, the achievable region is always contained in the single-dashed area in Figure 3a and, up to one additional bit, always contains the double-dashed area.

Proof: Two Huffman codes: one describing X and transmitted by $P_{\mathcal{X}}$, the other transmitted by $P_{\mathcal{Y}}$ and describing Y prove the upper bound. Since we did not formally define protocols, we cannot rigorously prove of the lower bound. The intuitive reasoning, however, is clear. As in Lemma 2, $P_{\mathcal{X}}$ must transmit at least $H(X|Y)$ bits on the average even if P knows Y, similarly for $P_{\mathcal{Y}}$, also, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ together must transmit at least $H(X, Y)$ bits on the average. To turn this reasoning into a proof, one needs to prove the analog of Lemma 1: that no information is gained by the parsing of the sequences. $\square$

The results of Section 3 imply that when the support set of (X, Y) is a Cartesian product, the upper bounds of Corollary 3 are almost tight.

Corollary 4 If the support set of (X, Y) is a Cartesian product then:

1. Every pair $(C^{\mathcal{X}}, C^{\mathcal{Y}})$ satisfying $C^{\mathcal{X}} \geq H(X)+1$ and $C^{\mathcal{Y}} \geq H(Y)+1$ is achievable.

2. All achievable pairs satisfy: $C^{\mathcal{X}} \geq H(X)$ and $C^{\mathcal{Y}} \geq H(Y)$.

Therefore, up to one additional bit, the achievable region is the shaded area in Figure 3b.

Proof: Part (1) is taken from Corollary 3. The proof of Part (2) resembles that of Corollary 1. Even if P knows Y, at least $H(X)$ bits must be transmitted between $P_{\mathcal{X}}$ and P. Similarly for $P_{\mathcal{Y}}$. $\Box$

We use the results of Section 4 to show that if (X,Y) is uniform, the Slepian-Wolf region can be achieved.

Corollary 5 If (X,Y) is uniformly distributed then

1. Every pair $(C^{\mathcal{X}}, C^{\mathcal{Y}})$ satisfying

$$C^{\mathcal{X}} \geq H(X|Y) + 3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 18 \ ,$$

$$C^{\mathcal{Y}} \geq H(Y|X) + 3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 18 \ ,$$

and

$$C^{\mathcal{X}} + C^{\mathcal{Y}} \geq H(X,Y) + 3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 20$$

is achievable.

2. All achievable pairs satisfy: $C^{\mathcal{X}} \geq H(X|Y)$, $C^{\mathcal{Y}} \geq H(Y|X)$, and $C^{\mathcal{X}} + C^{\mathcal{Y}} \geq H(X,Y)$.

Therefore, up to $3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 20$ additional bits, the achievable region is the shaded area in Figure 3c.

Proof: The second claim is taken from Corollary 3; we prove the first. The "corner point,"

$$\pi_1 \stackrel{\text{def}}{=} \left(H(X)+1 \ , \ H(Y|X) + 3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 19\right)$$

is achieved by a simple application of Theorem 2. Using a Huffman code, $P_{\mathcal{X}}$ conveys X to P while transmitting at most $H(X)+1$ bits on the average. Then, $P_{\mathcal{Y}}$ and P use the protocol of Theorem 2 to convey Y to P who already knows X. The other "corner point,"

$$\pi_2 \stackrel{\text{def}}{=} \left(H(X|Y) + 3\log\left(H(X|Y)+1\right) + 3\log\left(H(Y|X)+1\right) + 19 \ , \ H(Y)+1\right) \ ,$$

is achieved similarly. To achieve the point $\alpha \cdot \pi_1 + (1-\alpha)\pi_2 + (1,1)$ on the line connecting $\pi_1 + (1,1)$ and $\pi_2 + (1,1)$, the communicators imitate "time sharing." P flips a coin that is "heads" with probability α and "tails" with probability $1-\alpha$. He transmits the outcome of the coin-flip to $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$. If the coin showed "heads" the communicators use the protocol achieving π_1, otherwise, they use the protocol achieving π_2.

Note however that this protocol is *randomized*. The same input may result in different transmitted bits, depending on the outcome of the coin flip. If randomized protocols are not desirable, the coin-flip effect can be simulated by partitioning $S_{\mathcal{X}}$ into two sets, one with probability of about α, the other with probability of about $1-\alpha$. Now, $P_{\mathcal{X}}$ informs $P_{\mathcal{Y}}$ which set X belongs to. If X is in the first set,they use the protocol achieving π_1, otherwise, they use the protocol achieving π_2. However, due to the granularity of the probability distribution $p(x,y)$, this protocol may require slightly more transmitted bits. □

Comparison with the Slepian-Wolf theorem

As we noted in the introduction, this problem is a single-event analog of one studied by D. Slepian and J. Wolf [2]. There, $(X_1,Y_1),\ldots,(X_n,Y_n)$ is a sequence of independent, identically distributed (*iid*) pairs of random variables. $P_{\mathcal{X}}$ knows $\overline{X} \stackrel{\text{def}}{=} (X_1,\ldots,X_n)$ and $P_{\mathcal{Y}}$ knows $\overline{Y} \stackrel{\text{def}}{=} (Y_1,\ldots,Y_n)$. A person P wants to learn $\overline{X}$ and $\overline{Y}$ with probability at least $1-\epsilon$ of being correct.

To that end, $P_{\mathcal{X}}$, $P_{\mathcal{Y}}$, and P agree on a one-way protocol. When $P_{\mathcal{X}}$ is given $\overline{X}$ and $P_{\mathcal{Y}}$ is given $\overline{Y}$, each transmits to P a single binary message. Neither $P_{\mathcal{X}}$ nor $P_{\mathcal{Y}}$ knows the other's message and no transmission errors are incurred.

A pair $(R^{\mathcal{X}}, R^{\mathcal{Y}})$ of real numbers (*rates*) is *achievable* if for every permissible error, $\epsilon > 0$, there is an integer n for which the expected[9] number of bits transmitted by $P_{\mathcal{X}}$, normalized by n, is at most $R^{\mathcal{X}}$ and the expected number of bits transmitted by $P_{\mathcal{Y}}$, normalized by n, is at most $R^{\mathcal{Y}}$. The Slepian-Wolf Theorem says that the *achievable region,* the closure of the set of achievable rate pairs, is the set shown in Figure 5a:

$$\{(R^{\mathcal{X}}, R^{\mathcal{Y}}) : R^{\mathcal{X}} \geq H(X_1|Y_1), R^{\mathcal{Y}} \geq H(Y_1|X_1), \text{ and } R^{\mathcal{X}} + R^{\mathcal{Y}} \geq H(X_1,Y_1)\} .$$

Our result demonstrate that with interaction (transmissions from P to $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$) roughly the same region is achievable even in the single-event analog of this problem, provided that the random pair is uniformly distributed. Additionally, our results have following implications on the original Slepian-Wolf problem.

[9]The Slepian-Wolf theorem applies to both expected and worst-case number of bits. For comparison with our results, it is convenient consider expectations.

Error-free Slepian-Wolf

Suppose that the support set of (X_1, Y_1) (and, by iid'ness, of each (X_i, Y_i)) is a Cartesian product, and that P wants to know $\overline{X}$ and $\overline{Y}$ with no probability of error ($\epsilon = 0$). It is easy to see that $S_{\overline{X},\overline{Y}}$ is a Cartesian product. Therefore, Corollary 4 implies that *even with interaction* the achievable region is as shown in Figure 5b:

$$\{(R^{\mathcal{X}}, R^{\mathcal{Y}}) : R^{\mathcal{X}} \geq H(X_1), R^{\mathcal{Y}} \geq H(Y_1)\} \ .$$

Therefore, in the Slepian-Wolf problem, admission of $\epsilon > 0$ error is essential to achieving the Slepian-Wolf region of figure 5a; when $\epsilon = 0$, the achievable region is smaller: that of figure 5b.

Remark: This contrasts with communication of a single sequence $X_1, \ldots, X_n$ to P. There, an expected rate of $H(X_1)$ bits per source symbol is necessary and sufficient regardless of whether errors are permitted or not, hence allowing error does not increase the set of achievable rates.

ϵ-error Slepian-Wolf

Suppose, as in the original Slepian-Wolf problem, that $\overline{X}$ and $\overline{Y}$ are arbitrary (their support set is not necessarily a Cartesian product) and that $\epsilon > 0$ error is allowed. For simplicity, we analyze the case where $P_{\mathcal{X}}$ wants to convey X to $P_{\mathcal{Y}}$ who knows Y. Similar statements hold when $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ want to convey X and Y to P.

The standard proof of the Slepian-Wolf theorem (cf. [16]), proceeds as follows. As n increases, $(\overline{X}, \overline{Y})$ approaches a uniform distribution over a set of "jointly typical sequences." Therefore, for large enough values of n, the probability that $(\overline{X}, \overline{Y})$ is not jointly typical is smaller than $\epsilon/2$. Assuming that $(\overline{X}, \overline{Y})$ is jointly typical, the set of all $\overline{X}$s is partitioned into subsets called *bins*. Given $\overline{X}$, $P_{\mathcal{X}}$ transmits its bin to $P_{\mathcal{Y}}$. $P_{\mathcal{Y}}$ decides that $\overline{X}$ is a sequence in that bin which is jointly typical with $\overline{Y}$. $P_{\mathcal{Y}}$ can err only if there is more than one sequence in $\overline{X}$'s bin that is jointly typical with $\overline{Y}$. If the number of bins is large enough then, for almost all choices of bin partitions, this happens with probability of at most $\epsilon/2$.

We can use our results to construct a protocol that commits errors of the first kind only (those resulting from occurrence of non typical $(\overline{X}, \overline{Y})$ sequences) and still requires at most $H(X_1|Y_1)$ bits per source symbol. As $(\overline{X}, \overline{Y})$ is uniformly distributed over the typical set, Theorem 2 implies that $\overline{X}$ can be communicated to $P_{\mathcal{Y}}$ while the expected number of bits exchanged by $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ is at most

$$H(\overline{X}|\overline{Y})+3\log\left(H(\overline{X}|\overline{Y})+1\right)+17\leq nH(X_1|Y_1)+3\log n+3\log\left(H(X_1|Y_1)+1\right)+17\ .$$

Normalized by n, the expected number of bits exchanged by $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ per source symbol is at most

$$H(X_1|Y_1)+3\frac{\log n}{n}+\frac{3\log\left(H(X_1|Y_1)+1\right)+17}{n}\ .$$

Therefore, the rate $H(X_1|Y_1)$ is in the achievable region even if we require error-free communication of jointly-typical sequences. Note that, in this case, all of $P_{\mathcal{X}}$'s ambiguity sets are of roughly the same size, hence our techniques can be simply modified to show that this bound applies to the worst-case number of bits as well. (We can derive a worst-case bound even when both $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ want to convey X and Y to P: there is a sequence of random pairs, hence we can use real time sharing to avoid the expectation resulting from the coin flip.)

In order to achieve error-free communication, we allowed interaction: transmissions from $P_{\mathcal{Y}}$ to $P_{\mathcal{X}}$. With interaction allowed, the standard binning protocol can be modified to achieve error-free communication of jointly-typical $(\overline{X},\overline{Y})$ sequences as well. $P_{\mathcal{Y}}$ checks if the bin containing $\overline{X}$ contains more than one sequence jointly-typical with $\overline{Y}$. If so, he asks $P_{\mathcal{X}}$ to transmit a message that identifies $\overline{X}$. The rationale is that with high probability the bin containing $\overline{X}$ contains no other sequences jointly typical with $\overline{Y}$, therefore, additional transmissions are unlikely. Yet, if the standard random-binning analysis is applied to this protocol, it yields a disappointing upper bound: $2H(X|Y)$. The remark below draws a further distinction between our method and this simple modification of random-binning.

The preceding discussion assumes that interaction is allowed. What if communication is permitted only from $P_{\mathcal{X}}$ to $P_{\mathcal{Y}}$? Could we still communicate the jointly-typical sequences with no errors and achieve average rate of $H(X_1|Y_1)$ bits per symbol? Not always. The lower bound argument for $\bar{C}_1(K|I,J)$ in Example 1 can be modified to show that in many cases, an average rate of $H(X_1)$ bits per symbol is required when no interaction is allowed.

Remark: To demonstrate why 'random-binning with simple correction' is not effective for the current problem, consider applying it to the League problem of Example 1.

Assume that the t teams are partitioned into b random bins. With probability $1/b$, the two playing teams belong to the same bin. With that probability, $P_{\mathcal{Y}}$ cannot determine the winner, hence $P_{\mathcal{X}}$ has to transmit further messages identifying the winner. With a simple-minded approach where $P_{\mathcal{X}}$ then transmits $\lceil\log t\rceil$ additional bits, the expected number of bits is roughly $\log b + \frac{\lceil \log t \rceil}{b}$; arbitrarily more than $\bar{C}_2(K|I,J)$ which Example 1 showed is at most 3.

6 Open Problems

We mention two open problems.

1. Our bounds on $\bar{C}_\infty(X|Y)$ are tight in only two cases: when (X,Y) is uniform, and when its support set is a Cartesian product. For other random pairs, these bounds may be far asunder.

   Can tight bounds on $\bar{C}_\infty(X|Y)$ be derived for arbitrary random pairs? Can bounds be derived if some $\epsilon > 0$ probability of error is allowed?

2. In Equation (10), we have shown that if (X,Y) is uniform then

$$\bar{C}_2(X|Y) \le 2H(X|Y) + 2\log\left(H(X|Y)+1\right) + 7.5 \ .$$

   Can the factor of two be improved, or is there a family of (X,Y) pairs with increasing $H(X|Y)$ such that for each pair in the family,

$$\bar{C}_2(X|Y) \ge 2H(X|Y) - o(H(X|Y) \ ?$$

   Note that for *worst-case* complexities there are random pairs for which two messages require twice the minimum number of bits (cf., [5]).

7 Acknowledgements

I thank Abbas El Gamal for many ideas emanating from our joint work on [1] that have been put to use here and Max Costa for sharing his league-problem protocol. I also acknowledge helpful discussions with Larry Shepp and Aaron Wyner and a thorough review by Sergio Verdú.

References

[1] 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.

[2] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19:471–480, 1973.

[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] M. Costa. Private Communication, 1989.

[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. Interactive communication of balanced distributions and of correlated files. *SIAM Journal of Discrete Mathematics*, 6(4):548–564, November 1993.

[7] 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.

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

[9] 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.

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

[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] P. Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 21(2):194–203, March 1975.

[16] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, 1991.

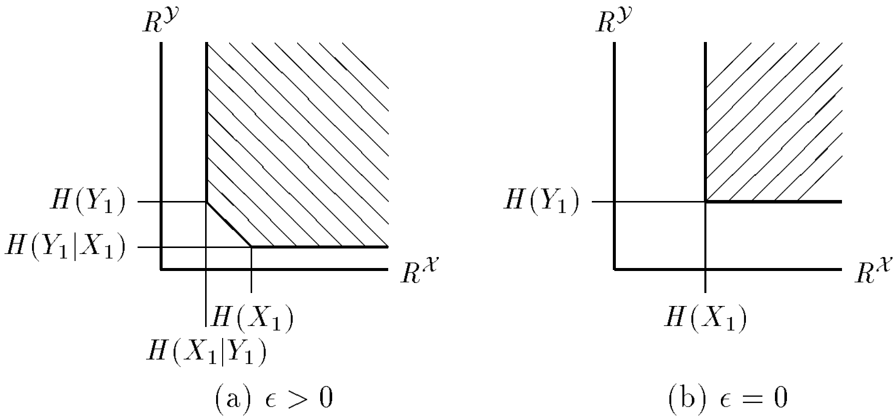


Figure 5: Achievable regions for the Slepian-Wolf Theorem