

QR コードの多色化

誤り訂正能力が保証される条件

伊達 虎太郎

数学応用数理専攻 楯研究室 修士 2 年

2024 年 2 月 2 日

はじめに

- QR コードは汚れなどで一部が認識できなくても正しく読み取れる。
- その仕組みはリード・ソロモン符号 (以下 RS 符号) による。
- RS 符号は任意の有限体上で扱うことができる。
- QR コードは白と黒の 2 色なので標数 2 の体上で扱われている。

はじめに

- QR コードは汚れなどで一部が認識できなくても正しく読み取れる。
- その仕組みはリード・ソロモン符号 (以下 RS 符号) による。
- RS 符号は任意の有限体上で扱うことができる。
- QR コードは白と黒の 2 色なので標数 2 の体上で扱われている。
- この研究の目的は、QR コードを多色化する (標数 3 や 5 の有限体上で考える) ことによる恩恵がどのような条件下ならばあるのかを確かめることである。
- 体の標数を 2 のままにして QR コードを多色化する研究 [12] は存在する。
- (株)SHIFT によってカメレオンコードという多色化された 2 次元バーコードが開発・運用されている。



カメレオンコード (引用元:[10])

符号化と復号

符号化

$q = p^e$ ($e \geq 1$) とする。

長さ k の情報を $\mathbb{F}_q^k$ の元としたとき、 $\mathbb{F}_q^k$ の元の符号化とは $n > k$ に対し

$$E : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$$

である。また、 $C = E(\mathbb{F}_q^k) \subset \mathbb{F}_q^n$ を符号といい、 C の元を符号語という。

復号

$D : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^k$ で

$$D \circ E = \text{id}_{\mathbb{F}_q^k}$$

となるものを復号という。

最小距離

ハミング距離

$x, y \in \mathbb{F}_q^n$ に対して

$$d(x, y) = \#\{i \mid x_i \neq y_i, 1 \leq i \leq n\}$$

で定義される $d(x, y)$ をハミング距離という。

最小距離

ハミング距離

$x, y \in \mathbb{F}_q^n$ に対して

$$d(x, y) = \#\{i \mid x_i \neq y_i, 1 \leq i \leq n\}$$

で定義される $d(x, y)$ をハミング距離という。

最小距離

符号 C の最小距離 d を

$$d = \min\{d(x, y) \mid x \neq y \in C\}$$

で定義する。

最小距離が $d \geq 2t + 1$ ならば $y = c + e$ ($c \in C, d(e, 0) \leq t$) のとき、 y に最も近い符号語は c ただ一つとなる。なので $D(y) = E^{-1}(c)$ とする。

$\mathbb{F}_q^n$ と $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$ のベクトル空間としての同型を

$$(c_0, c_1, \dots, c_{n-1}) \longleftrightarrow c_0 + c_1\bar{x} + \dots + c_{n-1}\bar{x}^{n-1}$$

の対応によって同一視する。

RS 符号

$\alpha : \mathbb{F}_q - \{0\}$ の乗法における生成元

$n = q - 1$, $g = (x - \alpha) \cdots (x - \alpha^{n-k}) \in \mathbb{F}_q[x]$ としたとき、

$$C := \langle g \rangle / \langle x^n - 1 \rangle \subset \mathbb{F}_q[x] / \langle x^n - 1 \rangle \simeq \mathbb{F}_q^n$$

で定義される符号 C を (n, k, d) RS 符号という。

RS 符号

RS 符号

$\alpha : \mathbb{F}_q - \{0\}$ の乗法における生成元

$n = q - 1$, $g = (x - \alpha) \cdots (x - \alpha^{n-k}) \in \mathbb{F}_q[x]$ としたとき、

$$C := \langle g \rangle / \langle x^n - 1 \rangle \subset \mathbb{F}_q[x] / \langle x^n - 1 \rangle \simeq \mathbb{F}_q^n$$

で定義される符号 C を (n, k, d) RS 符号という。

例えば $\mathbb{F}_{256}$ 上の RS 符号は必ず $n = 255$ となってしまう。

短縮 RS 符号

整数 $\mu < k$ について

$$C' = \{c_0 + c_1 \bar{x} + \cdots + c_{n-\mu-1} \bar{x}^{n-\mu-1} \in C\}$$

で定義される $C' \subset C$ を $(n - \mu, k - \mu, d)$ 短縮 RS 符号という。

QR コードの規格と構造 (JIS-X-0510)



1 型



2 型



3 型

- 型番は 1 から 40 まで存在。
- 型番ごとにデータを入れられるセル数は決まっている。
- 例えば 1 型であればデータを入れられるセル数は $21^2 - 7^2 \times 3 - 45 - 41 = 208$ である。通常の QR コードの場合 8 セルで 1 つのデータを表すのでこの場合語長 n は $208/8 = 26$ となる。

QRコードの規格と構造 (JIS-X-0510)



1 型



2 型



3 型

- 型番は 1 から 40 まで存在。
- 型番ごとにデータを入れられるセル数は決まっている。
- 例えば 1 型であればデータを入れられるセル数は $21^2 - 7^2 \times 3 - 45 - 41 = 208$ である。通常の QR コードの場合 8 セルで 1 つのデータを表すのでこの場合語長 n は $208/8 = 26$ となる。

誤り訂正レベル	L	M	Q	H
復元可能な損傷割合 (%)	7	15	25	30
n (語長)	26	26	26	26
k (情報量)	19	16	13	9

実験の流れ

- STEP1** 語長 n 、情報量 k 、色の数 $\text{color}(= p)$ 、1つのデータを表すセル数 $\text{cell}(= e)$ 、1つのデータが誤って伝わる確率 prob を入力。
- STEP2** ランダムに符号語 c と誤り e (汚れや光の当たり方などによって発生) を生成して受信語を $y = c + e$ とする。
- STEP3** リード・ソロモン符号の復号アルゴリズムに従い復号を行う。
(誤りの個数が t より大きいと復号に失敗する。)
- STEP4** STEP2,3 を 10000 回繰り返し、何回復号に成功したかを記録する。

プログラムによる実験例

singular 上での計算

```
> int n = 26;  
> int k = 19;  
> int color = 2;  
> int cell = 8;  
> int prob = 10;  
> <"RS.txt";
```

⋮

```
success 7658  
failure 2342
```

この実験では 10000 回中復号に 7658 回成功して 2342 回失敗した。

実験用のプログラム (RS.txt)

```
int countsuccess = 0;
int countfailure = 0;
int countloop = 1;

if((n-k)%2 == 1){
    n = n-1;
}
ring A = (color^cell,a),x,(lp,c);
int t = (n-k) div 2;
int i = 1;
poly g = 1;
while(i<=n-k){
    g = g*(x-a^(i));
    i++;
}

while(countloop<=10000){
    poly c0 = 0;
    int i = n-1;
    while(i>=n-k){
        c0 = c0+a^(random(0,color^cell-2))*x^(i);
        i--;
    }
    poly c = c0-reduce(c0,g);
    int i = 0;
    while(i<=n-1){
        if(random(1,100)<=prob){
            e = e+a^(random(0,color^cell-2))*x^(i);
        }
        i++;
    }
}
```

```
poly y = c+e;

poly S = 0;
int i = 1;
while(i<=n-k){
    S = S+subst(y,x,a^(i))*x^(i-1);
    i++;
}
vector v1 = [S,1];
vector v2 = [x^(2*t),0];
module K = v1,v2;
module M = groebner(K);
int i = 1;
while(1){
    if(M[i][1]<M[i][2]){
        int m = i;
        break;
    }
    i++;
}
poly r = subst(M[m][2],x,0);
if(r==0){
    r = 1;
}
poly M1 = M[m][1]/r;
poly M2 = M[m][2]/r;

int i = 0;
int j = 0;
vector er = (0);
while(i<=(color^cell-2){
```

実験用のプログラム (RS.txt)

```
if(subst(M2,x,a^(i))==0){
    j++;
    er = er+(1/a^(i))*gen(j);
}
i++;
}
int i = 1;
while(i<=j){
    poly P(i) = 1;
    int j' = 1;
    while(j'<=j){
        if(j' != i){
            P(i) = P(i)*(1-er[j']/er[i]);
        }
        j'++;
    }
    poly e(i) = subst(M1,x,(1/er[i]))/(er[i]*P(i));
    i++;
}
int i = 1;
while(i<=j){
    int j' = 0;
    while(j'<=(color^cell)-2){
        if(a^(j') == er[i]){
            int num(i) = j';
        }
        j'++;
    }
    i++;
}
```

```
poly E = 0;
int i = 1;
while(i<=j){
    E = E+e(i)*x^(num(i));
    i++;
}
if(y-E == c){
    countsuccess++;
}
else{
    countfailure++;
}

countloop++;
}

printf("success %p", countsuccess);
printf("failure %p", countfailure);
```

- 2 色のときは 8 セルで 1 つのデータを表していた。
- つまり $2^8 = 256$ 通りのデータを表すことができる。
- 256 通りのデータを表すために、以下のように cell の数を $\text{color}^{\text{cell}} \geq 256$ となる最小の整数に変更する。

color	2	3	5	7
cell	8	6	4	3
$\text{color}^{\text{cell}} (= q)$	256	729	625	343

実験 1

誤り訂正レベルを M(15 %訂正可能) とする。このとき色の数ごとの型番と送ることのできる k の最大値は下の表のようになる。

色の数	1 型	2 型	3 型	4 型	5 型
2	16	28	44	64	86
3	21	37	58	85	114
5	32	56	88	128	172
7	42	74	117	170	229

$k = 70$ のデータを送る場合を考える。

実験 1

プログラムで実際に実験した結果が以下の表である。

n	k	color	cell	color ^{cell}	prob = 5	10	20	時間 [s]
134	86	2	8	256	10000	10000	9982	144
134	85	3	6	729	10000	10000	9987	331
141	88	5	4	625	10000	10000	9994	322
119	74	7	3	343	10000	10000	9981	161

- prob が一定で n と k がほぼ同じなのであればほぼ同じ誤り訂正能力のまま小さい QR コードにできることがわかった。(理由は後程説明)
- 計算時間には最大で約 2.3 倍ほどの差が出る。(体のサイズ $q = \text{color}^{\text{cell}}$ に依存する。)

$p := \text{prob}$ とすると復号誤り確率は

$P_{\text{fail}} = 1 - \sum_{j=0}^t \binom{n}{j} p^j (1-p)^{n-j} = \sum_{j=t+1}^n \binom{n}{j} p^j (1-p)^{n-j}$ である。

s : 誤りの個数, $p \ll t/n = (n-k)/2n =: T$ (定数) とする。

n を十分大きいとみなすと誤りの割合 $s/n \sim N(p, \frac{p(1-p)}{n})$ なので標準化すると

$$Z := \frac{\frac{s}{n} - p}{\sqrt{\frac{p(1-p)}{n}}} \sim N(0, 1) \text{ (正規標準分布)}$$

$p := \text{prob}$ とすると復号誤り確率は

$P_{\text{fail}} = 1 - \sum_{j=0}^t \binom{n}{j} p^j (1-p)^{n-j} = \sum_{j=t+1}^n \binom{n}{j} p^j (1-p)^{n-j}$ である。

s : 誤りの個数, $p \ll t/n = (n-k)/2n =: T$ (定数) とする。

n を十分大きいとみなすと誤りの割合 $s/n \sim N(p, \frac{p(1-p)}{n})$ なので標準化すると

$$Z := \frac{\frac{s}{n} - p}{\sqrt{\frac{p(1-p)}{n}}} \sim N(0, 1) \text{ (正規標準分布)}$$

よって

$$\begin{aligned} (T =) t/n < s/n &\Leftrightarrow \frac{T - p}{\sqrt{\frac{p(1-p)}{n}}} < Z \\ &\Leftrightarrow \frac{\sqrt{n}(T - p)}{\sqrt{p(1-p)}} < Z \end{aligned}$$

つまり、 k/n が一定のとき n が大きいほうが復号誤り確率は低くなる。

誤り訂正レベルが同じ $\implies k/n$ はほぼ一定
なので誤り訂正レベルを固定したとき特に

- 型が大きい方が復号成功確率が高い。
- color が大きい方が復号成功確率が高い。

ということが言える。

実験 2

色の数を 2 色に固定して 3 型、5 型、7 型で、各誤り訂正レベルの 10000 回中 10000 回復号に成功した最大の prob (以下 $Q_{\text{color, 型, 誤り訂正レベル}}$ とする) をまとめる。

誤り訂正レベル	L	M	Q	H
最大の prob(3 型)	2	4	10	14
最大の prob(5 型)	3	8	14	18
最大の prob(7 型)	3	9	16	22

実験 2

色の数を 2 色に固定して 3 型、5 型、7 型で、各誤り訂正レベルの 10000 回中 10000 回復号に成功した最大の prob (以下 $Q_{\text{color}, \text{型}, \text{誤り訂正レベル}}$ とする) をまとめる。

誤り訂正レベル	L	M	Q	H
最大の prob(3 型)	2	4	10	14
最大の prob(5 型)	3	8	14	18
最大の prob(7 型)	3	9	16	22

準備の内容より、たとえば

$$9 = Q_{2,7,M} \leq Q_{2,12,M} \leq Q_{3,12,M}$$

である。よって

P_3 : 3 色のときの 1 つのデータが誤って伝えられる確率

とすると 3 色で 12-M 型の QR コードは少なくとも $P_3 \leq 9$ の環境であれば必ず復号ができることがわかる。

ここで 2 色 14-M 型の QR コードを、3 色にして 12-M 型にできるかを考える。 k の値は以下のようにになっている。

	2 色 14-M	3 色 12-M
k	365	386

よって、 k の値のみを考えると問題なく 3 色 12-M 型にできる。
実験の結果より $P_3 \leq 9$ である環境で使われている場合は誤り訂正能力も保証される。

以上と同様にして、以下の条件を満たすとき 2 色の QR コードは 2 つ小さな型の 5 色の QR コードにできることがわかる。(型によっては 3 つ以上小さくできる可能性もある。)

5 色にするとき							
レベル (2 色)	L		M		Q		H
型番 (2 色)	5 以上	7 以上	5 以上	10 以上	5 以上	7 以上	5 以上
P_5 の条件	$P_5 \leq 2$	$P_5 \leq 8$	$P_5 \leq 4$	$P_5 \leq 16$	$P_5 \leq 10$	$P_5 \leq 18$	$P_5 \leq 14$
レベル (5 色)	L	M	M	Q	Q	H	H

- P_3 や P_5 、 P_7 の条件次第では QR コードを多色化することで、誤り訂正能力を保証したまま小さな QR コードにすることができる。
- 一方で読み取りの時間は 3 色や 5 色だと約 2.3 倍、7 色だと 1.1 倍ほどになってしまう。

- [1] D. コックス, J. リトル, D. オシー著: グレブナー基底 2. 大杉英史, 北村知徳, 日比孝之訳. 丸善出版. 2012.
- [2] 池田和興. 例題が語る符号理論. 共立出版. 2007.
- [3] イエルン・ユステセン, トム・ホーホルト著: 誤り訂正符号入門 [第 2 版]. 阪田省二郎, 栗原正純, 松井一, 藤沢匡哉訳. 森北出版. 2019.
- [4] tech-jp, QR コードの仕組み. <https://www.tech-jp.com/QRCode/index.html> (閲覧日: 2024 年 1 月 13 日)
- [5] KEYENCE, バーコード講座.
<https://www.keyence.co.jp/ss/products/autoid/codereader/basic2d-qr-types.jsp> (閲覧日: 2024 年 1 月 13 日)
- [6] QRcode.com, <https://www.qrcode.com/en/index.html> (閲覧日: 2024 年 1 月 13 日)
- [7] デンソー技術情報.
<https://www.denso-wave.com/ja/adcd/fundamental/2dcode/qrc/index.html> (閲覧日: 2024 年 1 月 13 日)
- [8] 濱屋進. 符号理論入門: 数学的な基礎知識から「QR コード」の作成まで. 工学社. 2008.
- [9] QR のススメ, QR コード作成サイト. <https://qr.quel.jp/> (閲覧日: 2024 年 1 月 10 日)
- [10] いろあと, <https://www.iroato.com/cameleoncode/> (閲覧日: 2024 年 1 月 23 日)
- [11] JIS-X-0510, <https://kikakurui.com/x0/X0510-2018-01.html> (閲覧日: 2024 年 1 月 25 日)
- [12] 助川 修司, 伊藤 正都, 近藤 圭佑, 大園 忠親, 新谷 虎松, QR コードの多色化による 2 次元コードの大容量化について, 全国大会講演論文集, 第 70 回 (コンピュータと人間社会)(2008), 845-846