6.897 Algorithmic Introduction to Coding Theory	September 19, 2001
Lecture 4	
Lecturer: Madhu Sudan	*Scribe: Joe Aung*

Today's topics:

- Singleton bound and Maximum Distance Separable (MDS) codes.
- Reed Solomon codes.
- Reed Muller codes.
- Hadamard codes.

1 The Singleton Bound

Our first result is a simple lower bound on the block length of a codeword, given a fixed distance and message length. This bound is due to R. C. Singleton [4] and is hence named the Singleton bound. To motivate this result, recall that in lecture 2, we saw an $[n, n, 1]_2$ code and an $[n, n-1, 2]_2$ and wondered if we could generalize these results to a $[n, n-d+1, d]_2$ code (and the Hamming bound ruled this out for $d = 3$ in the binary case). A more elementary question is why should we only ask for $k \leq n-d+1$ and not better! The Singleton bound shows that this is indeed the best possible, over *any* alphabet.

Theorem 1 **([4])** *If C is an $(n, k, d)_q$ code then $d \leq n-k+1$.*

Proof Let Σ be the q-ary alphabet of C. Consider the projection map $\pi : \Sigma^n \to \Sigma^{k-1}$ that projects every word in Σ^n to its first $k-1$ coordinates. Since the range of π has only q^{k-1} elements and $|C| = q^k > q^{k-1}$, we see that there must exist two distinct codewords $\mathbf{x}, \mathbf{y} \in C$ such that $\pi(\mathbf{x}) = \pi(\mathbf{y})$. Since $\mathbf{x}$ and $\mathbf{y}$ agree on their first $k-1$ coordinates, it follows that they may differ on at most all remaining $n-(k-1)$ coordinates, and thus we have $\Delta(\mathbf{x}, \mathbf{y}) \leq n-k+1$. It follows that the minimum distance of C is at most $n-k+1$. ■

Codes that meet the Singleton bound, i.e., satisfy $k = n-d+1$, are called *Maximum Distance Separable (MDS)* codes. Last time we defined *Perfect Codes* as codes that meet the Hamming bound, and we said that the only perfect codes were the Hamming codes and two codes discovered by Golay. MDS codes and perfect codes are incomparable: i.e., there exist perfect codes that are not MDS and MDS codes that are not perfect. Each meets an incomparable optimality criterion. Today we will see a simple but large family of MDS codes, namely the Reed-Solomon codes.

2 Reed Solomon codes

Reed-Solomon codes were introduced in a paper by Reed and Solomon in 1959 [3]. They are based on properties of univariate polynomials and in particular the following property of univariate polynomials as introduced in the last lecture (see Lecture Notes on Algebra).

Fact 2 *Two distinct polynomials p_1, $p_2 \in F_q[x]$ of degree strictly less than k, agree in strictly less than k points in F_q. I.e., there exist at most $k-1$ points $\alpha \in F_q$ s.t. $p_1(alpha)$=$p_2(alpha)$.*

2.1 Construction of Reed-Solomon Codes

We describe the Reed-Solomon codes by giving the encoding function for them. Note that the encoding function is not unique. Our choice is made simply to ease the exposition.

Given a prime power q and $n \leq q$, and $k \leq n$, a Reed-Solomon code $\mathrm{RS}_{q,n,k}$ is constructed as follows:

1. Generate the field $\mathbb{F}_q$ explicitly (say via an irreducible polynomial over the underlying prime).

2. Pick n distinct elements $\alpha_1, \ldots, \alpha_n \in F_q$. Note this is where we need the property $n \leq q$.

3. To define the encoding, we first pick a convenient representation for the messages. Note that the message is k elements of $\mathbb{F}_q$, say $c_0, \ldots, c_{k-1}$. We let the message define the polynomial $C(x) \stackrel{\text{def}}{=} \sum_{j=0}^{k-1} c_j x^j$.

4. The encoding of the message $C(x)$ is its evaluation at $\alpha_1, \ldots, \alpha_n$, i.e., the sequence $\langle C(\alpha_1), \ldots, C(\alpha_n) \rangle$.

2.2 Parameters acheived by the code $\mathrm{RS}_{q,n,k}$

First we note the the Reed Solomon codes are linear.

Proposition 3 *The Reed-Solomon code* $\mathrm{RS}_{q,n,k}$ *is linear.*

Proof Suppose we are given two codewords $\langle C(\alpha_1), \ldots, C(\alpha_n) \rangle$ and $\langle D(\alpha_1), \ldots, D(\alpha_n) \rangle$ and suppose $\beta \in \mathbb{F}_q$. We need to show that the sequences $\langle C(\alpha_1)+D(\alpha_1), \ldots, C(\alpha_n)+D(\alpha_n) \rangle$ and $\langle \beta C(\alpha_1), \ldots, \beta C(\alpha_n) \rangle$ are also codewords. Note that the former sequence is the evaluations of the polynomial $(C+D)(x)$ at the points $\alpha_1, \ldots, \alpha_n$, while the latter is the evaluations of the polynomial βC at the same points. Further, note that if C, D are polynomials of degree at most $k-1$ then the polynomials $C+D$ and βC are also polynomials of degree at most $k-1$. Thus the resulting sequences are also codewords of the Reed-Solomon code. ■

It is obvious from the construction that the Reed-Solomon code $\mathrm{RS}_{q,n,k}$ has block length n and message length k. The only parameter that does not follow by definition is the distance, but that is easily argued.

Proposition 4 *The Reed-Solomon code* $\mathrm{RS}_{q,n,k}$ *has distance* $n-k+1$.

Proof By the Singleton bound we know that the distance $d \leq n-k+1$. So it suffices to prove $d \geq n-k+1$. Suppose we have two distinct polynomials $C(x)$ and $D(x)$ of degree at most $k-1$. Then by Fact 2 we have that $C(x)$ and $D(x)$ agree on at most $k-1$ points of $\mathbb{F}_q$ and hence disagree on at least $n-k+1$ points of the set $\{\alpha_1, \ldots, \alpha_n\}$. The distance follows. ■

As a result we get the following theorem.

Theorem 5 *For every prime power* q*, and every pair of positive integers* k, n *such that* $k \leq n \leq q$*, there exists an* $[n, k, n-k+1]_q$ *code.*

2.3 Applications of the Reed-Solomon codes

By playing games with the alphabet size, we've managed to construct codes that meet the Singleton bound. But a natural question to ask at this stage is: "How useful is it to have a code over large alphabets?".

To answer the question, we first invoke empirical evidence! Reed-Solomon codes are possibly the most commonly used codes in practical applications. In particular they are used to store information/music/video in compact discs (CDs) and digital video discs (DVDs), making them the most deployed family of codes! How do these technologies end up using codes over large alphabets? We describe the basic idea below. (Warning: The numbers used below are mostly for example. They are close to, but not exactly equal to the numbers used in practice.)

CDs and DVDs store information as a sequence of bits. The actual sequence of bits is quite long. The usual error-correcting methods break this long sequence into a collection of small chunks and encode each chunk separately. For example, we may pick each chunk to contain 240 bytes each (where one byte equals 8 bits). This gives a message sequence of 240 bytes where we now interpret each byte as an element of $\mathbb{F}_{256}$, the field on 256 elements. Using $n = q = 256$, one may encode this sequence using a Reed-Solomon code $\mathrm{RS}_{256,256,240}$ to get a sequence of 256 bytes which are then recorded on the CD. Thus in actuality we have described a *binary error-correcting code* of message length 240×8 bits, and block length 256×8 bits. What is the distance of this code? To analyze the distance, first let us recall that the underlying Reed-Solomon code over $\mathbb{F}_{256}$ has distance 17 — i.e., we must change at least 17 bytes of the encoded message to get an encoding of some other message. In turn this implies that we need to flip at least 17 bits in the binary encodings to get from one codeword to another. Abstracting this idea for arbitrary n, k and q, we get the following implication for binary codes.

Proposition 6 *For every $k \leq n \leq q$, where q is a prime power, there exists a family of $(n\lceil \log_2 q \rceil, k \log_2 q, n - k + 1)_2$ code.*

Exercise: Show that if $q = 2^l$, then the above code construction can be made linear.

How good is such a code? Written slightly differently, and throwing away some floors and ceilings, we see that the above amounts to codes of the form $(K + (1 + o(1))d \log K, K, d)_2$ codes. In contrast the Hamming bound says a code with message length K and distance d must have block length at least $K + (1 - o(1))\frac{d}{2} \log K$, for any fixed d and $K \to \infty$. So these codes based on Reed-Solomon codes are not too bad compared to the impossibility result. As we mentioned last time, better codes are known. In particular for the same block length and distance, BCH codes could encode 248 bytes of data. But analyzing those codes is somewhat harder, which is why we don't present them here.

Still the complexity of analyzing the distance of the code can not possibly a reason not to use them in practice. So do people prefer to use a weaker Reed-Solomon code as opposed to a potentially better BCH code? The main reason is the nature of the error. Typical errors on storage devices tend to happen in bursts. So when, say, 30 bits on the chunk are flipped it is quite likely that these 30 bit errors are not distributed uniformly over the chunk, but are localized to five or six bytes. In such as case, the Reed-Solomon code can actually correct all these errors (since it can correct upto 8 byte errors)! This enhanced performance of the Reed-Solomon codes in case of bursty error patterns is the main reason why it is so commonly used.

3 Codes based on Multivariate Polynomials

The major bottleneck with the Reed Solomon codes is the restriction $q \geq n$. In this section, we use minor algebraic extensions of such codes to get codes which work over smaller alphabets, including one non-trivial family of codes over the binary alphabet.

3.1 Bivariate polynomials

We start by generalizing the idea behind Reed Solomon codes in a simple way using bivariate polynomials instead of univariate polynomials. This will already give codes over an alphabet of size $q = \sqrt{n}$. We proceed as follows:

For prime power q and integer $l < q$ the bivariate polynomial code $B_{q,l}$ is defined as follows:

- Messages consist of $(l + 1)^2$ field elements which we view as an $(l + 1) \times (l + 1)$ matrix of coefficients $\langle m_{ij} \rangle_{i=0,j=0}^{l,l}$. We identify this message with the bivariate polynomial $M(x, y) = \sum_{i=0}^{l} \sum_{j=0}^{l} m_{ij} x^i y^j$.

- The encoding of a message corresponding to $M(x,y)$ is its evaluation at all field elements. Thus the encoding of M is $\langle M(\alpha,\beta)\rangle_{\alpha\in\mathbb{F}_q,\beta\in\mathbb{F}_q}$.

This gives us an $[n,k,d]_q$ code with $n=q^2$ and $k=(l+1)^2$. How much is the distance? It follows from Theorem 17 of Lecture 2.5 (Lecture notes on algebra) that its distance is $d=(q-l)^2$. In contrast, the Singleton bound allows $d=q^2-(l+1)^2$. The difference, approximately, $2l(q-l)$, is the price we pay for the smaller alphabet size.

3.2 Multivariate polynomial codes: Reed-Muller codes

We now extend the generalization of the previous section fully, to multivariate polynomials with an arbitrary number of variables, say m. These codes are termed Reed-Muller codes after their discoverers: These codes were discovered by D. E. Muller [1] and then I. S. Reed gave a decoding procedure for them [2]. The codes as described here are generalized to a range of parameters that were not covered originally, which seems to have focussed on codes over $\mathbb{F}_2$ only. (In particular, the way we describe them, these will be strict generalizations of Reed-Solomon codes, while Reed-Solomon codes were actually discovered much later!)

Here we will work with the notion of the total degree of a polynomial. Recall this is the maximum, over all monomials with non-zero coefficients, of the total degree of the monomial, where the total degree of a monomial is the sum of the degrees of all variables in it. E.g. the total degree of the monomial $x^3y^4z^3$ is 10, and the total degree of the polynomial $3x^9+4y^8+2x^3y^4z^3$ is also 10. Recall Theorem 18 from Lecture 2.5 shows that a polynomial of total degree l is zero on at most l/q fraction of the inputs — we will use this fact below.

We will start by presenting a computer scientist's view of Reed-Muller codes, which only consider polynomials of degree $l<q$.

Reed-Muller Codes - Case 1. For positive integers m,l and prime power q with $l<q$, the Reed-Muller code $\mathrm{RM}_{m,l,q}$ is defined as follows:

- The message is a sequence of coefficients $\langle m_{i_1,\dots,i_m}\rangle_{i_1+\cdots+i_m\le l}$. The message represents the polynomial
$$M(x_1,\dots,x_m)=\sum_{i_1+\cdots+i_m\le l} m_{i_1,\dots,i_m}x_1^{i_1}\dots x_m^{i_m}.$$
- The encoding of a polynomial $M(\mathbf{x})$ is the sequence $\langle M(\alpha)\rangle_{\alpha\in\mathbb{F}_q^m}$.

It is obvious that the block length of the code $\mathrm{RM}_{m,l,q}$ is $n=q^m$. The message length equals the number of m-long sequences of non-negative integers that sum to at most l, and this number turns out to be $\binom{m+l}{m}$. Finally, from Theorem 18 of Lecture 2.5, the distance of the code is at least (actually exactly) $(1-\frac{l}{q})n$. We will summarize the properties of the code shortly, but before doing so, let us consider a choice of parameters which is somewhat illustrative of the powers of this code.

Sample setting of parameters: Suppose we wish to encode k elements of some alphabet. It seems reasonable to ask for codes of length $n=\mathrm{poly}(k)$ that have large minimum distance (say $n/2$) with as small an alphabet as possible. It turns out Reed-Muller codes can give such codes with alphabet size $\mathrm{poly}(\log k)$, by the following setting of parameters: We choose $m=\frac{\log k}{\log\log k}$ and $q=\log^2 k$ and l such that $\binom{m+l}{l}=k$. For this choice of parameters, we note that the code $\mathrm{RM}_{m,l,q}$ has block length $n=q^m=k^2$ which was one of our goals. To estimate the distance, note the $\binom{m+l}{l}\ge(l/m)^m$. Thus we have $l\le mk^{1/m}=m\log k=\log^2 k/\log\log k=o(q)$ as $k\to infty$. Thus this family of codes has distance $(1-o(1))\cdot n$.

Reed-Muller codes - Case 2. Now we consider the case where the total degree $l > q$. In such case, we associate messages with polynomials of total degree at most l and individual degree at most $q-1$ in every variable. Let $S(m,l,q) = \{(i_1,\ldots,i_m) | \sum_j i_j \leq l, 0 \leq i_j < q\}$ and let $K(m,l,q) = |S(m,l,q)|$. The Reed-Muller codes $\mathrm{RM}_{m,l,q}$ are described as follows:

- The messages are a sequence of $K(m,l,q)$ elements of $\mathbb{F}_q$ denoted $\langle m_{\mathbf{i}} \rangle_{\mathbf{i} \in S(m,l,q)}$. This message is associated with the polynomial

$$M(x_1,\ldots,x_m) = \sum_{\mathbf{i} \in S(m,l,q)} m_{\mathbf{i}} x_1^{i_1} \cdots x_m^{i_m}.$$

- The encoding of the message is its evaluation at all points $\alpha \in \mathbb{F}_q^m$, i.e., the sequence $\langle M(\alpha) \rangle_{\alpha \in \mathbb{F}_q^m}$.

This yields a code of block length q^m and message length $K(m,l,q)$. To estimate the distance of the code write $l = a(q-1)+b$ with $b < q-1$. Then by Theorem 19 of the Lecture 2.5, the distance of this code is at least $q^{m-a}(1-b/q)$. Again the setting of parameters may be somewhat confusing. So we give an example setting of parameters to illustrate the power of this code:

Sample setting of parameters: (This is the original setting of the parameters in the papers of Reed and Muller.) We let $q = 2$ and pick $l < m$. Then $K(m,l,2) = \mathrm{Vol}(l,m)$ (that's right! - the volume of the Hamming ball in $\{0,1\}^m$ of radius l.) We may lower bound this quantity by $\binom{m}{l}$. The distance of the code is 2^{m-l}. Thus the $\mathrm{RM}_{m,l,2}$ code gives a $[2^m, \binom{m}{l}, 2^{m-l}]_2$ code!

4 Hadamard codes

Before concluding today's lecture we give one more example of codes, which again turn out to be special cases of Reed-Muller codes. (The presentation here is different from the way we did it in lecture.)

Jacques Hadamard was interested in some constructions of self-orthogonal matrices with all entries being $+1$ or -1. The ensuing constructions lead to nice error-correcting codes and we describe this connection here.

Definition 7 *An $n \times n$ matrix $\mathbf{H} = \{\mathbf{h}_{ij}\}$ is a* Hadamard matrix *if $\mathbf{h}_{ij} \in \{+1,-1\}$ for all i,j and $\mathbf{H}\mathbf{H}^T = n\mathbf{I}$, where $\mathbf{I}$ is the $n \times n$ identity matrix, and all arithmetic is regular integer arithmetic.*

Viewed appropriately, the rows of an $n \times n$ Hadamard matrix give a binary code of block length n, with n codewords (i.e., a message length of $\log n$). To get this view, note that every row of H is just a binary vector (where the binary alphabet just happens to be $\{+1,-1\}$ rather than the usual $\{0,1\}$). Thus clearly the rows form a binary code (of message length $\log n$ and block length n). The most interesting aspect of this code is the distance. The fact that $\mathbf{H}\mathbf{H}^T = n\mathbf{I}$ is equivalent to saying that the codewords are at distance exactly $n/2$ from each other! To see this, note that the (i,j)th entry of $\mathbf{H}\mathbf{H}^T$ is $\sum_{k=1}^n \mathbf{h}_{ik}\mathbf{h}_{jk}$. For any k, the quantity $\mathbf{h}_{ik}\mathbf{h}_{jk}$ is either $+1$ or -1, with -1 indicating $h_{ik} \neq h_{jk}$ and $+1$ indicating $h_{ik} = h_{jk}$. For $i \neq j$, we have $\sum_{k=1}^n \mathbf{h}_{ik}\mathbf{h}_{jk} = 0$, and this implies that exactly $n/2$ of the summands are $+1$ and exactly -1 of the summands are $+1$. In turn this yields that for that half the coordinates k, $\mathbf{h}_{ik} = \mathbf{h}_{jk}$ and so the codewords corresponding to the ith and jth rows agree on exactly $n/2$ coordinates.

There are several ways to augment this obvious code and all of these are interchangeably referred to as Hadamard codes. For our purpose, we will fix the following codes to be Hadamard codes based on an $n \times n$ Hadamard matrix $\mathbf{H}$.

Definition 8 *Given an $n \times n$ Hadamard matrix $\mathbf{H}$, the Hadamard code of block length n, Had_n, is the binary code whose codewords are the rows of $\mathbf{H}$ (with $+1$s replaced by 0s and -1s replaced by 1s), and the complements of the rows of $\mathbf{H}$.*

Proposition 9 *For every n such that an $n \times n$ Hadamard matrix exists, the Hadamard code* Had_n *is a* $(n, \log(2n), \frac{n}{2})_2$*-code.*

Proof The block length and message length follow by definition. We have also argued that two codewords that correspond to distinct rows of the Hadamard matrix differ in $n/2$ places. Now if $\mathbf{c}$ is a codeword corresponding to a row of the matrix and $\mathbf{c}'$ is a codeword corresponding to the complement of a row of the matrix, then if the corresponding rows are the same, then the codewords differ everywhere; and if the corresponding rows are different then the codewords disagree whenever the corresponding rows agree, but this also happens exactly $n/2$ times. The proposition follows. ■

We now point to one family of codes with the above parameters that we actually know of! Note that if we take the Reed-Muller code $\text{RM}_{m,1,2}$ with m variables and total degree 1 over a binary alphabet, then we get a $[2^m, m+1, 2^{m-1}]_2$ code, which is of the form $[n, \log(2n), n/2]_2$. So it is worthwhile asking if this is a Hadamard code, i.e., is there an underlying Hadamard matrix.

It turns out that the $\text{RM}_{m,1,2}$ codes do come from an underlying Hadamard matrix. To do so recall that the messages of the Reed-Muller code correspond to coefficients $c_0, \ldots, c_m$ representing the polynomial $C(\mathbf{x}) = c_0 + \sum_{i=1}^{m} c_i x_i$. Now if we consider only those codewords corresponding to $c_0 = 0$, then we get a collection of codewords that differ in exactly half the places, and using them as the rows yields the Hadamard matrix.

As an aside note that the usual construction of Hadamard matrices with n being a power of 2 is inductive, with $\mathbf{H}_m$, the $2^m \times 2^m$ Hadamard matrix being defined as:

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_{m+1} = \begin{bmatrix} \mathbf{H}_m & \mathbf{H}_m \\ \mathbf{H}_m & -\mathbf{H}_m \end{bmatrix}.$$

5 Summary

We've seen a number of different codes with incomparable merits. The Reed-Solomon codes have optimal distance to message length behaviour but require large alphabets. Hadamard codes work over a binary alphabet, but have very poor relationship between message length and block length. Hamming codes have a good relationship between message and block length, but only offer a distance of 3. In going forward we will look for families of codes which maintain a constant ratio between message length and block length, while also maintaining a constant ratio between distance and block length.

References

[1] D. E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

[2] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

[3] Irving S. Reed and Gustav Solomon. Polynomial codes over certain finite fields. *J. SIAM*, 8:300–304, 1960.

[4] Richard C. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10:116–118, April 1964.