Identification via Channels and Constant-Weight Codes

Krishnan Eswaran

1 Introduction

In the standard problem of transmission, the goal is to encode a message in a way such that after it passes through a noisy channel, the message can be successfully decoded at the other end. For this case, it turns out that one can send messages that scale exponentially with the blocklength and have the error probability go to 0. For this case, error control coding provides ways of adding redundancy into messages so one can still determine the intended message.

In the problem of identification via channels, introduced by Ahlswede and Dueck [1], the receiver is only interested in testing whether a particular message was sent, but the encoder does not know which message the decoder wants. Errors are now considered in terms of false alarm and missed identification. It turns out that for this case, one can design systems such that the number of different messages one can identify grows doubly exponentially with the blocklength. The trick is that each message can map into a list of codewords and the encoder selects one randomly. As long as the fraction of the pairwise overlap of these lists is small, the error probabilities will be small.

In this report, we will survey existing approaches to solve this problem. We first state the problem of identification via channels and motivate its study. To solve the problem, we consider the design of constant-weight codes using Reed-Solomon codes, which is based on the papers of Verdú and Wei [2] as well as Kurosawa and Yoshida [3]. Following this, we study constant-weight codes for finite blocklengths and compare the performance of codes constructed in [2] with those in [3]. Then, we examine how constant-weight codes can be used for a special case of the one-way communication problem considered by Orlitksy in [4].

2 Identification Codes: Problem Statement

Before proceeding, we provide some definitions that will be useful in the problem setup. To distinguish between the definition of codes used in for error control and codes for identification, the following two definitions will be useful.

Definition 1. *A (n, k) transmission code on A^n is a subset of A^n with $|A|^k$ elements.*

Definition 2. *[1] An $(n, N, \lambda_1, \lambda_2)$ identification code (ID code) is a collection of N probability distributions Q_i on $\mathcal{X}^n$ and N decoding subsets $\mathcal{D}_i \subset \mathcal{Y}^n$ such that for a channel $p(y^n|x^n)$, $x^n \in \mathcal{X}^n, y^n \in \mathcal{Y}^n$, the following is true for all $i \neq j$:*

$$\sum_{y^n \in \mathcal{D}_i} \sum_{x^n \in \mathcal{X}^n} Q_i(x^n)p(y^n|x^n) \geq 1 - \lambda_1 \tag{1}$$

$$\sum_{y^n \in \mathcal{D}_i} \sum_{x^n \in \mathcal{X}^n} Q_j(x^n)p(y^n|x^n) \leq \lambda_2 \tag{2}$$

We illustrate the use of an ID code in the framework of our problem. Alice is interested in sending a message $a \in \{1, \ldots, N\}$ to Bob. Bob wants to know if Alice sent a particular message $b \in \{1, \ldots, N\}$, but Alice does not the value of b. There is a noisy channel between them. The difference between this and the transmission problem is summarized in Figure 1

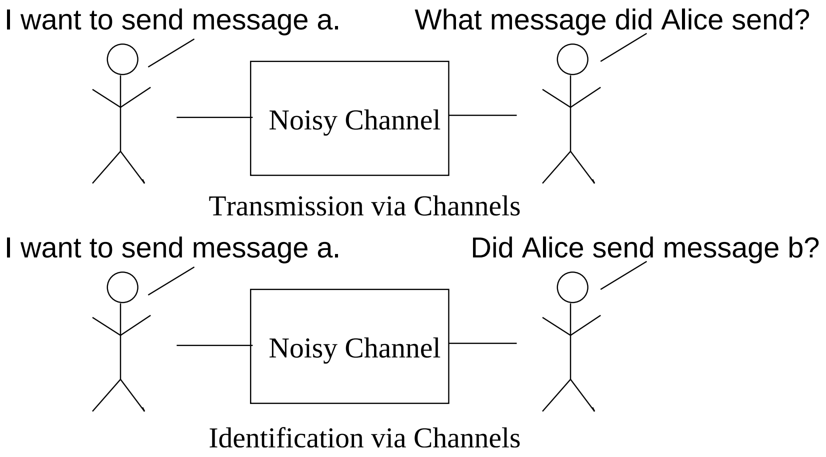
I want to send message a.
What message did Alice send?
Noisy Channel
Transmission via Channels
I want to send message a.
Did Alice send message b?
Noisy Channel
Identification via Channels

Figure 1: The difference between transmission and identification

There are two types of errors that can occur. First, Bob can conclude that his message (b) was not the one Alice sent (a) when it in fact was ($b = a$), which we call a missed identification error. Second, Bob can conclude that his message (b) was the one Alice sent (a) when in fact it was not ($b \neq a$), which we call a false alarm error. We want to have missed identification and false alarm errors less than λ_1 and λ_2, respectively. That is,

$$P[\text{Bob decides } b \text{ was not sent}|\text{Alice sends } a] \leq \lambda_1, b = a \tag{3}$$

$$P[\text{Bob decides } b \text{ was sent}|\text{Alice sends } a] \leq \lambda_2, b \neq a \tag{4}$$

Our goal is to design schemes such that these constraints are met while minimizing the number of channel uses. Alice can encode her message using an $(n, N, \lambda_1, \lambda_2)$ ID code. Suppose that each Q_i is distributed over subsets $\mathcal{C}_i$. Alice simply selects one of the strings in subset $\mathcal{C}_a$ according to Q_a and sends it across the channel to Bob. If the output Bob sees is in the set $\mathcal{D}_b$, he decides that his message b was sent. This results in missed identification and false alarm probabilities governed by equations (1) and (2), respectively.

Thus, our problem is to design an $(n, N, \lambda_1, \lambda_2)$ identification code.

3 Problem Motivation

Given the above problem statement, one might notice that starting with a (n, k) transmission code, we can construct an $(n, 2^{nR}, \lambda_1, \lambda_2)$ ID code for $R = k/n$ and some λ_1, λ_2, as follows: each Q_i corresponds to probability distribution that is 1 for the ith codeword and 0 otherwise, and λ_1 and λ_2 can be determined by setting an Neyman-Pearson decision rule (see, e.g. [5, p.22]). Thus, one might ask what benefit encoder randomization provides.

In fact, the benefit of allowing for randomization is that it allows for encoding sets to overlap (see Figure 2), which significantly increases the number of objects one can identify. As long as the pairwise overlap is small, we can drive the error probability to 0 with the blocklength while having the number of objects scale doubly exponentially. The following statements make this precise.

Definition 3. *R is an achievable ID rate if, for all $\lambda_1, \lambda_2, \epsilon > 0$ and sufficiently large n, there exists an $(n, N, \lambda_1, \lambda_2)$ ID code such that*

$$\frac{1}{n} \log \log N > R - \epsilon. \tag{5}$$

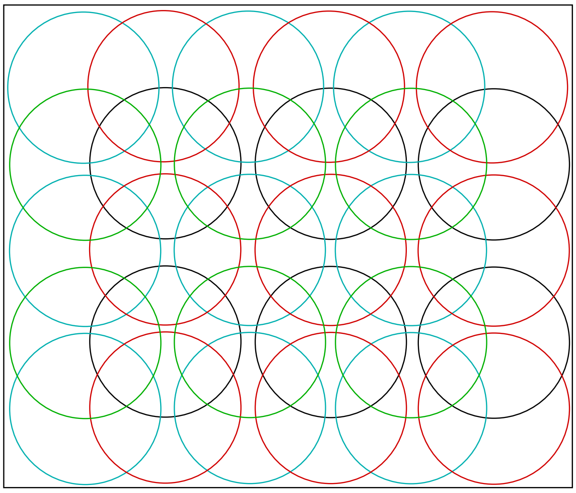

Figure 2: Overlap can increase the number of sets we can place.

Theorem 1. *[1] For a discrete memoryless channel $W(y|x)$, R is an achievable rate if and only if*

$$R \leq C, \tag{6}$$

where C is the Shannon capacity of the discrete memoryless channel. That is,

$$C = \max_{p(x)} \sum_{x,y} p(x)W(y|x) \log \frac{W(y|x)}{\sum_{\tilde{x}} p(\tilde{x})W(y|\tilde{x})}. \tag{7}$$

Note that the above result holds under less restrictive assumptions, for certain cases in which the channel is not necessarily memoryless or discrete [6], [7], as long as we replace (7) with the appropriate formula for capacity. In other cases, it can be even larger [8], [7], but Shannon capacity is always a lower bound for achievable ID rates [1].

What is more important to observe is that N can grow doubly exponentially with the blocklength when we allow for randomization. It should be clear from counting that this is not possible when randomization is not allowed since this would preclude overlaps.

In addition to this technically interesting motivation, ID codes have also been considered for application in watermarking problems [9], [10]. Thus, we return to the question of how to construct such a code.

4 Identification Codes: Design Principles

One simplification of the problem that will allow us to design codes is the following. We return to the example of Alice and Bob to describe it. Suppose the decoding sets $\mathcal{D}_i$ that Bob has at his disposal are determined by the following algorithm. After correcting for any errors that may have occurred across the channel, Bob determines whether the string he decodes to is in the set $\mathcal{C}_b$.

Under this framework, the missed identification error depends solely on whether the subset $\mathcal{C}_a$ is robust to errors. For example, if we consider channels that make at most one error, and $\mathcal{C}_a \subseteq \mathcal{L}, \forall a$, where $\mathcal{L}$ is some $(n, k, 3)$ linear code, then using nearest neighbor decoding to correct for errors, our identification code satisfies $\lambda_1 = 0$.

For the false alarm error, in addition to how robust $\mathcal{C}_a$ is to errors, we must also consider the pairwise overlap between $\mathcal{C}_a$ and $\mathcal{C}_b$. For example, if we consider channels that make at most one error, and $\mathcal{C}_i \subseteq \mathcal{L}, \forall i$, where $\mathcal{L}$ is some $(n, k, 3)$ linear code, then the identification code satisfies $\lambda_2 = \frac{\max_{i \neq j} |\mathcal{C}_i \cap \mathcal{C}_j|}{2^n}$.

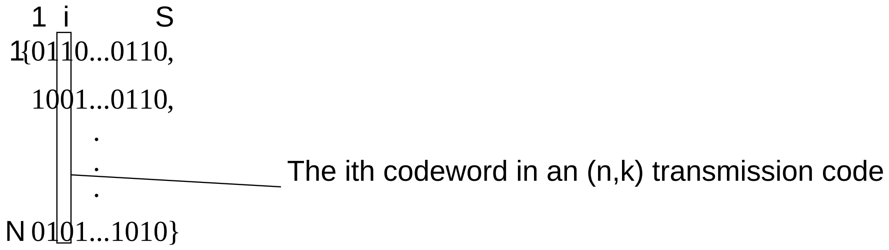

Figure 3: Each column corresponds to a codeword in some transmission code.

Since we can embed codes for a noiseless channel into error control codes, we will consider codes for the noiseless channel in what follows. We now constrain our distributions Q_i to be equiprobable over subsets $\mathcal{C}_i$ of the same size. While this may limit some flexibility, Ahlswede and Dueck [1] showed identification codes that attained the Shannon capacity could be found under these constraints. Under these restrictions, we can represent the subsets $\mathcal{C}_i$ as codewords, with value 1 if the ith codeword is in the set and 0 if it is not (see Figure 3). We call these representations constant-weight codes. The remainder of this section is based on [2], unless stated otherwise.

4.1 Constant-Weight Codes

Definition 4. *[2] An (S, N, M, K) constant-weight code is a set of N binary strings of length S and Hamming weight M such that the number of coincident 1's between any two codewords does not exceed K. $\frac{K}{M}$ is the overlap fraction.*

The location of 1's correspond to the embedded (n, k) transmission code codewords with a uniform chance of being selected for that designed object.

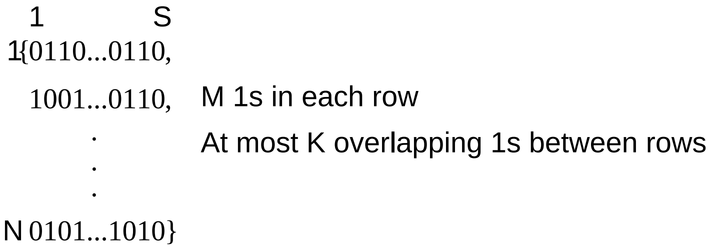

Figure 4: An (S, N, M, K) constant weight code.

In our construction, we think of the blocklength of the constant-weight code as the total number of transmission codewords that we can transmit. Thus, each position in a constant-weight codeword corresponds to a codeword in a $(n, \log S)$ transmission code, and is 1 if that transmission codeword is in this set $\mathcal{C}_i$ and 0 if it is not. With this scheme, each codeword in a constant-weight code corresponds directly to a uniform probability distribution on codewords for some $(n, N, \lambda_1, \lambda_2)$ ID code.

Assuming we have a good transmission code, we have reduced the problem to finding a good ID code for the noiseless channel. By good, we mean an code that is approximately 2^{2^n} with the blocklength with the false alarm probability going to 0. Recall that under our simplified assumptions about decoding, the missed identification error probability will always be 0 for the noiseless channel. Therefore, the corresponding constant-weight code should have the following two properties. First, the noiseless rate

$$R_{ID}^{noiseless} = \frac{\log \log N}{\log S} \approx 1. \tag{8}$$

Second, the overlap fraction (i.e. noiseless false alarm error probability), is

$$P_{FA}^{noiseless} = \frac{K}{M} \approx 0. \tag{9}$$

A third condition is included in [2] for IT codes, in which the random codeword selected conveys information about some auxiliary message. Our constructions will also apply to that case, and the interested reader is referred to [6] and [2] for more information.

Now that we have enforced some structure by introducing constant-weight codes, the next step is to understand how to design such codes. Observe that constant-weight codes are not linear codes, except in the trivial case when $M = 0$. However, we want to leverage our knowledge about error control codes to design constant-weight codes. Drawing one analogy between the two, we note that K in a (S, N, M, K) constant-weight code is loosely connected to minimum distance of a (n, k) transmission code. For instance, for a (n, k) transmission code with a binary alphabet and minimum distance d, there are at most $n - d$ coincident 1's. We will use this fact to provide an initial construction of constant-weight codes.

4.2 Recursively Generating Constant-Weight Codes

Our approach to constructing constant weight codes will be to combine an existing constant-weight code with a transmission code. To combine them, we will use concatenated codes. We introduce concatenated codes with an example. Consider a code C_i with blocklength $n_i = 5$, alphabet $A_i = \{0, 1\}$, $N_i = 4$. In particular,

$$C_i = \{01110, 00111, 11100, 10011\}.$$

Now, let C_o be a second code blocklength $n_o = 2$, $A_o = \{a, b, c, d\}$, and $N_o = 8$. Specifically,

$$C_o = \{aa, bb, cc, dd, ab, bc, bd, da\}.$$

Notice that A_o and C_i contain the same number of elements. Suppose we used the codewords in C_i as labels for our elements instead of the letters in A_o. Assigning the labels in the same order as they are listed above produces the code (spacing provided for readability)

$$\begin{aligned} C = \{&01110\ 01110,\\ &00111\ 00111,\\ &11100\ 11100,\\ &10011\ 10011,\\ &01110\ 00111,\\ &00111\ 11100,\\ &00111\ 10011,\\ &10011\ 01110\} \end{aligned}$$

of blocklength $n = 10$, size $N = 8$, and alphabet $A = \{0, 1\}$. The code C is a concatenated code with inner code C_i and outer code C_o. In general, we have the following definition.

Definition 5. *[2] Let C_i and C_o be codes with blocklengths n_i, n_o, sizes N_i, N_0, and alphabets A_i, A_0, respectively. If $|A_o| = N_i$, then the concatenated code $C = C_i \circ C_o$ with blocklength $n_i n_o$, size N_o and alphabet A_i is constructed using any one-to-one function $h : A_o \rightarrow C_i$:*

$$C = \{(h(y_1), \ldots, h(y_{n_o}) : (y_1, \ldots, y_{n_o}) \in C_o, y_k \in A_o\}.$$

Recursion Assume we already have a (S, N, M, K) constant-weight code. Then, the following lemma will allow us to construct a new one. The lemma generalizes an argument in the proof of [2, Proposition 2].

Lemma 1. *For a (S, N, M, K) constant-weight code C_i and a (n, k) transmission code C_o with alphabet size N and minimum distance d, the concatenated code $C = C_i \circ C_o$ is a $(Sn, N^k, Mn, M(n-d) + Kn)$ constant-weight code.*

Proof. From the Definition 5, we know the blocklength of the concatenated code will be Sn and its size will be N^k. Further, since each codeword in the inner code is of constant weight M, each codeword in C will have constant weight Mn. To bound the overlap in C, note that at most $n - d$ positions in the outer code will be identical, each have total overlap of M, for a subtotal of $M(n - d)$. When there is not overlap in the outer code (in at most n positions), the overlap contributed by the inner code is at most K, for a subtotal of Kn. Thus, the total overlap is at most $M(n - d) + Kn$. □

Lemma 1 has important consequences on our design considerations. For instance, notice that

$$\frac{M(n-d) + Kn}{Mn} \geq \frac{K}{M}, \tag{10}$$

which could potentially give us a worse false alarm probability with the concatenated constant-weight code over the original one, depending on how good our bound was in Lemma 1. Thus, we want the transmission code we concatenate with our initial constant-weight code to have a large minimum distance. Further, we want N^k to be exponentially larger than Sn.

This structure also allows us to implement randomized encoding recursively to prevents one from generating the entire constant-weight codeword corresponding to message we plan to send. We now outline such a procedure to randomly select a codeword. First, we generate the codeword in our outer code corresponding to our message. After getting a our uniform random number, we select the position in the outer code (each position has the same number of 1s in the inner code) where the transmission codeword will be, and find the label corresponding to that position in the code.

Initialization Now that we have established that we can recursively generate constant-weight codes by concatenation, we provide an initial constant-weight code from [2]. In fact, this constant-weight code corresponds to the degenerate case in which our ID code is just a transmission code. We observed earlier that this does not attain a doubly exponential rate. While an alternate design of initial constant-weight codes is the subject of [3], we start with this one because it is conceptually simple to understand.

Definition 6. *A (q) pulse position modulation (PPM) code C is a $(q, q, 1, 0)$ constant-weight code. That is,*

$$\begin{aligned} C_{PPM}^{q} = \{&10\cdots00, \\ &01\cdots00, \\ &\ddots \\ &00\cdots10, \\ &00\cdots01\}. \end{aligned}$$

Applying Lemma 1 to the (q) PPM code concatenated with a (n, k) transmission code with minimum distance d and alphabet size q gives a $(qn, q^k, n, n - d)$ constant-weight code.

From the observations above, we want transmission codes with large minimum distances. Therefore, Reed-Solomon codes, which are maximum distance separable, are reasonable candidates to concatenate with our PPM code. Verdú and Wei [2] considered both Reed-Solomon codes and algebraic geometry codes, but we restrict our attention to the former. Recall that a Reed-Solomon code on $GF(q)$ is an $(q-1, k)$ code with minimum distance $q - k$.[1] This gives us a $(q^2 - q, q^k, q - 1, k - 1)$ constant-weight code when concatenated with our (q) PPM code.

[1] This is based on the definition in [11]. The definition of a RS code is slightly different in [2], which results in a minor change to the blocklength.

When q is large compared to k, we have a small overlap fraction, but the number of codewords is only polynomially larger than the blocklength. This is due to the fact that to drive down the overlap, we must drive down the rate of the Reed-Solomon code. To correct this, we concatenate with another Reed-Solomon code which will have a rate that scales with the size of the field. Thus, we now consider a Reed-Solomon code on $GF(q^k)$ that is $(q^k - 1, q^t)$, with $t < k < q$, yielding a minimum distance $q^k - q^t$.

Concatenating our new constant-weight code with this Reed-Solomon code gives, by Lemma 1, a $((q^k - 1)(q-1)q, q^{kq^t}, (q^k - 1)(q-1), (q-1)(q^t - 1) + (k-1)(q^k - 1))$ constant-weight code. Now, the overlap fraction for this code is

$$P_{FA}^{noiseless} = \frac{(q-1)(q^t-1) + (k-1)(q^k-1)}{(q^k-1)(q-1)} = \frac{q^t-1}{q^k-1} + \frac{k-1}{q-1}. \tag{11}$$

Recalling our design goal in (9), we observe that as q increases, the overlap fraction goes to 0. Further, for t close to k, we also see that as q increases, the number of codewords will be exponential in the blocklength. That is, as q gets large, we can approximate

$$R_{ID}^{noiseless} = \frac{\log\log q^{kq^t}}{\log(q^k-1)(q-1)q} = \frac{t\log q + \log k + \log\log q}{\log(q^k-1) + \log(q-1) + \log q} \approx \frac{t}{k+2}. \tag{12}$$

Setting $t = k - 1$, then we can make the above ID rate (for the noiseless channel) arbitrarily close to 1 by fixing a large enough k, meeting our design goal in (8). Thus, we have reduced the problem to finding a transmission code (i.e. error correcting code) that achieves the Shannon capacity of the channel. We have already seen that certain LDPC codes get close to Shannon capacity of the binary input AWGN channel [12]. Summarizing the above result gives us a variant of [2, Proposition 2].

Theorem 2. *Concatenating a (q) PPM code with a $(q-1, k)$ Reed-Solomon code, and concatenating the result with a $(q^k - 1, q^t)$ Reed-Solomon code yields a (S, N, M, K) constant-weight code with the following parameters:*

$$S = (q^k - 1)(q-1)q, \tag{13}$$

$$N = q^{kq^t}, \tag{14}$$

$$M = (q^k - 1)(q-1), \tag{15}$$

$$K = (q-1)(q^t-1) + (k-1)(q^k-1). \tag{16}$$

So,

$$\frac{K}{M} = \frac{q^t-1}{q^k-1} + \frac{k-1}{q-1}. \tag{17}$$

4.3 Nonasymptotic Behavior of Constant-Weight Codes

We will now see how our constant-weight codes perform for relatively small block lengths. Substituting $q = 2^8$, $t = 1, k = 2$ into the construction summarized in Theorem 2 gives us that

$$S \leq 2^{32}, \tag{18}$$

$$N = 2^{4096}, \tag{19}$$

$$\frac{K}{M} \approx 0.0078 \tag{20}$$

4.4 Alternate Initialization for Constant-Weight Construction

As stated earlier, Kurosawa and Yoshida [3] provided a different initialization for constant-weight codes than the (q) PPM code. Using the theory of $\epsilon-$*almost strongly universal* hash functions and a result in [13], they show that the following construction gives a (q^2, q^{k+1}, q, k) constant-weight code.

q	$\log_2 S$	N	K/M
2^5	20	2^{320}	0.063
2^6	24	2^{768}	0.031
2^7	28	2^{1792}	0.016
2^8	32	2^{4096}	0.0078

Table 1: Constant-Weight Codes from Theorem 2 , $k = 2, t = 1$

The codewords correspond to some enumeration of all elements in $GF(q^{k+1})$. The positions correspond to some enumeration of all elements in $GF(q^2)$. We now consider c_{ij}, the jth position in the ith codeword of our code. Let $(a_0, a_1, \ldots, a_k)$ represent a $k+1$-dimensional vector representation of the ith element in our enumeration of $GF(q^{k+1})$, where each $a_k \in GF(q)$. Similarly, let (e_0, e_1) represent a 2-dimensional vector representation of the jth element in our enumeration $GF(q^2)$, with $e_0, e_1 \in GF(q)$. Then,

$$c_{ij} = \begin{cases} 1 & a_0 = e_0 + a_1 e_1 + a_2 e_1^2 + \cdots + a_k e_1^k \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Using Lemma 1, we find that the $(q^2, q^k, q, k-1)$ constant-weight code code generated by this procedure concatenated with the $(q^k - 1, q^t)$ Reed-Solomon code leads to a (S, N, M, K) constant-weight code with parameters

$$S = q^{k+2} - q^2, \tag{22}$$

$$N = q^{kq^t}, \tag{23}$$

$$M = q^{k+1} - q, \tag{24}$$

$$K = q(q^t - 1) + (k-1)(q^k - 1), \tag{25}$$

$$\frac{K}{M} = \frac{q^t - 1}{q^k - 1} + \frac{k-1}{q}. \tag{26}$$

$$\tag{27}$$

We summarize the performance for relatively small blocklengths in Table 2. The main benefit is that the false alarm probabilities decay faster.[2]

q	$\log_2 S$	N	K/M
2^5	20	2^{320}	0.031
2^6	24	2^{768}	0.015
2^7	28	2^{1792}	0.0082
2^8	32	2^{4096}	0.0039

Table 2: Constant-Weight Codes from (21) , $k = 2, t = 1$

5 Constant-Weight Codes and the League Problem

We now turn our attention to the league problem, which was introduced and considered by Orlitsky in [4]. In this problem, there are T baseball teams. Bob is listening to the radio and hears that two teams U and V

[2] The authors in [3], using the definition of Reed-Solomon codes used in [2], claimed that their initialization was better. Using our definition of Reed-Solomon codes, however, shows that the $\log_2 S$ term is in fact larger than in our original code, but both round to the same integer for the values considered in Tables 1 and 2.

played against each other. Before he can hear which team won, Alice takes the radio from him and hears the name of the winning team. Alice, unfortunately, did not hear the name of the teams playing. It turns out that if Bob is allowed to send a message to Alice (noiseless) before she communicates to him (also noiseless), the total number of bits required in the worst case for Bob to learn the winning team with zero error is $\lceil \log\log T \rceil + 1$. If one requires Alice to communicate to Bob immediately, then in the worst case, the number of bits required is $\lceil \log T \rceil$ for zero error.

However, what if Bob could tolerate some error ϵ? Then Alice could communicate directly to Bob using ID codes to obtain a scaling that is also doubly logarithmic in the number of teams. Orlitsky [4] made a similar observation and construction for a more general class of problems, and for his construction, the message uses at most $2(\log\log T + \log(1/\epsilon)) + 3$ bits in the league problem. This construction, while applying far more generally than to just the league problem, requires the ability to find prime numbers efficiently.

Using a (S, N, M, K) constant-weight code would yield a noiseless ID code with the following properties in the league problem. Bob has the choice of learning the answer to one of the following two questions without error if it is true: did team U win or did team V win (equivalently, did team V lose or did team U lose). In the event it is not true, there is a probability K/M that he accidentally decides it is true, anyway. Alice will only require $\lceil \log S \rceil$ bits to send this information to him. Thus, we just need to bound $\log S$. Recall our (S, N, M, K) constant-weight code in Theorem 2. For this code, we have the following facts.

$$\frac{K}{M} \leq \frac{1}{q^{k-t}} + k/q \leq \frac{k+1}{q}. \tag{28}$$

$$\log\log N = \log k + t\log q + \log\log q. \tag{29}$$

$$\log N = kq^t \log q \geq k\log q \geq \log q. \tag{30}$$

From (30), we note that a trivial bound on $\log S$ is $3\log\log N$. With these facts, we can now get a bound on the number of bits needed for Alice to notify Bob of which team won with error probability K/M. For $t = k - 1$,

$$\begin{aligned} \log S &\leq (k+2)\log q && (31)\\ &\leq 3\log\log N && (32) \end{aligned}$$

where (31) follows from Theorem 2 and (32) follows from (30). Note that due to our construction, N may not correspond to T exactly, and this may not achieve the desired ϵ.

We now assume that $q = 2^m$. We know that $T \leq N$, so from (29), $\log\log T \leq \log k + tm + \log m$, and reinterpreting this in terms of m, we get the bound

$$m \geq \log\log T, \tag{33}$$

which holds for all t, but is tightest for $t = 1$. Similarly, from (28), we know that $k \leq \epsilon 2^m - 1$. Choosing m and k as close to these bounds as possible, we find from (32) and (29) that for $\epsilon 2^{\lceil \log\log T \rceil} > 1$, $\log\log T > 0$, and $t = 1$,

$$\log S \leq 3(\log(\epsilon 2^{\lceil \log\log T \rceil} - 1) + \lceil \log\log T \rceil + \log\lceil \log\log T \rceil). \tag{34}$$

Note that we can loosen the restriction $\epsilon 2^{\lceil \log\log T \rceil} > 1$ by a careful choice of t. Thus, we have a bound in (34) that an ID code approach to the league problem attains a $\log\log T$ performance like Orlitsky's one-way communication construction with errors, except we do not require a prime-number algorithm.

6 Conclusions

We described the problem of identification via channels and surveyed how random encoding procedures can be constructed explicitly to allow one to identify one of doubly exponentially many objects in the blocklength. Further, we showed how these codes could be used to reduce communication in the league problem and gave a bound on the number of bits required to attain a particular error probability.

References

[1] R. Ahlswede and G. Dueck, “Identification via channels,” *IEEE Transactions on Information Theory*, vol. 35, pp. 15–29, Jan. 1989.

[2] S. Verdú and V. Wei, “Explicit construction of optimal constant-weight codes for identifications via channels,” *IEEE Transactions on Information Theory*, vol. 39, pp. 30–36, 1993.

[3] K. Kurosawa and T. Yoshida, “Strongly universal hashing and identification codes via channels,” *IEEE Transactions on Information Theory*, vol. 45, pp. 2091–2095, 1999.

[4] A. Orlitsky, “Worst-case interactive communication I: Two messages are almost optimal.,” *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 1111–1126, 1990.

[5] H. V. Poor, *Introduction to Signal Detection and Estimation.* Texts in Electrical Engineering, New York: Springer-Verlag, 1994.

[6] T. Han and S. Verdú, “New results in the theory of identification via channels,” *IEEE Transactions on Information Theory*, vol. 38, pp. 14 – 25, 1992.

[7] T. Han and S. Verdú, “Approximation theory of output statistics,” *IEEE Transactions on Information Theory*, vol. 39, pp. 752 – 772, May 1993.

[8] M. V. Burnashev, “On identification capacity of infinite alphabets or continuous-time channels,” *IEEE Transactions on Information Theory*, vol. 46, pp. 2407 – 2413, 2000.

[9] Y. Steinberg and N. Merhav, “Identification in the presence of side information with application to watermarking,” *IEEE Transactions on Information Theory*, vol. 47, p. 1410, 2001.

[10] Y. Jiang and F.-W. Sun, “‘Watermarking’ for convolutionally/turbo coded systems and its applications,” in *IEEE Globecom*, 2001.

[11] S. Lin and D. J. Costello, *Error Control Coding, Second Edition.* Upper Saddle River, NJ, USA: Prentice-Hall, 2004.

[12] S.-Y. Chung, J. G. David Forney, T. J. Richardson, and R. Urbanke, “On the design of Low-Density Parity-Check codes within 0.0045 dB of the Shannon limit,” *IEEE Communications Letters*, vol. 5, pp. 490–498, 2001.

[13] B. den Boer, “A simple and key-economical unconditional authentication scheme,” *Journal of Computer Security*, vol. 2, pp. 65 – 71, 1993.