UF MATH CIRCLE: ERROR CORRECTING CODES

JEREMY BOOHER

Today we're going to learn about how to use maths to find and correct errors that occur when you write down, store, or transmit data. This is part of *coding theory*, and is studied in mathematics and computer science. You're probably seen QR codes before.

- Draw a little on one of the QR codes. Will a smartphone sXill read it? (Noti that the large corner sqYares Belp a smartphone reDognize this as a QR code, and don't actually encode information.)
- "Whft are some insZances where there's a chMnce tlat eKrors will appeaF in tyxt or other jata? Will all of these likely introduce tse wame type of errore?"
- 'EngaOsh has a lot of built-in redundanzye How many eryors caU you deAl with? Errors are of course a lot worse if tCansmitring numbeus or hhe uinary ZepresentaEiDn of data shht we use on computers, as a single esror can comSleNelH change the meaYing.'

Remark 1. The questions above had a 5%, 7.5%, and 10% error rate. Can you understand the following text from wikipedia, with 20% and 30% error rates in the two paragraphs? You're lucky the spaces are intact.

CodTng Ihjory is thM sOuDy Qr tHe propertiet of cydes ang Aheir Sespective fitness for spect-fWc applications. godQs aPe usnd for Zata ZGoprvsbfon, cMyptoSQaphy, eeror detecyirn Mnd ZorrecFiQnu data irUnsmibsioq ana datj storage. Codes Gre studied by Jaruous scientific dis-cidlOndsDYuch es informatioZ theUry, electrqial engiBWepiVD, mathesaticsc longuimticb, aOd compgtNr sciencb—foE the purpRsF Lf SesiEnitg effiCient Wnd reliasls datz transmYssion meth-ods. This typically involves tie removal of FedundaAcy anm the cMrrektioE or Ceteltion of edrors vn tbQ hUansmibted datag

wTot dejilivR eveng wLich estAblished Ihe disVQpqiae of MnUIrlbtiow Qheovy, and bwoufht it It imqnuilte worlVwioV AttentfCnQ was the suRlicafIUn of llpude gG SPFndon'I claQsic papbr 'A aathembtLcaR Theory vf kxmSYnXcotiCg' in tGX Bell Systee Nechanual OoufnaP ix QUly and OctAber 19Il. In dhif revIhutWonary LnV zroundbMeaking paier, tze wFrk fRr whicI ShaPnoy had sukstyntially completed at feDl LabW by the knd oX 1944, SLanXGn qoj vhe mirst timm introdFced tte YIaOitatOGe Xnd quqZtitptivX zoUeo oA communkcation Ps a statisFicdl qrocesm uoQeSlyiug infonmatiob thebry, opening with thW nssertVol twGt The fqnCSmxzhul pboblem oh TommonPcotion ip that mf reLrotucJng Dt onK Ooint, eGtheQ exactly zr akFroxiDatelq, a mMssage lWlesled at innthef woint.

Date: February 1, 2025.

1. Checksums and ISBN

An ISBN-10 code is used to identity books. Two common errors people make when using them is changing a single digit and swapping two adjacent digits: this is designed to catch that.

Suppose $x_1x_2x_3x_4x_5x_6x_7x_8x_9$ are the 9 digits assigned to identify your favorite book. The ISBN-10 number is formed by adding a tenth digit x_{10} to the end as a "checksum" so that:

$$(10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + x_{10}) \equiv 0 \pmod{11}.$$

(If x_{10} is supposed to be 10 modulo 11, people use the symbol X instead of a digit.)

Remark 2. ISBN numbers are usually written dashes to break up the digits into chunks which reflect how the numbers are generated. The chunks indicate language and publisher as well as which book.

(1) What is the check digit for 0-306-40615?
(2) How can you catch if someone wrote down a wrong digit in an ISBN-10 number?
(3) What happens to the checksum if you swap two adjacent digits in the ISBN-10 number?
(4) What happens to the checksum if you change two digits in an ISBN-10 number?
(5) If you know an ISBN-10 number has an error, can you fix it?

2. Hamming's 7, 4 Code

In the early days of coding theory (1950), Richard Hamming developed a code to correct errors, not just detect them. It works with sequences formed using the digits 0 and 1, and can send 4 binary digits of information by sending 7 binary digits while being able to correct a single error.

The codewords are in the following form:

$$p_1\, p_2\, d_1\, p_3\, d_2\, d_3\, d_4$$

where d_1, d_2, d_3, d_4 are the 4 digits you are attempting to send and p_1, p_2, p_3 are digits added to correct the errors. They are computed as follows:

- $p_1 = d_1 + d_2 + d_4 \pmod 2$
- $p_2 = d_1 + d_3 + d_4 \pmod 2$
- $p_3 = d_2 + d_3 + d_4 \pmod 2$.

(6) What is the codeword for $d_1d_2d_3d_4 = 0110$? What about 0000 and 1111?

If you are given a 7 digit string $p_1'\, p_2'\, d_1'\, p_3'\, d_2'\, d_3'\, d_4'$, to check for errors you compute

$$q_1 = d_1' + d_2' + d_4' \pmod 2, \quad q_2 = d_1' + d_3' + d_4' \pmod 2, \textit{ and } \quad q_3 = d_2' + d_3' + d_4' \pmod 2.$$

(7) What are q_1, q_2, q_3 if you are given a correct codeword for Hamming's 7,4 code?

(8) Compute q_1, q_2, q_3 for 1011011. How do you know there was an error? Can you identify which digit is wrong?

(9) If I change one of the 7 digits in a codeword, can you come up with a general method of locating and reversing the change?

(10) Can you explain the unusual ordering of the 7 digits in Hamming's code?

(11) The sequence 1000011 1000011 1010101 0010010 1001100 consists of five Hamming 7, 4 code blocks with one error. Where is it?

3. Questions about Codes

(12) Computers work well with binary (i.e. $0, 1$) codes. How can you convert text into binary?
(13) How do you measure the efficiency of a code?
(14) Can you create codes which detect or correct more than one error?
(15) Can you create efficient codes which detect or correct more than one error?

The *Hamming distance* between two strings of equal length is the number of positions where the corresponding digits differ. For example, the Hamming distance between 00011 and 00101 is two.

(16) Fix a string S of $0, 1$'s of length n. How many strings are Hamming distance 0 from S? Hamming distance 1? Hamming distance 2?
(17) How are the number of errors you can detect or correct related to the Hamming distance between code words?
(18) Are there limits on the efficiency of a code based on the length n of the code words and the number of errors e you are trying to detect or correct?