An investigation into the mathematical theory behind check-digits for the purpose of error-detection*

Ng Weng Leong
Raffles Junior College

§1. Introduction

A check-digit is a mathematical device that attempts to detect errors in a numerical string. It is arrived at by a calculation based on the digits of the string. A transcription error in a string gets detected when a calculation based on the transcribed string does not give the check-digit. Many cataloguing systems, like the ISBN (International Standard Book Number), make use of check-digits for error-detection [3]. The errors may be due to faulty transcription or an outsider trying to break into a system by inventing a password.

The basic concept that enables a check-digit to function is the classification of all the relevant strings into residue classes modulo some integer m. This concept is expressed in the following theorem [1].

Theorem 1: *The m residue classes* $[0], [1], [2], \ldots, [m-1]$ *are disjoint and their union is the set of all integers.*

Theoretically, when a numerical string and its erroneous variant fall into different residue classes, the check-digit will be able to detect the error, otherwise, the error simply goes undetected.

The mathematical procedure used to calculate a check-digit is significant in that it is like a mechanical process that assigns to all numerical

---

* Paper presented to the Science Research Congress (November, 1989) as part of the 1989 Science Research Programme organised jointly by National University of Singapore and Ministry of Education; supervisor Professor A J Berrick.

strings their respective residue classes. Each different mathematical procedure produces a different type of check-digit, and each type of check-digit is useful for detecting certain types of error.

The aim of this project is to investigate the mathematical mechanism that enables check-digits to function, and to devise specific mechanisms for detecting errors. The three types of errors under investigation are single-digit, transposition and 3-cycle errors. The principal method used is the Multiplier method, and variations of this method are investigated.

§2. Digit-sum method

Procedure to find check-digit for a k-digit string : All the digits in the string are added up and reduced ($mod\ m$) to c such that $0 \leq c < m$. Then c is the required check-digit. For example we have an 8-digit string 23843521. Summing up and reducing (mod 10) to c, i.e. $c \equiv 28\ (mod\ 10), 0 \leq c < m$, we have $c = 8$. So the check-digit for 23843521 is 8, and it is usually attached to the end of the string like this : 23843521-8.

Detection of single-digit errors

Definition: Single-digit errors occur when one digit in a numerical string is copied wrongly.

k-digit string,	S	:	$b_1, b_2, \ldots, b_r, \ldots, b_k$
erroneous string,	S'	:	$b_1, b_2, \ldots, b'_r, \ldots, b_k$

String S' is the transcription error of S, with b'_r being listed in place of b_r. Suppose we use the Digit-Sum method and the single-digit error does not get detected, then the check-digits calculated from S and S' must be equal, hence we have

$$b_1 + b_2 + b_3 + \cdots + b_r + \cdots b_k \equiv b_1 + b_2 + b_3 + \cdots + b'_r + \cdots + b_k \quad (mod\ m)$$

$$b_r \equiv b'_r, \quad (mod\ m) \ldots (1)$$

To ensure that every single.-digit error gets detected, we need

$$b_r \not\equiv b'_r, \quad (mod\ m) \ldots (2)$$

From here, we derive conditions on m such that (2) is satisfied.

Theorem 2: *If* $a \equiv b \ (mod\ m)$ *and* $0 \leq |b - a| < m$ *then* $a = b$.

From Theorem 2 we obtain

Proposition (A): *Every single-digit error gets detected when* $m > 9$.

Transposition errors

Definition : Transposition errors occur when two digits within a string switch positions. (This commonly occurs as a result of transcription).

Clearly, transposition errors do not alter the sum of digits of a string. This means that the same check-digit will be obtained by the Digit-Sum method despite the error. It follows that the Digit-Sum method cannot detect transposition errors.

§3. The Multiplier Method

Procedure :
Multipliers : $a_1, a_2, \ldots, a_k$
k-digit string : $b_1, b_2, \ldots, b_k$

As shown above, we have a k-digit string with multipliers attached to each digit, so a_i corresponds to b_i for $i = 1, 2, \ldots, k$. The idea then is to take the product $a_i b_i$ and sum over every i. i.e. $a_1 b_1 + a_2 b_2 + \ldots + a_i b_i + \ldots + a_k b_k = s$, and then reduce s $(mod\ m)$ to obtain the check-digit c i.e. $c \equiv s \ (mod\ m)$ $0 \leq c < m$.

(It can be observed that the Digit-Sum method is a special case of the Multiplier method when $a_1 = a_2 = a_3 = \cdots = a_k = 1$.)

Detection of single-digit errors

Multipliers : $a_1, a_2, a_3, \ldots, a_r, \ldots, a_k$
k-digit string, S : $b_1, b_2, b_3, \ldots, b_r, \ldots, b_k$
Erroneous string, S' : $b_1, b_2, b_3, \ldots, b'_r, \ldots, b_k$
(b'_r was transcribed in place of b_r)

Using the Multiplier procedure, suppose that the error is not detected, then

$$a_1 b_1 + \cdots + a_r b_r + \cdots + a_k b_k \equiv a_1 b_1 + \cdots + a_r b'_r + \cdots + a_k b_k \pmod{m}$$

$$a_r b_r \equiv a_r b'_r \pmod{m} \ldots (3)$$

To detect all single-digit errors we need

$$a_r b_r \not\equiv a_r b'_r \pmod{m} \ldots (4)$$

Theorem 3: *The Cancellation Law. If* $ac \equiv bc \pmod{m}$ *and if* $(m, c) = d$ *i.e.* d *is the greatest common divisor of* m *and* c*, then* $a \equiv b \pmod{(m/d)}$.

Corollary: *If* $(m, c) = 1$*, i.e.* m *and* c *are relatively prime, then* $ac \equiv bc \pmod{m}$ *implies* $a \equiv b \pmod{m}$. *By combining this corollary with Theorem 2 we obtain*

Proposition (B): *If* $(a_r, m) = 1$ *and* $m > 9$ *then every single-digit error gets detected.*

There are obviously many ways to choose m and a_r such that proposition (B) is satisfied. But certainly one of the easiest ways is to have conditions

1) $a_n = n$ for $n = 1, 2, \ldots, k$
2) m a prime greater than 9 and k
1) and 2) together imply $(a_n, m) = 1$

Detection of Transposition errors

Multipliers : $a_1, a_2, \ldots, a'_r, \ldots, a_r, \ldots, a_k$
k-digit string, S : $b_1, b_2, \ldots, b'_r, \ldots, b_r, \ldots, b_k$
erroneous string, S' : $b_1, b_2, \ldots, b_r, \ldots, b'_r, \ldots, b_k$

Here a transposition error has occurred, with b'_r and b_r being transposed. Assuming that the error is not detected, then

$$a_1 b_1 + a_2 b_2 + \cdots + a'_r b'_r + \cdots + a_r b_r + \cdots + a_k b_k$$
$$\equiv a_1 b_1 + a_2 b_2 + \cdots + a'_r b_r + \cdots + a_r b'_r + \cdots + a_k b_k \pmod{m}$$
$$a'_r b'_r + a_r b_r \equiv a'_r b_r + a_r b'_r \pmod{m}$$
$$(b'_r - b_r) a'_r \equiv (b'_r - b_r) a_r \pmod{m} \ldots (5)$$

To detect all transposition errors we need

$$(b'_r - b_r)a'_r \not\equiv (b'_r - b_r)a_r \qquad (mod\ m) \ldots (6)$$

Proposition (C): *Suppose the greatest possible absolute difference between any two multipliers is D. All transposition errors are detected provided $(m, (b'_r - b_r)) = 1$, $m > D$, and $a'_r \neq a_r$.*

Proof: When $(m, (b'_r - b_r)) = 1$, then by the corollary of Theorem 3, (5) implies that $a'_r \equiv a_r (mod\ m) \cdots (7)$. By Theorem 2, $m > D$ and (7) imply $a'_r = a_r$ (a contradiction). So (6) must hold whenever a transposition error occurs. □

Note that the conditions of Proposition (C) hold whenever conditions 1) and 2) above hold. In fact, this is precisely the method used for the ISBN check-digit, where $k = 9$ and $m = 11$, see [3].

§4. Variations of the Multiplier Method

In the original method, the multipliers were directly multiplied to the corresponding digits. But useful variations can be made by taking the product of the multipliers against some function of the corresponding digits.

Multipliers	:	a_1,	a_2,	$\ldots$,	a_k
k-digit string	:	b_1,	b_2,	$\ldots$,	b_k
f(digits)	:	$f(b_1)$,	$f(b_2)$,	$\ldots$,	$f(b_k)$

Detection of 3-cycle errors

Definitions: (a) A 3-cycle is a permutation of three elements in a sequence such that none of them remains in its original position. (b) A 3-cycle error occurs when a 3-cycle is performed on three distinct digits in a string.

S = original string, S' = erroneously transcribed string

Multipliers	:	a_1,	a_2,	$\dots$,	a_i,	$\dots$,	a_j,	$\dots$,	a_r,	$\dots$,	a_k
f(digits)	S :	$f(b_1)$,	$f(b_2)$,	$\dots f(b_i)$,	$\dots$,	$f(b_j)$,	$\dots f(b_r)$,	$\dots f(b_k)$
f(digits)	S' :	$f(b_1)$,	$f(b_2)$,	$\dots f(b_j)$,	$\dots f(b_r)$,	$\dots f(b_i)$,	$\dots f(b_k)$

Suppose that the error goes undetected, then we have

$$a_1 f(b_1) + a_2 f(b_2) + \cdots + a_i f(b_i) + \cdots + a_j f(b_j) + \cdots + a_r f(b_r) + \cdots + a_k f(b_k)$$
$$\equiv a_1 f(b_1) + a_2 f(b_2) + \cdots + a_i f(b_j) + \cdots + a_j f(b_r) + \cdots + a_r f(b_i) + \cdots + a_k f(b_k) \pmod m$$

$$a_i f(b_i) + a_j f(b_j) + a_r f(b_r) \equiv a_i f(b_j) + a_j f(b_r) + a_r f(b_i) \pmod m$$

$$(a_i - a_r) f(b_i) \equiv (a_i - a_j) f(b_j) + (a_j - a_r) f(b_r) \pmod m \dots (8)$$

Combination of two check-digits

We use the functions $y = x^2$ and $y = x$ to obtain two different check-digits, which are then combined to form one single check-digit.
From (8), we obtain for $y = x^2$ and $y = x$ respectively

$$(a_i - a_r) b_i^2 \equiv (a_i - a_j) b_j^2 + (a_j - a_r) b_r^2 \pmod m \dots (9)$$
$$(a_i - a_r) b_i \equiv (a_i - a_j) b_j + (a_j - a_r) b_r \pmod m \dots (10)$$

Theorem 4: *Let m be a prime greater than 9 and D, where D is the maximum absolute difference between any two multipliers. If (9) and (10) are simultaneously satisfied then $b_i = b_r = b_j$.*

Proof: Squaring (10),

$$(a_i - a_r)^2 b_i^2$$
$$\equiv (a_i - a_j)^2 b_j^2 + (a_j - a_r)^2 b_r^2 + 2(a_i - a_j)(a_j - a_r) b_j b_r \pmod m \dots (11)$$

Multiplying (9) by $(a_i - a_r)$ and subtracting (11)

$$[(a_i - a_j)^2 - (a_i - a_r)(a_i - a_j)] b_j^2 + [(a_j - a_r)^2 - (a_i - a_r)(a_j - a_r)] b_r^2$$
$$\equiv 2(a_i - a_j)(a_r - a_j) b_r b_j \pmod m$$

$$(a_i - a_j)(a_r - a_j) b_j^2 + (a_i - a_j)(a_r - a_j) b_r^2 \equiv 2(a_i - a_j)(a_r - a_j) b_r b_j \pmod m$$

But $[(a_i - a_j)(a_r - a_j), m] = 1$, so by Theorem 3,

$$b_j^2 + b_r^2 \equiv 2b_r b_j \qquad (mod\ m)$$
$$(b_j - b_r)^2 \equiv 0 \qquad (mod\ m)$$

And since m is prime, $b_j - b_r \equiv 0\ (mod\ m)$

$$b_j \equiv b_r \qquad (mod\ m) \ldots (12)$$

By Theorem 2 we have $b_j = b_r$

Substitute $b_j = b_r$ into (10),

$$(a_i - a_r)b_i \equiv (a_i - a_r)b_r \qquad (mod\ m)$$

$$b_i \equiv b_r \qquad (mod\ m)$$

So $b_j = b_r = b_i$. □

If (9) and (10) are satisfied simultaneously, then Theorem 4 states that $b_j = b_r = b_i$, which implies that no error has occurred, and Proposition (D) thus follows.

Proposition (D): *When a 3-cycle error occurs, at least one of (9) and (10) cannot hold.*

It follows from proposition (D) that if the two check-digits produced by the two digit-functions are combined, then the resulting check-digit can detect all 3-cycle errors. For each digit-function, there are m possible check-digits, therefore the number of distinct combined check-digits is m^2. Since proposition (D)will be satisfied when the conditions for detection of single-digit and transposition errors are met, therefore all three types of error can be detected. For example, if 1) and 2) hold, then all three types of error are detected, and m^2 check-digits are required. In the case of strings of length 9, working with (*mod* 11) (as for ISBN) gives 121 possible check-digits.

Another method to detect 3-cycle errors

Definition: Consecutive errors are 3-cycle errors that occur on 3 digits which are consecutive in position.

In practice, consecutive errors are those 3-cycle errors which occur most frequently. Here we develop another method to detect consecutive errors. Firstly, we assign distinct values to each of the digits from 0 to 9. Then we use these values in place of the original digits to calculate the check-digit. This is a generalization of the first method in that the function used is based on the following general definition:

Definition: If A and B are sets, a function $f : A \to B$ is a rule which assigns to each element a in A an element $f(a)$ in B.

Assuming that a consecutive error is not detected when the multiplier method is applied and assuming that $a_n = n$ for $n = 1, 2, 3, \ldots, k$ then we have, by (8),

$$2f(b_i) \equiv f(b_j) + f(b_r) \qquad (mod\ m)$$

We need

$$2f(b_i) \not\equiv f(b_j) + f(b_r) \qquad (mod\ m) \quad (13)$$

for all i, j, r

The problem then is to select a set of 10 integers $(mod\ m)$ (one integer for each digit) such that no three integers are in arithmetic progression. Such a set can be found by trial and error, like the following:

$$\{0, 1, 3, 4, 9, 10, 12, 13, 27, 28\} \qquad (mod\ 43)$$

Therefore (13) is satisfied if the above set is used.

This strengthens our previous result, for this method gives a check-digit for strings of length of up to 42 digits, detecting all single-digit, transposition and consecutive errors, and using just 43 possible characters for the check-digit. These characters could possibly come from the upper and lower case characters of the Roman letter set.

This method leads to the following problem:

Problem: *Given k, find the least m such that, for some set $b_1, b_2, \ldots, b_k$ of distinct residues (mod m), (13) holds.*

The above problem is analogous to the famous unsolved problem of planar difference sets [2]. In a planar difference set, it is required that all possible absolute differences between two elements be distinct (*mod* m).

Acknowledgements

The author of this paper would like to thank the following people who have in some way or other helped to make this project possible. Many thanks to Professor A. J. Berrick for his patient guidance throughout the project; Choo Meng Kiam for encouraging me to join the programme in the first place; fellow participants, and organizers of the SRP.

References

[1] Tom M. Apostol, *Introduction to Analytic Number Theory*, Springer Verlag (1980).

[2] Ryser, Combinatorial Mathematics, *The Carus Mathematical Monographs (1963).*

[3] Publishers in the United Kingdom and their addresses, *J Whitaker and Sons (London, 1986).*