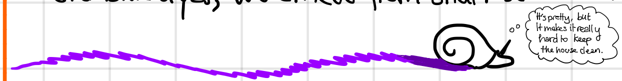
CS250/EE387 – LECTURE 6 – MAKING RS CODES BINARY

AGENDA

1. BCH Codes
2. Reed-Muller Codes [part I]
3. Concatenated Codes [part I]

GASTROPOD FACT. Some snails produce a secretion that is a color-fast natural dye. For example, "Tyrian purple" as well as other purple and blue dyes, were made from snail secretions.



The story so far is:

Reed-Solomon Codes have the optimal trade-off between rate and distance, and they have efficient decoding algorithms!

That's a pretty good story. (Maybe we should just stop here?)

HOWEVER, there are some downsides to RS codes. A big one is the alphabet size.

GOAL. Obtain EXPLICIT (aka, efficiently constructible), ASYMPTOTICALLY GOOD families of BINARY CODES, ideally with fast algorithms.

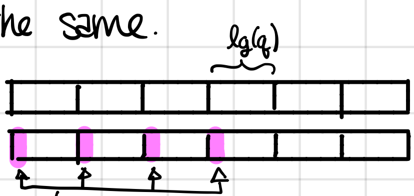
for today

Today we will see a few ways to approach this problem. [The first several won't work, but they are independently interesting and will come back later.

STRAWMAN. Replace every symbol of $\mathbb{F}_q$ with $\log_2(q)$ bits.
If we start w/ n', k' RS code:

NOTE: This is actually done in practice!

$$\text{Rate} = \frac{k' \cdot \log_2(q)}{n' \cdot \log_2(q)} = k'/n' \quad \text{stays the same.}$$



$$\text{Relative Distance} = \frac{n'-k'+1}{n' \cdot \log_2(q)} \sim \frac{1-R}{\log(n')}$$

if $n' = q-1$

corrupting d bits also can corrupt d symbols. So the distance is $d \geq n'-k'+1$

$$\sim \frac{1-R}{\log\left(\frac{n}{\lg(n)}\right)} \sim \frac{1-R}{\log(n)}$$

where $n = n' \lg(n')$ is the block length of the binary code.

So the STRAWMAN is **NOT** asymptotically good.
If R is constant, then $\delta \to 0$.

① BCH Codes.

BCH = Bose and Ray-Chaudhuri, Hocquenghem

What if we just take $RS(n,k) \cap \{0,1\}$?

DEF. Let $n = 2^m - 1$, let γ be a primitive element of $\mathbb{F}_{2^m}$.
Then for $d \le n$, define

$$BCH(n,d) = \left\{ (c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n \;\middle|\; c(\gamma^j) = 0 \quad \forall j = 1, \ldots, d-1 \right\}$$

$c(X) = \sum_{i=0}^{n-1} c_i \cdot X^i$

Notice that this is exactly the same as our def. of RS codes, except that we restrict $(c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n$ instead of $\mathbb{F}_{2^m}^n$. In particular, $\text{dist}(BCH(n,d)) = d$.

b/c $BCH(n,d) \subseteq RS(n, n-d+1)$ and $RS(n, n-d+1)$ has dist d.

NOTE. BCH codes make sense if you replace "2" with "q" [any prime power].
We focus on binary codes for today.

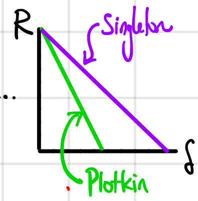
FALSE

CLAIM. BCH codes are linear codes with dimension $\ge n - d + 1$.

BAD **Proof.** $c(\gamma^j) = 0$ is a linear constraint; there are $d-1$ such constraints, so the dimension is at least $n - (d-1)$.

GREAT! So, BCH codes are binary codes that meet the Singleton bound!

Erm....



doesn't that violate the Plotkin bound??

(Yes).

What's wrong?!

The problem is that the constraints $c(\gamma^j)=0$ are linear over $\mathbb{F}_{2^m}$, not over $\mathbb{F}_2$. Fortunately, BCH codes ARE still linear over $\mathbb{F}_2$:

CLAIM. $BCH(n,d)$ is $\mathbb{F}_2$-linear with dimension $\geq n-(d-1)\lg(n+1)$.

proof. Each constraint $c(\gamma^j)=0$ is actually $m=\log_2(n)$ linear constraints over $\mathbb{F}_2$. To see this, we'll use the fact that $\mathbb{F}_{2^m}$ is actually a vector space over $\mathbb{F}_2$ of dimension m:

$\mathbb{F}_{2^m}$ has the same additive structure as $\mathbb{F}_2^m$.

So it makes sense to write elements $\alpha\in\mathbb{F}_{2^m}$ as vectors $v_\alpha\in\mathbb{F}_2^m$, as long as we're only going to be adding them or multiplying by scalars in $\mathbb{F}_2$.

Then $c(\gamma^j)=0$ means:

$c_i\in\{0,1\}$

$$\sum_{i=0}^{n-1} c_i\gamma^{ij}=0 \iff \sum_{i=0}^{n-1} c_i\cdot \underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{v(\gamma^{ij})} = 0$$

$\in\mathbb{F}_2^m$

$$\iff m\Big\{\underbrace{\begin{bmatrix} | & | & | & & | \\ v_1 & v_{\gamma^i} & v_{\gamma^{2i}} & \cdots & v_{\gamma^{(n-1)i}} \\ | & | & | & & | \end{bmatrix}}_{n}\begin{bmatrix} \\ \vec{c} \\ \\ \end{bmatrix}=0$$

binary matrix

$\vec{c}\in\mathbb{F}_2^n$

So each $\mathbb{F}_{2^m}$-linear constraint is actually m $\mathbb{F}_2$-linear constraints, and altogether we have $m\cdot(d-1)$ $\mathbb{F}_2$-linear constraints.

$$n-m(d-1)=n-\underbrace{\log_2(n+1)}_{\substack{n=2^m-1\\ \Rightarrow m=\lg(n+1)}}\cdot(d-1),$$

as claimed.

In fact, we can do even better:

CLAIM. BCH(n,d) is $\mathbb{F}_2$-linear with dimension $\geq n - \lfloor \frac{d-1}{2} \rfloor \lg(n+1)$.

Proof. We'll show that the linear constraints $c(\gamma^j)=0$ are actually redundant: $c(\gamma^j)=0 \Leftrightarrow c(\gamma^{2j})=0$. This cuts the number of constraints in half, which gives the bound.

SUB CLAIM. For any $c \in \mathbb{F}_2[X]$, $\alpha \in \mathbb{F}_{2^m}$, $c(\alpha)=0 \Leftrightarrow c(\alpha^2)=0$.

pf.

$$
\begin{aligned}
& c(\alpha) = 0 & \\
\Leftrightarrow\ & [c(\alpha)]^2 = 0 & \text{Since } 0^2=0 \\
\Leftrightarrow\ & \Big[\sum_{i=0}^{n-1} c_i \alpha^i\Big]^2 = 0 & \text{Def. of } c(X) \\
\Leftrightarrow\ & \sum_{i=0}^{n-1} c_i^2 \alpha^{2i} = 0 & \text{Since in } \mathbb{F}_{2^m},\ 1+1=0, \text{ so } (a+b)^2 = a^2+ab+ab+b^2 = a^2+b^2 \\
\Leftrightarrow\ & \sum_{i=0}^{n-1} c_i \alpha^{2i} = 0 & \text{Since } c_i \in \{0,1\}, \text{ so } c_i^2 = c_i. \\
\Leftrightarrow\ & c(\alpha^2) = 0 & \text{Def. of } c(X) \text{ again.}
\end{aligned}
$$

and more generally $(\sum_i a_i)^2 = \sum_i a_i^2$.

and by the above reasoning the SUB CLAIM proves the CLAIM.

Is this good? $R \geq \frac{n - \lfloor \frac{d-1}{2} \rfloor \cdot \lg(n+1)}{n} \sim 1 - \frac{\delta}{2}\lg(n)$, aka

BCH codes

$$\boxed{\text{Rate } R, \quad \text{distance } \delta \sim \frac{2}{\lg(n)} \cdot (1-R).}$$

So BCH codes do <u>slightly</u> better than our STRAWMAN:

STRAWMAN

$$\boxed{\text{Rate } R, \quad \text{distance } \delta \sim \frac{1-R}{\lg(n)}}$$

But still not asymptotically good :(.

However! Note that BCH codes can be decoded with either Berlekamp-Welch or Berlekamp-Massey! So that's pretty cool.

② BINARY REED-MULLER CODES

(Dumb) idea: just do RS codes over $\mathbb{F}_2$ directly! $\quad RS_2(n,k) = \left\{ (f(\alpha_1), \ldots, f(\alpha_n)) \;\middle|\; \begin{array}{l} f \in \mathbb{F}_2[X] \\ \deg(f) < k \end{array} \right\}$

This is obviously dumb since (a) $\deg(f) \leq q-1 = 1$ to be interesting

(b) $\alpha_1, \ldots, \alpha_n$ should be distinct pts in $\mathbb{F}_2$, so $n \leq 2$.

However, one fix is to add more variables.

DEF. The BINARY m-VARIATE REED-MULLER CODE of DEGREE r is

$$RM_2(m,r) = \Big\{ \underbrace{\big(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{2^m})\big)}_{\substack{\{\alpha_1, \ldots, \alpha_{2^m}\} = \mathbb{F}_2^m, \\ \text{in any pre-determined order.}}} : \underbrace{f \in \mathbb{F}_2[X_1, \ldots, X_m]}_{\substack{m\text{-variate polynomials over } \mathbb{F}_2. \text{ eg,} \\ f(X_1,X_2) = 1 + X_1X_2 + X_1}}, \underbrace{\deg(f) \leq r}_{\substack{\deg \text{ means total degree. eg,} \\ \deg(X_1X_2) = 2.}} \Big\}$$

Parameters:

Block length $n = 2^m$

Dimension $k = \sum_{j=0}^{r} \binom{m}{j} = \mathrm{Vol}_2(r,m)$. ← This is the number of coefficients in $f(X_1,\ldots,X_m) = \sum_{\substack{S \subseteq [m] \\ |S| \leq r}} c_S \cdot \prod_{i \in S} X_i$, a generic degree $\leq r$ m-variate polynomial over $\mathbb{F}_2$.

Distance $d = ?$

LEMMA (Binary Schwartz-Zippel)

Let $f \in \mathbb{F}_2[X_1, \ldots, X_m] \neq 0$, with $\deg(f) \leq r$.

Then $\sum_{\alpha \in \mathbb{F}_2^m} \mathbb{1}\{f(\alpha) \neq 0\} \geq 2^{m-r}$.

We may do the proof later for a more general version, but if you haven't seen this before it's a FUN EXERCISE!

So $\mathrm{dist}(RM_2(m,r)) \geq 2^{m-r}$. This is because $RM_2(m,r)$ is linear, and so as usual we only need to look at the minimum wt of a codeword.

And, it turns out this is the correct answer: consider $f(X_1,\dots,X_m) = X_1 \cdot X_2 \cdots X_r$.
This vanishes whenever any of $X_1,\dots,X_r = 0$, and so

$$|\{\alpha \in \mathbb{F}_2^m : f(\alpha) \neq 0\}| = |\{\alpha \in \mathbb{F}_2^m : \alpha_1 = \cdots = \alpha_r = 1\}| = 2^{m-r}.$$

So for $RM_2(m,r)$: $n = 2^m$

$$k = \mathrm{Vol}_2(r,m) \qquad \Rightarrow \qquad R = \mathrm{Vol}_2(r,m)/2^m$$

$$d = 2^{m-r} \qquad\qquad \delta = 1/2^r$$

RM codes also admit efficient decoding algs. We'll see some of these later in the course.

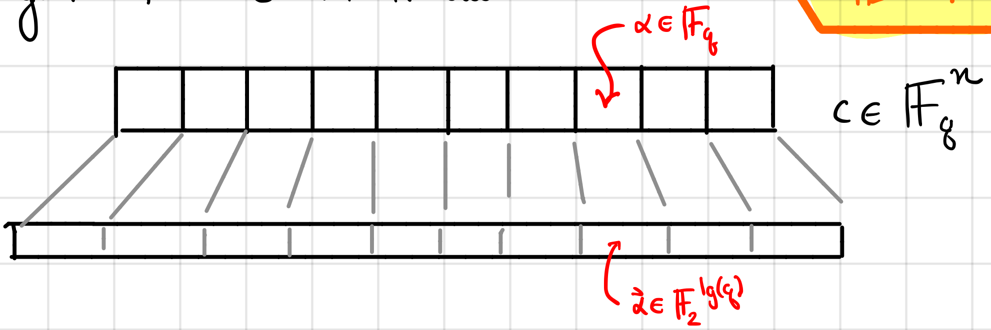
Unfortunately, this isn't asymptotically good either.
If $\delta = \Theta(1)$, then r is constant but $m \nearrow$, so $R \searrow 0$.

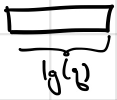
So this doesn't acheive the GOAL either... :(

③ CONCATENATED CODES.

NOTE: WE BARELY STARTED TALKING ABOUT THESE IN CLASS, AND WILL PICK UP HERE NEXT TIME.

Let's go back to our STRAWMAN code:

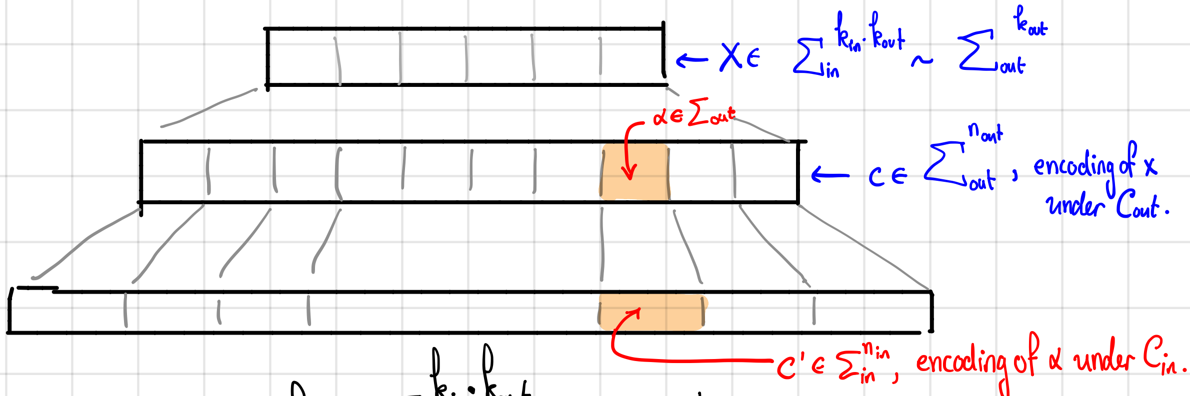


This wasn't a good idea, because if I were a bad guy, I'd mess up one bit from each of $d+1$ of the length $\lg(q)$ blocks. So these  blocks were not very robust.

IDEA. Let's encode each block $\vec{\alpha} \in \mathbb{F}_2^{\lg_2(q)}$ with a good binary ECC!

DEF. Let $C_{out} \subseteq \Sigma_{out}^{n_{out}}$ be a q_{out}-ary code of dimension k_{out} and distance d_{out}.
Let $C_{in} \subseteq \Sigma_{in}^{n_{in}}$ be the same thing with "in" subscripts, so that $q_{out} = q_{in}^{k_{in}}$.

The CONCATENATED CODE $C_{in} \circ C_{out} \subseteq \Sigma_{in}^{n_{out} \cdot n_{in}}$ is defined [by picture...] by



Formally, the encoding of $x \in \mathbb{F}_{q_{in}}^{k_{in} \cdot k_{out}}$ is given by:

- Treat $x \in \left[\mathbb{F}_{q_{in}}^{k_{in}}\right]^{k_{out}} \cong \mathbb{F}_{q_{out}}^{k_{out}}$
- Let $y = C_{out}(x) \in \mathbb{F}_{q_{out}}^{n_{out}} \cong \left[\mathbb{F}_{q_{in}}^{k_{in}}\right]^{n_{out}}$
- Let $c = \underbrace{Enc_{in}(y_1) \circ \cdots \circ Enc_{in}(y_{n_{out}})}_{n_{in} \cdot n_{out} \text{ bits}}$ where $Enc_{in} : \mathbb{F}_{q_{in}}^{k_{in}} \to \mathbb{F}_{q_{in}}^{n_{in}}$, and $\circ$ is concatenation.

Parameters of Concatenated Codes:

alphabet: Σ_{in}

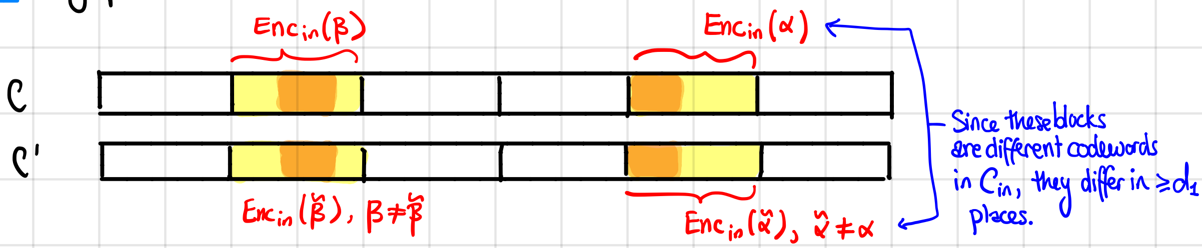
message length: $k_{in} \cdot k_{out}$

Codeword length: $n_{out} \cdot n_{in}$

So the rate is $\dfrac{k_{in} \cdot k_{out}}{n_{in} \cdot n_{out}} = R_{in} \cdot R_{out}$

PROPOSITION. The distance of $C_{in} \circ C_{out}$ is at least $d_{in} \cdot d_{out}$.

pf. by picture: Let $c, c' \in C_{in} \circ C_{out}$:



- At least d_{out} blocks of c, c' are encoding different symbols.
- In each of those, there are at least d_1 symbols in Σ_{in} that differ.
- So that's $d_{out} \cdot d_{in}$ differences total.

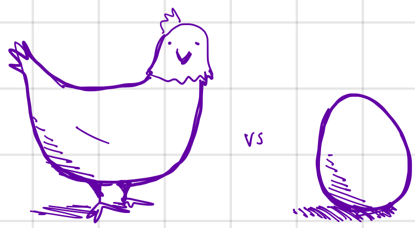
Finally! Progress to our GOAL.

To obtain an EXPLICIT, ASYMPTOTICALLY GOOD BINARY CODES:

1. Set C_{out} = Reed-Solomon Code
2. Set C_{in} = EXPLICIT, ASYMPTOTICALLY GOOD BINARY CODE.

D'oh.

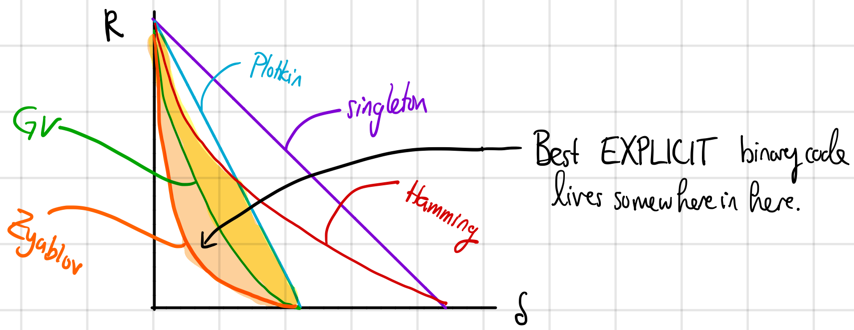
But actually it's OKAY!
The secret is that C_{in} will be short enough that we can do exhaustive stuff efficiently.



THM. For any $\varepsilon > 0$, there is a family of explicit binary linear codes of rate R and distance δ, satisfying

$$R \geq \sup_{0 < r < 1 - H_2(\delta) - \epsilon} \left(r \cdot \left(1 - \frac{\delta}{H_2^{-1}(1 - r - \epsilon)} \right) \right)$$

That is called the Zyablov Bound.



pf. The construction will be:

- C_{out} = RS code with rate R_{out}, dist. $\delta_{out} = 1 - R_{out}$
- C_{in} = Binary linear code on the GV bound, with rate $r \geq 1 - H_2(\delta_{in}) - \epsilon$.

We still need to say how to get this... we'll get there.

- The concatenated code has:

RATE: $R_{out} \cdot r$

DISTANCE: $\delta_{out} \cdot \delta_{in} = (1 - R_{out})\, H_2^{-1}(1 - r - \epsilon)$.

$R_{out} = 1 - \frac{\delta_{in}\delta_{out}}{H_2^{-1}(1-r-\epsilon)}$

Hence $R = R_{out} \cdot r = r \cdot \left(1 - \frac{\delta_{in} \cdot \delta_{out}}{H_2^{-1}(1 - r - \epsilon)} \right)$,

and then we get to choose r.

So that gives us the combinatorial bound.
Next, the algorithmic bit.

continued...

proof continued...

Suppose the evaluation pts for the RS code are $\mathbb{F}_q^*$, where $q = 2^{k_{in}}$.

So $k_{in} = \lg(q)$, and $n_{out} = q-1$

ALG 1. Search over all $\mathbb{F}_2$-linear codes of rate r and dimension k_{in}.

There are approximately $2^{n_{in} \cdot k_{in}} = 2^{k_{in}^2/r} = 2^{\lg^2(q)/r}$ such codes

final block length

$$q = n_{out} + 1 = \frac{n}{n_{in}} + 1 = \frac{r\,n}{k_{in}} + 1 \Rightarrow n \sim \frac{1}{r}\, q \lg(q)$$

So $\lg^2(q) = \Theta(\lg^2(n))$, so that's $2^{\Theta(\lg^2(n))} = n^{\Theta(\lg(n))}$, which is **NOT** polynomial time. :(

ALG 2. On your homework, you will give an alg to construct binary linear codes on the GV bound w/ rate r, dim k_{in} in time $2^{O(k_{in})}$, instead of $2^{O[k_{in}^2]}$. This will fix the above, and proves the theorem.

So we have acheived (most of) our goal.

HOWEVER, this version of "explicit" [can compute it in polynomial time] may be unsatisfying.

WHAT IF I WANTED "explicit" meaning: "Give me a short, useful description"
Formally, I'd like to be able to compute any entry G_{ij} in time polylog(n).

IDEA: Instead of using the <u>same</u> inner code at every position and requiring it to be good, we'll use a different inner code in each position.

We won't actually know which of these inner codes is good, but we'll know that enough of them are good.

THM. Let $\varepsilon > 0$, fix any k. There is an ensemble of binary linear codes

$$C_{in}^1, C_{in}^2, \dots, C_{in}^N \subseteq \mathbb{F}_2^{2k}$$

of rate $\frac{1}{2}$, with $N = 2^k - 1$, so that for at least $(1-\varepsilon)N$ values of i, C_{in}^i has distance at least $H_q^{-1}(\frac{1}{2} - \varepsilon)$.

This is called the **WOZENCRAFT ENSEMBLE**.

<u>proof idea.</u> For $x \in \mathbb{F}_2^k$, treat it as an element of $\mathbb{F}_{2^k}$.
Then for each $\alpha \in \mathbb{F}_{2^k}^*$, let the α^{th} code C_{in}^α be the image of the encoding map

$$E_{in}^\alpha : x \longmapsto \underbrace{(x, \underbrace{\alpha \cdot x}_{\text{multiplication in } \mathbb{F}_{2^k}})}_{\text{treat these as } 2k \text{ bits.}}$$

FUN EXERCISE: finish the proof!

Using the Wozencraft ensemble, we can implement the idea above to obtain the JUSTESAN CODE.

DEF (JUSTESEN CODE)

Let $k > 0$ [k will be the dimension of the inner codes in the Wozencraft Ensemble]

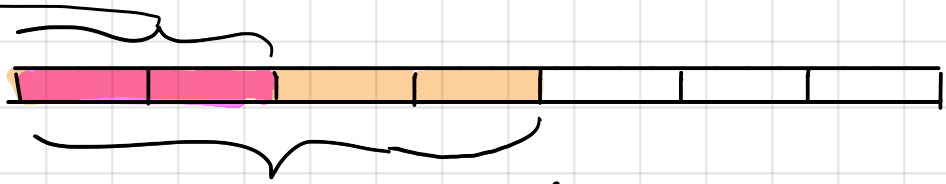
Let $C_{out} = RS_{2^k}\left(\mathbb{F}_{2^k}^*,\ 2^k-1,\ R_{out}\cdot(2^k-1)\right)$

Use the Wozencraft Ensemble as the inner code:

$$C = \left\{ \left\langle E_{in}^{\alpha}\left(f(\alpha)\right) \right\rangle_{\alpha\in\mathbb{F}_{2^k}^*} : f\in \mathbb{F}_{2^k}[X],\ \deg(f) < R_{out}(2^k-1) \right\}$$

CLAIM. Let R_{out} be constant, choose $\varepsilon > \frac{1-R_{out}}{2}$. Then C is asymptotically good!

pf. (sketch) The rate is $R_{out}/2$, and it's a binary linear code, so we just have to consider the minimum wt to compute the distance. Consider any codeword:



- At least $(1-R_{out}) \geq 2\epsilon$ fraction of the chunks are the encodings of nonzero symbols.
- At most an ε-fraction of chunks have "bad" inner codes, so at least an $2\varepsilon - \varepsilon = \varepsilon$-fraction of chunks are the encodings of nonzero symbols with a "good" inner code.

For each of those, since the inner code has distance $\geq H_2^{-1}(\frac{1}{2}-\epsilon) = \Theta(1)$, a constant fraction of the bits in each of a constant fraction of blocks are nonzero.

$\Rightarrow$ Each nonzero codeword has relative weight larger than some constant. Thus the code is asymptotically good.

So the JUSTESEN CODE is "EXPLICIT" in the way we wanted.
The α'th block is given by $(f(\alpha), \alpha \cdot f(\alpha)) \in \mathbb{F}_q^2 \sim \mathbb{F}_2^{2\lg(q)}$.
That's pretty explicit!

FUN EXERCISE. What is the best rate/distance trade-off you can get w/ the Justesen code?

QUESTIONS to PONDER

1. How would you efficiently decode a concatenated code?
2. How would you efficiently decode Reed-Muller codes.
3. What happens to the Wozencraft ensemble if you do

$$x \mapsto (x, \alpha \cdot x, \alpha^2 \cdot x, \ldots, \alpha^r \cdot x) \ ?$$