

Lecture 30: Achieving the BSC capacity (II)

Tuesday, November 6, 2007

Lecturer: Atri Rudra

Scribe: Nathan Russell

1 Error probabilities and time complexity

In the last lecture, we started with the description of our BSC_p capacity achieving code C^* , which is a concatenated code $C_{\text{out}} \cdot C_{\text{in}}$, where C_{out} and C_{in} satisfy certain properties. In today's lecture, we will analyze the properties of C^* and also see how to get our bounds on C_{out} and C_{in} with the desired properties.

For notational convenience, we define $b' = \frac{b}{1-H(P)-\frac{\epsilon}{2}}$. Note that b' is the block length of C_{in} .

By the properties of D_{in} , for any fixed i , there is an error at g'_i with probability $\leq \frac{\gamma}{2}$. Each such error is independent, since errors in BSC_p itself are independent by definition. Because of this, and by linearity of expectation, the expected number of errors in y' is $\leq \frac{\gamma n}{2}$.

Taken together, those two facts allow us to conclude that, by the Chernoff bound, the probability that the total number of errors will be more than γn is at most $e^{-\frac{\gamma n}{3 \cdot 2}}$. Since the decoder D_{out} fails only when there are more than γn errors, this is also the decoding error probability. Expressed in asymptotic terms, the error probability is $2^{-\Omega(\frac{\gamma N}{b})}$, where N is the final block length of C^* .

We find C_{in} with the required properties by an exhaustive search among linear codes of dimension b with block length b' that achieve the BSC_p capacity by Shannon's theorem.

With those parameters, if b is $\Omega\left(\frac{\log(\frac{1}{\gamma})}{\epsilon}\right)$, Shannon's theorem implies the existence of a linear code with decoding error probability at most $\frac{\gamma}{2}$ (which is what we need).

Note, however, that since Shannon's proof uses maximum likelihood decoding on the inner code, the decoding time for this code with dimension b is $2^{O(b)}$. The construction time is even worse. There are $2^{O(b^2)}$ generator matrices; for each of these, we must check the error rate for each of 2^b possible transmitted codewords, and for each codeword computing the decoding error probability requires time $2^{O(b)}$. The overall construction time is the product of these terms, which is $2^{O(b^2)}$.

2 Finding Outer Codes

We need an outer code with the required properties. There are several ways to do this.

One option is to set C_{out} to be a Reed-Solomon code, and set $b = \Theta(\log n)$. Then the decoding algorithm for C_{out} , which we can call D_{out} , could be the BW algorithm, setting $\gamma = \frac{\epsilon}{2}$ and the decoding time is $T_{\text{out}}(n) = n^3$. The problem here is that the construction time is $N^{O(\log N)}$.

2.1 Using a binary code as the outer code

We could also use an outer code which is some explicit code which lies on the Zyablov bound and can be corrected from errors up to half its design distance. We have seen that such a code can be constructed in polynomial time.

Note that even though C_{out} is a binary code, we can think of C_{out} as a code over \mathbb{F}_{2^b} in the obvious way. If C_{out} can correct β fraction of bit errors, it can correct that same fraction of errors when the domain is considered to be \mathbb{F}_{2^b} . From this line of reasoning, intuitively, D_{out} appears to be on the “right track” with this option. We will choose b to be $\Theta\left(\frac{\log(\frac{1}{\gamma})}{\varepsilon}\right)$.

The Zyablov Bound gives $\delta_{\text{out}}(1 - R)H^{-1}(1 - r)$. We note that C_{out} itself is an RS code concatenated with an inner code that achieves the GV bound. Now we can set $1 - R = 2\sqrt{\gamma}$, $H^{-1}(1 - r) = \sqrt{\gamma}$, and it can be shown that r is $O(\sqrt{\gamma} \log \frac{1}{\gamma})$ (the proof is left as an exercise). Now if we pick D_{out} to be the GMD algorithm, then it can correct $\frac{\delta_{\text{out}}}{2} = \gamma$ fraction of errors in polynomial time.

The overall rate of C_{out} is simply $R \cdot r = (1 - 2\sqrt{\gamma})(1 - O(\sqrt{\gamma} \log \frac{1}{\gamma}))$. This simplifies to $1 - O(\sqrt{\gamma} \log(\frac{1}{\gamma}))$.

We would be done here if we could show that ε is $O(\sqrt{\gamma} \log \frac{1}{\gamma})$, which would follow by setting $\gamma = \varepsilon^3$. This implies that the rate of C_{out} is at least $1 - \frac{\varepsilon}{2}$, as desired.

The construction time, meanwhile, is $2^{O(b^2)}$, which substituting for b , is $2^{O((\frac{1}{\varepsilon} \log(\frac{1}{\varepsilon}))^2)}$. The construction time for C_{out} , meanwhile, is only $n^{O(1)}$.

As we have seen in the last few lectures, the encoding time for this code is $O(n^2)$, and the decoding time is $n^{O(1)} + n \cdot 2^{O(b)} = n^{O(1)} \cdot 2^{O(\frac{1}{\varepsilon} \log(\frac{1}{\varepsilon}))}$.

We also have that the decoding error probability is exponentially small: $2^{-\Omega(\frac{\gamma N}{b})} = 2^{-\Omega(\varepsilon^\gamma N)}$.

Thus, we now have an explicit, polynomial time construction for codes which achieve reliable communication at Shannon capacity over BSC_p with polynomial time encoding and decoding algorithms. This answers in the affirmative the central open question from Shannon’s work.

2.2 Some Remarks

To review, the decoding times are around $\text{poly}(n) \cdot 2^{\text{poly}(\frac{1}{\varepsilon})}$. This suggests an open question:

Question: Can we bring the high dependence on ε down to $\text{poly}(\frac{1}{\varepsilon})$?

For the binary erasure channel, it can be brought down to $n \text{poly}(\frac{1}{\varepsilon})$ using LDPC codes, specifically a class known as Tornado codes developed by Luby et.al. [1]. The question for binary symmetric channels, however, is still open and we are left with a high dependence on ε .

2.3 Using Expander Codes

We have a theorem due to Spielman’s 1991 work:

Theorem 2.1. *For every small enough $\beta > 0$, there exists an explicit C_{out} of rate $\frac{1}{1+p}$ which can correct $\Omega\left(\frac{\beta^2}{(\log \frac{1}{\beta})^2}\right)$ errors, and has linear time encoding and decoding in n .*

Clearly, in terms of time complexity, this is superior to the previous option. Such codes are termed “Expander codes”. The rest is the same as in subsection 2.1.

The decoding time for these codes is still $n \cdot 2^{\text{poly}(\frac{1}{\epsilon})}$, and so the question above is still open.

References

- [1] M. Luby, M. Mitzenmacher, M. A. Shekrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, 2001.