More on the Reliability Function of the BSC

Alexander Barg
DIMACS, Rutgers University

Andrew McGregor
University of Pennsylvania

ISIT 2003, Yokohama

Some Definitions

Some Definitions

- Communicating over a binary symmetric channel with cross-over probability p.

Some Definitions

- Communicating over a binary symmetric channel with cross-over probability p.
- We use a length n binary code $C=\{x_1, x_2, \ldots x_{|C|}\}$ with rate ≥ R ie.

Some Definitions

- Communicating over a binary symmetric channel with cross-over probability p.
- We use a length n binary code $C=\{x_1, x_2, \ldots x_{|C|}\}$ with rate ≥ R ie.

$$|C| \geq 2^{nR}$$

Some Definitions

- Communicating over a binary symmetric channel with cross-over probability p.
- We use a length n binary code $C=\{x_1, x_2, \dots x_{|C|}\}$ with rate ≥ R ie.

$$|C| \geq 2^{nR}$$

- No matter what code we use there is the possibility of making errors - for a given rate of transmission there is some degree of error that is inherent to the channel itself.

Making Decoding Errors

- Maximum Likelihood Decoding: When we receive a word y we'll guess that the sent codeword is the codeword that lies closest to it.
- For each codeword x we define the Voronoi region:
- Let $P_e(x)$ be the probability that, when codeword x is transmitted, this decoding procedure leads to an error. Therefore we have

Making Decoding Errors

- Maximum Likelihood Decoding: When we receive a word y we'll guess that the sent codeword is the codeword that lies closest to it.
- For each codeword x we define the Voronoi region:

$$D(x) = \{y \in \{0,1\}^n : d(x,y) < d(x_j,y) \forall x_j \in C \setminus x\}$$

- Let $P_e(x)$ be the probability that, when codeword x is transmitted, this decoding procedure leads to an error. Therefore we have

Making Decoding Errors

- Maximum Likelihood Decoding: When we receive a word y we'll guess that the sent codeword is the codeword that lies closest to it.
- For each codeword x we define the Voronoi region:

$$D(x) = \{y \in \{0,1\}^n : d(x,y) < d(x_j,y) \forall x_j \in C \setminus x\}$$

- Let $P_e(x)$ be the probability that, when codeword x is transmitted, this decoding procedure leads to an error. Therefore we have

$$P_e(x) = P_x(\{0,1\}^n \setminus D(x))$$

The Reliability Function

- The average error probability of decoding is

- We're interested in

- We present a new lower bound for this quantity, or equivalently, an upper bound on the reliability function or error exponent of the channel:

The Reliability Function

- The average error probability of decoding is

$$P_e(C) = \frac{1}{|C|}\sum_{x\in C} P_e(x)$$

- We're interested in

- We present a new lower bound for this quantity, or equivalently, an upper bound on the reliability function or error exponent of the channel:

The Reliability Function

- The average error probability of decoding is

$$P_e(C) = \frac{1}{|C|}\sum_{x\in C} P_e(x)$$

- We're interested in

$$P_e(R) = \min_{C:Rate(C)>R} P_e(C)$$

- We present a new lower bound for this quantity, or equivalently, an upper bound on the reliability function or error exponent of the channel:

The Reliability Function

- The average error probability of decoding is

$$P_e(C) = \frac{1}{|C|} \sum_{x \in C} P_e(x)$$

- We're interested in

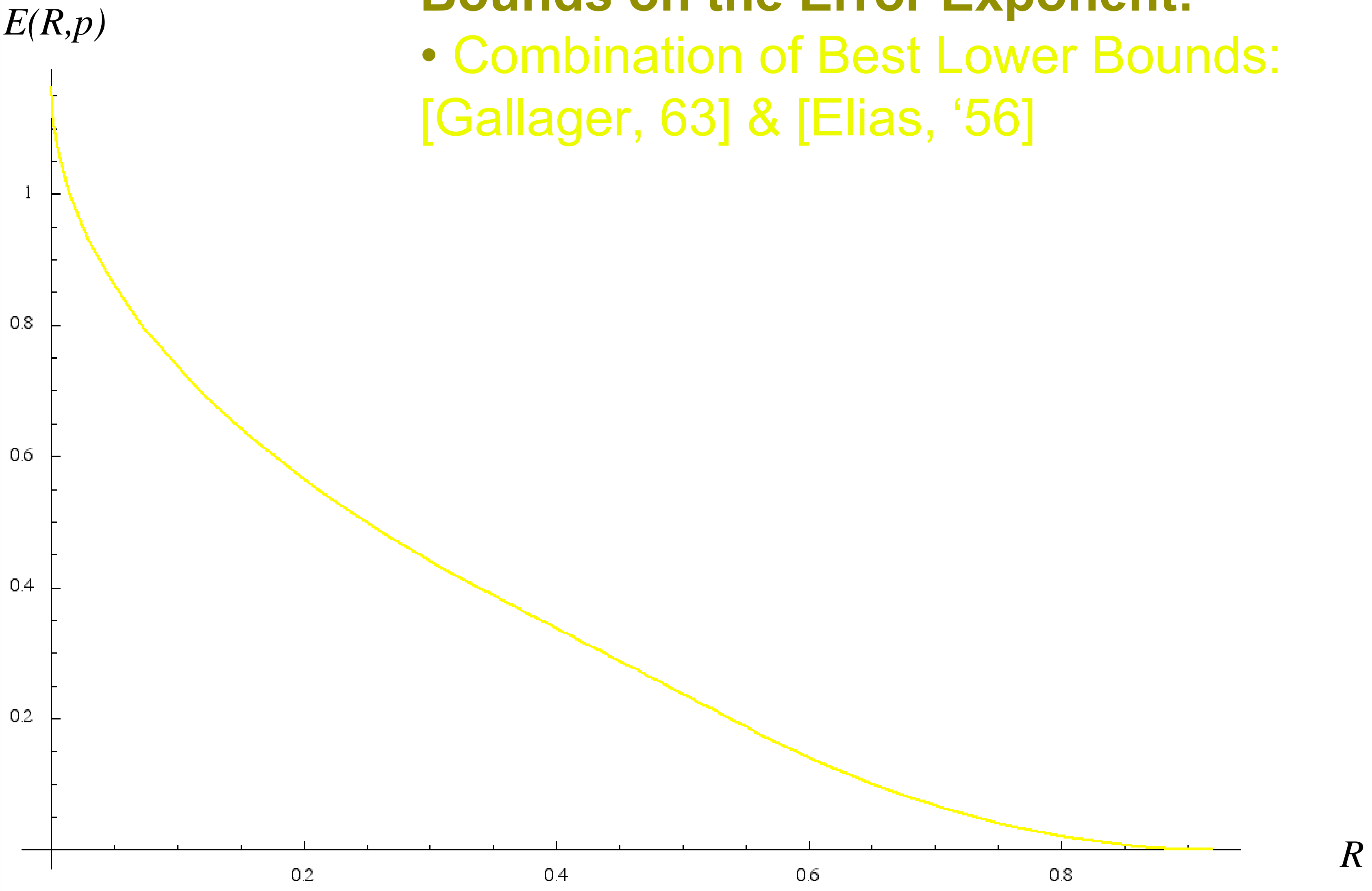
$$P_e(R) = \min_{C: Rate(C) > R} P_e(C)$$

- We present a new lower bound for this quantity, or equivalently, an upper bound on the reliability function or error exponent of the channel:

$$E(R,p) = -\lim_{n \to \infty} \frac{1}{n} \log\left[\min_{C: R(C) > R} P_e(C)\right]$$

Bounds on the Error Exponent:

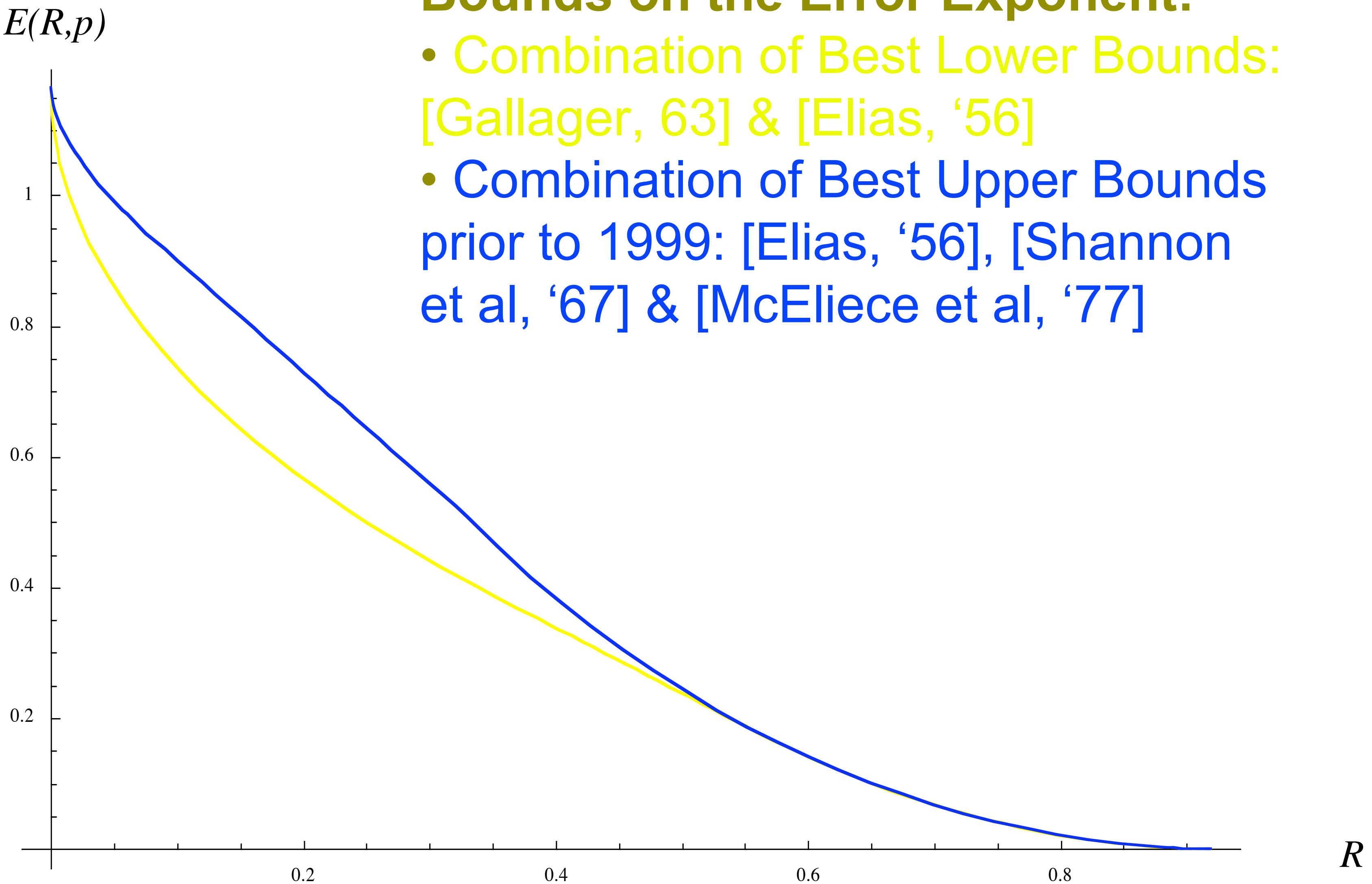
- Combination of Best Lower Bounds: [Gallager, 63] & [Elias, '56]



p=0.01

Bounds on the Error Exponent:

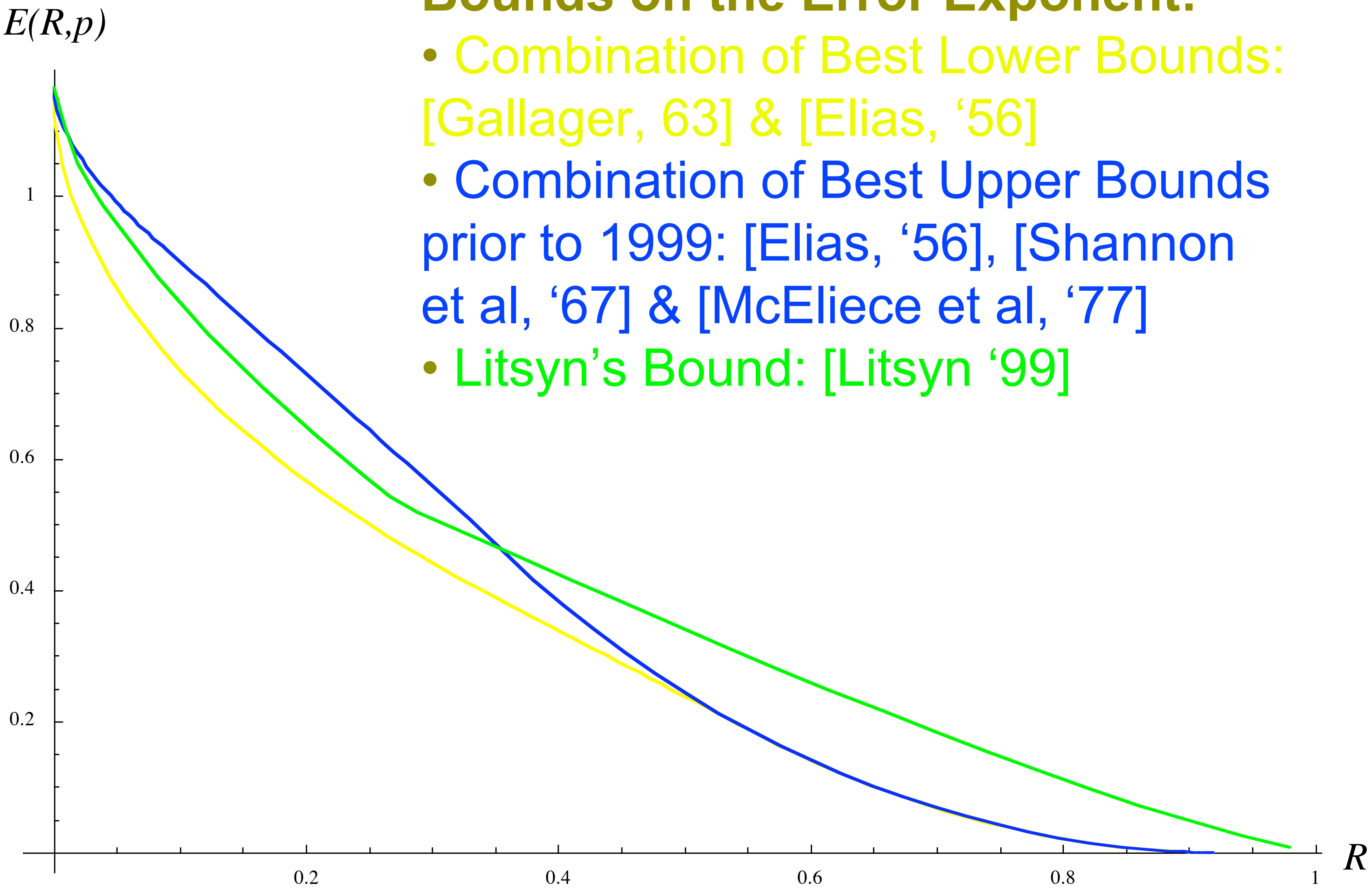
- Combination of Best Lower Bounds: [Gallager, 63] & [Elias, ‘56]
- Combination of Best Upper Bounds prior to 1999: [Elias, ‘56], [Shannon et al, ‘67] & [McEliece et al, ‘77]



$p=0.01$

Bounds on the Error Exponent:

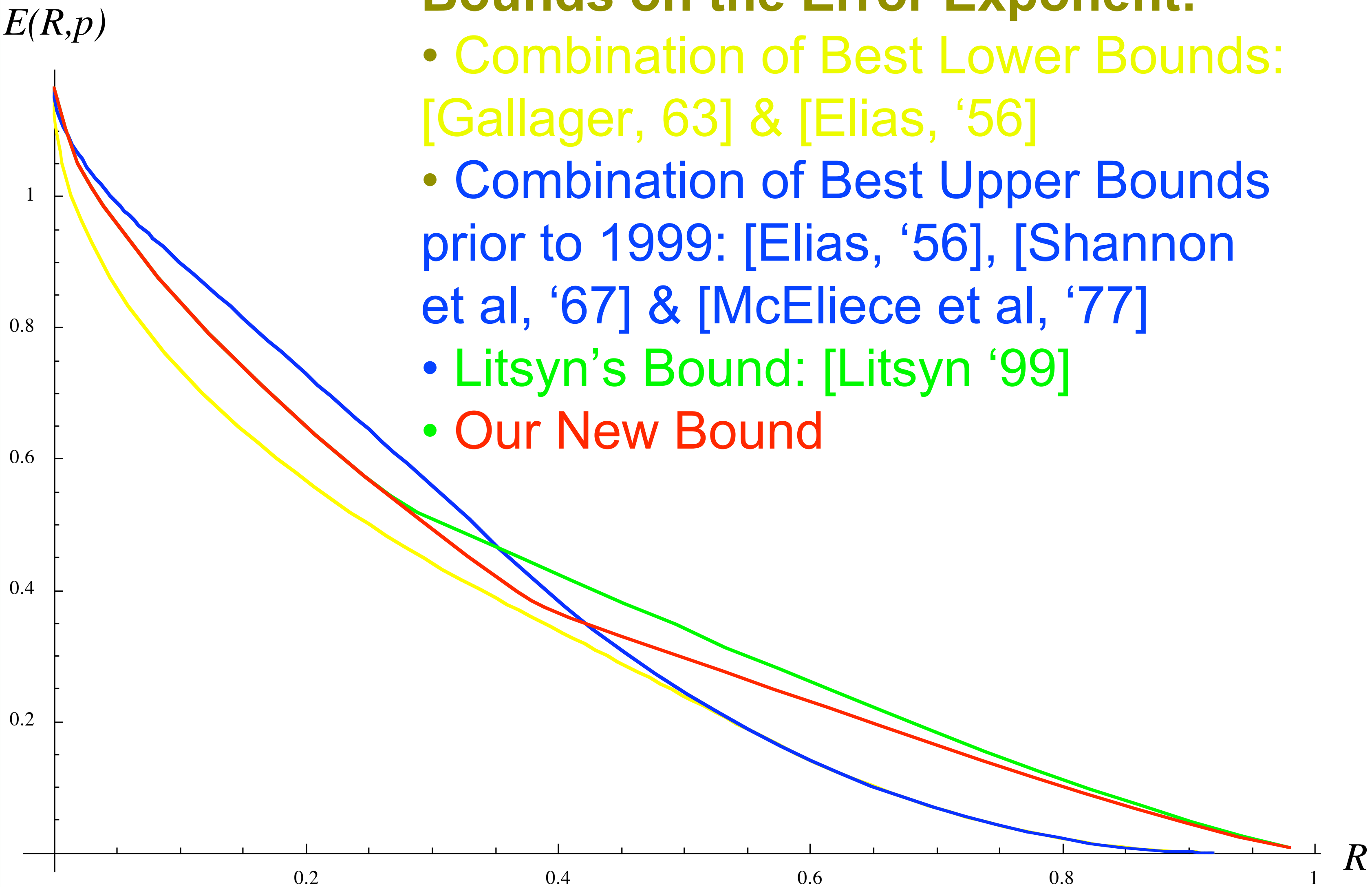
- Combination of Best Lower Bounds: [Gallager, 63] & [Elias, ‘56]
- Combination of Best Upper Bounds prior to 1999: [Elias, ‘56], [Shannon et al, ‘67] & [McEliece et al, ‘77]
- Litsyn’s Bound: [Litsyn ‘99]



p=0.01

Bounds on the Error Exponent:

- Combination of Best Lower Bounds: [Gallager, 63] & [Elias, ‘56]
- Combination of Best Upper Bounds prior to 1999: [Elias, ‘56], [Shannon et al, ‘67] & [McEliece et al, ‘77]
- Litsyn’s Bound: [Litsyn ‘99]
- Our New Bound



p=0.01

Litsyn’s Distance Distribution Bound

- Define

- Litsyn’s Distance Distribution Bound: For any code C of rate R there exists a w such that

Litsyn's Distance Distribution Bound

- Define

$$B_w(x) = |\{x_j : d(x\ , x_j) = w\}|$$

- Litsyn's Distance Distribution Bound: For any code C of rate R there exists a w such that

Litsyn’s Distance Distribution Bound

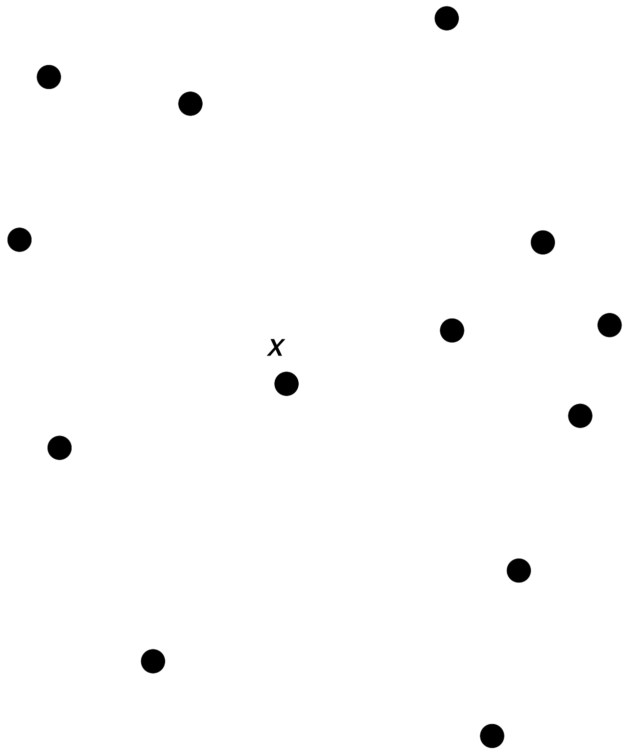
- Define

$$B_w(x) = |\{x_j : d(x ,x_j) = w\}|$$

- Litsyn’s Distance Distribution Bound: For any code C of rate R there exists a w such that

$$B_w(x) \geq \mu(R,w)$$

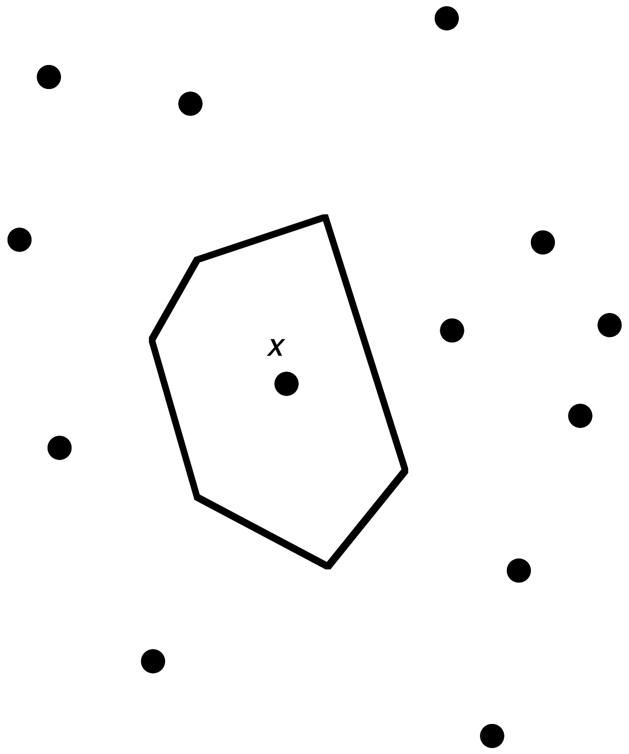
Estimating $P_e(x)$



$$P_e(x) = P_x(\{0,1\}^n \setminus D(x))$$

Estimating $P_e(x)$

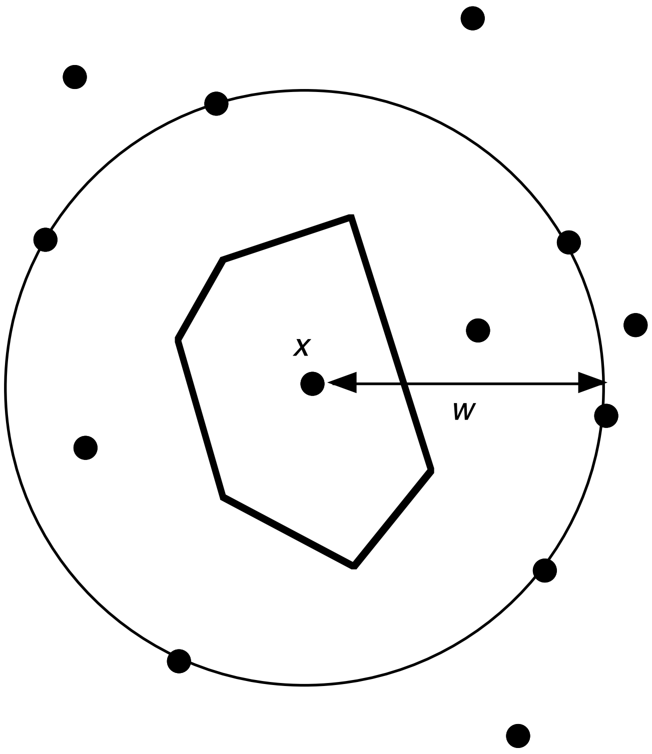
The Voronoi Region



$$P_e(x) = \sum_{y \in C:\, d(y,x_j) \le d(y,x) \text{ for some } x_j \in C} p^{d(y,x)} (1-p)^{n-d(y,x)}$$

Estimating $P_e(x)$

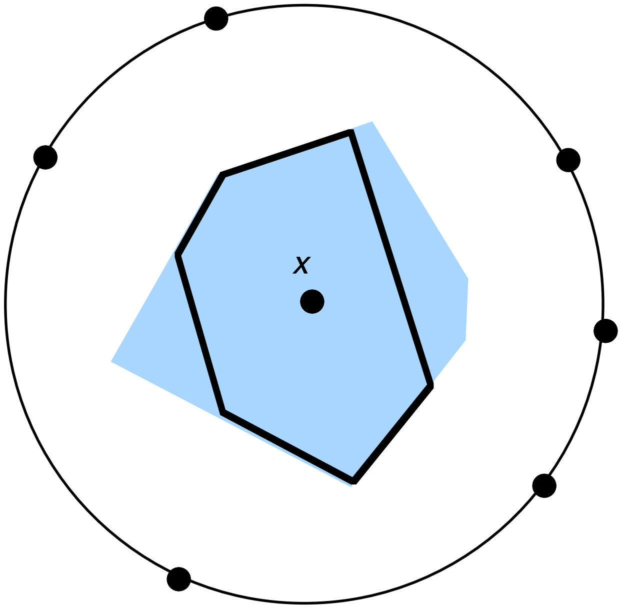
Use the distance distribution result…



$$P_e(x) = \sum_{y \in C:\, d(y,x_j) \le d(y,x) \text{ for some } x_j \in C} p^{d(y,x)}(1-p)^{n-d(y,x)}$$

Estimating $P_e(x)$

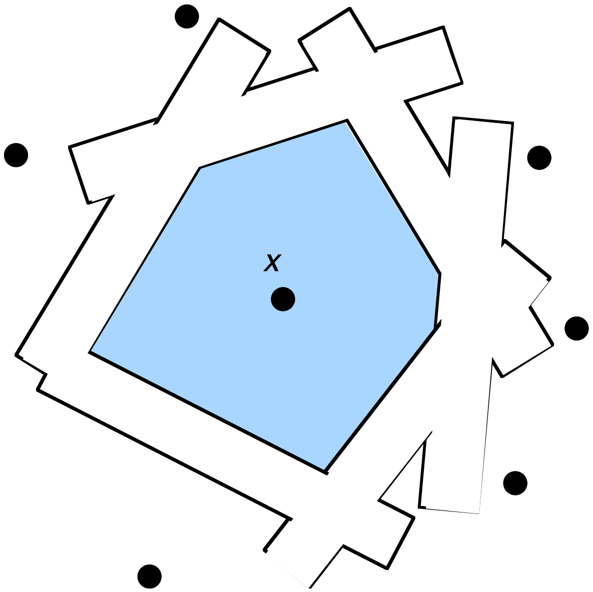
Approximating the Voronoi Region…



$$P_e(x) \geq \sum_{y\in C:\, d(y,x_j)\leq d(y,x) \text{ for some } x_j\in C \text{ where } d(x,x_j)=w} p^{d(y,x)}(1-p)^{n-d(y,x)}$$

Estimating *$P_e(x)$*

Introducing the X_j…



For each neighbour x_j define a set X_j such that

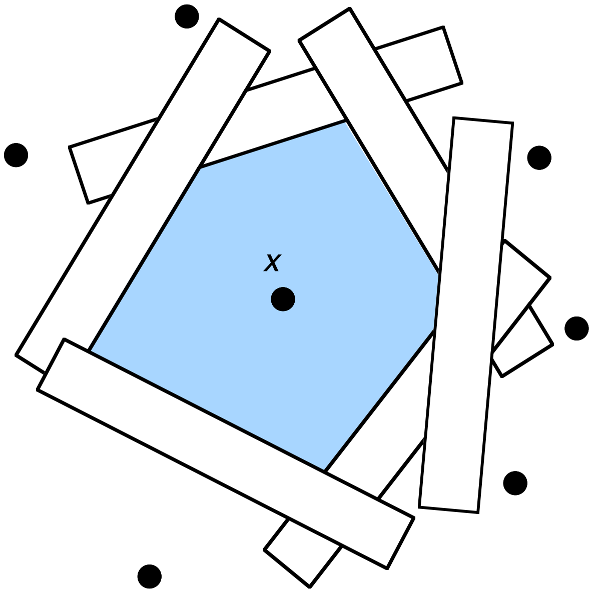
$$y \in X_j \Rightarrow$$

$$d(y,x_j) \le d(y,x)$$

$$P_e(x) \ge P_x\Big(\bigcup_{j:d(x,x_j)=w} X_j\Big)$$

Estimating $P_e(x)$

"Pruning" the X_j...



For each neighbour x_j assign a priority n_j at random. Let

$$Y_j = X_j \setminus \bigcup_{k:n_k>n_j} X_k$$

$$P_e(x) \geq \sum_{j:d(x,x_j)=w} P_x(Y_j)$$

Estimating $P_e(x)$

Applying the Reverse Union Bound…

The Reverse Union Bound:

Giving us our final shape of our bound:

Estimating $P_e(x)$

Applying the Reverse Union Bound…

The Reverse Union Bound:

$$P_x(Y_j) = P_x(X_j \setminus \bigcup_{k:n_k>n_j} X_k)$$

$$\geq P_x(X_j)(1 - \sum_{k:n_k>n_j} P_x(X_k \mid X_j))$$

Giving us our final shape of our bound:

Estimating $P_e(x)$

Applying the Reverse Union Bound…

The Reverse Union Bound:

$$P_x(Y_j) = P_x(X_j \setminus \bigcup_{k:n_k>n_j} X_k)$$

$$\geq P_x(X_j)(1 - \sum_{k:n_k>n_j} P_x(X_k \mid X_j))$$

Giving us our final shape of our bound:

$$P_e(x) \geq \sum_{j:d(x,x_j)=w} P_x(X_j)(1 - \sum_{k:n_k>n_j} P_x(X_k \mid X_j))$$

- Now look across the entire code. Let X_{ij} and Y_{ij} be the sets for the neighbourhood of codeword x_i.
- Therefore we have:

and

where, the amount of “pruning” is

- What we do now depends on the values of the K_{ij}…

- Now look across the entire code. Let X_{ij} and Y_{ij} be the sets for the neighbourhood of codeword x_i.
- Therefore we have:

$$P_e(x_i) \geq \sum_{j:d(x_i,x_j)=w} P_i(Y_{ij})$$

and

where, the amount of “pruning” is

- What we do now depends on the values of the K_{ij}…

- Now look across the entire code. Let X_{ij} and Y_{ij} be the sets for the neighbourhood of codeword x_i.
- Therefore we have:

$$P_e(x_i) \geq \sum_{j:d(x_i,x_j)=w} P_i(Y_{ij})$$

and

$$P(Y_{ij}) \geq P_i(X_{ij})(1-K_{ij})$$

where, the amount of “pruning” is

- What we do now depends on the values of the K_{ij}…

- Now look across the entire code. Let X_{ij} and Y_{ij} be the sets for the neighbourhood of codeword x_i.
- Therefore we have:

$$P_e(x_i) \geq \sum_{j:d(x_i,x_j)=w} P_i(Y_{ij})$$

and

$$P(Y_{ij}) \geq P_i(X_{ij})(1-K_{ij})$$

where, the amount of “pruning” is

$$K_{ij} = \sum_{k:n_{ik}>n_{ij}} P_i(X_{ik} \mid X_{ij})$$

- What we do now depends on the values of the K_{ij}…

- Consider the set of codewords

- Consider the set of codewords

$$S=\{x_j : K_{ij} > 1/2 \text{ for some } i\}$$

- Consider the set of codewords

$$S=\{x_j : K_{ij} > 1/2 \textit{ for some } i\}$$

- Either this is a “substantially” sized subcode or it isn’t.

- Consider the set of codewords

$$S=\{x_j : K_{ij} > 1/2 \text{ for some } i\}$$

- Either this is a “substantially” sized subcode or it isn’t.
- Ie, either we had to do a lot of pruning or we didn’t have to do a lot of pruning.

If S was not substantially sized…

- Just remove codewords in S from the code!
- Then in the remaining code we have for all Y_{ij}

$$P_i(Y_{ij}) \geq P_i(X_{ij})/2$$

- Hence, modulo constant factors, the average error probability satisfies

$$P_e(C,p) \geq A(w)\mu(w)$$

- where $A(w) = P_i(X_{ij})$

If *S* was substantially sized…

- Consider

where

- Consider a codeword x_j such that $K_{ij}>1/2$. Then there exists an l' such that

$$B_{l'}(x_j) > 1/(2nB(w,l'))$$

- The upshot of S being substantial is that we discover a nuisance level l_1, such that

$$P_e(x_j) \geq A(w)/B(w,l_1)$$

  and a substantial number of codewords have the

$$B_{l_1}(x_j) > 1/B(w,l_1)$$

If *S* was substantially sized…

- Consider

where
$$K_{ij} = \sum_{k:n_{ik}>n_{ij}} P_i(X_{ik} \mid X_{ij}) = \sum_{l=0}^{n} \left(\sum_{k:n_{ik}>n_{ij}, d(x_j, x_k)=l} B(w,l) \right)$$

- Consider a codeword x_j such that $K_{ij}>1/2$. Then there exists an l' such that
$$B_{l'}(x_j) > 1/(2nB(w,l'))$$
- The upshot of S being substantial is that we discover a nuisance level l_1, such that
$$P_e(x_j) \geq A(w)/B(w,l_1)$$
and a substantial number of codewords have the
$$B_{l_1}(x_j) > 1/B(w,l_1)$$

If *S* was substantially sized…

- Consider

where

$$K_{ij} = \sum_{k:n_{ik}>n_{ij}} P_i(X_{ik} \mid X_{ij}) = \sum_{l=0}^{n} \left(\sum_{k:n_{ik}>n_{ij}, d(x_j,x_k)=l} B(w,l) \right)$$

$$B(w,l) = P_i(X_{ik} \mid X_{ij}) \text{ where } d(x_i,x_j) = d(x_i,x_k) = w,\ d(x_j,x_k) = l$$

- Consider a codeword x_j such that $K_{ij}>1/2$. Then there exists an l' such that

$$B_{l'}(x_j) > 1/(2nB(w,l'))$$

- The upshot of S being substantial is that we discover a nuisance level l_1, such that

$$P_e(x_j) \geq A(w)/B(w,l_1)$$

and a substantial number of codewords have the

$$B_{l_1}(x_j) > 1/B(w,l_1)$$

- A priori we don’t know whether we required a lot or a little pruning. We therefore take the weaker of the two bounds:

- But if there existed a nuisance level l_1 then we know that for a substantial number codewords such that

- Hence we can repeat the process with this new bound on the distribution.

- A priori we don’t know whether we required a lot or a little pruning. We therefore take the weaker of the two bounds:

$$P_e(C,p) \geq \min\left[A(w)\mu(w), \tfrac{A(w)}{B(w,l_1)}\right]$$

- But if there existed a nuisance level l_1 then we know that for a substantial number codewords such that

- Hence we can repeat the process with this new bound on the distribution.

- A priori we don’t know whether we required a lot or a little pruning. We therefore take the weaker of the two bounds:

$$P_e(C,p) \geq \min\left[A(w)\mu(w), \tfrac{A(w)}{B(w,l_1)}\right]$$

- But if there existed a nuisance level l_1 then we know that for a substantial number codewords such that

$$B_{l_1}(x) \geq \frac{1}{B(w,l_1)}$$

- Hence we can repeat the process with this new bound on the distribution.

Our Bound

- Continuing in this way we eventually get

$$P_e(C,p) \geq \min\left[A(w)\mu(w), \tfrac{A(l)}{B(w,l)}\right]$$

$$\text{where } 0 \leq l \leq w \leq \delta_{LP} n$$

- Minimizing over l and w gives us our final bound.

Random Linear Codes

- It can be shown that, with high probability, the weight distribution of a random linear code converges to

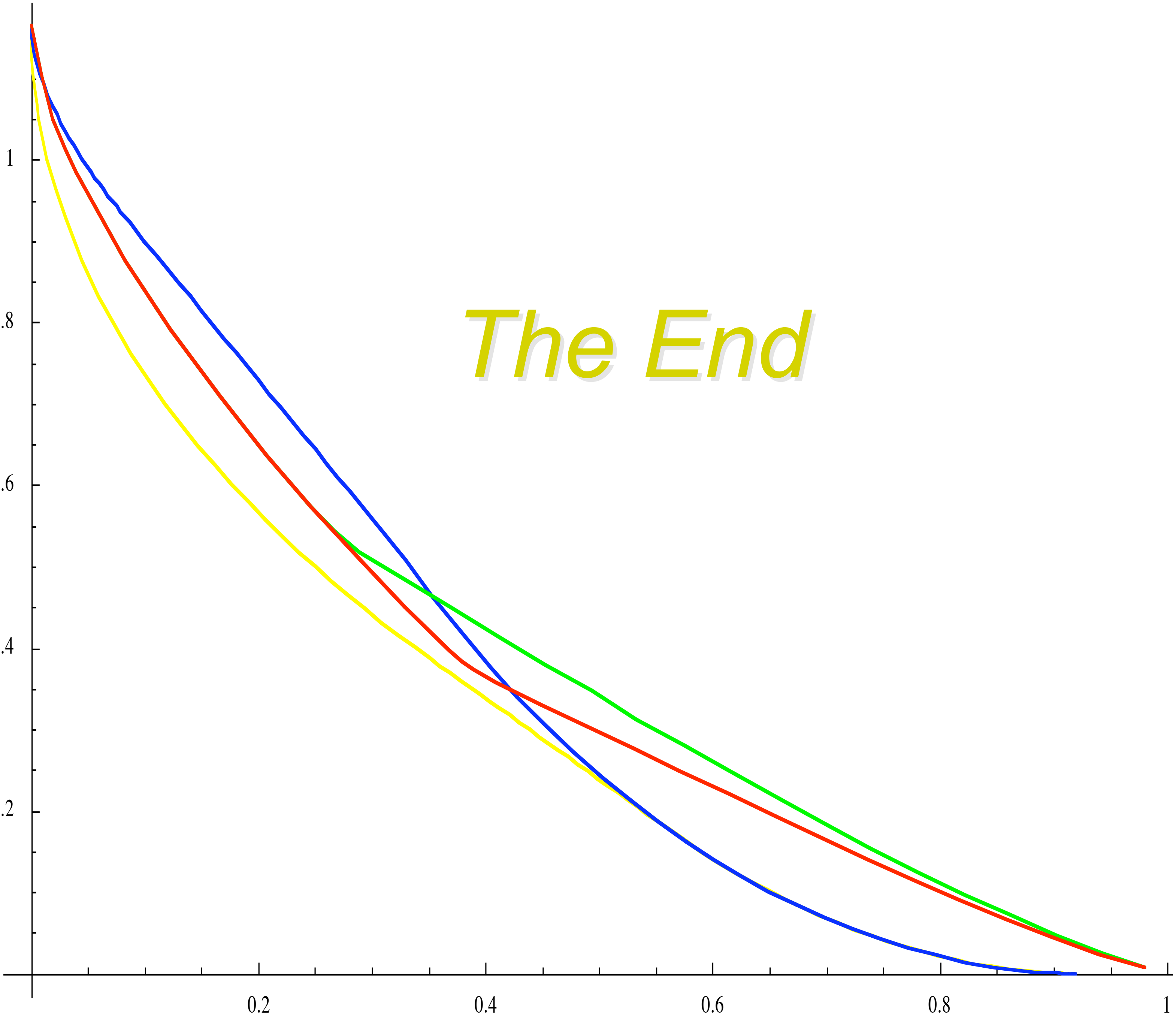
$$B_w = \exp[n(R+h(w)-1)]$$

- Using this instead of Litsyn's expression μ leads us to believe that the expurgation bound

$$E(R,p) \geq -\delta_{GV}(p)/2 \log 2p(1-p)$$

is tight for a random linear code for very low rates.

The End


1
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1