SE Codierungstheorie
Leitung: Dr. Amin Coja-Oghlan

Reed-Solomon-Codes

Jörg Pohle, 140114

15. Dezember 2006

---

1 Einleitung

Im gleichen Jahr, in dem R. C. Bose, D. K. Ray-Chaudhuri und A. Hocquenghem die später nach ihnen benannten BCH-Codes beschrieben, entwickelten Irving S. Reed und Gustave Solomon, die zu dieser Zeit zusammen am MIT Lincoln Laboratory arbeiteten, in ihrer Arbeit „Polynomial Codes over Certain Finite Fields“ eine wichtige Teilmenge der BCH-Codes – die Reed-Solomon-Codes. Mangels Fähigkeiten der damaligen Computertechnologie waren die von ihnen vorgeschlagenen Codierungs- und Decodierungsalgorithmen ob ihrer Komplexität jedoch nicht praktisch einsetzbar. Dies änderte sich erst, als Elwyn Berlekamp 1968 einen Algorithmus entwickelte, der für eine gegebene Ausgabesequenz das kürzeste linear rückgekoppelte Schieberegister (engl. Linear Feedback Shift Register) berechnete, und James L. Massey diesen Algorithmus 1969 für lineare Codes adaptierte.

Reed-Solomon-Codes finden aufgrund ihrer guten Fehlerkorrektureigenschaften, und weil inzwischen ein relativ einfacher Decodieralgorithmus existiert, breite Anwendung. So basiert die Fehlerkorrektur gewöhnlicher Audio-CDs auf einem Reed-Solomon-Code, und auch im Mobilfunk, im Digital Video Broadcasting (DVB), im Digital Audio Broadcasting (DAB) sowie zur Kommunikation mit Raumsonden wurden und werden Reed-Solomon-Codes benutzt.

2 Definitionen

Zu Beginn sei hier noch einmal auf die Definition von BCH-Codes verwiesen: Sei α eine primitive n-te Einheitswurzel über $\mathrm{GF}(q)$ und $b \in \mathbb{N}$. Der Code

$$C = C(\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}) \tag{1}$$

heißt ein BCH-Code zum (definierten) Abstand δ.

2.1 Reed-Solomon-Code

Ein BCH-Code der Länge $n = q - 1$ über $\mathrm{GF}(q)$ heißt **Reed-Solomon-Code**.

Sei α ein primitives Element von GF(q)*, dann hat ein Reed-Solomon-Code zum Abstand δ das Generatorpolynom

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+\delta-2}). \tag{2}$$

Für $b = 1$ ist es ein Reed-Solomon-Code im engeren Sinne.

Weil RS-Codes die BCH-Schranke erfüllen, gilt $d \geq \delta = n+k-1$, und gemäß der Singleton-Schranke $k \leq n - d + 1$. Daraus folgt $d = \delta$ und somit, dass alle RS-Codes die Singleton-Schranke mit Gleichheit erfüllen, also $k = n - d + 1$. RS-Codes sind daher MDS-Codes („Maximum Distance Separable“).

Jeder Reed-Solomon-Code zum Abstand $\delta = q - k$ über GF(q) ist ein zyklischer $[q-1, k, q-k; q]$-Code. Es gilt insbesondere, dass für alle $k = 1, \ldots, q-1$ ein zyklischer $[q-1, k, q-k; q]$-Code existiert.

2.2 Erweiterter Reed-Solomon-Code

Sei C ein Code über GF(q). Dann ist der erweiterte Code C' von C definiert als:

$$C' = \{(c_0, \ldots, c_n) : (c_0, \ldots, c_{n-1} \in C) \text{ und } c_0 + \cdots + c_n = 0\}. \tag{3}$$

Ein solcher erweiterter Code existiert auch für RS-Codes. Sei dazu C ein $[q-1, k, q-k; q]$-RS-Code im engeren Sinne (also $b = 1$) und C' der erweiterte Code. Dann ist C' ein $[q, k, q-k+1; q]$-Code, insbesondere also auch ein MDS-Code.

3 Codierung und Decodierung

3.1 Codierung

Wie alle zyklischen Codes können RS-Codes mit Hilfe des Generatorpolynoms $g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+\delta-2})$ codiert werden. Dazu wird für die Nachricht $m = m_{n-k} \ldots m_{n-1}$ dann $c'(m) = m_{n-k}x^{n-k} + \cdots + m_{n-1}x^{n-1}$ in GF(q)[x] durch g dividiert. Das zu übertragende Code-Wort $c(m)$ wird dann aus der Differenz von $c'(m)$ und dem bei der Division entstandenen Rest $r = r_0 + r_1 x + \cdots + r_{n-k-1}x^{n-k-1}$ gebildet, also $c(m) = c'(m) - r$.

Reed und Solomon haben in ihrer Arbeit vorgeschlagen, die Codierung über die diskrete Fourier-Transformation vorzunehmen. Diese ist jedoch im Gegensatz zur Codierung mit Hilfe eines Generatorpolynoms nicht systematisch, die Code-Wörter zerfallen also nicht in Informations- und Kontrollsymbole.

Dazu wird für jede Nachricht $m = (m_0, \ldots, m_{k-1})$ für $1 \leq k \leq q-1$ mit dem zugehörigen Polynom $m(x) = m_0 + m_1 x + \cdots + m - k - 1x^{k-1}$ der zugehörige Fouriervektor von $m(x)$ gebildet. Dieser ist dann das Code-Wort $c(m)$ mit $c(m) = \big(m(1), m(\alpha), \ldots, m(\alpha^{q-2})\big)$.

Die Menge $\{c_u\}$ ist dann der $[q-1, k, q-k; q]$-RS-Code im engeren Sinne.

3.2 Decodierung

Drei Verfahren, die Nachricht aus dem übertragenen Code-Wort wiederherzustellen, sollen hier kurz angesprochen werden.

Eine mittels diskreter Fourier-Transformation codierte Nachricht kann durch eine inverse diskrete Fourier-Transformation wieder decodiert werden.

Wurde für die Codierung der Nachricht ein Generatorpolynom g benutzt, für das das Polynom $h = (x^n -)/g$ gerade das Kontrollpolynom über GF(q) ist, kann die Nachricht m aus dem empfangenen Wort m' wie folgt decodiert werden: Für das empfangene Wort $m' = c'(x) = c(x) + e(x)$, wobei $e(x)$ der Fehlervektor ist, werden die Syndrome S_i von $a(x)$ für $i = 1, \ldots, 2t$ berechnet. Dabei ist t die maximale Anzahl von zu korrigierenden Fehlern durch diesen Code C, also falls für C gilt: $d = 2t + 1$. Falls $S_i = 0$ für alle $i = 1, \ldots, 2t$ gilt, dann ist kein Fehler aufgetreten und $m = c(x) = c'(x) = m'$. Andernfalls muss das Gewicht w des Fehlerpolynoms $e(x)$ berechnet werden, um danach die Koeffizienten $\lambda_1, \ldots \lambda_w$ des Fehlerlokatorpolynoms $\sigma(z) = \prod_{i=1}^{w}(1 - X_i z)$ mit den Fehlerlokatoren X_i zu bestimmen. Die Inversen der Nullstellen von $\sigma(z)$ sind dann gerade die Fehlerlokatoren X_i von $e(x)$, wodurch die Fehlergrößen Y_i von $e(x)$ – und damit $e(x)$ selbst – eindeutig bestimmt werden können. Die Nachricht m lässt sich dann mit $m = c'(x) - e(x)$ decodieren.

Das dritte Verfahren, das von Reed und Solomon beschriebene „Majority Logic Decoding“, folgt prinzipiell der gleichen Idee wie „Minimum Error Probability Decoding“ (MED). Das heißt, das empfangene Code-Wort entscheidet selbst, welches das gesendete ist.

Empfangen wird $m' = c(x) + e(x)$ mit Irrtumsvektor $e = (e_0, \ldots, e_{q-2})$. Der Empfänger kennt also

$$\begin{aligned} m'_0 &= e_0 + m_0 + \cdots + m_{k-1} \\ m'_1 &= e_1 + m_0 + \alpha m_1 + \cdots + \alpha^{k-1} m_{k-1} \\ &\vdots \\ m'_{q-2} &= e_{q-2} + m_0 + \alpha^{q-2} m_1 + \cdots + \alpha^{(k-1)(q-2)} m_{k-1} \end{aligned}$$

und berechnet für beliebiges e die Lösungen der $\binom{q-1}{k}$ Teilsysteme aus je k dieser Gleichungen, wobei jedes Teilsystem eine eindeutige Lösung für k Unbekannte liefert. Falls keine Fehler aufgetreten sind, ist $e = 0$ und jedes der Teilsysteme kann nach $m_0, \ldots, m_{k-1}$ aufgelöst werden.

Falls $e \neq 0$ ist, also Fehler aufgetreten sind, wird dasjenige m gewählt, das am häufigsten als Lösung der Teilsysteme auftritt.

Wenn $2w(e) < d$ ist, wobei $w(e)$ das Gewicht des Fehlervektors e ist, liefert die Decodierung auch das richtige Ergebnis, weil genau dann gilt:

$$\binom{q-1-w}{k} > \binom{w+k-1}{k} \tag{4}$$

.

Dies kann folgendermaßen verdeutlicht werden:

Sei m eine inkorrekte Lösung. Also kann m dann höchstens $\binom{w+k-1}{k}$ Teilsysteme erfüllen, nämlich w Gleichungen mit $e_i \neq 0$ – wegen des Fehlergewichtes w – und $k-1$ Gleichungen mit $e_i = 0$. Würde es mehr als $k-1$ Gleichungen lösen, wäre es demgegenüber korrekt.

Hingegen kann ein korrektes m genau $\binom{q-1-w}{k}$ Teilsysteme erfüllen, nämlich gerade diejenigen, die nur Gleichungen mit $e_i = 0$ enthalten. In diesem Fall gilt dann (4) und m kann korrekt decodiert werden.

4 Projizierte Reed-Solomon-Codes

4.1 Definition der Projektion

Reed-Solomon-Codes haben für die Nutzung in der Informationstechnik einen Nachteil – sie sind oftmals über sehr langen Alphabeten definiert, während die Informationstechnik mit einem binären Alphabet arbeitet.

Es existiert jedoch eine einfache Methode, um aus bestimmten RS-Codes solche über kürzeren Alphabeten zu erstellen. Bedingung dafür ist, dass $q = p^m$ ist.

Grundsätzlich gilt: Falls $q = p^m$ ist, kann ein $[q-1, k, d; q]$-RS-Code der Länge $n = q-1$ über $\mathrm{GF}(q)$ in einen $[m(q-1), mk, d'; q]$-Code der Länge $n' = m(q-1)$ über $\mathrm{GF}(p)$ projiziert werden mit $d' \geq \delta = q - k$. Dabei wird jedes $x \in \mathrm{GF}(q)$ für eine Basis b von $\mathrm{GF}(q)$ über $\mathrm{GF}(p)$ mit $x = x_1 b_1 + \cdots + x_m b_m$ ersetzt durch seinen Koordinatenvektor $x' = (x_1, \ldots, x_m) \in \mathrm{GF}(p)^m$.

Eine zweite Möglichkeit ist, mittels Projektion aus einem RS-Code der Länge $n = q-1$ über $\mathrm{GF}(q)$ einen Code der Länge $n' = (m+1)(q-1)$ herzustellen. Die Elemente x von $\mathrm{GF}(q)$ werden dabei jeweils ersetzt durch $x' = (x_1, \ldots, x_m, x_1 + \cdots + x_m)$.

Bei der Projektion gilt zu beachten, dass solche durch Projektion konstruierten Codes im allgemeinen nicht mehr zyklisch sind.

4.2 Praktische Bedeutung

Die praktische Bedeutung von projizierten RS-Codes ist dennoch groß. Sie besteht in ihrer Eignung zur Korrektur von Fehlerbündeln, sogenannten Burst-Fehlern.

Sei C ein RS-Code über $\mathrm{GF}(q)$ mit $q = p^m$, der t Fehler korrigieren kann. C hat demnach die Länge $n = q - 1 = p^m - 1$ und Dimension $q - 2t = p^m - 2t$.

Der projizierte Code C' über $\mathrm{GF}(p)$ hat die Länge $mn = m(p^m - 1)$ und Dimension $m(p^m - 2t)$. Auch C' kann nur t Fehler korrigieren.

C' kann aber darüberhinaus alle Fehlerbündel der Länge $b \leq (t-1)m + 1$ korrigieren, weil jedes solcher Fehlerbündel höchstens t aufeinanderfolgende Symbole in C verändert, die C jedoch korrigieren kann.

An einem Beispiel soll dies verdeutlicht werden:

Sei C ein $[3, 2, 2; 4]$-RS-Code und C' der projizierte $[6, 4, 2; 2]$-Code. Sei c ein Code-Wort in C in der Form $c = xyz$, dann hat c' in C' die Form $c' = x_1x_2y_1y_2z_1z_2$. C kann $t = 2$ Fehler korrigieren. Ein Fehlerbündel der Länge $b \leq (t-1)m + 1 = 3$ kann daher in c' höchstens die Sequenzen $(x_1x_2y_1)$, $(x_2y_1y_2)$, $(y_1y_2z_1)$ oder $(y_2z_1z_2)$ verändern. Daher ist leicht zu sehen, dass im ursprünglichen Code C maximal 2 Fehler auftreten können, und diese Fehler sind korrigierbar.

5 Literatur

Jungnickel, Dieter, Codierungstheorie, Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1995.