List Decoding with Side Information

Venkatesan Guruswami

Department of Computer Science
University of Washington
Seattle, WA 98195.
venkat@cs.washington.edu

Abstract

Under list decoding of error-correcting codes, the decoding algorithm is allowed to output a small list of codewords that are close to the noisy received word. This relaxation permits recovery even under very high noise thresholds. We consider one possible scenario that would permit disambiguating between the elements of the list, namely where the sender of the message provides some hopefully small amount of side information about the transmitted message on a separate auxiliary channel that is noise-free. This setting becomes meaningful and useful when the amount of side information that needs to be communicated is much smaller than the length of the message.

We study what kind of side information is necessary and sufficient in the above context. The short, conceptual answer is that the side information must be randomized and the message recovery is with a small failure probability.

Specifically, we prove that deterministic schemes which guarantee correct recovery of the message provide no savings and essentially the entire message has to be sent as side information. However, there exist randomized schemes which only need side information of length logarithmic in the message length. In fact, in the limit of repeated communication of several messages, the amortized amount of side information needed per message can be a constant independent of the message length or the failure probability. Concretely, we can correct up to a fraction $(1/2 - \gamma)$ *of errors for binary codes using only* $2\log(1/\gamma) + O(1)$ *amortized bits of side information per message, and this is in fact the best possible (up to additive constant terms).*

1 Introduction

1.1 Problem Statement and Motivation

Consider the problem of communicating a message x (say, a binary string of length k) across a noisy channel so that errors effected on any adversarially chosen fraction p of the transmitted symbols can be corrected. Error-correcting codes provide a way to deal with this problem: the message x is first encoded into a longer string $C(x) \in \{0,1\}^n$ which is the transmitted on the channel. For two n-bit strings y_1 and y_2, define $\delta(y_1, y_2)$ to be the fraction of coordinates where they differ. If the received string y (a potentially corrupted version of $C(x)$) satisfies $\delta(y, C(x)) \leq p$, then we wish to be able to correctly determine x. An error-correcting code C is said to have relative distance γ if for any two distinct messages x_1, x_2, their encodings $C(x_1)$ and $C(x_2)$ differ in at least a fraction γ of positions. It is well-known and simple to see that an error-correcting code of relative distance more than $2p$ is necessary and sufficient to solve the above problem of handling an adversarial noise threshold of p (of course one would also like the code to have efficient procedures for performing the necessary encoding and decoding operations).

For concreteness, we focus on binary codes in this paper. By a well-known bound in coding theory, a binary error-correcting code of relative distance more than $1/2$ can encode at most a constant number of messages. Therefore, it seems that the largest noise threshold we can handle is a fraction $1/4$ of errors. However, a relaxation of the decoding requirement called list decoding allows one to go beyond the half-the-distance bottleneck and handle noise levels $p > 1/4$. Under list decoding, the decoding is considered successful if the receiver is able to output a small list of messages which includes the original transmitted message. We refer the reader to the survey by Sudan [15] for background and pointers on list decoding.

A code $C : \{0,1\}^k \to \{0,1\}^n$ is said to be (p, L)-list decodable if for every $y \in \{0,1\}^n$, the set $\{x \in \{0,1\}^k : \delta(y, C(x)) \le p\}$ has at most L elements. For every $p < 1/2$, explicit constructions of families of (p, L_p)-list decodable codes are known for a fixed constant L_p that depends only on p and *not* on the message length (the exact dependence is $L = O(1/\gamma^2)$ when $p = (1/2 - \gamma)$ for any constant $\gamma > 0$). Moreover, there are constructions known for which there is a polynomial time algorithm to find and output the list of at most L codewords that differ from y in a fraction p or less of positions. Therefore, even when constrained to output a relatively small list, list decoding permits one to efficiently decode up to any fraction $p < 1/2$ of errors.

Of course we have (inevitably) lost the luxury of obtaining an unambiguous decoded message, but in many applications reducing the candidate messages to belong to a small list suffices. However one can imagine settings, including the basic communication problem mentioned above, where one would like to unambiguously decode the correct message. Presumably, one can disambiguate between elements in the list using some context information and thereby pick the correct message from the list. In this paper, we formalize and study one scenario, called "list decoding with side information", in which this becomes possible. We assume that the sender can send some side information on an auxiliary channel which is noise-free, and the hope is that the receiver can disambiguate between elements of the list based on this information.[1] Now the sender can simply send the original message itself as side information thereby trivializing the decoding problem. But this setting becomes meaningful and useful if there are schemes where the amount of side information communicated is much shorter than the message length.

In this paper, we study what kind of and how much side information is required to unambiguously pin-point the correct codeword among the candidates obtained by list decoding. We mention that our results, both positive and negative, are generic in the sense that they apply to any error-correcting code and do not refer to any specifics of the code itself (beyond its (p, L)-list decodability property).[2] We describe our results next and then discuss the relation to well-studied questions in communication complexity.

1.2 Our Results

The conceptual or qualitative message of our work can be captured in one sentence: for a small amount of side information to be useful, it must be randomized, and recovery of the correct codeword occurs with some small failure probability. We describe the quantitative aspects of our work below. Throughout the paper, we will use S and R to refer to the sender and receiver respectively.

We prove that for deterministic schemes in which we require that R be *always* able to decode the correct message, essentially the entire message has to be sent as side information and thus list decoding offers no significant savings. Specifically, for k-bit messages, we prove a lower bound of $(k - c_p)$ for a constant c_p that depends only on p.

However, we present probabilistic polynomial time strategies for S and R using at most $2\log(kL/\varepsilon) + O(1)$ bits of side information that permits R to recover the correct message with probability at least $(1-\varepsilon)$. Moreover, our scheme has the nice property that R never makes a mistake in decoding; formally, either R correctly decodes the message that was sent by S, or at worst detects that it cannot do so and reports a failure and just returns the "standard list" (the latter failure event happening with probability at most ε). The amount of side information communicated in our scheme is is tight up to constant factors as we show a lower bound of (approximately) $\max\{\log k, \log L, \log(1/\varepsilon)\}$ for randomized schemes. (The most "interesting" of these lower bounds is the one in terms of the failure probability ε; in particular, if one seeks failure probability that is exponentially small in k, then one has to communicate $\Omega(k)$ bits of side information.)

We also consider (in Section 3) the repeated communication setting where a large number N of messages $x_1, x_2, \ldots, x_N$ are each encoded by C and transmitted. Suppose R gets received words $y_1, y_2, \ldots, y_N$ where $\delta(y_i, C(x_i)) \le p$ for every i. The goal for R is to correctly recover all the N messages $x_1, x_2, \ldots, x_N$. Note that we are assuming that no more than a fraction p of errors occur in the encoding of each x_i. We are interested in understanding the amount of side information needed per message in the limit of large N. One possibility would just be to use the one-message scheme N times (with separate side information for each of the N messages), but it is known that in certain situations sig-

[1] The noise-freeness assumption might seem arbitrary, but we can imagine the side channel to be an expensive one with very low noise thresholds which we will use to only send a small number of bits. Moreover, we can always simulate a noiseless channel from one with noise threshold less than $1/4$ by just using unambiguous decoding on that channel; this will only incur a constant factor overhead in the side information to be communicated.

[2] For the negative results we study binary codes for noise levels $p > 1/4$ since for smaller values of p one can just pick a code that permits unambiguous decoding up to radius p. For positive results, when we claim probabilistic polynomial time algorithms for S and R, we assume that the code C is *efficiently* (p, L)-list decodable, i.e. there exists a polynomial time list decoding for C that corrects up to a fraction p of errors.

nificant savings can be obtained in the repeated communication setting. This motivates the question of whether one can come up with cleverly chosen side information that depends on the whole sequence of messages and do much better compared to using the one-message side information N times over.

Again, for deterministic schemes, we show that nothing significantly better than simply sending the N messages can be achieved; specifically we show that at least $\Omega(kN)$ bits of side information, or in other words $\Omega(k)$ bits of side information per message, are required. We can also show a stronger lower bound of $(k - O(1))N$ under certain special conditions.

We also give a randomized scheme where S and R are both probabilistic polynomial time procedures that works with only $\log L + O(1)$ bits of side information per message in an amortized sense. In particular, in the limit of communicating a large number of messages, the per message overhead is therefore a constant independent of the message length k (assuming the list size L is a constant that does not depend on k). In fact, up to additive $O(1)$ terms, the bound on amortized bits of side information that we achieve is the best possible ($logL - o(1)$ is a lower bound). It is interesting to note that the amortized cost is independent of the failure probability ε, and once again our scheme has the desirable property of never decoding incorrectly, i.e. either R correctly decodes all the x_i's or discovers, with probability at most ε, that it cannot do so.

1.3 Techniques

Our lower bound technique is based on a simple relation of the side information problem to the chromatic number of certain graphs which capture when a pair of messages (or message sequences, for the repeated communication setting) can be "confused" with one another. To get the best constants in the lower bound of $\Omega(kN)$ for deterministic schemes in the repeated communication setting, we make use of certain properties of the Lovász theta function of a graph and in particular an upper bound for the theta function in terms of the maximum independent set size due to Alon and Kahale [3].

Our randomized schemes are obtained using hash families with low collision probability for subsets of certain size of the set of messages (or message sequences, for the repeated communication setting).

1.4 Relation to Communication Complexity

Our basic set-up is an instance of the general problem of communication complexity with partial information (cf. [9, Chap. 4]) which is defined as follows. Two parties Alice and Bob possess x and y respectively which satisfy $(x, y) \in D$ for some relation D known to both parties. The aim is for Bob to learn x using as little communication as possible. In our setting, we also restrict the communication to consist of just one message from Alice to Bob (though Section 4 briefly discusses the multiple rounds case).

Questions similar to the ones we study have been investigated in this more abstract setting [16, 13, 12, 5] (see [1] for a nice survey of the various results). The proofs of the "general" communication complexity results we need are all rather simple, so for sake of readability, we just go ahead and give complete proofs. Some of the results concerning randomized communication complexity with partial information do not seem to have been stated explicitly in the literature, the only work along this line being [13].

1.5 Notation

As defined earlier, $\delta(y, y')$ denotes be the fractional distance between two strings y and y'. Denote $[n]$ to be the set $\{1, 2, \ldots, n\}$. We use $\log a$ to denote the logarithm of a to the base 2. We assume the side information sent by S is always of the same length. This is only for convenience: the schemes we construct will have this property and requiring this property, which is equivalent to a "prefix-freeness" property if we allow communication of varying length strings, will affect the lower bounds by only 1 bit.

2 Communicating one message

2.1 Deterministic schemes do not help

Suppose the sender S communicated $x \in \{0,1\}^k$ by encoding it by an $[n, k]$ (p, L)-list decodable code C, and $y \in \{0,1\}^n$ was received by R. Here, the noise parameter p satisfies $1/4 < p < 1/2$, and we are thinking of $L = L_p$ as a constant that only depends on p. The goal is for S to send some side information I so that R can figure out x from y and I with certainty whenever y satisfies $\delta(y, C(x)) \leq p$. It is a well-known fact in the communication complexity literature (cf. [9, Sec. 4.7]) that the amount of information needed in a setting like ours is exactly $\lceil \log \chi \rceil$ bits where χ is the chromatic number of a certain graph. Since this fact is simple to prove, below we give a self-contained proof.

Theorem 1 *For the setting as above, for $p > 1/4$, S needs to send at least $k - \log\left(\frac{4p}{4p-1}\right)$ bits of side information I if R can always correctly determine x from any y which satisfies $\delta(C(x), y) \leq p$ and I.*

Proof: Since S is unaware of y, its strategy is simply to send auxiliary information, say $f(x)$, that depends only on x. Now suppose $x, x' \in \{0,1\}^k$ are two messages such that $\delta(C(x), C(x')) \leq 2p$. Then clearly there exists a y for which $\delta(C(x), y) \leq p$ and $\delta(C(x'), y) \leq p$. Now if $f(x) = f(x')$, then the receiver R when he has input y cannot tell whether the original message was x or x'. Hence we must have $f(x) \neq f(x')$ whenever $\delta(C(x), C(x')) \leq 2p$. Consider a graph with vertex set $\{0,1\}^k$ and where (x, x') is an edge whenever $\delta(C(x), C(x')) \leq 2p$. We have argued that f gives a legal coloring of G.

We now prove that G has large chromatic number and hence the amount of auxiliary information has to be large. Let J be an independent set in G. For every $x, x' \in J$, we have $\delta(C(x), C(x')) > 2p > 1/2$. Thus the set $\{C(x) \mid x \in J\}$ forms an error-correcting code of relative distance at least $2p > 1/2$. By a well-known bound in coding theory (the Plotkin bound), this implies that $|J| < 4p/(4p-1)$. Hence the size of the largest independent set in G, $\alpha(G)$, satisfies $\alpha(G) < 4p/(4p-1)$. Consequently the chromatic number of G, $\chi(G)$, satisfies $\chi(G) \geq 2^k \left(\frac{4p-1}{4p}\right)$.

Since $\{f(x) : x \in \{0,1\}^k\}$ has at least $\chi(G)$ elements by the earlier argument, we get a lower bound of $\log \chi(G) \geq k - \log\left(\frac{4p}{4p-1}\right)$ on the number of bits of side information that S must send. $\square$

2.2 A Randomized Scheme with Low Communication

We now prove that if we allow a small error probability with which R might not recover the correct message x and instead report a decoding failure, then there are schemes in which the sender sends side information that is significantly shorter than the message length k. As before S communicates a message x by encoding it by an (p, L)-list decodable code C of block length n and R receives a string $y \in \{0,1\}^n$ such that $\delta(y, C(x)) \leq p$, and the goal is to recover x.

Theorem 2 *For every $p < 1/2$ and every $\varepsilon > 0$, there are probabilistic polynomial time strategies for S and R under which S sends $2\log k + 2\log L + 2\log(1/\varepsilon) + O(1)$ bits of side information which enables R to recover the correct x with probability at least $(1-\varepsilon)$ when it receives an arbitrary y that satisfies $\delta(y, C(x)) \leq p$. Furthermore, if unable to recover the correct x, R will report a failure (as opposed to outputting a wrong message).* [3]

The proof is based on a family of hash functions guaranteed by the following lemma.

Lemma 1 *For every integer m the following holds: for all integers ℓ and for all $\varepsilon > 0$ (ℓ, ε are allowed to depend on m), there exists an integer $q \leq 2m\ell/\varepsilon$ and an explicitly constructible family of hash functions $\mathcal{F}(m, \ell, \varepsilon) = \{f : \{0,1\}^m \to [q]\}$ of size q such that for every $x \in \{0,1\}^m$ and subset $\{z^{(1)}, z^{(2)}, \ldots, z^{(\ell-1)}\} \subseteq \{0,1\}^m \setminus \{x\}$,*

$$\mathbf{Pr}\left[f(x) \notin \{f(z^{(1)}), f(z^{(2)}), \ldots, f(z^{(\ell-1)})\}\right] \geq 1-\varepsilon\,,$$

where the probability is taken over a uniformly random choice of f from $\mathcal{F}(m, \ell, \varepsilon)$.

Constructions of hash functions as above are well-known: for example, one can take a Reed-Solomon code $C_{\mathrm{RS}} : \{0,1\}^m \to \mathbb{F}_q^q$ over a field with q elements (where q is a power of two) of block length q, dimension $m/\log q$ and relative distance at least $(1 - m/q) \geq (1 - \frac{\varepsilon}{2\ell})$. The i'th function in the hash family, for $1 \leq i \leq q$, is then defined by $f_i(x) = C_{\mathrm{RS}}(x)_{|i}$, i.e. the i'th symbol of the encoding of x by C_{RS}.

Proof of Theorem 2: Let x be the message that S wishes to communicate and let R receive y where $\delta(y, C(x)) \leq p$. Since C is a (p, L)-list decodable code, R can compute in polynomial time a list $\{z^{(1)}, z^{(2)}, \ldots, z^{(\ell)}\}$ of all messages whose encodings are within fractional distance p of y for some $\ell \leq L$. Clearly x will be included in this list and let j^*, $1 \leq j^* \leq \ell$ be such that $z^{(j^*)} = x$ (of course neither S nor R know j^*).

The strategy for S is quite simple: S picks a function f from the hash family $\mathcal{F}(k, L, \varepsilon)$ given by Lemma 1 at random and sends $\langle f, f(x)\rangle$ to R as side information on the error-free auxiliary channel. The strategy for R is now the following. It checks if there is a unique $j \in \{1, 2, \ldots, \ell\}$ for which $f(z^{(j)}) = f(x)$; if so it outputs $z^{(j)}$, and otherwise it outputs a decoding failure. Note that since $x \in \{z^{(1)}, z^{(2)}, \ldots, z^{(\ell)}\}$, it follows that whenever R outputs an answer it is the correct one. By the property of the hash family, with probability at least $(1 - \varepsilon)$ over the choice of f, we have that $f(x) = f(z^{(j^*)})$ is distinct from all values in $\{f(z^{(i)}) \mid i \neq j^*\}$, and therefore R recovers the correct message successfully with probability at least $(1 - \varepsilon)$.

The proof is now complete by noting that the number of bits communicated as side information equals $\log|\mathcal{F}(k, L, \varepsilon)| + \log q = 2\log q = 2\log k + 2\log(L/\varepsilon) + O(1)$, as claimed. $\square$

2.3 Lower bound for randomized schemes

We now show that the bound on side information achieved by Theorem 2 is the best possible up to constant factors. Specifically, we prove the following.

[3] By using a different construction of the required hash functions, one can also achieve a side-information communication complexity of $\log k + 4\log(L/\varepsilon) + O(1)$ bits.

Theorem 3 *Let $p > 1/4$. Assume that there is a probabilistic strategy for S to send randomized side information $I(x)$ consisting of b bits based on its message x such that for every x, given y where $\delta(y, C(x)) \le p$ and $I(x)$, the receiver R can correctly determine x with probability at least $(1-\varepsilon)$ where $2^{-\Omega(k)} \le \varepsilon < 1/2$. Then*

$$b \ \ge \ \max\Big\{\log\Big(\frac{k}{\log k}\Big) - \log\log\frac{1}{(1/2-\varepsilon)} - O(1), \quad \log L - \log\Big(\frac{1}{1-\varepsilon}\Big), \ \log(1/\varepsilon) - O(1)\Big\} \quad (1)$$

where L is smallest integer for which C is (p, L)-list decodable.

Remark: We are typically thinking of L as a fixed constant even as the message length k increases. However, this need not be the case and one can imagine list decoding with lists whose size depends on k (eg. is a polynomially growing function of k). In such cases the second lower bound of $\log L$ in (1) can be better. In particular, it shows that when R spends no effort in performing list decoding at all, i.e., the case where $L = 2^k$, one has to essentially send the entire message as the side information. Thus randomization alone cannot solve the problem, but together with good list decodable codes, it does the job with significantly shorter side information.

The third lower bound above of $\log(1/\varepsilon) - O(1)$ is the most "interesting" one. The first lower bound is an instance of the general principle that the randomized communication complexity is at least logarithmic in the deterministic communication complexity. The second lower bound follows form the fact that the side information that S sends must enable R to pin-point the correct message among L choices, and therefore one cannot get away with much less than $\log L$ bits of communication. For completeness, we include a proof of all three lower bounds, though only the third one is novel to our setting.

Proof of Theorem 3:

<u>Lower Bound 1</u>: Suppose there is a randomized scheme that uses b bits. We will obtain a deterministic scheme that uses side information of about 2^b bits and then appeal to Theorem 1 to obtain a lower bound on b. The idea is the following (and is quite standard). Suppose S sent x and wants to convey this to R by sending as side information a string z picked randomly according to a distribution D_x on $\{0,1\}^b$. We can construct a deterministic scheme where S sends the value of $D_x(z)$ for every $z \in \{0,1\}^b$ and R can then compute the probability that it would have output each of the elements in its list (i.e. the messages $x^{(1)}, x^{(2)}, \ldots, x^{(L)}$ whose encodings are close to y) and choose the one for which this probability is at least $(1-\varepsilon)$. Of course sending the distribution exactly is not possible (or necessary) and communicating each probability with precision of $t = \log(1/2-\varepsilon)^{-1} + b + 1$ bits suffices. This guarantees that the value sent for each probability differs from the true value by at most $(1/2-\varepsilon)2^{-b-1}$. Therefore, the probability computed by R will differ from the actual value by less than $(1/2-\varepsilon)$, and therefore if R picks the $x^{(i)}$ which has probability greater than $1/2$, this will be the correct message that S transmitted. Thus there is a deterministic scheme in which S communicates at most $2^b(\log(1/2-\varepsilon)^{-1} + b + 1)$ bits of side information. Since this quantity has to be at least $k - O(1)$ by Theorem 1, we get the desired lower bound on b.

<u>Lower Bound 2</u>: We now prove the second lower bound: $b \ge \log L + \log(1-\varepsilon)$. In fact this lower bound holds even if S knows the string y that R receives. By the definition of L there must be a $y \in \{0,1\}^n$ such that a Hamming ball of radius pn around y contains exactly L codewords of C, say $C(x^{(1)}), C(x^{(2)}), \ldots, C(x^{(L)})$. In the case when R receives y, S must send side information which will permit R to recover the correct $x^{(i)}$ among the L possibilities with probability at least $(1-\varepsilon)$. To give a lower bound, it suffices to consider a deterministic S and show a lower bound on the amount of information S must communicate if R recovers the correct $x^{(i)}$ with probability $(1-\varepsilon)$ when the probability is taken over the choice of $x^{(i)}$ picked uniformly at random from $\{x^{(1)}, x^{(2)}, \ldots, x^{(L)}\}$. But in this case we have a deterministic scheme which communicates $(1-\varepsilon)L$ of the $x^{(i)}$'s correctly. Clearly this means that S must communicate at least $\log L + \log(1-\varepsilon)$ bits.

<u>Lower Bound 3</u>: We now prove that $b \ge \log(1/\varepsilon) - O(1)$ when $\varepsilon \ge 2^{-\Omega(k)}$. Since there is only one round of communication, S's strategy can be abstracted via a function H as follows: S picks a string r uniformly at random from $\{0,1\}^M$ for some integer M and uses it to (deterministically) compute the string $z = H(x, r) \in \{0,1\}^b$ that is sent as the side information. We will prove the claimed lower bound under the "public coins" model where we assume that R knows S's random coin tosses r for free. Clearly this only makes the lower bound proof stronger. Define $Q = 2^b$, and we view strings in $\{0,1\}^b$ as elements of $[Q]$ as per some fixed encoding.

Now define the function $E : \{0,1\}^k \to [Q]^{2^M}$ as follows: $E(x) = \langle H(x,r)\rangle_{r\in\{0,1\}^M}$. We will prove that E has a "code-like" property which will imply that its "alphabet size" Q must be $\Omega(1/\varepsilon)$, and therefore $b = \log Q$ must be at least $\log(1/\varepsilon) - O(1)$.

Suppose $x, x' \in \{0,1\}^k$ are such that $\delta(C(x), C(x')) \le 2p$. Then there exists $y \in \{0,1\}^n$ for which $\delta(C(x), y) \le p$ and $\delta(C(x'), y) \le p$. Now consider the following two situations: (i) S wishes to communicate message x (and sends $C(x)$) and R receives y, and (ii) S wishes to communicate message x' (and sends $C(x')$) and R receives y. Since R can correctly determine x and x' respectively in these situations with probability at least $(1-\varepsilon)$, in particular R can distinguish between these two cases with probability at least $(1-\varepsilon)$ over the choice of (the public random string) r. For each random choice r for which $H(x,r)=H(x',r)$, R must make a mistake on at least one of the situations (i) or (ii). It follows that $H(x,r)\neq H(x',r)$ for at least a $(1-2\varepsilon)$ fraction of values of r. Or in other words, $\delta(E(x),E(x'))\geq 1-2\varepsilon$ whenever $\delta(C(x),C(x'))\leq 2p$.

Now consider the graph G with vertex set $\{0,1\}^k$ defined in the proof of Theorem 1. Vertices x and x' were adjacent in G iff $\delta(C(x),C(x'))\leq 2p$, and we proved that $\alpha(G)<4p/(4p-1)$. By a well-known Ramsey-theoretic fact (namely that the clique number $\omega(G)\geq m^{1/a}$ in an m-vertex graph if $\alpha(G)\leq a$), it follows that there is a clique Π in G of size at least $2^{k(4p-1)/4p}$. Consider the encoding E when its inputs are restricted to elements of Π, call it E_Π. Then E_Π defines an error-correcting code containing at least $2^{k(4p-1)/4p}$ elements over alphabet size Q and the code has relative distance at least $(1-2\varepsilon)$.

We will now prove that $Q\geq\frac{1}{4\varepsilon}$. Suppose not, and $Q<\frac{1}{4\varepsilon}$. In this case, by the Plotkin bound, the number of codewords in E_Π is at most $\frac{Q(1-2\varepsilon)}{1-2Q\varepsilon}\leq 2Q(1-2\varepsilon)\leq 2Q$.[4] But we know that E_Π has at least $2^{k(4p-1)/4p}$ codewords. This implies that

$$\frac{1}{4\varepsilon}>Q\geq 2^{\frac{k(4p-1)}{4p}-1}\ .$$

Therefore if $\varepsilon\geq 2^{-k(4p-1)/4p}$, we would have a contradiction. We conclude that $Q\geq 1/(4\varepsilon)$ if $\varepsilon\geq 2^{-k(4p-1)/4p}$, and therefore $b=\log Q\geq\log(1/\varepsilon)-O(1)$ under the same condition. □

3 Repeated Communication

We now consider the situation of repeated communication where S desires to communicate N k-bit messages $x_1,x_2,\ldots,x_N$ to R. As before, on the "main" noisy channel, S sends the encodings of the messages $C(x_1),C(x_2),\ldots,C(x_N)$ by a (p,L)-list decodable and R gets a sequence of received words $y_1,y_2,\ldots,y_N$, where for *each* i, $1\leq i\leq N$, y_i satisfies $\delta(y_i,C(x_i))\leq p$. The goal is for R to determine *all* the x_i's correctly using some side information that S sends. We refer to this as the *N-repeated communication problem for the code C* and will use this notation throughout this section.

Now, why do we worry about this setting instead of treating the N messages together as one message? This is in order to avoid the delay that will be involved in receiving messages if all of them are treated as a single unit and then encoded. The receiver can begin list decoding the messages as and when it receives them instead of waiting for the entire string. For the messages whose encodings arrive with no (or even very small fraction of) errors, R can recover them right away. In general this is the reason why one studies block codes with the understanding that long messages will be broken up into blocks and then each of these blocks encoded and communicated.

Why is this not just repeating the one-message communication scheme N times? Because the side information can depend on the whole sequence of messages. Of course, this means that R has to wait for the side information till it receives all the encoded messages, but this information will (hopefully) be much shorter in length and can be used to quickly disambiguate between elements in each of the already decoded lists (and it could even be that for several of the messages, y_i was very close to $C(x_i)$ and x_i was already unambiguously decoded correctly).

We mention here that questions of this nature are called "direct-sum" problems in communication complexity literature. In the context we are interested in, similar questions have been studied by Feder *et al* [5] and Naor *et al* [12].

3.1 Lower bound for deterministic schemes

We have shown that for communicating just one message one needs to send essentially the entire message as side information. We now show that even if N messages have to be sent (and they are known in advance), deterministic schemes will have to communicate $\Omega(k)$ bits of side information per message.

Theorem 4 *Consider the N-repeated communication problem for a (p,L)-list decodable code C where R is guaranteed to receive $y_1,y_2,\ldots,y_N$ such that $\delta(y_i,C(x_i))\leq p$ for every i, $1\leq i\leq N$. Suppose that there is a deterministic scheme in which S sends B bits of side information using which R can correctly determine all the x_i's from the y_i's. Then $B\geq N\Big(2k\left\lceil\frac{4p}{4p-1}\right\rceil^{-1}-O(1)\Big)$.*

Proof: Recall the graph G defined in the proof of Theorem 1. Now define the "N'th AND product" $G^{\wedge N}$ of G as follows. The vertex set of $G^{\wedge N}$ is $\big(\{0,1\}^k\big)^N$ and two distinct N-tuples $\vec{x}=\langle x_1,x_2,\ldots,x_N\rangle$ and

[4]The Plotkin bound states that if a code over alphabet size Q has relative distance δ where $\delta>1-1/Q$, then the number of codewords in such a code is at most $\frac{Q\delta}{Q\delta-Q+1}$.

$\vec{z} = \langle z_1, z_2, \ldots, z_N \rangle$ are adjacent in $G^{\wedge N}$ if for *every* i, either $x_i = z_i$ or (x_i, z_i) is an edge in G.

If $\vec{x}$ and $\vec{z}$ are adjacent in $G^{\wedge N}$, then clearly there exists a $\vec{y} = \langle y_1, \ldots, y_N \rangle \in (\{0,1\}^n)^N$ such that $\delta(y_i, C(x_i)) \le p$ and $\delta(y_i, C(z_i)) \le p$ for every $i = 1, 2, \ldots, N$. Thus, if R receives $\vec{y}$ then it has to rely on the side information S sends to determine if $\vec{x}$ or $\vec{z}$ was transmitted. Hence the side information that S sends has to be different for every pair $\vec{x}$, $\vec{z}$ which are adjacent in $G^{\wedge N}$. It follows that S has to send at least $B = \log \chi(G^{\wedge N})$ bits of side information.

It remains to show a lower bound on $\chi(G^{\wedge N})$. Unlike the inclusive graph product case where $\chi(G^{\vee N})$ was essentially just $\chi(G)^N$, the behavior of the chromatic number with respect to the AND product is more complicated (see [4] for some surprising results and some basic questions that still remain open). However, we not only know that $\chi(G)$ is large, we in fact know that $\alpha(G)$ is small; specifically $\alpha(G) < a \stackrel{\text{def}}{=} \lceil \frac{4p}{4p-1} \rceil$ for a small integer constant a. We exploit this upper bound on $\alpha(G)$ to get a lower bound on $\chi(G^{\wedge N})$.

As in the argument in the proof of the third lower bound of Theorem 3, a Ramsey-theoretic fact implies that $\omega(G) \ge 2^{k/(a-1)}$. It is easy to see that $\omega(G^{\wedge N}) \ge \omega(G)^N$ (in fact $\omega(G^{\wedge N}) = \omega(G)^N$), and therefore we get $\chi(G^{\wedge N}) \ge \omega(G^{\wedge N}) \ge 2^{kN/(a-1)}$.

This gives us a $\Omega(kN)$ lower bound on the side information needed, but we can improve the constant factor in the bound slightly using some bounds on the theta function of graphs. Alon and Kahale [3] show that if a graph G on M vertices satisfies $\alpha(G) < a$ for a positive integer a, then its theta function satisfies $\theta(G) \le AM^{1-2/a}$ for an absolute constant A. It is also well-known that $\theta(G^{\wedge N}) = \theta(G)^N$ (we point the reader to [8] for a good survey on the various properties of Theta function). Therefore $\theta(G^{\wedge N}) \le A^N M^{N(1-2/a)}$, which implies that the theta function of the complement of $G^{\wedge N}$ satisfies $\theta((G^{\wedge N})^c) \ge M^{2N/a} A^{-N}$ (since $\theta(H) \cdot \theta(H^c) \ge m$ for any m-vertex graph H). Since $\theta(H^c)$ is a lower bound on the chromatic number $\chi(H)$ of H for any graph H, we get $\chi(G^{\wedge N}) \ge 2^{2kN/a} A^{-N}$. Taking logarithms and recalling that $a = \lceil \frac{4p}{4p-1} \rceil$, we get the claimed lower bound on the number of bits of side information needed. $\square$

Though the above gives a $\Omega(kN)$ lower bound, one can hope to prove an even stronger lower bound of $(k - O(1))N$ (this would then be a strict generalization of Theorem 1 and will also mean that essentially the best possible scheme is the trivial one which sends all the N messages as side information). We can show such a result for $p \ge 1/3$, though the proof only works for binary codes (though stated for binary codes all other results have analogous statements that hold for q-ary codes for larger q as well). In this regard it is unsatisfactory, but we state it nevertheless.

Lemma 2 *When $p \ge 1/3$, for binary codes, the lower bound of Theorem 4 can be improved to $(k - \log(L+1))N$.*

Proof Sketch: The idea is to prove that $\omega(G)$ is really large, specifically $\omega(G) \ge 2^k/(L+1)$, when $p \ge 1/3$. Since $\chi(G^{\wedge N}) \ge \omega(G^{\wedge N}) \ge \omega(G)^N$, we will get the claimed lower bound. To get the lower bound on $\omega(G)$, we prove that $\Delta(G^c)$, the maximum degree of the complement of G, is at most L, and use the simple fact that $\omega(G) = \alpha(G^c) \ge \frac{2^k}{\Delta(G^c)+1}$.

Indeed, suppose that $x \in \{0,1\}^k$ is adjacent to Δ nodes $z_1, z_2, \ldots, z_\Delta$ in G^c. By the definition of G, $\delta(C(x), C(z_j)) > 2p$ for every j. Therefore, if $\overline{C(x)}$ denotes the bitwise complement of $C(x)$, then $\delta(\overline{C(x)}, C(z_j)) < 1 - 2p \le p$ for every j where the second inequality follows because $p \ge 1/3$. By the (p, L)-list decodability property of C applied to "received word" $y = \overline{C(x)}$, we conclude that $\Delta \le L$. $\square$

In Section 3.3, we consider a nastier noise model for the N-repeated communication problem under which we can prove a lower bound of $(k - O(1))N$ (this is the result of Theorem 6).

3.2 Randomized schemes with low communication per message

Theorem 5 *For every p, $1/4 < p < 1/2$, every integer $L \ge 2$, and all (p, L)-list decodable codes $C : \{0,1\}^k \to \{0,1\}^n$, the following holds. For every $\varepsilon > 0$ and all $N \ge 1$, for the N-repeated communication problem for the code C, there are probabilistic polynomial time procedures (with run time polynomial in N, n) for S and R under which S sends at most*

$$\left(\frac{N}{\log N} + 1\right) \cdot \left(\log\left(\frac{kN^{1+\log L}}{\varepsilon}\right) + O(1)\right) \qquad (2)$$

bits of side information such that with probability at least $(1 - \varepsilon)$, using this side information, R can decode the correct x_i for **every** *i.*

Proof: We use a hashing based strategy similar to the one message case. Let $x_1, x_2, \ldots, x_N$ where each $x_i \in \{0,1\}^k$ be the N messages that S wishes to communicate and let R receive $y_1, y_2, \ldots, y_N$ where $\delta(y_i, C(x_i)) \le p$ for each i. Since C is a (p, L)-list decodable code, R can compute in polynomial time a list $\{z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(L)}\}$ of all messages whose encodings are within fractional distance p of y_i (there could actually be fewer than L such messages but we assume there are exactly L for simplicity as this is the "hardest" case anyway). Clearly x_i will be included in this list for every i.

The strategy for S is the following. Break the N messages into $s = N/\log N$ blocks containing $\log N$ messages each (we assume N is a power of two for simplicity). For $1 \leq r \leq s$, denote by $X_r \in \{0,1\}^{k \log N}$ the concatenation of the x_i's in the r'th block, i.e. $X_r = x_{(r-1)\log N+1} \cdots x_{r \log N}$. Pick a function f from the hash family $\mathcal{F}(k\log N, L^{\log N}, \varepsilon \log N/N)$ given by Lemma 1 at random and send $\langle f, f(X_1), f(X_2), \ldots, f(X_s)\rangle$ to R as side information on the error-free auxiliary channel.

Let us first estimate the number of bits of side information needed by the above scheme. This equals

$$(s+1)\log\left(\frac{2k\log N L^{\log N}}{\varepsilon \log N/N}\right)$$

which matches the bound (2) claimed in the statement of the theorem.

The strategy for R is now the following. For every r, $1 \leq r \leq s$, R checks if there is a *unique* $\log N$-tuple $(j_{r,1}, j_{r,2}, \ldots, j_{r,\log N}) \in \{1,2,\ldots,L\}^{\log N}$ for which $f(Z_r) = f(X_r)$ where Z_r is the concatenation of the $\log N$ strings $z_{(r-1)\log N+l}^{(j_{r,l})}$ for $l = 1,2,\ldots,\log N$. Note that since L is a constant, $L^{\log N} = N^{\log L}$ is polynomially bounded in N and so R can do this in polynomial time. For each r for which it finds such a unique tuple, it outputs the corresponding Z_r as the messages in the r'th block X_r of messages.

By hypothesis, for every $1 \leq r \leq s$, there is a $\log N$-tuple of choices from the lists from the list decoding step such that the corresponding messages from the lists equal the messages in the block X_r. (This is because for each i, $1 \leq i \leq N$, $x_i \in \{z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(L)}\}$.) Thus if R finds a unique $\log N$-tuple for which $f(Z_r) = f(X_r)$, then it *must be* the case that $X_r = Z_r$. Therefore R never makes a mistake. For each fixed r, the probability that it fails to output the messages in the r'th block is at most the collision probability $\varepsilon \log N/N$ of the hash family. Therefore, with probability at least $(1-\varepsilon)$, R outputs all blocks of messages, and thus all messages, correctly, as desired. □

Corollary 1 *For the N-repeated communication problem using a (p,L)-list decodable code, in the limit of large N, there is a randomized scheme in which S sends an amortized $\log L + 1 + o(1)$ bits of side information per message such that given this side information, with high probability R can correctly recover all the messages.*

Proof: Follows by dividing the bound (2) by N and taking the limit as $N \to \infty$. □

Similar to the second lower bound in Theorem 3, we can show the following which proves an almost matching lower bound for the amount of amortized side information that is needed.

Lemma 3 *Let L be the smallest integer for which a code C is (p,L)-list decodable. Then in any randomized scheme for the N-repeated communication problem using C which enables R to correctly recover all the messages, in the limit of large N, the sender S must communicate an amortized* $\log L - o(1)$ *bits of side information* per message.

When $p = (1/2-\gamma)$, there are binary codes known which are efficiently $((1/2-\gamma), O(1/\gamma^2))$-list decodable, and in fact one needs a list size of $\Omega(1/\gamma^2)$, cf. [7]. Therefore, we can correct from a fraction $(1/2-\gamma)$ of errors using only $2\log(1/\gamma) + O(1)$ bits of side information per message, in an amortized sense, and this is the best possible up to additive constant terms (no matter which code is used).

3.3 Repeated Communication under a stronger noise model

We have so far considered the model of repeated communication where R is *guaranteed* to receive y_i such that $\delta(y_i, C(x_i)) \leq p$ for every message x_i, $1 \leq i \leq N$. A stronger noise model to deal with is one where there is no guarantee that $\delta(y_i, C(x_i)) \leq p$ and R has to correctly recover every x_i for which $\delta(y_i, C(x_i)) \leq p$ (and it can output failures or wrong answers for the rest of the messages). This is a much harder model to deal with, and intuitively in this case amortization of side information over several messages will not help that much, since when all encodings other than $C(x_i)$ get severely corrupted, the side information must have enough information about just x_i to permit correct recovery of x_i.

Notice that our randomized scheme from Theorem 5 does have the nice property that it can recover all blocks of $\log N$ messages that satisfy $\delta(y_i, C(x_i)) \leq p$ for every i in that block, irrespective of what happens in other blocks. (This is simply because each block of $\log N$ messages is hashed separately.) For a similar scheme to work in the stronger model, we would have to use the hash function to hash each x_i separately. But this amounts to a $N\log(kLN/\varepsilon)$ bits of side information, and the amortized per message side information is no longer a constant that is independent of N (recall that we are thinking of N as much bigger than k). Since R must succeed with probability ε in recovering *all* x_i's for which $\delta(y_i, C(x_i)) \leq p$, the error probability in recovering each individual x_i must be ε/N (for a simple union bound argument to work out) and this leads to a $\log(N/\varepsilon)$ "barrier" for the amortized side information owing to the third lower bound of Theorem 3. Note though that this is **not** a formal lower bound proof. We leave showing such a lower bound as an open problem.

We do not know how to exploit randomization and do better than deterministic schemes in this setting. Below, prove a formal lower bound for deterministic schemes which suggests that in this model, essentially the best one can do is to send all the N messages as side information. The proof is quite standard and based on graph product ideas from [5].

Theorem 6 *Let $p > 1/4$. For the N-repeated communication problem for a (p, L)-list decodable binary code $C : \{0,1\}^k \to \{0,1\}^n$ under the stronger noise model where R is no longer guaranteed to receiver y_i satisfying $\delta(y_i, C(x_i)) \le p$ for every i, the following holds. For any deterministic scheme in which R is guaranteed to recover every x_i for which $\delta(y_i, C(x_i)) \le p$, the sender S must communicate at least*

$$\left(k - \log\Big(\frac{4p}{4p-1}\Big) - \log\log\Big(\frac{4p}{4p-1}\Big) - O(1)\right)N$$

bits of side information.

Proof: The proof is similar to that of Theorem 1: we define a certain graph G_N on all possible N-tuples of messages whose chromatic number will give us the desired lower bound. Recall the graph $G = (V, E)$ from the proof of Theorem 1: the vertex set of G was $\{0,1\}^k$ and two nodes x_1, x_2 were adjacent in G whenever $\delta(C(x_1), C(x_2)) \le 2p$. We argued that when $p > 1/4$, $\alpha(G) \le 4p/(4p-1)$ and hence $\chi(G) \ge 2^k\big(1 - \frac{1}{4p}\big)$.

Let us now define the graph G_N. Let $V_N = \big(\{0,1\}^k\big)^N$ be the vertex set of G_N. We connect two N-tuples $\vec{x} = \langle x_1, \ldots, x_N\rangle$ and $\vec{z} = \langle z_1, z_2, \ldots, z_N\rangle$ in G_N if for at least one value of i, $1 \le i \le N$, $x_i \ne z_i$ and $(x_i, z_i) \in E$, i.e. $\delta(C(x_i), C(z_i)) \le 2p$. The graph thus constructed is the "inclusive graph product" of G taken with itself N times, in the terminology of [10]. Given graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the vertex set of the inclusive graph product $G_1 \times G_2$ is $V_1 \times V_2$ and its edge includes all the edges $((u_1, u_2), (v_1, v_2))$ such that $(u_1, v_1) \in E_1$ **or** $(u_2, v_2) \in E_2$. Let $G^{\vee N}$ denote the inclusive product of G taken with itself N times (i.e. define $G^{\vee 1} = G$ and $G^{\vee (i+1)} = G^{\vee i} \times G$ for $i \ge 1$). It is easy to check that the graph G_N is in fact just $G^{\vee N}$. It is known (see for example [6, 11]) that

$$\chi(G_N) = \chi(G^{\vee N}) \ge \frac{\chi(G)^N}{(1 + \ln(\alpha(G)))^N}\,. \tag{3}$$

Since $\alpha(G) \le 4p/(4p-1)$, it follows that

$$\chi(G_N) \ge 2^{kN}\Big(\frac{4p-1}{4p}\Big)^N\Big(1 + \ln\frac{4p}{4p-1}\Big)^{-N} \tag{4}$$

Denote by $f(\vec{x})$ the side information that S sends when it wishes to communicate the N-tuple $\vec{x}$. We claim if $\vec{x} = \langle x_1, \ldots, x_N\rangle$ and $\vec{z} = \langle z_1, \ldots, z_N\rangle$ are adjacent in G_N, then $f(\vec{x}) \ne f(\vec{z})$.

Let i be such that (x_i, z_i) is an edge in G. Now consider the two situations (a) S transmits $C(x_1), \ldots, C(x_N)$, and (b) S transmits $C(z_1), \ldots, C(z_N)$. Consider what happens when R receives $\vec{y}$ where $y_j = C(x_j)$ for $j \ne i$ and y_i is such that $\delta(y_i, C(x_i)) \le p$ and $\delta(y_i, C(z_i)) \le p$ (such a y_i must exist since (x_i, z_i) is an edge in G) in both situations (a) and (b). Since y_i is within distance p of both $C(x_i)$ and $C(z_i)$, R must correctly output x_i in situation (a) and output z_i in situation (b). If $f(\vec{x}) = f(\vec{z})$, then R has no way of distinguishing between situations (a) and (b), and therefore we must have $f(\vec{x}) \ne f(\vec{z})$.

We conclude that f must give a legal coloring of G_N, and using (4) we get the claimed lower bound on the amount of side information that S has to send. □

4 Concluding Remarks

We have studied one scenario where list decoding can be used to correct a fraction of errors greater than $1/4$, which is the maximum fraction of errors we can unambiguously decode (for binary codes). Specifically, we assume that the receiver gets side information on an auxiliary, error-free channel using which it can disambiguate between members of the list output by the list decoding procedure.

We considered the simplest scenario where the sender, along with the encoded message, also sends the side information. That is, we did not allow any feedback from the receiver or multiple rounds of information exchange between the receiver and the sender. Savings that can be obtained by even few rounds of back-and-forth exchange are not that attractive in our opinion since one major advantage offered by using error-correction is avoiding requests for retransmissions or other information. However, we do mention here what can be achieved with interactive back-and-forth communication of the side information necessary to pick the correct message from the list. Known results on interactive communication (cf. [13, 12]) imply the following two bounds for our problem:

1. There is a deterministic scheme with two messages of side communication (i.e. after receiving y, R sends a message to S and then receives a message from S in response) with a total of $2\log k + 2\log L + O(1)$ bits exchanged, that enables R to correctly determine the message x whose encoding $C(x)$ was transmitted by S [13, 12].[5] The idea is quite simple: one proceeds

[5]One can also achieve a communication complexity of $\log k +$

as in the randomized scheme (fixing failure probability ε to be $1/2$ or any quantity less than 1), and R computes the list $\{z^{(1)}, z^{(2)}, \ldots, z^{(L)}\}$ of candidate messages. Now instead of S sending $\langle f, f(x)\rangle$ as side information for a random f drawn from the hash family, we first have R find a hash function g for which $g(z^{(1)}), g(z^{(2)}), \ldots, g(z^{(L)})$ are all distinct and send it to S, and then S responds with the value of $g(x)$ (it is easily seen that the hash family of Lemma 1 guarantees the existence of such a function for every L-tuple of $z^{(i)}$'s). This involves two rounds of communication with total amount of information the same as the randomized scheme of Theorem 2.

2. There is a deterministic scheme with four messages of side communication with a total of $\log k + \log L + 3 \log \log L + O(1)$ bits exchanged that enables R to correctly determine the message x sent by S. This follows from Theorem 1 in [12].

Both $\log k - O(1)$ and $\log L$ are lower bounds on the amount of communication needed, regardless of the number of rounds (this follows from Lemma 3 in [13]). Therefore, the first result shows that for interactive communication two messages are almost optimal (within constant factors) and the second result implies that four messages are optimal in an amortized sense for the repeated communication setting.

We conclude with open questions concerning the repeated communication setting. Can the lower bound of Theorem 4 be improved to $(k - O(1))N$ for every error rate $p > 1/4$? Or, is it possible to have a deterministic scheme for some noise rate $p > 1/4$ for which one can communicate at most $(1 - \zeta)kN$ bits of side information for some $\zeta > 0$ using which R can correctly determine all the x_i's ? For the stronger noise model of Section 3.3 where the receiver is no longer guaranteed that at most a fraction p of errors occur in each $C(x_i)$, can one take advantage of randomness to give a scheme achieving better than $\Omega(k)$ amortized bits of side information per message in the limit of large N?

Acknowledgments

Many thanks to Oded Goldreich and the Complexity'03 Program Committee for numerous useful comments and suggestions which greatly improved the quality and presentation of the paper.

References

[1] R. Ahlswede, N. Cai and Z. Zhang. On Interactive Communication. *IEEE Transactions on Information Theory*, 43(1): 22-37, January 1997.

[2] N. Alon, J. Bruck, J. Naor, M. Naor and R. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. on Information Theory*, 38 (1992), pp. 509-516.

[3] N. Alon and N. Kahale. Approximating the independence number via the θ-function, *Math. Programming*, 80:253–264, 1998.

[4] N. Alon and A. Orlitsky. Repeated communication and Ramsey graphs. *IEEE Transactions on Information Theory*, 41(5): 1276–1289, September 1995.

[5] T. Feder, E. Kushilevitz, M. Naor and N. Nisan. Amortized communication complexity. *SIAM Journal on Computing*, 24(4):736-750, 1995.

[6] U. Feige. Randomized Graph Products, Chromatic Numbers, and the Lovasz theta Function. *Combinatorica*, 17(1):79–90, 1997.

[7] V. Guruswami, J. Håstad, M. Sudan, and D. Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5): 1021–1034, May 2002.

[8] D. E. Knuth. The Sandwich Theorem. *The Electronic Journal of Combinatorics*, 1:1–48, 1994.

[9] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[10] N. Linial and U. Vazirani. Graph products and chromatic numbers. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 124-129, 1989.

[11] L. Lovász. On the ratio of the optimal integral and fractional covers. *Discrete Mathematics*, 13: 383-390, 1975.

[12] M. Naor, A. Orlitsky and P. Shor. Three results on interactive communication. *IEEE Transactions on Information Theory*, 39: 1608-1615, 1993.

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

[14] Ba-Zhong Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory*, Vol. 39, pp. 239-242, 1993.

[15] M. Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31(1):16–27, March 2000.

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

$3 \log L + O(1)$ using a different choice of hash functions – in fact this is the bound stated in Theorem 4 of [13].