2. Lineare Codes und Syndrom-Decodierung

Codierungstheorie
Anton Malevich
15.02.2018

Definitionen

Linearer Code, Erzeuger- und Kontrollmatrix

Sei K ein Körper. Ein Unterraum $C \leqslant K^n$ heißt ein *linearer Code*. Sei dim $C = k$ und $d(C) = d$. Dann sagt man, dass C ein $[n, k, d]$- oder ein $[n, k]$-Code ist. Die Zahlen n, k und d heißen Parameter des Codes.

Ist $\{c_1, c_2, \ldots, c_k\}$ eine Basis von C, so heißt die $k \times n$ Matrix $G = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix}$ die *Erzeugermatrix* von C. Eine *Kontrollmatrix* H ist eine $(n - k) \times k$ Matrix mit der Eigenschaft $H\, c^{\mathsf{T}} = 0$ für alle $c \in C$. (Die Vektoren betrachen wir als Zeilenvektoren betrachtet.)
Es gilt: $C = K^k\, G$ und $C = \{c \in K^n : H\, c^{\mathsf{T}} = 0\}$, d.h. C ist der Lösungsraum für das homogene lineare Gleichungssystem mit Matrix H.

Gewicht

Das Gewicht eines Vektors $v \in K^n$ ist die Anzahl von 0 unterschiedlichen Stellen in v.

```
wt (v) = ⌗ {i : v_i ≠ 0} = d (v, 0).
```

Für lineare Codes ist die Minimaldistanz von C gleich dem Minimalgewicht:

```
d (C) = wt (C) = min {wt (c) : 0 ≠ c ∈ C}.
```

Duale Codes

Für zwei Vektoren $v = (v_1, \ldots, v_n)$, $w = (w_1, \ldots, w_2)$ ist $< v, w > = v\, w^{\mathsf{T}} = v_1\, w_1 + \ldots + v_n\, w_n$.

Der code $C^{\perp} = \{v \in K^n : < v, c > = 0 \text{ für alle } c \in C\}$ heißt der zu C *duale* Code.
Hat C Erzegermatrix G und Kontrollmatrix H, so ist G eine Kontroll- und H eine Erzeugermatrix für $C^{\perp}$.

Syndrom-Decodierung

Vorbereitung

Sei C ein $[n, k]$-Code über K mit Kontrollmatrix H.
Für $u \notin C$ ist $u + C = \{u + c : c \in C\}$ die *Nebenklasse* von C bezüglich u.
Für $v \in K^n$ ist $s_H(v) = v\,H^{\mathsf{T}}$ das *Syndrom* von v. Es gilt

$$
\begin{aligned}
& s_H(v) = s_H(u) \\
\Leftrightarrow\ & H\,v^{\mathsf{T}} = H\,u^{\mathsf{T}} \\
\Leftrightarrow\ & H\,(v-u)^{\mathsf{T}} = 0 \\
\Leftrightarrow\ & v - u \in C \\
\Leftrightarrow\ & v \in u + C \\
\Leftrightarrow\ & v + C = u + C,
\end{aligned}
$$

d.h. $s_H(v)$ hängt nur von der Nebenklasse von v ab.
Ein Element $f_v \in v + C$ mit kleinstem Gewicht in $v + C$

$$\mathrm{wt}(f_v) = \min\{\mathrm{wt}(v + c) : c \in C\}$$

ist der *Nebenklassenanführer* von $v + C$.
Jeder Nebenklassenanführer f_v mit $\mathrm{wt}(f_v) \leqslant \frac{d(C)-1}{2}$ ist eindeutig in $v + C$.

Syndrom-Decodierung

Decodiere $v \in K^n$ zu $c = v + f_v \bmod 2$.

Damit Syndrom-Decodierung effizient wird, macht man im Vorfeld eine Tabelle, in der zu jeder Nebenklasse das Syndrom und die Nebenklassenanführer gespeichert werden. Diese Tabelle hat dann $\left|K^{n-k}\right|$ Einträge der Form $[s_H(v), \{f_v\}]$.

Hamming-Codes

Es sei $n = 2^m - 1$. Ein $[n, n - m, 3]$-Code mit Kontrollmatrix, deren Spalten alle von Nullvektor unterschiedlichen m-Tupeln enthalten, heißt ein binärer Hamming-Code Ham(m). (Vertauschen der Spalten liefert einen *äquivalenten* Code.)

Alle Hamming-Codes sind perfekt.

```
m = 3;
Rest@Tuples[{0, 1}, m];
MatrixForm[%^T]
```

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Aufgaben

Aufgabe 1

Zeigen Sie, dass der binäre Code C, der aus den folgenden Codeworten besteht, nicht linear ist:

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

```
{{1, 0, 0, 0, 0, 0, 1}, {1, 0, 0, 1, 1, 1, 1}, {0, 1, 1, 0, 0, 0, 0}, {1, 0, 1, 0, 1, 1, 0},
 {1, 1, 0, 0, 1, 0, 0}, {0, 1, 1, 1, 1, 1, 0}, {1, 0, 1, 1, 0, 0, 0}, {1, 1, 0, 1, 0, 1, 0},
 {0, 1, 0, 0, 1, 1, 1}, {0, 0, 1, 0, 1, 0, 1}, {1, 1, 1, 0, 0, 1, 1}, {0, 1, 0, 1, 0, 0, 1},
 {0, 0, 1, 1, 0, 1, 1}, {1, 1, 1, 1, 1, 0, 1}, {0, 0, 0, 0, 0, 1, 0}, {0, 0, 0, 1, 1, 0, 0}}
```

Aufgabe 2

Beweisen Sie, dass der binäre Code C, der aus den folgenden Codeworten besteht, linear ist:

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

```
{{0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 1, 1, 0}, {1, 1, 1, 0, 0, 0, 1}, {0, 0, 1, 0, 1, 1, 1},
 {0, 1, 0, 0, 1, 0, 1}, {1, 1, 1, 1, 1, 1, 1}, {0, 0, 1, 1, 0, 0, 1}, {0, 1, 0, 1, 0, 1, 1},
 {1, 1, 0, 0, 1, 1, 0}, {1, 0, 1, 0, 1, 0, 0}, {0, 1, 1, 0, 0, 1, 0}, {1, 1, 0, 1, 0, 0, 0},
 {1, 0, 1, 1, 0, 1, 0}, {0, 1, 1, 1, 1, 0, 0}, {1, 0, 0, 0, 0, 1, 1}, {1, 0, 0, 1, 1, 0, 1}}
```

Geben Sie eine Erzeuger- und eine Kontrollmatrix von C an.

Zeigen Sie, dass C ein Hamming-Code ist.

Finden Sie den zu C dualen Code $C^{\perp}$. Welche Gewichte

Aufgabe 3

Wir wollen einen binären ($K = \{0, 1\}$) Code konstruieren, der es uns erlaubt, die 26 Buchstaben des lateinischen Alphabets zu übertragen, und der zwei Fehler korrigieren kann. Um die 26 Buchstaben codieren zu können, brauchen wir ein Code der Dimension mindestens 5. Um zwei Fehler korrigieren zu können, muss die Minimaldistanz des Codes auch mindestens 5 sein. Die kleinste Länge, die das erlaubt, ist 13.

Sei darum C der folgende lineare Code mit Erzeugermatrix

```
G = {{1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0},
     {0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1},
     {0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0},
     {0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1},
     {0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1}};
```

Die Kontrollmatrix finden wir als

```
H = NullSpace[G, Modulus → 2];
MatrixForm[H]
```

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Warum?

Zeigen Sie, dass $\dim C = 5$ und $d(C) = 5$ ist.

Wir schreiben alle Codewörter des Codes C in die Liste L

```
Prepend[Rest@Subsets[G], {Array[0 &, 13]}];
L = Mod[Plus @@@ %, 2];
```

Wir benutzen die ersten 26 Codewörter, um die 26 Buchstaben zu codieren. Das erste Codewort entspricht dem Buchtaben A, das zweite dem B, usw., das 26-te dem Z.

```
TableForm[CodeTable = MapThread[#1 ↔ #2 &, {Alphabet[], L[[ ;; 26]]}]]
```

```
a ↔ {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
b ↔ {1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0}
c ↔ {0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1}
d ↔ {0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0}
e ↔ {0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1}
f ↔ {0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1}
g ↔ {1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1}
h ↔ {1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0}
i ↔ {1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1}
j ↔ {1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1}
k ↔ {0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1}
l ↔ {0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0}
m ↔ {0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0}
n ↔ {0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1}
o ↔ {0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1}
p ↔ {0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0}
q ↔ {1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1}
r ↔ {1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0}
s ↔ {1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0}
t ↔ {1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1}
u ↔ {1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1}
v ↔ {1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0}
w ↔ {0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0}
x ↔ {0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0}
y ↔ {0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1}
z ↔ {0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0}
```

Zur Decodierung brauchen wir die Liste Weight2Less aller Vektoren aus K^{13} mit Gewicht höchstens 2 (die Zahl der Fehler, die wir korrigieren wollen) und die Liste Syndromes ihrer Syndrome.

```
Weight2Less = Join[
    {Array[0 &, 13]},
    Permutations@Join[Array[0 &, 12], {1}],
    Permutations@Join[Array[0 &, 11], {1, 1}]
   ];
Syndromes = Mod[H.#, 2] & /@ Weight2Less;
```

Beispiel

Alice hat das 15-te Element der Liste *L* (also den Buchstaben O) gesendet

```
c = L[[15]]
FromLetterNumber[15]
```

```
{0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1}
```

```
o
```

Bob hat aber den Vektor

```
x = {0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1};
```

empfangen, der kein Codewort ist. Um zu decodieren, berechnet Bob das Syndrom von *x* und sucht dieses in der Liste der Syndrome der möglichen Fehlervektoren

```
Mod[H.x, 2]
Position[Syndromes, %]
```

```
{1, 0, 0, 1, 0, 1, 0, 0}
```

```
{{55}}
```

Dann ist der 55-te Eintrag der Liste Weight2Less

```
Weight2Less[[55]]
```

```
{0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0}
```

der aufgetretene Fehler und

```
Mod[x + Weight2Less[[55]], 2]
c == %
```

```
{0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1}
```

```
True
```

das gesendete Codewort.

Aufgaben

Decodieren Sie die folgenden 5 empfangenen Vektoren unter der Voraussetzung, dass jeweils höchstens 2 Fehler aufgetreten sind.
Was ist die Nachricht?

```
{0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}
{1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0}
{1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1}
{0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0}
{0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0}
```

Es werden die Buchstaben *i* und *j* versandt. Es passieren aber jeweils drei Fehler, und es werden folgende Vektoren empfangen:

```
x = Mod[L⟦LetterNumber["i"]⟧ + {0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0}, 2]
y = Mod[L⟦LetterNumber["j"]⟧ + {0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1}, 2]
```

```
{1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1}
```

```
{1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0}
```

Decodieren Sie. Was ist passiert? Warum?

Benutzen Sie die restlichen unbenutzten 6 Codeworte ($L⟦27 ;; 32⟧$) um die Symbole " ", ",", "." ":", "(" und ")" zu codieren.
Codieren Sie nun die Textnachricht "Coding is cool. :)". Übersenden Sie die Codeworte über ein Kanal mit Symbolfehlerwahrscheinlichkeit $p = 0.01$, ein mit $p = 0.1$ und ein mit $p = 0.2$. Decodieren Sie jeweils die empfangenen Worte. Wie sieht es nun aus?

```
p = 0.1;
Channel[x_, p_ /; RandomReal[] < p] := Mod[x + 1, 2]
Channel[x_, _] := x
SetAttributes[Channel, Listable]
```

```
Total@Table[Channel[0, p], {1000}]
```

```
102
```

- Wie wir oben gesehen haben, kann der Code C einige Fehler vom Gewicht 3 erkennen.
  Sei M die Menge aller Vektoren aus K^{13} von Gewicht 3. Bestimmen Sie, welche von diesen Vektoren in den Nebenklassen mit Anführer aus der Liste Weight2Less liegen (d.h., für welche $v \in M$ ein $e \in$ Weight2Less existiert mit $v \in e + C$) und welche nicht. Welche von den Vektoren vom Gewicht 3 sind eindeutige anführer ihrer Nebenklassen?

```
M = Permutations@Join[Array[0 &, 10], {1, 1, 1}];
```

- Modifizieren Sie den Decodierer. Der soll nun auch die Fehler vom Gewicht 3 korrigieren können, deren Anführer nicht in der Liste Weight2Less liegen. Falls der Anführer nicht eindeutig ist, soll eine Liste der möglichen Codeworten ausgegeben werden.
  Testen Sie den modifizierten Decodieren an den Fehlervektoren:

```
{0,0,0,0,0,0,0,0,0,0,1,1,1}
{1,0,0,0,1,0,0,0,0,0,0,1,0}
{0,0,0,1,0,0,0,0,1,1,0,0,0}
```