Research in Designs & Codes

Kimball Martin

April 27, 2009

The idea of coding theory

Coding theory is the theory of communicating accurately over noisy channels. It lies in the intersection of mathematics, computer science and electrical engineering. Coding theory is used in many diverse applications such as in satellite transmission and CD recording. Let me illustrate the basic ideas with the following example. Suppose you're sending emails back and forth with your friend and you receive:
DO YOU WNT TO SEE A OVIE THIS WEEKND?
So you notice there are typos in his email. This is called *error-detection*. You are able to do this because English has a certain structure to it. If he was just typing a random string of characters (say a password or scientific data), you would have no idea if there were any errors (until you tried it). There are two ways to understand his message. First you could ask him for confirmation, i.e., you could ask
DID YOU ASK IF I WANTED TO SEE A MOVIE THIS WEEKEND?, or
WHAT DID YOU SAY?
Of course in this example, if you did that, he'd think you were stupid. But still, there is such a protocol for *error-detecting codes*. These are codes with which you can detect errors but not correct them. The simplest example is the "parity check code," where the sum of all the digits in each word is even. If you detect an error in the data you receive, you simply ask the other party to resend it. In some applications, this may be all you want to do—e.g., to make sure a credit card number or license number is valid. Of course, this could be quite inconvenient if you're telling a satellite something and by the time you hear it say "WHAT?" it's on the other side of the earth, so you need to wait until tomorrow to transmit again. (Perhaps this is a bit of an exaggeration, but you get the point.) It would be much better to be able to correct the errors in real-time. Error-detecting codes are well enough known about, so most of the current research focuses on "error-correcting" codes, as we will do.

In our example, this means you would simply interpret his email to be:
DO YOU WANT TO SEE A MOVIE THIS WEEKEND?
Here we simply correct the apparent three errors by common sense. Of course, there's a small possibility that we've made a mistake. For example, there could have been four errors and the original message was
DO YOU WANT TO SEE A BOVINE THIS WEEKEND?
But this is highly unlikely, though grammatically correct. So in error-correcting codes you correct errors you notice into the most probable intended word. Thus you want to find a code with high probability of proper correction for your noisy channel.

We will be concerned with the following general model. First we will be interested in *binary (block) codes* of length n. This means you have a set of codewords C which are strings of length n of 0's and 1's. The idea is to take the original data, encode it into codewords with some encoding algorithm, transmit your encoded data, receive it, correct errors and decode it. We will assume the only possible errors that can occur are that some bits might get flipped across the noisy channel. This is sufficient for nearly all applications I know of.

Let me illustrate the process the repetition code of length 3. Say you want to send the following nibble (string of 4 bits): 1010. To encode, you repeat each bit 3 times to get: 111 000 111 000. You send this over the noisy channel and say you receive: 110 100 111 001. To decode, you just take the majority vote on each set of 3 bits, and get 1010 back. Notice there were three errors total and two of them in a row. Fortunately they were on different codewords. If we had two errors on the same codeword, say we received: 100 000 111 010, then we would incorrectly decode this as 0010. So the repetition code of length 3 can detect up to 2 errors per word but can (properly) correct only 1 error per word. Now you can define a repetition code of any odd length $2k+1$ which will detect up to $2k$ errors and correct up to k errors per word.

Generally you choose your code based on what kind of noise you expect in your channel: What percentage of bits will be flipped? Is the noise sparse or well spread out, or does it come in bursts? Often errors come in bursts, i.e., 7 out of 10 bits in a row get flipped, but then out of the next 120 bits, only 2 or 3 get flipped. So you wouldn't want

a code of length 10, you might want one of length 128 that could correct up to 15 errors per word to be safe. Now you could get by with a repitition code of length 31, but if you have a lot of data to send then your code isn't very efficient–for every 31 bits you receive, you only get one bit of data. We can do much better with a little work.

Linear codes

We'll only consider *linear* codes. The theory of linear codes is very rich and there's still a lot of research left to do in it. In fact, one of the most interesting aspects of coding theory in my opinion is that, about 10 years ago, even some of the non-linear codes with good properties (Kerdock, Preparata) were shown to be linear over $\mathbb{Z}/4$.

Definition 1 *Let $\mathbb{F}_2 = \mathbb{Z}/2 = \{0,1\}$ and V be the vector space $\mathbb{F}_2^n$. For any $v \in V$, define its* weight $w(v)$ *to be the number of coordinates which are 1. Let C be a linear subspace of V of dimension k. Then we say C is a* (linear) code *of* length n *and* dimension k *and the elements of C are* codewords. *The* minimum distance d *of C is $d = \min\{w(c-c') : c, c' \in C, c \neq c'\} = \min\{w(c) : 0 \neq c \in C\}$. We also say C is an (n,k,d)-code.*

So $|V| = 2^n$ and $|C| = 2^k$. Then you encode data words of length k as elements as codewords of length n. For example, let $V = \mathbb{F}_2^3$ and $C = \{(0,0,0),(1,1,1)\}$, the repetition code of length 3. This is a [3,1,3] code. Notice you can detect up to $d-1 = 2$ errors per word and correctly correct up to $(d-1)/2 = 1$. In general you can detect up to $d-1$ and correct up to $(d-1)/2$ errors per word.

Why do we look at linear codes? There are easy constructions. It is easy to compute the (worst case) error-correction capability $(d-1)/2$ and the *information rate* k/n (i.e., the data length efficiency). It is easy to encode and decode, and to detect and correct errors. Most importantly, they turn out to give some very good codes in practice! However, it's not known if there are classes of linear codes which are *asymptotically* good. At the end of the last section, I said we might want a code with $n = 128, d = 31$. The best linear code there is with desired n and d is a (128, 40, 31)-code (see *http://www.win.tue.nl/~aeb/voorlincod.html*). Then encoding will only multiply the size of the data by 3.2, instead of 31 as in the repitition code. There is also a well known so-called second-order Reed-Muller code with parameters (128, 29, 32), which will only increase the data size by a factor of about 4.4.

Example 1 *The Hamming code, the smallest interesting example. Let*

$$G = \begin{pmatrix} 1&1&0&1&0&0&0\\ 0&1&1&0&1&0&0\\ 0&0&1&1&0&1&0\\ 0&0&0&1&1&0&1\\ 1&0&0&0&1&1&0\\ 0&1&0&0&0&1&1\\ 1&0&1&0&0&0&1 \end{pmatrix}, \quad H = \begin{pmatrix} 1&0&0\\ 1&1&0\\ 1&1&1\\ 0&1&1\\ 1&0&1\\ 0&1&0\\ 0&0&1 \end{pmatrix}.$$

The (7,4,3) Hamming code C is the rowspace of G. It's also the kernel of H. This means that $c \in C$ iff $Hc = 0$. Let G' be the matrix G with the bottom three rows deleted. To encode a 4-bit piece of data x, multiply by G' to get the 7-bit codeword $G'x$. Say someone sends you $G'x$ and you receive $c = G'x + e$ where e is the "error" vector. Suppose $w(e) \leq 1$. If $Hc = H(G'x+e) = He = 0$, there's no error. Error-correction and decoding can be done with a little more linear algebra—I'll leave you to think about that later.

Now where does this come example from? It's the incidence matrix of the Fano plane *(the* projective plane *of order 2). It's also the incidence matrix of the* quadratic residue difference set *in $\{1,2,4\}$ in $\mathbb{Z}/7$. G is also a (truncated) table of the* autocorrelation sequence $x_{t+3} = x_{t+1} + x_t + 1$.

Designs

The last example generalizes in various ways. Projective planes (and more generally, finite geometries) give rise to codes. Difference sets (from quadratic residues and beyond) give rise to codes also, and these notions don't coincide in general (we'll look at when they do when we get to difference sets). However, both of these generalizations fit into the context of designs. I'll only introduce Symmetric Balanced Incomplete Block Designs (SBIBDs).

Definition 2 *Let $\mathcal{P}$ be a set of v* points *and $\mathcal{B}$ a set of v* blocks *B, where each B is a set of k points in $\mathcal{P}$. Suppose any 2 points lie in exactly λ blocks. Further suppose each point lies in exactly k blocks (symmetry). Then $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ is an (v,k,λ)-*SBIBD*. The* incidence matrix *for $\mathcal{D}$ is the 0-1 matrix with rows labeled by $\mathcal{P}$ and column labeled by $\mathcal{B}$ such that an entry $(P,B) = 1$ iff $P \in B$.*

You can think of a SBIBD as a geometry where the points are points and the blocks are lines, each made up of k points and there are k lines through each point. If $\lambda = 1$, then two points determine a line, so we get a plane. For example, the Fano plane (projective plane of order 2) is a (7,3,1)-SBIBD. A design $\mathcal{D}$ with incidence matrix A gives rise to the code which is the rowspace of A (over $\mathbb{F}_2$).

The *order* of a (v, k, λ)-SBIBD is $n = k - \lambda$. Codes from designs are only interesting if the design is of even order. (Don't confuse the n and k of the design with the n and k of the code—they're not the same, but the notation's standard.) A *Hadamard design* is a $(4n-1, 2n-1, n-1)$ or $(4n-1, 2n, n)$ SBIBD. All SBIBDs satisfy $4n - 1 \leq v \leq n^2 + n + 1$. Projective planes attain the upper bound. Hadamard designs satisfy the lower one.

Planes

Definition 3 *An* affine plane of order n *is a set of* n^2 *points* $\mathcal{P}$ *and* $n^2 + n$ *lines* $\mathcal{B}$ *such that*
(i) 2 points determine a line
(ii) each line contains n *points*
(iii) each point lies in $n + 1$ *lines*
(iv) 2 lines intersect in at most one point.

The "at most" part in condition (iv) makes us give up some symmetry in the definition by allowing for parallel lines (which at times makes them harder to work with). We can remedy this however. Given an affine plane $(\mathcal{P}, \mathcal{B})$, we can form the corresponding *projective plane of order* n by adding one "point at infinity" for each of the $n + 1$ classes of parallel lines. Add this point at infinity onto each of these parallel lines. Then join all the points at infinity with one "line at infinity". This gives a system $(\mathcal{P}', \mathcal{B}')$ where $\mathcal{P}'$ has $n^2 + n + 1$ points and $n^2 + n + 1$ lines such that
(i') 2 points determine a line
(ii') each line contains $n + 1$ points
(iii') each point lies in $n + 1$ lines
(iv') *any* two lines intersect in exactly one point.
This is the general definition of a projective plane of order n. An equivalent definition is that a projective plane of order n is an $(n^2 + n + 1, n + 1, 1)$-SBIBD.

Note that in addition to the extra symmetry from (iv'), there is now a duality between points an lines: for instance there are the same number of points as lines, and (ii') now mirrors (iii'). Just as any affine plane yields a projective plane, removing one line and all the points on it from a projective plane of order n yields an affine plane of order n.

In general, not all (either affine or projective) planes of order n are isomorphic. However by the symmetry of the projective plane, any two affine planes which come from a given projective plane must be isomorphic. Hence, an affine plane of order n exists if and only if a projective plane of order n exists, and the number of affine planes of order n (up to isomorphism) is the same as the number of such projective planes of order n.

Facts about projective planes of order n
1. They exist if $n = q$ is a prime power.
2. There's more than one if $q = p^s > 8, s > 1$. The number of planes $\to \infty$ as $p \to \infty$.
3. (Bruck-Ryser Theorem) If $n \equiv 1, 2 \pmod 4$ and the squarefree part of n is divisible by a prime $p \equiv 3 \pmod 4$, then there is no projective plane of order n.
4. One exists iff there are $n - 1$ *mutually orthogonal Latin squares of order* n.
5. There is no **known** plane of non-prime power order.
6. Order 10 was ruled out by coding theory + computer (Lam, '88).
7. There is only one plane of order n for $0 \leq n \leq 5$ (easy to show), $n = 7$ (not so easy) and $n = 8$ (computer proof).
8. There are 4 planes of order 9 (constructed in early 20-th century; proved no more than 4 by computer).

Question 1 *Does there exist a projective plane of order 12? More generally, of any non-prime-power order?*

This is the smallest order not known, for Fact 3 says there's no plane of order $2p$ if $p \equiv 3 \pmod 4$, e.g. $n = 6, 14, 22$, etc. It's a very interesting problem. We can't just apply the program for the order 10 case on super-fast computers. The basic idea for $n = 10$ is: the corresponding code of such a plane can be proven to have so many codewords of certain weights. Then you need to determine conditions on configurations for planes could possibly result in this, and search for them with a computer. Lam started on this problem in 1980 and his final search ran for 2 or 3 years (intermittantly) on a CRAY at IDA. For order 12, you'd need to start by proving some properties of such a plane.

It's conjectured that planes only exist for prime-power orders. This is due to the fact that it's easy to construct them for prime-power orders, but no one knows how to for non-prime-power orders. Also the Bruck-Ryser theorem, rules out infinitely man non-prime power orders. The next most outstanding question in this subject is:

Question 2 *Is there more than one projective plane of order 11? Of any prime order?*

Difference sets

A *cyclic* (v,k,λ) *difference set* is a set D of k integers (mod v) such that any non-zero x can be written as the difference of 2 elements of D in exactly λ ways. A cyclic difference set gives rise to a circulant incidence matrix for a (v,k,λ)-SBIBD. The i-th row will correspond to the i-th translate $D+i$. See the Hamming code example. You can define difference sets more generally by, instead of working with $\mathbb{Z}/v$, working with an arbitrary group of order v, and you get (v,k,λ)-SBIBDs. More precisely,

Definition 4 *Let G be a (multiplicative) group of order v and D a subset of k elements. If any non-zero $g \in G$ can be written as $g = xy^{-1}$ exactly λ ways with $x, y \in D$, then D is a (v,k,λ) difference set.*

Example 2 *Let p be a prime. For $p = 4n+3$, the nonzero quadratic residues mod p form a $(4n+3, 2n+1, n)$ difference set. Let x be odd. For $p = 4x^2+1$, the nonzero fourth powers mod p form a difference set. For $p = 4x^2+9$, the fourthpowers mod p form a difference set. If $p+2$ is prime, then $\{(a,0)\} \cup \{(a,b) : a, b$ both (non)squares$\}$ is a difference set in $\mathbb{Z}/p \times \mathbb{Z}/(p+2)$.*

A (v,k,λ) difference set is also a (v,k,λ) design. It is easy to see that for a (v,k,λ) difference set we have $k^2 - k = (v-1)\lambda$. When $\lambda = 1$, $(v,k,\lambda = (n^2+n+1, n+1, 1)$ where $n = k-1$, i.e., a *planar* ($\lambda = 1$) difference set is a projective plane. Planes obtained this way give ideal network designs. Planar difference sets exist for any prime-power order. In fact, any *Desarguesian* plane comes from a difference set.

Question 3 *Which projective planes can be constructed from difference sets?*

Question 4 *For which groups, or which parameters even, do difference sets exist?*

Hadamard Matrices

Definition 5 *An $n \times n$ ± 1-matrix H is called* Hadamard *(of order n) if $H^T H = nI$.*

Hadamard matrices are also used in other subjects such as radar, communications, spectrometry, image processing, pattern recognition and genetic algorithms. They also make pretty patterns if you color them black and white squares instead of ± 1's. Examples:

$$[1]\ ,\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\ ,\ \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}\ ,\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Hadamard matrices of order n can only exist if $n = 1$, $n = 2$ or $n = 4m$. There are two famous and still outstanding conjectures about Hadamard matrices. First we have the

Hadamard conjecture There is a Hadamard matrix of each order $n = 4m$.

Now notice the third matrix in the example above is *circulant*, i.e., each row is obtained by the previous by a cyclic shift to the right. This is very special! It brings us to the second conjecture:

Hadamard circulant conjecture If $n > 1$, then there's only one Hadamard circulant (up to equivalence), the 4×4 above.

What's known:
–Hadamard matrices exist for all 2^e (Sylvester) and $2^e(p^m+1)$, p odd prime (Paley)
–The tensor (Kronecker) product of Hadamard matrices is Hadamard (so m, n exist $\implies mn$ exists)
–The smallest case not known is 668 (the last case, 428, was done just this year!)
–Many construction of (infinite families) of Hadamard matrices are known.

–All Hadamard circulants have order $4n^2$ (or 1).
–A Hadamard circulant of order $4n^2$ implies the existence of a cyclic $(4n^2, 2n^2-n, n^2-n)$ difference set
–There's no Hadamard circulant of order $4 < n \leq 10^{11}$ except possibly if $n = 4u^2$, $u = 165, 11715, 82005$

Let H be a Hadamard matrix of order $4n$. By multiplying row and columns by -1's we may assume the first row and column are all 1's. Then deleting the first row and column and changing the ± 1's to 0's and 1's gives the

incidence matrix for a *Hadamard design*, i.e., a $(4n-1, 2n-1, n-1)$ or $(4n-1, 2n, n)$ design. This process is reversible. *Hadamard difference sets*, i.e., those with Hadamard design parameters or parameters $(4n^2, 2n^2 \pm n, n^2 \pm n)$, also give rise to Hadamard matrices.

Question 5 *Which groups have Hadamard difference sets?*

If $|G| = 2^{t+2}$ it is conjectured that G has a Hadamard difference set iff G/K is not cyclic or dihedral for any normal subgroup K of index greater than 4. This is known if G is abelian. If $|G| = 2^{2t+2}$ and G is abelian, there is a Hadamard difference set iff $\exp G \leq 2^{t+2}$.

Question 6 *Classify the cyclic* $(4n-1, 2n-1, n-1)$ *Hadamard difference sets.*

Several non-existence results are known. The only known cases are when $v = 4n-1$ is (non-Mersenne) prime (quadratic/sextic residues), v is a product of twin primes—Jacobi symbols, and $v = 2^m - 1$ (Singer, GMW + several new constructions). It's been conjectured that these are only possible v's.

This has been a very active field recently due to several recent conjectured constructions with some proofs. Many things are still very mysterious. For example, if $(k, 2^m - 1) = 1$ then the (non-zero) image of the "additive" 2-to-1 map $x + x^k$ on $\mathbb{F}_{2^m}$ gives a difference set in the multiplicative group $\mathbb{F}_{2^m}^*$. The interplay between the additive and multiplicative groups of $\mathbb{F}_{2^m}$ seems quite deep and is really not understood.