# Lecture 1: Basic problems of coding theory

Error-Correcting Codes (Spring 2016)
Rutgers University
Swastik Kopparty
Scribes: Abhishek Bhrushundi & Aditya Potukuchi

Administrivia was discussed at the beginning of this lecture. All the information about the course can be found on the course web page: **http://www.math.rutgers.edu/ sk1233/courses/codes-S16/**.

## 1 Basic concepts, definitions, and facts

### 1.1 Definitions

For most part of the course, $n$ will denote a parameter taking values in $\mathbb{N}$.

**Definition 1** (Hamming distance)**.** *The Hamming distance between $x, y \in \{0,1\}^n$ is defined as the number of coordinates in which $x$ and $y$ differ. We shall denote the Hamming distance between $x$ and $y$ by $\Delta(x, y)$.*

A related notion is that of *Hamming weight*:

**Definition 2** (Hamming weight)**.** *For $x \in \{0,1\}^n$, the Hamming weight of $x$, denoted by $wt(x)$ or $|x|$, is defined as $\Delta(x, 0)$, where $0$ denotes the all zero vector.*

Even though we defined the above concepts for strings over $\{0,1\}$, one can extend them to strings over an arbitrary alphabet $\Sigma$ in a natural manner.

The reader can easily check the following:

**Fact 3.** *For $x, y, z \in \{0,1\}^n$, $\Delta(x, y) + \Delta(y, z) \geq \Delta(x, z)$. Also, $\Delta(x, y) \geq 0$, and equality holds iff $x = y$. Another way of saying this is that $\Delta(.,.)$ is a metric on $\{0,1\}^n$.*

We are now in a position to define what a *code with minimum distance $d$* is:

**Definition 4** (Code with minimum distance $d$)**.** *A code with minimum distance $d$ in $\{0,1\}^n$ is a subset $C \subseteq \{0,1\}^n$ with the property that $min_{x,y \in C, x \neq y} \Delta(x, y) = d$. $d$ is called the minimum distance of the code $C$.*

Note that the notion of minimum distance is well defined for any subset of $\{0,1\}^n$. Before we go on, we introduce yet another useful concept.

**Definition 5** (Hamming ball)**.** *The Hamming ball of radius $r$ centered around $c \in \{0,1\}^n$ is the set $\{y \in \{0,1\}^n,\ \Delta(y, c) \leq r\}$. We will denote it by $B_n(c, r)$, though it's typical to drop the $n$ in the subscript when it's clear that we are working over $\{0,1\}^n$.*

## 1.2 A short discussion on error correction

What does error correction have to do with the combinatorial objects defined above? Let's consider the following scenario: Alice is on Earth and Bob's on the Moon. They are sending messages to each other, where the messages come from a subset $C$ of $\{0,1\}^n$. Given the huge distance between them, it's conceivable that whenever a message is sent, some of its bits get corrupted (flipped). As we shall see later, it's reasonable to assume that whenever a message $x \in C \subseteq \{0,1\}^n$ is transmitted, at most $t$ of its bits get corrupted. Alice and Bob want to be able to recover the original message from a corrupted one.

Let us suppose that the set $C$ had minimum distance $d$. How much error can be tolerated, i.e. how large can $t$ be?

**Claim 6.** $t \leq \lfloor \frac{d-1}{2} \rfloor$

*Proof.* Suppose $d < 2t+1$. Let $c_1$ and $c_2$ be in $C$ such that $\Delta(c_1, c_2) = d$. It follows that there is a string $x \in \{0,1\}^n$ such that $\Delta(x, c_1) \leq t$ and $\Delta(x, c_2) \leq t$. Now suppose Alice sends $c_1$ to Bob and it gets corrupted with $\leq t$ errors and becomes $x$ when it reaches Bob. Bob has no way to tell whether Alice had sent $c_1$ or $c_2$ since both can be corrupted with $\leq t$ errors and be made into $x$. □

But what happens when $t \leq \lfloor \frac{d-1}{2} \rfloor$? Can the original codeword be recovered? The following fact, which follows trivially from the triangle inequality, tells us that recovery is possible in this regime:

**Fact 7.** *Let $C$ be a code with minimum distance $d$. For any $c, c' \in C$, $c \neq c'$, we have that $B(c, \lfloor \frac{d-1}{2} \rfloor)$ is disjoint from $B(c', \lfloor \frac{d-1}{2} \rfloor)$.*

*Proof.* We will give a proof by contradiction. Suppose $x \in B(c, \lfloor \frac{d-1}{2} \rfloor) \cap B(c', \lfloor \frac{d-1}{2} \rfloor)$. Then $\Delta(x, c) \leq \lfloor \frac{d-1}{2} \rfloor$ and $\Delta(x, c') \leq \lfloor \frac{d-1}{2} \rfloor$, and by triangle inequality, we have $\Delta(c, c') \leq d - 1 < d$, which is a contradiction. □

Thus, we want the minimum distance to be as large as possible in order to be able to tolerate a large number of errors. But then one might argue, why not just take $C$ to be the set containing the all zero string and the all one string, making the distance $n$? The reason is that we not only want large distance, but also want $|C|$ to be large. Why?
Let's say that Alice and Bob have come up with a way to identify the integers $1, \ldots, |C|$ with the elements of $C$. Now whenever Alice has a number in her mind that she wants to convey to Bob, all she has to do is to send the element of $C$ corresponding to that number. But, for doing this, she has to send a string of length $n$. It is easy to see that integers between 1 and $|C|$ can be represented using roughly $\log |C|$ bits, and so if $|C| << 2^n$, then Alice and Bob are being wasteful - the "effective" amount of information being sent in the $n$ bits is $\log |C| << n$.

## 1.3 The main questions

The above discussion raises the following questions:

1. Given $n, d$, how large can $C \subseteq \{0,1\}^n$ be such that $C$ has minimum distance $d$?

2. Given $n, d$, how can one "efficiently" construct a code in $\{0,1\}^n$ with minimum distance $d$?

3. Given $n, d$, a code $C \subseteq \{0,1\}^n$ with minimum distance $\geq d$, and an $x \in \{0,1\}^n$ with the property that there is a (unique) $c \in C$ such that $x \in B(c, \frac{d-1}{2})$, how can one "efficiently" decode $x$, i.e. how does one efficiently find $c$?

For this lecture, we will mostly focus our attention on the first question.

# 2 Basic results

We begin by giving some basic existence and impossibility results about the number of codewords in a code with minimum distance $d$. We will always be interested in the asymptotics of $n \to \infty$. Particularly interesting choices of $d$ are (1) constant distance $d = O(1)$, and (2) constant relative distance $d = \delta n$.

## 2.1 The volume bound

Set $r = \lfloor \frac{d-1}{2} \rfloor$. It follows from Fact 7 that

$$|\bigcup_{c \in C} B(c, r)| = \sum_{c \in C} |B(c, r)| = |C| \cdot |B(r)|.$$

Also

$$\bigcup_{c \in C} B(c, r) \subseteq \{0,1\}^n$$

and so we have

$$|\bigcup_{c \in C} B(c, r)| \leq 2^n$$

which gives us

$$|C| \leq \frac{2^n}{|B(r)|}.$$

Thus,

$$|C| \leq \frac{2^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}}$$

This bound is known as the *volume bound* or the *Hamming bound*.

## 2.2 The greedy construction

**A greedy approach**: Consider the following greedy process:

- Set $S = \{0,1\}^n$ and let $C = \emptyset$.
- As long as $S \neq \emptyset$, pick a point $v \in S$ and add $v$ to $C$. Remove $B(v, d-1)$ from $S$, i.e. $S = S \setminus B(v, d-1)$.

Notice that we are removing at most $\sum_{i=0}^{d-1} \binom{n}{i}$ points from $S$ in every iteration. Since we add one element to $C$ in each iteration, we have that

$$|C| \geq \frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$$

Using the fact that $d = O(1)$, we get

$$|C| = \Omega(\frac{2^n}{n^{d-1}})$$

Note that this bound does not match the volume bound that we proved. It's conceivable that a modification of this approach might give better bounds though this has eluded everyone till date.

### 2.3 A probabilistic construction

It turns out that randomly chosen codes also achieve roughly the same parameters as the greedy construction. However, some care is needed in the random choice, as shown by the following failed attempt. See the homework and the next lecture for more on this.

**Naive probabilistic approach**: A naive probabilistic approach is to pick $K$ vectors, $x_1, \ldots, x_K$, in $\{0,1\}^n$ uniformly and independently, and set $C = \{x_1, \ldots, x_K\}$.

**Fact 8** (Birthday Paradox)**.** *If we pick $\Theta(\sqrt{N})$ numbers between 1 and $N$ uniformly and independently then, with probability $\geq 0.9$, at least two of them will be the same.*

Thus, if $K = \Theta(2^{n/2})$, with probability 0.99, it would be the case that $x_i = x_j$ for some $i \neq j$. In fact, it's not hard to see that, even if we don't worry about duplications, with high probability, two of the strings will end up being in the same ball of radius $d-1$, which would thwart our attempt.

**A probabilistic approach with alterations**: This approach involves picking strings uniformly and independently as above, followed by doing some modifications to the set in order to get the distance property. [See the first homework set]

## 3 Constant distance codes

Consider the case when $n$ is a growing parameter, and $d$ is a fixed constant, i.e. $d = O(1)$. In this case, the volume packing bound says that:

$$|C| \leq O\left(\frac{2^n}{n^{\lfloor (d-1)/2 \rfloor}}\right).$$

The greedy code construction produces a code $C$ with

$$|C| \geq \Omega\left(\frac{2^n}{n^{d-1}}\right).$$

These bounds are roughly in the same ball-park, but they are still significantly off. In a later lecture we will see that in fact the volume packing bound is tight: this is by a remarkable algebraic family of codes known as BCH codes.

## 4 The $d = 3$ case: the Hamming code

For $d = 3$, the volume bound becomes

$$|C| = O\left(\frac{2^n}{n}\right)$$

We will show a construction that gives a code with $|C| = \Theta(2^n/n)$, thus resolving our question for $d = 3$. This code is known as the *Hamming code*, and is due to Richard Hamming who also showed the volume bound.

We identify $\{0, 1\}$ with the field $\mathbb{F}_2$, and think of the code as a subset of $\mathbb{F}_2^n$. Let $\ell \in \mathbb{N}$ be a parameter, and $H$ be the $\ell \times (2^\ell - 1)$ matrix whose columns consist of all non-zero vectors in $\mathbb{F}_2^\ell$. We set $n = 2^\ell - 1$ and define our code as

$$C = \{c \in \mathbb{F}_2^n;\ Hc = 0\}$$

The reader can check that $C$ is a subspace of $\mathbb{F}_2^n$ whose dimension is $n - \ell$[1]. Thus, we have

$$|C| = \frac{2^n}{2^\ell} = \frac{2^n}{n+1} = \Theta\left(\frac{2^n}{n}\right)$$

All that remains to be shown is that the minimum distance of $C$ is 3. We will do this by showing that the minimum distance of $C$ cannot be 1 or 2. Before we show this, we will take a slight detour to talk about some properties of codes that are subspace in $\mathbb{F}_2^n$.

The following is an easy fact:

**Fact 9.** *Let $u, v, w \in \mathbb{F}_2^n$. Then $\Delta(u, v) = \Delta(u \oplus w, v \oplus w)$.*

If a code is a subspace of $\mathbb{F}_2^n$, it is called a *linear* code. We will study more properties of linear codes in the next lecture, but here is a fact that we will need for now:

**Fact 10.** *If $C$ is a linear code with minimum distance $d$ then $min_{c \in C} wt(c) = d$, where $wt(.)$ denotes the Hamming weight function.*

*Proof.* We will first show that $min_{c \in C} wt(c) \geq d$. Since $C$ has minimum distance $d$ there are codewords $c_1, c_2 \in C$ such that $\Delta(c_1, c_2) = d$. Using Fact 9, $\Delta(c_1 \oplus c_2, 0) = d$, which is the same as $wt(c_1 \oplus c_2) = d$. Since $C$ is a linear code $c_1 \oplus c_2 \in C$ and we have $min_{c \in C} wt(c) \geq d$.
We will now show the other direction. Suppose there is a $c \in C$ such that $wt(c) < d$. Since $C$ is a linear code and hence a subspace, we have that $0 \in C$. Then, $\Delta(c, 0) < d$, which would be a contradiction since the minimum distance of $C$ is $d$. □

We now get back to showing that the minimum distance of the Hamming code is 3. By Fact 10, this is the same as showing that there are no codewords of Hamming weight less than 3.
**Case i - there is a codeword $c$ such that $wt(c) = 1$:** From the definition of the Hamming code, we have $Hc = 0$. But this is not possible since $Hc$ is a column of $H$, and all the columns of $H$ are non-zero.

[1] This follows from the rank-nullity theorem over $\mathbb{F}_2$. The reader is urged to try and prove the theorem over $\mathbb{F}_2$ if he or she hasn't done so before.

**Case ii - there is a codeword $c$ such that $wt(c) = 2$:** Again, $Hc = 0$, and $Hc = v \oplus w$, where $v, w$ are distinct columns of $H$. This means $v \oplus w = 0$ or $v = w$, which is a contradiction.
This shows that the minimum distance of the Hamming code is at least 3.

Codes that match the volume bound are called *perfect* codes and the Hamming code is such a code for the $d = 3$ case. We now try and answer the third question in Section 1.3 for the Hamming code, i.e. given $x \in \{0,1\}^n$ such that $\Delta(x, c) \leq 1$ for some $c \in C$, how do we "efficiently" find $c$?

Typically, "efficiently" here means that the running time of our recovery algorithm should be polynomial in $n$, the input size. In the case of the Hamming code, polynomial time is trivial, since one can try all 1-bit modifications of $x$ and check if that resulting string is in the code. This gives a quadratic time decoding algorithm.

Below we give a linear time decoding algorithm.

1. Given $x$, compute $Hx$. This involves multiplying a $\Theta(\log n) \times n$ matrix with an $n \times 1$ vector, which can be done in time. $\Theta(n \log n)$.
2. If $Hx = 0$, then $x \in C$. Otherwise $x = c + e_i$ for some $c \in C$, where $e_i$ is the vector in $\mathbb{F}_2^n$ which is 0 everywhere except in the $i^{th}$ coordinate. Then $H(c + e_i) = Hc + He_i = He_i$.
3. Note that $He_i$ is the $i^{th}$ column of $H$. We now want to compute $i$ given $He_i$. This can be done by comparing $He_i$ with all the columns of $H$, and takes time $O(n \log n)$.
4. Once $i$ is determined, we can compute $c$ as $x + e_i$.

Thus, we have an algorithm to recover $c$ that takes time $\Theta(n \log n)$. This can be made $\Theta(n)$ by divide and conquer: exercise!

# 5 More on linear Codes

Recall that a code $C \subseteq \mathbb{F}_2^n$ is said to be linear if it is a subspace of $\mathbb{F}_2^n$. The definition of linear codes allows us to succinctly represent the code by selecting a basis $v_1, v_2, \ldots, v_k$ of $\mathcal{C}$ where $k \leq n$. It also allows a bijection between $\{0,1\}^k$ and $\mathcal{C}$ as follows: for a $\sigma \in \{0,1\}^k$, the corresponding codeword is $c(\sigma) = \sum_{i=1}^{k} \sigma_1 v_i \in \mathbb{F}_2^n$. To this extent, let $G$ be a matrix with $k$ rows and $n$ columns, where the $i^{th}$ row is $v_i^T$.

$$G = \begin{pmatrix} \text{———} & v_1^T & \text{———} \\ \text{———} & v_2^T & \text{———} \\ \vdots & \vdots & \vdots \\ \text{———} & v_k^T & \text{———} \end{pmatrix}$$

The matrix $G$ is called the *Generator Matrix* for $\mathcal{C}$. The map $E : \mathbb{F}_2^k \to \mathbb{F}_2^n$ given by $E(m) = mG$ is called the *Encoding Map*. Here, one normally thinks of $m$ as a message (given by a binary string) and $E(m)$ as the encoding of $m$.

$$\mathbb{F}_2^k \xrightarrow{Encoding} \mathbb{F}_2^n \xrightarrow{Corrupt} \mathbb{F}_2^n \xrightarrow{Decoding} \mathbb{F}_2^k$$

Since linear codes are nothing but vector spaces, we define a *Dual Code* $\mathcal{C}^\perp$ of a linear code $\mathcal{C} \subset \mathbb{F}_2^n$, analogous to dual spaces, as follows:

$$\mathcal{C}^\perp := \{y \in \mathbb{F}_2^n \mid \sum_{i=1}^{n} x_i y_i = 0 \text{ for all } x \in \mathcal{C}\}$$

**Observation 11.** $(\mathcal{C}^\perp)^\perp = \mathcal{C}$

**Observation 12.** *From the* rank-nullity theorem*, we have* $\dim(\mathcal{C}) + \dim(\mathcal{C}^\perp) = n$.

A *Parity Check Matrix* for a linear code $\mathcal{C} \subset \{0,1\}^n$ is an $(n-k)\times n$ matrix with rows $w_1^T, w_2^T, \ldots, w_{n-k}^T$, where $w_1, w_2, \ldots, w_{n-k}$ are the basis vectors for $\mathcal{C}^\perp$

$$H = \begin{pmatrix} \text{———} & w_1^T & \text{———} \\ \text{———} & w_2^T & \text{———} \\ \vdots & \vdots & \vdots \\ \text{———} & w_{n-k}^T & \text{———} \end{pmatrix}$$

**Observation 13.** $c \in \mathcal{C}$ *iff* $Hc = \mathbf{0}$

**Observation 14.** $GH^T = \mathbf{0}$

The above observation is the main property of the Parity Check Matrix.

**Proposition 15.** *The following are equivalent:*

1. $\mathcal{C}$ *has minimum distance at least* $d$
2. *Every nonzero element of* $C$ *has at least* $d$ *non zero entries*
3. *Every* $d-1$ *columns of* $H$ *are linearly independent.*

*Proof.* (*1* $\Rightarrow$ *2*): Since $0 \in \mathcal{C}$, we have that for every $x \in \mathcal{C}$, where $x \neq 0$, $\Delta(x, 0) \geq d$. Therefore, $x$ has at least $d$ nonzero entries.

(*2* $\Rightarrow$ *3*): Since $Hx = 0$ implies $x \in \mathcal{C}$, if the sum of some $t < d$ columns in $H$ was equal to 0, then we have a codeword that has $t < d$ nonzero entries, which is a contradiction.

(*3* $\Rightarrow$ *1*): Suppose there existed $x, y \in \mathcal{C}$ such that $\Delta(x, y) < d$, we have $Hx = Hy = 0$, which gives us $H(x + y) = 0$. But since $x + y$ had less than $d$ nonzero entries, the sum of $< d$ columns in $H$ equal to 0, which is a contradiction

□

## 5.1 Larger Alphabets

So far, we have thought of all of our messages, and their encodings as binary strings. However, in general, we need not be restricted by this condition. Let $\Sigma$ be any finite set of alphabet which

make up the content of the messages. We can extend the definition of Hamming distance between two strings $x$ and $y$ in this alphabet to be the number of points where they differ. We can also get a similar 'volume packing' bound as before. Let $\mathcal{C}$ be code over the alphabet $\Sigma$ that can correct at most $r$ errors. Let $\underset{\Sigma}{B_n}(r)$ be the set of strings in $\Sigma^n$, that differ in at most $r$ points from a given string. Then, we have:

$$|\mathcal{C}| \leq \frac{|\Sigma|^n}{\left|\underset{\Sigma}{B_n}(r)\right|}$$

We can also pick words greedily to get a lower bound on the number of code words with minimum distance $d$:

$$|\mathcal{C}| \geq \frac{|\Sigma|^n}{\left|\underset{\Sigma}{B_n}(d)\right|}$$

However, to talk about linear codes, we can only use certain sizes of alphabet, in particular, if $|\Sigma|$ is either a prime, or a prime power, we can identify $\Sigma$ with a field $\mathbb{F}_p$ or $\mathbb{F}_{p^l}$, and talk about linear codes as subspaces of $\mathbb{F}_p^n$ or $\mathbb{F}_{p^l}^n$.

We can define a Generator Matrix similar to the binary case as a $k \times n$ matrix $G$, where the rows are a basis of $\mathcal{C}$. We also define a dual code $\mathcal{C}^\perp$ as before, and a Parity Check Matrix as an $(n-k) \times n$ matrix whose rows are a basis of $\mathcal{C}^\perp$.

A few remarks:

1 Even if you only care about binary codes, the study codes over larger alphabet sizes is very important, since some of the best known constructions of binary codes are by using codes over larger alphabet sizes.

2 Imposing the condition of linearity on codes does not seem to come with any disadvantages. We do not know any considerable differences between codes and linear codes.

# 6 Erasure Correction

In the regime of Erasure Correction, a codeword is given by a string $x \in \{0,1\}^n$. However, instead of being flipped, some bits are replaced with the symbol '?'. These can be thought of as *erasures*, i.e., the bits in those location have been erased.

Our goal is to design a code that is resilient to at most $t$ erasures. In other words, if $\mathcal{C}$ is a $t-$erasure code, then for every $c \in \mathcal{C}$, and for every set of $t$ coordinates, if we erase these $t$ coordinates, then $c$ should not match with any other string in $\mathcal{C}$ in the remaining $n-t$ coordinates.

**Observation 16.** *A code $\mathcal{C}$ is $t-$erasure iff it has minimum distance at least $t+1$.*

**Observation 17.** *Hamming code is $2-$erasure.*

# 7 Constant relative distance

Now we study the regime where the distance of the code is $\delta n$.

The rate of a binary code $\mathcal{C}$, given by $\frac{\log_2 |\mathcal{C}|}{n}$, is a measure of how much information each coordinate contains (there are $\log_2 |\mathcal{C}|$ bits of information needed to specify a codeword in $\mathcal{C}$, and this information is then written in the form of $n$ bits). For a code over a general alphabet $\Sigma$, the rate is given by $\frac{\log_{|\Sigma|} |\mathcal{C}|}{n}$.

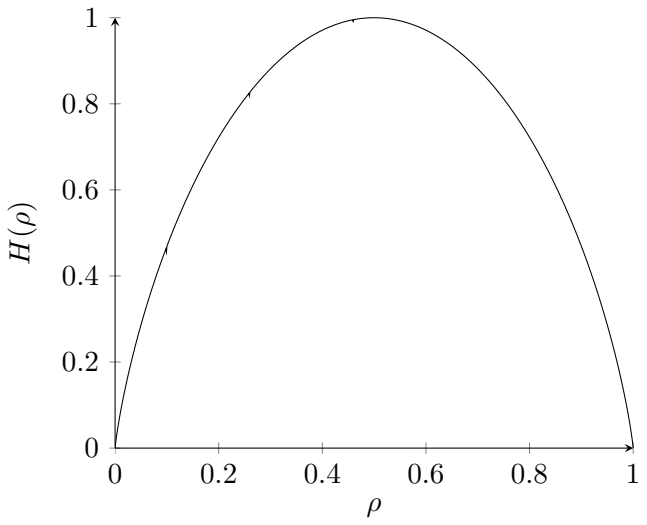
The *Relative Minimum Distance* (or simply relative distance) of a code $\mathcal{C} \subseteq \{0,1\}^n$ with minimum distance $d$ is given by $\frac{d}{n}$.

Before we study the relation between the rate and minimum distance of a code, we establish the following useful estimate for the volume of the Hamming ball. For any positive constant $\rho < \frac{1}{2}$, we have:

$$|B_n(\rho n)| = \Theta\left(\frac{2^{H(\rho)n}}{\sqrt{n}}\right)$$

where $H(\rho)$ is called the *Shannon Entropy* given by $H(\rho) = \rho \log_2 \frac{1}{\rho} + (1-\rho)\log_2 \frac{1}{1-\rho}$. *This intimate relation to the volumes of Hamming balls is the reason that the function $H$ shows up so much in the study of codes.*

*The following plot shows the relation between $\rho$ and $H(\rho)$. It is symmetric about $\rho = 1/2$, where it takes its maximum value, $H(1/2) = 1$, and is $0$ at $\rho = 0$, and $\rho = 1$.*



### 7.1 Upper Bound

*From the Volume Packing Bound, we have*

$$|\mathcal{C}| = O\left(\frac{2^n}{\frac{2^{H(\delta/2)n}}{\sqrt{n}}}\right)$$

*Taking logarithms on both sides and dividing by $n$, we have*

$$R \leq 1 - H\left(\frac{\delta}{2}\right) + o(1)$$

### 7.2 Lower Bound

*Using the greedy approach as before, we get*

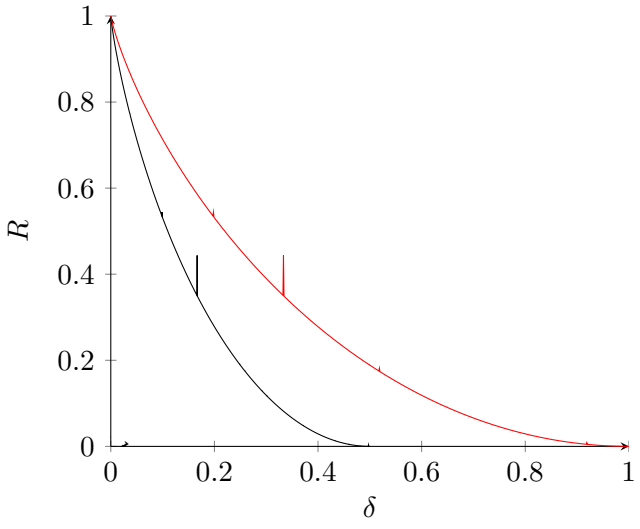
$$|\mathcal{C}| \geq \frac{2^n}{|B_n(\delta n - 1)|} = \Omega\left(\frac{2^n}{\frac{2^{H(\delta)n}}{\sqrt{n}}}\right)$$

*Taking logarithms on both sides and diving by $n$,*

$$R \geq 1 - H(\delta) + o(1)$$

### 7.3 The story so far..

*So far, we know that for a positive constant $\delta < \frac{1}{2}$, there cannot exist codes of rate more than $1 - H(\delta/2) + o(1)$ (Volume Packing Bound), and that there exist codes with rate at least $1 - H(\delta) + o(1)$. These are illustrated in the following plot.*



*The upper bound is not tight for every $\delta$. For instance, we know that there do not exist codes with nonzero rate with relative distance $3/4$ (Exercise!). In the next lecture, we will see that in fact, there are no codes with nonzero rate with relative distance $\geq 1/2$, and give a somewhat better overall upper bound.*