Capacity Upper Bounds on Binary Deletion Channels

Ray Li

December 7, 2014

Abstract

In this paper, we discuss the problem of bounding the capacity of binary deletion channels in light of the paper, "Tight Asymptotic Bounds for the Deletion Channel with Small Deletion Probabilities" (Kalai, Mitzenmacher, Sudan, 2010), which proves an upper bound of $C \leq 1 - (1 - o(1))H(p)$ for the capacity of a binary deletion channel for p approaching 0. We present a brief history surrounding the bounds on the Binary Deletion Channels (BDC). We then explain the proof presented in the paper, highlighting its key ideas, its connections to other results for deletion channels and its limitation to small deletion probabilities.

1 Introduction

The binary symmetric channel with crossover probability p (BSC_p), where each bit is flipped with some fixed probability p, and the binary erasure channel with erasure proability p (BEC_p), where each bit is erased with some fixed probability p, are both examples of discrete memoryless channels. Fundamental properties of both were studied by Shannon in the 1940s [5], and both are very well understood.

The *binary deletion channel with deletion probability* p (BDC_p), however, is much less understood. In the binary deletion channel, each bit is *deleted* with some fixed probability p. In contrast to the erasure channel, we do not know where the deleted elements are in the deletion channel. For example, in the binary erasure channel, if we transmitted the string 00 and one bit was deleted, we would receive either 0? or ?0 depending on which bit was deleted, but in the binary deletion channel we would only receive the single bit 0.

The big questions in discussing noisy channels such as the BSC, BEC, and BDC are a) determining the capactiy of the channel (i.e. theoretical limits) b) constructing good codes c) finding efficient encoding and decoding algorithms. For symmetric channels and erasure channels, these questions are fairly well understood, but for deletion channels we know comparatively little. Constructions and efficient algorithms for deletion codes are heavily discussed in [3], and we will briefly touch on some of these results. The capacity of deletion channels is largely unknown, and this problem will be the focus of the paper.

We will begin by examining existing bounds of BDC_p beginning with the most basic ones, and use that as a basis for discussing intuition for bounds on the BDC_p. We conclude with a summary of the proof and ideas in a tight upper bound for BDC_p of $1 - (1 - o(1))H(p)$ when p is small [1].

1.1 Definitions and Notation

Definition 1.1. For transmission of a string $X \in \{0,1\}^n$ according to a binary deletion channel, the *deletion pattern* A is an increasing subsequence of $[n] = \{1, \ldots, n\}$ representing the bits that are *not* deleted.

Notation 1.2. For string $X \in \{0,1\}^n$, X_A represents the transmission of X through a deletion channel with deletion pattern $A = (a_1, \ldots, a_n)$. The ith bit of transmission is X_{a_i}. This may also be thought of as X restricted to the index set A.

Notation 1.3. We denote $D_{q,n}$ as the set of deletion pattens for $[n]$ with q deletions.

The following definition formally expresses the idea that deletion channels only encode the successfully transmitted bits.

Definition 1.4. Two transmissions X_A, Y_B accross a binary deletion are identical if and only if $\text{len}\,(A) = \text{len}\,(B)$ and $X_{a_i} = Y_{b_i}$ for $i = 1, \ldots, \text{len}\,(A)$.

Example 1.5. Suppose we send $X = 101010$ across a deletion channel and position 3,4,5 are deleted. Then,

$$\begin{aligned} X &= 101010 \\ A &= [1,2,6] \\ X_A &= 100 \end{aligned}$$

We will also define a similar channel to the BDC_p, the (q,n) deletion channel. This channel will be mentioned throughout the paper and will appear in our discussion of the proof for [1]. As we will see, it is presented as a key step in the proof to link our combinatorial intuition with the actual behavior of BDC_p

Definition 1.6. A (q,n) *deletion channel* is a channel that deletes *exactly* q bits of an n bit message, with the set of deleted bits chosen uniformly at random over $D_{q,n}$.

Example 1.7. In a $(1,4)$ deletion channel, when the word 1000 is sent across, the word 100 will be received with probability $3/4$.

2 Existing Bounds

We begin our overview of existing capacity bounds with the simplest upper bound for BDC_p.

2.1 Easy Upper Bound: $1-p$

Proposition 2.1. *The capacity for BDC_p is at most $1-p$ for all p.*

Intuitively this is clearly true because $1-p$ is capacity of BEC_p, and as we've seen, BDC_p carries less information that BEC_p. Formally, suppose, we had a code rate greater than $1-p$

which could be recovered under transmission through BDC_p. Then, if we sent such a code through a BEC_p and then deleted all the erased bits, we would obtain the same distribution of words as if we sent the bit through BDC_p, so we could use the same decoding algorithm to recover the codeword. This would give us a code of rate greater than $1-p$ that can decode with high probability under transmission through BEC_p, contradicting the capacity of BEC_p. Thus the capacity of BDC_p cannot be more than that of BEC_p.

We can also argue this proposition as follows. Suppose the capacity were greater than $1-p$ and thus there were some $\epsilon > 0$ such that for all n, we could encode an $(1-p)n$ bit message encoded in a codeword of length $(1-\epsilon)n$. Then the expected length of the received word would be $(1-p)(1-\epsilon)n$. Thus, choosing $n > O(1/\epsilon^2)$ will give that with high probability, The length of received word will be $\leq (1-p)(1-\epsilon)n + O(\epsilon n) < n$, in which case recovering the codeword happens with low probability. This means the probability of recoving the codeword goes to 0 as $n \to \infty$.

This proposition illustrates several points about the binary deletion channel. The first is that we explicitly see how BDC_p is harder to decode than BEC_p. The second is the observation is the simple probabilistic fact that the lengths of the code will be concentrated around the expected value, pn, and thus it often suffices to restrict our analysis to when the number of deletions is fixed, either to a single number of deletions around pn, or to a range about pn. This idea will reappear in our proof of the theorem, where we reduce Theorem 4.1 about binary deletion codes to Theorem 4.5 about (q, n) deletion codes using this type of argument.

2.2 Lower Bound: $.1185(1-p)$

In 2008, Mitzenmacher constructed a code of rate $.1185(1-p)$ (approximately $(1-p)/9$) for the BDC_p. The idea was to use a Poisson repeat channel, and this is the best known constructable lower bound for large p. While this result is far from tight, it is remarkable because it gives an explicity construction for a code that is within a relatively small constant factor of optimal.

2.3 Lower Bound: $1 - H(p)$

There are several long-known works which prove an implicit lower bound of $1 - H(p)$ for deletion channels. In the 1960s, Gallager [2] analyzed codes over insertion/deletion/substitution channels, and Zigangirov [7] published a paper on insertion/deletion channels, both being more general version of deletion channels. They both constructed codes which yielded a lower capactiy bound of $1 - H(p)$ in the case of i.i.d. deletions.

2.4 Upper Bounds beating $1 - p$

In 2007, Diggavi, Mitzenmacher, and Pfister [4] proved several upper bounds which beat the $1-p$ bounds. Most notably, they improved the $1-p$ bound to $.7918(1-p)$ in the limit as $p \to 1$. In the same paper, they obtained a computer generated bound beating $1-p$ for $p \leq .9$, shown in the table below (taken from the paper).

d	LB	UB
0.05	0.7283	0.816
0.10	0.5620	0.704
0.15	0.4392	0.6188
0.20	0.3467	0.5507
0.25	0.2759	0.4943
0.30	0.2224	0.4466
0.35	0.1810	0.4063
0.40	0.1484	0.3711
0.45	0.1229	0.33987
0.50	0.1019	0.31082
0.55	0.08432	0.28382
0.60	0.06956	0.25815
0.65	0.05686	0.2331
0.70	0.04532	0.2083
0.75	0.03598	0.183
0.80	0.02727	0.157
0.85	0.01938	0.1298
0.90	0.01238	0.0999*
0.95	0.00574	0.064*

TABLE I

THE LOWER BOUND FROM [6] AND THE UPPER BOUND DERIVED FROM THEOREM 2. ENTRIES DENOTED * ARE WORSE THAN THE $1 - d$ BOUND.

3 Intuition for Upper Bounding BDCs

To establish some intuition for upper bounding deletion channels, we will first review a relevant proof of the well known $1 - H(p)$ capacity for binary symmetric channels. Several of these ideas are taken from [1].

3.1 Review: BSC Capacity $= 1 - H(p)$

Proposition 3.1. *A binary symmetric channel with crossover probability p has capacity at most $1 - H(p)$*

We've seen both proofs in class, but they included here for completeness. More detail can be found in [6]

Proof 1. Suppose we have a code of length n and size N that can decode BSC_p with high probability. This means that when every message message m can be decoded with probability at least some fixed value, say $\frac{1}{2}$.

For any γ, the Chernoff bound gives that the probability of a received word of sending m through the BSC having number of bit flips not in $[(1-\gamma)pn, (1+\gamma)pn]$ to be at most $2^{-\Omega(\gamma^2 n)}$.

Choosing γ such that this bound is less than $\frac{1}{4}$, we have by union bound that the probability that we can decode the received word *and* the number of bit flips is in $[(1-\gamma)pn, (1+\gamma)pn]$ is at least $\frac{1}{2} - 2^{-\Omega(\gamma^2 n)} \geq \frac{1}{4}$.

Since all words distance in $[(1-\gamma)pn, (1+\gamma)pn]$ from m have roughly equal probability of occuring (up to factor $(\frac{1-p}{p})^{\gamma pn}$) the above analysis tells us at least $\approx \frac{1}{4}\left(\frac{1-p}{p}\right)^{\gamma pn}$ fraction of the approximately $2^{H(p)}$ words with distance in $[(1-\gamma)pn, (1+\gamma)pn]$ from m will decode into m. Since each length n word can decode into at most one codeword, we observe

$$N\frac{1}{4}\left(\frac{1-p}{p}\right)^{\gamma pn} 2^{H(p)n} \leq 2^n \implies \frac{\log N}{n} \leq 1 - H(p) + o(1)$$

□

Proof 2. Suppose we can communicate successfully across a BSC_p. If we can successfully recover our codeword, then we will also have recovered the set of bits which were flipped. The former has $\log N$ bits of information, and the later has about $\log \binom{n}{pn} \approx h(p)n$ bits of information. Since the total number of bits which were transmitted is n, the total information we recover cannot be more than n, so it follows that $\frac{\log N}{n} \leq 1 - h(p) + o(1)$. □

In these two approaches, we see two approaches to interpretting the $1 - H(p)$ value: one see the $H(p)$ term to represent the number of words which must decode into a given codeword, and the other interprets it as the information captured in the set of flipped bits. We'll see below how the ideas help bound deletion channel capacity.

3.2 Ideas for BDC capacity when p is small

The key step to arguing simialrly to the symmetric channel bound is this: When p is small, if we have a code that can effectively correct deletions in a binary deletion channel, it's possible to recover the deletion pattern with nontrivial probability for "most" codewords. While in symmetric channels we can recover the flip pattern with probability 1, it is still possible to acheive comparable bounds in the deletion channels.

Using the same type of argument, each of the $2^{n(1-p)}$ recieved words should map to one of approximately $N2^{H(p)n}$ codeword-deletion pattern pairs. Then we also get $\frac{\log N}{n}$ is roughly going to be $\leq 1 - H(p)$. (Of course, we will work out the actual γs and δs and Chernoff bounds in the actual proof).

We can also use the see this in terms of information, as our $n(1-p)$ received bits can contain at most the information encoding the deletion pattern and the codeword, which will be about $H(p)n + \log N$ bits, giving the same $1 - (1 - o(1))H(p)$ bound.

We make two notes on the limitations of this argument: At the beginning of this subsection, we emphasized that p needed to be small for this type of argument to work. The following crude example illustrates this necessity.

Example 3.2. Suppose we have a deletion channel which deletes half of the transmitted bits. If we send across $1010\ldots10$ with n bits, and receive $1\ldots10\ldots0$ with $n/4$ 1s and $n/4$ 0s, there are approximately $\binom{n/2}{n/4} = 2^{\theta(n/2-\log(n))} \approx 2^{H(1/2)n/2}$ possible deletion patterns that can result. In this case, the log of number of possible patterns is within a constant

factor of the amount of information gained in the recovery of the deletion pattern. The low probability of deletion pattern recovery would offset any benefit in a bound we would get from information gain in deletion pattern recovery for obtaining a tight bound.

The second remark relates to the ability of recovering deletion patterns for "most" codewords. Deletion codes exhibit an asymmetry not present in the symmetric channel or the erasure channel, so it is not always possible to recover the deletion pattern with nontirival probability.

Example 3.3. Consider a $(1, n)$ deletion channel. If we send the codeword $1010\ldots10$ across and can successfully recover the codeword, we know exactly where the deleted bit is. On the other hand, if we send $00\ldots0$ across, we have absolutely no information about where the deleted bit is.

We observe that for purposes of our argument, codewords like "$1010\ldots10$" in which it is easy to recover the deletion pattern are "good" in some sense, and codewords like "$000\ldots0$" are "bad".

In the proof of the main theorem, we explicitly define notions of "good" and "bad" codewords. In order to prime the argument, a simpler example of working around this asymmetry is presented below to close off this section.

3.3 A simple result on deletion channels

To set up for the proof of the $1-(1-o(1))H(p)$ bound, we examine a simple (and tight) upper bound for the $(1, n)$ deletion channel, given in our homework.

Proposition 3.4. *A length n code that can be recovered in a $(1, n)$ deletion channel has at most $O(2^n/n)$ codewords.*

Proof. Define a length n word to be *bad* if can be deleted into at most $n/2$ different words when passed through the $(1.n)$ deletion channel, and *good* otherwise.

It suffices to prove that a) we can have at most $O(2^n/n)$ good words in our code, and b) the number of bad codes is $O(2^n/n)$.

Indeed, one can show that the number of bad words is $\approx 2^{H(1/4)n} = o(2^n/n)$. Furthermore, among the length 2^{n-1} words, each is correctly decoded into at most one good word, and since each good word must be correctly decoded by $n/2$ length-$n-1$ words, the total number of good words must be at most $2^{n-1}/(n/2) = O(2^n/n)$. □

4 Proof of $1-(1-o(1))H(p)$ Bound

At a high level, this theorem is fixes a small p and considers the dimension of the code as n goes to infinity. The claim is that when p is small, we can choose n sufficiently large so that the bound on the rate becomes close to $1-(1-o(1))H(p)$, and then we may make the $o(1)$ term vanish by taking $p \to 0$.

Theorem 4.1 (KMS2010, Main Theorem)**.** *Suppose there is a code C and a decoder which can successfully decode for BDC_p with probability at least δ, and suppose $n \geq 12\log(4/\delta)/p$. Let $\gamma = 3\log(4/\delta)/np$ and $q' = (1+\gamma)np$. Then the dimension of the code $\log|C|$ satisfies,*

$$\log|C| \leq n - np(1-\gamma) - \log\binom{n}{np(1-\gamma)} + \log\frac{4}{\delta} + \log\beta$$

where β is given by $\beta = t'(6t'/q')^{3q'+1}$ for $t' = \lceil 3q'\log\frac{ne}{q'} + \log 4\delta\rceil$.

In particular, the rate of the code satisfies

$$\frac{\log|C|}{n} \leq 1 - (1-o(1))H(p)$$

where the $o(1)$ term vanishes as $p \to 0$.

We can get a sense of how large the various terms are by verifying the last statement of 4.1. In particular, we may note that the $o(1)$ term varies as $p\log\log(1/p)$, in contrast to $H(p)$ which is about $p\log(1/p)$ as $p \to 0$:

Remark 4.2. If the main bound of 4.1 is true, then

$$\begin{aligned}\frac{\log|C|}{n} &\leq 1 - \frac{1}{n}\log\binom{n}{np(1-\gamma)} - p(1-\gamma) + \frac{1}{n}\log\frac{4}{\gamma} + \frac{1}{n}\log\beta \\ &\leq 1 - H(p(1-\gamma)) - 0 + o(1) + \frac{1}{n}\log\beta\end{aligned}$$

and,

$$\begin{aligned}\log\beta &= \log t' + (3q'+1)\log(6t'/q') \\ &\approx \left(\log pn + \log\frac{1}{p}\right) + O(pn)\log(O(\log(1/p))) \\ &= o(n) + O(np\log\log(1/p)) \\ &= O(np\log\log(1/p)) \\ &= o(nH(p))\end{aligned}$$

where the last equality comes from the fact that $p\log\log(1/p) = o(p\log(1/p)) = o(H(p))$ as $p \to 0$. Then, noting that $\gamma \to 0$ as $n \to \infty$, we have,

$$\frac{\log|C|}{n} \leq 1 - H(p(1-\gamma)) + o(H(p)) = 1 - (1-o(1))H(p)$$

Remark 4.3. When p is large, it is not true that $p\log\log(1/p) << H(p)$, as the $\log(1/p)$ term is roughly constant. Therefore, while the original bound may be true for larger p, the conclusion that the capacity is bounded by $1-(1-o(1))H(p)$ for $o(1)$ vanishing does not hold.

Remark 4.4. One may observe from the analysis in 4.3 that, when p is large β is too large to give us a tight upper bound. We will see in the proof that this lage β corresponds to the inability to sufficiently bound the number of candidate deletion patterns for a recovered codeword. Thus, when the deletion probability is high, our intuition that we should be able to recover a codeword's deletion pattern in addition to the codeword itself fails. Indeed, this matches our observation in Example 3.2.

In order to prove the theorem, we first move away from the binary deletion channel into a similarly constrained channel, the (q, n) deletion channel.

4.1 The Key Theorem

Recall definition 1.6 of the (q, n) channel. Now, instead of working with i.i.d deletions for each bit, we fix the number of deletions and can focus our analysis on the deletion patterns themselves. Note that we have already see a simple example of (q, n) deletion channels, i.e. the $(1, n)$ deletion channel in 3.4.

Theorem 4.5. *Suppose there exists a code C and a decoder Dec for C that succeeds on the (q, n) deletion channel with probability at least δ, where $n \geq 12\log(2/\delta)/p$. Then the dimension of the code satisfies*

$$\log |C| \leq n - q - \log(q) + \log \frac{2}{\delta} + \log \alpha$$

where α is given by $\alpha = t(6t/q)^{3q+1}$ for $t = \lceil 3q \log \frac{ne}{q} + \log \frac{2}{q} \rceil$

As we saw in the $(1, n)$ deletion channel, we will approach the generic (q, n) deletion channel using a similar trick: Denote words like $00\ldots0$ which can be transmitted into few words after q deletions as *bad*, and denote the remainder as *good*. Then we will show,

1. The number of bad words is relatively small
2. For good words, conditioned on recovering the word itself, we can additionally recover the deletion pattern with nontrivial probability.

Remark 4.6. We have used the pharses "transmit into many different words after q deletions" and "able to recover the deletion pattern conditioned on original word recovery" interchangably to describe codewords. Though these notions are not completely equivalent, they intuitively are similar: $00\ldots0$ transmits into only one possible length-$n-q$ word after q deletions and one gains no information about the deletion pattern from the received word. On the other hand $1010\ldots10$ transmits into many possible length-$n-q$ words after q deletions, and we are able to recover a large fraction of the deletion pattern just by observing the received sequence (Any 00 substring signals that a 1 was deleted in between, and similarly for 11) Furthermore, as we saw, for $q = 1$ the recovery is perfect.

We can also formally argue the relationship:

If one is able to recover the deletion pattern of a length n codeword with nontrivial probability δ, then because the deletion patterns are uniformly distributed, each $n-q$ word can be obtained from the original codeword in at most $1/\delta$ deletion patterns, so there must be at least $\frac{1}{\delta}\binom{n}{q}$ words that can result from n after q deletions.

Conversely, if a codeword is able to be mapped to many different words of length $n-q$, then on average each deleted word can not have many deletion patterns corresponding to it, so the probability of recovering the deletion pattern will be nontrivial in expectation.

For purposes of this proof, it will be easier to work with the later notion of deletion pattern recovery because of its strength.

Definition 4.7. The distance between two deletion patterns of equal length A, B is

$$\Delta(A, B) = |\{i | a_i \neq b_i\}|$$

Definition 4.8. A word $X \in \{0,1\}^n$ is called t-bad if there exist distinct deletion patterns A, B with q deletions each such that $\Delta(A, B) \geq t$ and $X_A = X_B$.

Examples 4.9. If $A = [1, 3, 4, 5], B = [1, 4, 5, 6]$, are deletion patterns for $n = 6$, then,

- $\Delta(A, B) = 3$.
- 11110000 is 6-bad but not 7-bad.
- 10101010 is not 1-bad.

Just as we did in the $(1, n)$ channel, we'll bound above the number of t-bad strings.

Lemma 4.10. *For any $t \geq 1$, there are at most $\binom{n}{q}^2 2^{n-t}$ different t-bad strings $X \in \{0,1\}^n$.*

Proof Sketch. For any pair of deletion patterns A, B with q deletions each and $\Delta(A, B) \geq t$, the probability that a random string $X \in \{0,1\}^n$ matches on A and B is at most $2^{-\Delta(A,B)} \leq 2^{-t}$. Doing a union bound over all pairs A, B gives the desired result. □

Using Lemma 4.10, we can choose

$$t = 3q \log \frac{ne}{q} + \log \frac{2}{\delta} \tag{1}$$

so that

$$\Pr_{Z \in C}[\text{Dec}(Z_A) = Z \wedge Z \text{ is not } t\text{-bad}] \geq \delta - \binom{n}{q}^2 \frac{2^{n-t}}{N} \geq \delta/2$$

We have chosen a sufficiently large t such that the number of t bad strings is small. Following our roadmap described earlier, we now would like show that the probability of recovering the deletion pattern is nontrivial. We'll begin with a lemma.

Lemma 4.11. *For any deletion pattern A with q deletions, the number of deletion patterns B such that $\Delta(A, B) \leq t - 1$ is at most $(t-1)\binom{2q+t}{2q+1}\binom{q+t-1}{q} < \alpha$.*

This result is entirely combinatorial in nature, and the full proof is descibred in [1]. The details do not concern us much, as we are primarily concerned here with the idea behind the result and it's use in helping the bound the probability of recovering the deletion pattern, but a proof sketch is provided below.

Proof Sketch. Call a bit $i \in [n]$ *clean* with respect to A and B if there is some j such that $a_j = b_j = i$, and call a bit dirty otherwise. Let $D(A, B)$ denote the set of dirty bits. Then, because $\Delta(A, B) \leq t$, one can show that $q \leq |D(A, B)| \leq q + t$. This gives us $(t + 1)$ choices for the number of dirty bits. Furthermore, one can show using standard techniques that for a fixed t, the number of possible distinct sets $D(A, B)$ over all choices of B is at most $\binom{2q+t+1}{2q+1}$. The q deletions of B must occur in $D(A, B)$, so the number of ways to choose B given $D(A, B)$ is at most $\binom{q+t}{q}$. Multiplying these bounds together gives a total of up to $(t + 1)\binom{2q+t+1}{2q+1} \cdot \binom{q+t}{q}$ choices for B given A, as desired. □

Take $\alpha = t(6t/q)^{3q+1}$, where t is chosen in equation 1. For any A, α is an upper bound on the number of B such that $\Delta(A, B) \leq t - 1$, as Lemma 4.11 gives

$$\begin{aligned}(t-1)\binom{2q+t}{2q+1}\binom{q+t-1}{q} &\leq t\left(e\frac{2q+t}{2q+1}\right)^{2q+1}\left(e\frac{q+t-1}{q}\right)^{q} \\ &\leq t\left(\frac{6t}{q}\right)^{3q+1} = \alpha\end{aligned}$$

Conditioned on decoding suceeding and codeword not being t-bad, each deletion pattern is equally likely, so we can recover the deletion pattern with probability at least α^{-1}. Formally, we can define a super-decoder $g : \{0,1\}^{n-q} \to C \times D_{q,n}$, such that for $Y \in \{0,1\}^{n-q}$, we have $Y \mapsto (\mathrm{Dec}\,(Y), A)$ where A is the lexiographically first deletion pattern such that $\mathrm{Dec}\,(Y)_A = Y$.

Then the probability that we can recover the codeword *and* the deletion pattern is

$$\begin{aligned}\mathop{\mathbf{Pr}}_{Z\in C, A\in D_{q,n}}[g(Z_A) = (Z, A)] &\geq \mathbf{Pr}[g(Z_A) = (Z, A) \wedge Z \text{ is good}] \\ &= \mathbf{Pr}[\mathrm{Dec}\,(Z_A) = Z \wedge Z \text{ is good}] \\ &\cdot \mathbf{Pr}[g(Z_A) = (Z, A) \mid \mathrm{Dec}\,(Z_A) = Z \wedge Z \text{ is good}] \\ &\geq \delta\alpha^{-1}/2\end{aligned}$$

But the probability of recovering deletion is at most $\frac{2^{n-q}}{N\binom{n}{q}}$ simply because g is mapping from a 2^{n-q} element set to an $N\binom{n}{q}$ element set, so the image has size at most $\frac{2^{n-q}}{N\binom{n}{q}}$ fraction of the entire codomain, and $g(Z_A)$ can equal (Z, A) only if (Z, A) is in the range.

Thus, it follows that $\frac{2^{n-q}}{N\binom{n}{q}} \geq \delta\alpha^{-1}/2$ and

$$\log N \leq n - q - \log(q) + \log\frac{2}{\delta} + \log\alpha$$

4.2 Proof of the Main Theorem

To finish, we will find a q^* near pn such that our decoder succeeds on the (q^*, n) deletion channel with nontrivial probability.

Choosing $\gamma = \sqrt{3\log(4/\gamma)/np}$ and $n \geq 12\log(4/\delta)/p$ gives $\gamma \leq \frac{1}{2}$. Then, from the Chernoff bound, it follows that there must be $q^* \in [(1-\gamma)pn, (1+\gamma)pn]$ such that the success probability of the (q^*, n) deletion channel is at least $\delta/2$.

Theorem 4.5 gives

$$\log N \leq n - q^* - \log\binom{n}{q^*} + \log\frac{4}{\delta} + \log\alpha^*$$

and using $(1-\gamma)pn \leq q^* \leq (1+\gamma)pn$ we can finish,

$$\log|C| \leq n - np(1-\gamma) - \log\binom{n}{np(1-\gamma)} + \log\frac{4}{\delta} + \log\beta.$$

5 Conclusion

In this paper, we explored the context and history for binary deletion channels and discussed a proof of a recent tight upper bound for small deletion probabilities p. We extracted key ideas from the argument which allowed us to understand the unique difficulties in working with binary deletion channels. While we do not present any new results in this paper, and while we were unable to extend the results of [1] to general probabilities p, we were able to bring together ideas from various ideas and problems related to noisy channels into a cohesive discussion.

References

[1] M. Mitzenmacher A. Kalai and M. Sudan. Tight asymptotic bounds for the deletion channel with small deletion probabilities. *Proceedings of International Symposium of Information Theory*, 2010.

[2] R. G. Gallager. Sequential decoding for binary channels with noise and synchronization errors. October 1961.

[3] M. Mitzenmacher. A survey of results for deletion channels and related synchronization channels. *Probability Surveys*, 6:1–33, 2009.

[4] M. Mitzenmacher S. Diggavi and H. D. Pfister. Capacity upper bounds for the deletion channel. *Proceedings of the International Symposium on Information Theory*, pages 1716–1720, June 2007.

[5] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27(3):379423, 1948.

[6] A. Rudra V. Guruswami and M. Sudan. *Essential Coding Theory.*

[7] K. S. Zigangirov. Sequential decoding for a binary channel with dropouts and insertions. *Problems of Information Transmission*, 5(2):17–22, 1969.