**Error Correcting Codes: Combinatorics, Algorithms and Applications** **(Fall 2007)**

# Lecture 29: Achieving the $BSC_p$ capacity

November 6, 2007

*Lecturer: Atri Rudra* *Scribe: Yang Wang & Atri Rudra*

## 1 Derandomized GMD algorithm

We introduced the GMD algorithm in the last lecture. Recall that we presented two randomized versions of the algorithm last time. Today we will present the derandomized version. Note that last time we proved that there exists a value $\theta \in [0, 1]$ such that the decoding algorithm works correctly. Obviously we can obtain such a $\theta$ by doing an exhaustive search for $\theta$. Unfortunately, there are uncountable choices of $\theta$ because $\theta \in [0, 1]$. However, this problem can be taken care of by the standard discretization trick.

Define $Q = \{0, 1\} \cup \{\frac{2w_1}{d}, \cdots, \frac{2w_N}{d}\}$. Then because for each $i$, $w_i = \min(\Delta(\mathbf{y'_i}, \mathbf{y_i}), d/2)$, we have

$$Q = \{0, 1\} \cup \{q_1, \cdots, q_m\}$$

where $q_1 < q_2 < \cdots < q_m$ for some $m \le \lfloor \frac{d}{2} \rfloor$. Notice that for every $\theta \in [q_i, q_{i+1})$, the **Step 1** of the second version of GMD algorithm outputs the same $\mathbf{y}''$. Thus, we need to cycle through all possible value of $\theta \in Q$, leading to the following algorithm:

**Input**: $\mathbf{y} = (y_1, \ldots, y_N) \in [q^n]^N$.

**Step 1**: For every $\theta \in Q$ do the following.

(a) Compute $y'_i = MLD_{c_{in}}(y_i)$ for $1 \le i \le N$.

(b) Compute $w_i = \min\left(\Delta(C_{in}(y'_i), y_i), \frac{d}{2}\right)$, for every $1 \le i \le N$.

(c) If $\theta < \frac{2w_i}{d}$, set $y''_i \leftarrow ?$, otherwise set $y''_i = y'_i$.

(d) Run errors and erasure algorithm for $C_{out}$ on $\mathbf{y}'' = (y''_1, \ldots, y''_N)$. Let $c_\theta$ be the codeword in $C_{out} \circ C_{in}$ corresponding to the output of the algorithm, if any.

**Step 2**: Among all the $c_\theta$ output in **Step 1**(d), output the one closest to $\mathbf{y}$.

Note that as $|Q| \le O(N)$ and each run of the algorithm in **Step 1** can be computed in polynomial time, the algorithm above can also be implemented in polynomial time. Thus we have shown the following:

**Theorem 1.1.** *For every constant rate, there exists an explicit linear binary code on the Zyablov bound. Further, the code can be decoded up to half the Zyablov bound in polynomial time.*

The following table summarizes the main results we have seen so far for binary codes:

| | Shannon | Hamming (Unique Decoding) | Hamming (List Decoding) |
|---|---|---|---|
| Existence | Capacity= $1 - H(p)$ | GV $\leq$ capacity $\leq$ MRRW | Capacity= $1 - H(p)$ |
| Explicit Codes | ? | Zyablov bound | ? |
| Efficient Algorithms | ? | half Zyablov bound | ? |

Next, we tackle the open questions in the first column of the table above.

# 2 Achieving capacity of $BSC_p$

Recall that there exist linear codes of rate $1 - H(p) - \varepsilon$ such that decoding error probability is not more than $2^{-\delta n}$, $\delta = \Theta(\varepsilon^2)$ on the $BSC_p$. (This follows from the Shannon's capacity proof for $BSC_p$ adopted to the linear code case.) This leads to the following natural question, which we had raised a few lectures back.

**Question 2.1.** *Can we achieve reliable transmission with polynomial time decoding over $BSC_p$ with explicit codes that have rate of $1 - H(p) - \varepsilon, \varepsilon > 0$?*

Forney answered the question above in the affirmative by using concatenated codes. (As was mentioned earlier, this was Forney's motivation for inventing code concatenation: the implication for the rate vs. distance question was studied by Zyablov later on.)

Next, we will present a positive answer to the question above by using a concatenated code $C_{out} \circ C_{in}$ with the following properties (where $\gamma > 0$ is a parameter that depends only on $\varepsilon$ and will be fixed later on):

(i) $C_{out}$: The outer code with block length $N$ and rate $1 - \frac{\varepsilon}{2}$ over $F_{2^k}$, with $k = O(\log N)$. Further, the outer code has a unique decoding algorithm $D_{out}$ that can correct at most $\gamma$ fraction of worst-case errors in time $T_{out}(N)$.

(ii) $C_{in}$: The inner code has dimension $k$, dimension $n$ and a rate of $1 - H(p) - \varepsilon/2$. Further, there is a decoding algorithm $D_{in}$ that runs in $T_{in}(k)$ time and has decoding error probability no more than $\frac{\gamma}{2}$ over $BSC_p$.

Suppose $C^* = C_{out} \circ C_{in}$. Then, it is easy to check that

$$R(C^*) = (1 - \frac{\varepsilon}{2})(1 - H(p) - \frac{\varepsilon}{2}) \geq 1 - H(p) - \varepsilon,$$

as desired.

The decoding algorithm for $C^*$ is the natural one. In particular, given the received word $\mathbf{y} = (y_1, \cdots, y_N) \in F_{q^n}^N$,

**Step 1**: Let $y_i' = D_{in}(y_i), 1 \leq i \leq N$.

**Step 2**: Run $D_{out}$ on $\mathbf{y}' = (y_1', \ldots, y_N')$.

Note that encoding $C^*$ takes time $O(N^2)+O(Nk^2) = O(N^2)$. Further, the decoding algorithm above takes time $NT_{in}(k) + T_{out}(N) = N^{O(1)}$ as long as $T_{out}(N) = N^{O(1)}$, and $T_{in}(k) = 2^{O(k)}$.

Next lecture, we will show that the decoding algorithm above has exponentially small decoding error probability over $BSC_p$. Further, we will use constructions that we have already seen in this course to instantiate $C_{out}$ and $C_{in}$ with the required properties.