Lineaire codes

Hoofdstuk III

LINEAIRE CODES

3.1. BLOK CODES

We nemen nu aan dat informatie wordt gecodeerd met behulp van een *alfabet* Q van q verschillende symbolen. Een code heet een *blok code* als de gecodeerde informatie verdeeld kan worden in rijtjes symbolen van vaste lengte die onafhankelijk van elkaar gedecodeerd kunnen worden. Deze blokken noemen we *codewoorden*; de lengte heet *blok lengte* of *woord lengte*. Alle voorbeelden in hoofdstuk II zijn blok codes. De symbolen van een woord noemen we weer de *letters* of ook wel coordinaten (zie § 3.2). Merk op dat de woorden in de nederlandse taal ook blokken zijn maar niet met vaste lengte. Een voor de practijk zeer belangrijke manier van coderen die we in dit boek helemaal niet beschouwen is een zgn. *convolutiecode*. Daarbij wordt een (evtl. oneindige) informatierij $i_0,i_1,i_2,\ldots$ gecodeerd (bij rate $\frac{1}{2}$) als $i_0,i_0',i_1,i_1',i_2,i_2',\ldots$ waarbij i_n' berekend wordt (m.b.v. een vooraf gegeven voorschrift) uit $i_0,i_1,\ldots,i_n$. Bij deze code is van blokken geen sprake.

Als generalisatie van (1.1.3) definieren we voor woorden $\underline{x}$ en $\underline{y}$ van n letters uit een alfabet Q met q letters:

(3.1.1) DEFINITIE. De *Hamming-afstand* $d_H(\underline{x},\underline{y})$ van $\underline{x}$ en $\underline{y}$ is

$$d_H(\underline{x},\underline{y}) := |\{i \mid 1 \leq i \leq n,\ x_i \neq y_i\}|.$$

Het *gewicht* $w(\underline{x})$ van $\underline{x}$ is $d_H(\underline{x},\underline{0})$, waarbij $\underline{0} = (0,0,\ldots,0)$, (= de *oorsprong*).

Hamming-afstand is een geschikt afstandsbegrip indien bij een fout in het i-de symbool alle mogelijke fouten op die positie even waarschijnlijk zijn en de fout in de i-de positie geen gevolgen heeft voor de andere posities. In hoofdstuk X leren we een ander afstandsbegrip kennen.

Een (blok-)code C met woordlengte n is een niet-lege deelverzameling van Q^n. We noemen C *triviaal* als $|C| = 1$. De code heet *binair* (zie hoofdstuk II) als $q = 2$, *ternair* als $q = 3$, etc. De volgende begrippen spelen

een centrale rol zoals uit de voorbeelden van hoofdstuk II duidelijk moet zijn:

(3.1.2) DEFINITIE. De *minimale afstand* van een niet-triviale code C is $\min\{d_H(\underline{x},\underline{y}) \mid \underline{x} \in C, \underline{y} \in C, \underline{x} \neq \underline{y}\}$. Het *minimale gewicht* van C is $\min\{w(\underline{x}) \mid \underline{x} \in C, \underline{x} \neq \underline{0}\}$.

We generaliseren (1.1.2) nu ook.

(3.1.3) DEFINITIE. Is $|Q| = q$ en $C \subset Q^n$ dan heet

$$R := n^{-1} \, {}_q\log|C|$$

de (*informatie-*) *rate* van C.

3.2. LINEAIRE CODES

We willen nu codes construeren met een algebraïsche structuur. Als het alfabet Q een groep is en de code C is een ondergroep van Q^n dan heet C een *groepcode*. In deze paragraaf eisen we nog iets meer. Laat Q het lichaam GF(q) zijn waarbij $q = p^f$ (p priem). De collectie Q^n is een n-dimensionale vectorruimte die we ook met $R^{(n)}$ aangeven.

(3.2.1) DEFINITIE. Een *lineaire code* V is een lineaire deelruimte van $R^{(n)}$. Als V dimensie k heeft wordt V een (n,k)-code over GF(q) genoemd. (N.B. niet verwarren met de notatie uit hoofdstuk II).

(3.2.2) DEFINITIE. Een *generator matrix* G (kort: generator) voor een lineaire code V is een matrix G waarvan de rijen een stelsel basisvectoren van V vormen.

Voor een (n,k)-code over GF(q) is een generator G een matrix met afmetingen k × n. De code bestaat uit alle vectoren $\underline{a}G$ met $\underline{a} \in R^{(k)}$. We zullen zeggen dat G de *standaardvorm* heeft als $G = (I_k, P)$, waarbij P een k × (n-k) matrix is. De in § 1.1 behandelde (6,3)-code over GF(2) had generator $G = (I, J-I)$, dus in standaardvorm. Merk op dat als G de standaardvorm heeft elk codewoord begint met k symbolen die willekeurig gekozen mogen worden (*informatiesymbolen*) gevolgd door n-k redundante symbolen die *parity-check symbolen* worden genoemd. Deze naam is afkomstig van het in hoofdstuk I genoemde voorbeeld van ponsband waar $G = (I_5 \underline{j}^T)$. Het zesde symbool van ieder woord controleert de pariteit.

(3.2.3) DEFINITIE. Twee codes C_1 en C_2 heten *equivalent* als er een permutatie π van $\{1,2,\ldots,n\}$ is zo dat

$$C_2 = \{c_{\pi(1)}, c_{\pi(2)}, \ldots, c_{\pi(n)}) \mid \underline{c} \in C_1\}.$$

Vaak wordt het equivalentie-begrip nog uitgebreid door ook nog toe te staan dat op elke plaats een permutatie van Q optreedt.

(3.2.4) STELLING. Bij iedere lineaire code is er een equivalente code die een generator in standaardvorm heeft.

BEWIJS. Dit is een bekende stelling uit de lineaire algebra. □

I.h.a. wordt een code *systematisch* genoemd als een aantal symbolen van elk woord vrij gekozen mag worden (weer: informatiesymbolen) en de andere symbolen dan bepaald zijn. In (3.2.4) staat dus dat iedere lineaire code (equivalent met) een systematische code is. Zoals we mochten verwachten is volgens (3.1.2) de rate van een (n,k)-code $\frac{k}{n}$ omdat de code q^k woorden bevat.

3.3. FOUTENVERBETERING

Bij het interpreteren van ontvangen signalen (bij gebruik van lineaire codes) passen we weer maximum-likelihood decoding toe. Als voor de code V de minimum afstand 2e + 1 is dan kunnen we foutenpatronen met ≤ e fouten corrigeren. Is de minimum afstand 2e dan wordt een foutenpatroon met e fouten wel ontdekt maar het is soms niet te verbeteren (e-*error-detecting code*).

(3.3.1) STELLING. *Voor een lineaire code* V *is de minimum-afstand gelijk aan het minimum gewicht.*

BEWIJS. $d_H(\underline{x},\underline{y}) = w(\underline{x}-\underline{y})$ en als $\underline{x} \in V$ en $\underline{y} \in V$ is ook $\underline{x}-\underline{y} \in V$. □

Uit deze stelling zien we dat de controle van de kwaliteit van een lineaire code aanzienlijk minder werk vergt dan voor een niet lineaire code waar men $d_H(\underline{x},\underline{y})$ voor alle paren $(\underline{x},\underline{y})$ moet uitrekenen.

(3.3.2) DEFINITIE. Is V een (n,k)-code over GF(q) dan is de *duale code* $V^\perp$ een (n,n-k)-code gedefinieerd door

$$V^{\perp} := \{\underline{y} \in R^{(n)} \mid \forall_{\underline{x} \in V}[<\underline{x},\underline{y}>=0]\}.$$

Hierin is $< \underline{x},\underline{y} >$ het inwendig product over GF(q), d.i. $x_1y_1 + \ldots + x_ny_n$. Merk op dat het feit dat $V^{\perp}$ een (n-k)-dimensionale lineaire deelruimte van $R^{(n)}$ is weer een bekende stelling uit de lineaire algebra is. We moeten wel bedenken dat over GF(q) i.h.a. niet geldt dat iedere $\underline{z}$ is te schrijven als $\underline{x} + \underline{y}$ met $\underline{x} \in V^{\perp}$ zoals we uit de lineaire algebra in Euclidische ruimten gewend zijn.

Is $G = (I_k, P)$ een generator van V in standaardvorm dan is $H = (-P^T, I_{n-k})$ een generator van $V^{\perp}$. Immers: H heeft de juiste afmetingen, de rang van H is n-k en $GH^T = 0$. Daar ieder codewoord $\underline{x} \in V$ de vorm $\underline{x} = \underline{a}\,G$ heeft kunnen we V ook beschrijven door

(3.3.3) $\quad \underline{x} \in V \iff \underline{x}\,H^T = \underline{0}.$

Dit is een stelsel van n-k lineaire vergelijkingen die V bepalen. Deze vergelijkingen heten *parity-check* vergelijkingen en H heet een *parity-check matrix* voor V. I.h.a. is voor iedere $\underline{y} \in V^{\perp}$ de vergelijking $< \underline{x},\underline{y} > = 0$ een parity-check vergelijking voor V. Voor de code uit § 1.1 zijn de vergelijkingen $a_4 = a_2 + a_3$, etc. waarmee de code werd gedefinieerd drie parity-check vergelijkingen, overeenkomend met H = (J-I, I).

(3.3.4) DEFINITIE. Is V een lineaire code met parity-check matrix H, dan noemen we voor iedere $\underline{x} \in R^{(n)}$ de vector $\underline{x}H^T$ het *syndroom* van $\underline{x}$.

De code V bestaat uit alle vectoren met syndroom $\underline{0}$. Daar V een ondergroep is van $R^{(n)}$ kunnen we $R^{(n)}$ splitsen in nevenklassen van V. Het is duidelijk dat twee vectoren $\underline{x}$ en $\underline{y}$ in dezelfde nevenklasse zitten als en alleen als ze hetzelfde syndroom hebben (immers $\underline{x}H^T = \underline{y}H^T \iff \underline{x}-\underline{y} \in V$). Hieruit zien we dat een ontvangen signaal $\underline{x}$ een foutenpatroon $\underline{e}$ uit dezelfde nevenklasse moet hebben (want $\underline{x}-\underline{e} \in V$). Om te decoderen moeten we dus een keuze doen uit de elementen van de nevenklasse van $\underline{x}$ die minimaal gewicht hebben. In de practijk gaat dit als volgt. We maken een lijstje van alle syndroomwaarden. Bij ieder daarvan behoort een nevenklasse. Uit deze nevenklasse kiezen we een representant (*coset-leader*) met minimaal gewicht. Wordt nu $\underline{x}$ ontvangen dan zoeken we bij $\underline{x}H^T$ de representant op en trekken deze van $\underline{x}$ af. De lezer kan nu zelf nagaan dat dit precies is wat we in § 1.1 hebben

gedaan met de binaire (6,3)-code. Voor 7 nevenklassen was de representant eenduidig bepaald; voor de laatste moesten we er één kiezen uit drie met hetzelfde gewicht. Het is duidelijk dat als V minimum afstand $d = 2e+1$ heeft twee vectoren met gewicht $\leq e$ niet in dezelfde nevenklasse kunnen zitten. In dat geval zijn deze vectoren dus allemaal representanten van verschillende nevenklassen.

Ook over een alfabet van q symbolen geldt (2.3.1). Bij een perfecte code zijn er geen andere representanten van nevenklassen dan de vectoren met gewicht $\leq e$. (Dit zijn er $\sum_{i=0}^{e} \binom{n}{i}(q-1)^i$). Een code (zoals ons voorbeeld uit § 1.1) waarvan de minimum afstand $d = 2e+1$ is en alle representanten van nevenklassen een gewicht $\leq e+1$ hebben heet *quasiperfect*.

3.4. HAMMING CODES

(3.4.1) STELLING. *Een lineaire code* V *over* GF(q) *heeft minimum afstand* ≥ 3 *als en alleen als de kolommen van de parity-check matrix* H *niet* $\underline{0}$ *zijn en paarsgewijs lineair onafhankelijk.*

BEWIJS. (i) Stel dat H de genoemde eigenschap heeft. De vergelijking $\underline{x}H^T = \underline{0}$ betekent dat de kolommen behorende bij coordinaten $x_i \neq 0$ lineair afhankelijk zijn. Is dus $\underline{x} \neq \underline{0}$ en $\underline{x}H^T = \underline{0}$ dan is $w(\underline{x}) \geq 3$.

(ii) Heeft H de genoemde eigenschap niet dan zien we op precies dezelfde manier dat er een $\underline{x}$ is met $1 \leq w(\underline{x}) \leq 2$ zó dat $\underline{x}H^T = \underline{0}$. □

Beschouw nu de r-dimensionale ruimte over GF(q). Bij iedere $\underline{x} \neq \underline{0}$ zijn er q-1 vectoren die veelvouden van $\underline{x}$ zijn. Er zijn dus $(q^r-1)/(q-1)$ paarsgewijs lineair onafhankelijke vectoren $\neq \underline{0}$. Noem dit aantal n. Kiezen we zo'n stelsel van n vectoren als kolommen van een r bij n matrix H dan heet de code met deze parity-check matrix een *Hamming code* en wel een (n,n-r)-*Hamming code* over GF(q). Is $q = 2$ dan **bestaat** H uit alle mogelijke kolommen $\neq \underline{0}$. Decoderen is dan heel eenvoudig. Orden de kolommen van H zo dat de i-de kolom de binaire schrijfwijze van het getal i is. Wordt $\underline{x}$ ontvangen en is het syndroom niet $\underline{0}$ dan is het syndroom de binaire schrijfwijze van een getal i. Vervang dan x_i door $x_i + 1$. Er ontstaat een codewoord. Hieruit zien we dat een binaire Hamming code perfect is. Dit geldt voor alle Hamming codes.

(3.4.2) STELLING. *De Hamming codes over* GF(q) *zijn perfect.*

BEWIJS. Zij $n := (q^r-1)/(q-1)$ en V een $(n,n-r)$-Hamming code. Is $\underline{v} \in V$ dan is $|B_1(\underline{v})| = 1 + n(q-1) = q^r$. De q^{n-r} disjuncte bollen $B_1(\underline{v})$ met $\underline{v} \in V$ bevatten dus q^n punten, d.w.z. dat ze $R^{(n)}$ overdekken. □

In § 2.1 hebben we gezien hoe uit de (7,4)-Hamming code door verlenging een code met woordlengte 8 en minimum afstand 4 kon worden gemaakt. Dit is een voorbeeld van een algemeen principe.

(3.4.3) DEFINITIE. Is C een code in $R^{(n)}$ dan wordt de *verlengde code* $\overline{C}$ (= *extended code*) in $R^{(n+1)}$ gedefinieerd door

$$(c_1,c_2,\ldots,c_{n+1}) \in \overline{C} \iff ((c_1,c_2,\ldots,c_n) \in C \wedge \sum_{i=1}^{n+1} c_i = 0).$$

Voor het geval dat C een lineaire code in $R^{(n)}$ is met generator G en parity-check matrix H vinden we voor $\overline{C}$ de matrices G^* en H^* door aan G een kolom toe te voegen zó dat de kolommen samen $\underline{0}$ zijn en dan

$$H^* := \begin{pmatrix} 1 & 1 & \ldots & 1 \\ & & & 0 \\ & H & & 0 \\ & & & \vdots \\ & & & 0 \end{pmatrix}.$$

(Vaak wordt de nieuwe letter van de verlengde code voorop geschreven).
Voor het binaire geval zien we dat in $\overline{C}$ alle woorden even gewicht hebben en dat $\overline{C}$ dus even minimum afstand heeft. Als dus C een oneven minimum afstand d heeft dan heeft $\overline{C}$ minimum afstand d + 1.

3.5. DREMPEL DECODERING

We geven nu een korte schets van een decodeermethode die voor vele lineaire codes wordt gebruikt. De methode heeft als voordeel de eenvoud en het feit dat vaak meer fouten worden verbeterd dan men verwacht op grond van de minimum afstand.

(3.5.1) DEFINITIE. Een stelsel parity-check vergelijkingen $< \underline{x}, \underline{y}^{(\nu)} > = 0$, $(1 \le \nu \le r)$ heet *orthogonaal* op positie i voor de code V als

(i) $y_i^{(\nu)} = 1 \quad (1 \le \nu \le r)$,

(ii) als $j \neq i$ dan is $y_j^{\nu} \neq 0$ voor ten hoogste één waarde van ν.

Laat $\underline{x}$ een woord zijn dat t fouten bevat waarbij $t \le \frac{1}{2} r$. Dan is

$$< \underline{x}, \underline{y}^{(\nu)} > \neq 0 \text{ voor } \begin{cases} \le t \text{ waarden van } \nu \text{ als } x_i \text{ goed is,} \\ \ge r-(t-1) \text{ waarden van } \nu \text{ als } x_i \text{ fout is.} \end{cases}$$

Daar $r-(t-1) > t$ beslist de meerderheid van de waarden van $< \underline{x}, \underline{y}^{(\nu)} >$ (nl. 0 of niet 0) of x_i goed is of fout. Bij een binaire code kan direct daarop x_i verbeterd worden. In de practijk gebruikt men een teller die zodra een bepaalde drempelwaarde wordt overschreden x_i verandert. Daarom noemt men dit procédé *"threshold decoding"*. Men moet wel voor iedere i over zo'n orthogonaal stelsel parity-checks beschikken.

We geven een voorbeeld. Zij V de duale code van de (7,4)-Hamming code. Deze code heeft generator

$$G := \begin{pmatrix} 0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1 \end{pmatrix}$$

De parity-check vergelijkingen

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_1 + x_4 + x_5 &= 0 \\ x_1 + x_6 + x_7 &= 0 \end{aligned}$$

zijn orthogonaal op de 1^{ste} positie. Als het woord $\underline{x}$ precies één fout bevat dan geven de drie vergelijkingen als uitkomst of drie keer 1 (als x_1 fout is) of één keer 1 (als x_1 goed is). Er zijn 6 even waarschijnlijke foutenpatronen van gewicht 2 die bij de drie vergelijkingen het resultaat 0,1,1 leveren. Slechts twee daarvan hebben een fout op de 1^{ste} plaats. Hier blijkt dus dat we als drempel $2\frac{1}{2}$ moeten kiezen omdat alleen de uitkomst 1,1,1 verandering van x_1 rechtvaardigt. We kunnen de zaak ook enigszins anders bekijken. Stel dat we het woord $\underline{y} = \underline{x} + \underline{e}$ ontvangen. Dan is blijkbaar

$$y_1 = x_1 + e_1$$
$$y_2 + y_3 = x_1 + e_2 + e_3$$
$$y_4 + y_5 = x_1 + e_4 + e_5$$
$$y_6 + y_7 = x_1 + e_6 + e_7.$$

De ontvanger kent de linkerleden. De meerderheid van de waarden is de waarde die we aan x_1 moeten toekennen. Bij staken van de stemmen nemen we $x_1 = y_1$ zoals boven is uitgelegd. Deze zienswijse verklaart de naam *majority-decoding* die ook wel voor dit procédé wordt gebruikt. In ons voorbeeld heeft V minimum afstand 4. We verwachten 1 fout te kunnen verbeteren. Het geschetste procédé verbetert vele foutenpatronen met 2 fouten ook goed.

3.6. DE WEIGHT ENUMERATOR EN DE MACWILLIAMS IDENTITEIT

Hoewel het minimum gewicht d van een lineaire code iets zegt over het aantal fouten dat we kunnen verbeteren is het mogelijk dat deze minimum afstand zelden optreedt. Dan zullen vele fouten patronen van gewicht > $\frac{1}{2}$ d ook nog goed gedecodeerd worden. Meer informatie over een code wordt gegeven door de zgn. *"weight enumerator"*.

(3.6.1) DEFINITIE. Is A_i het aantal woorden van gewicht i in een code met woordlengte n dan heet

$$A(z) := \sum_{i=0}^{n} A_i z^i$$

de *weight enumerator* van de code. De rij $(A_i)_{i=0}^{n}$ wordt de *weight distribution* van de code genoemd.

Een voorbeeld van berekening van A(z) is gegeven in (2.7.1).

Is de code klein genoeg dan kunnen we de getallen A_i door inspectie bepalen.

We berekenen de weight enumerator van de binaire Hamming codes. Beschouw i-1 kolommen van de parity check matrix H. Er zijn 3 mogelijkheden:

1) de som van deze kolommen is 0,
2) de som van deze kolommen is een van de gekozen kolommen,
3) de som van deze kolommen is gelijk aan een van de andere kolommen.

Het totale aantal manieren om i-1 kolommen te kiezen is $\binom{n}{i-1}$. Mogelijkheid 1) kan op A_{i-1} manieren optreden, mogelijkheid 2) op $(n-(i-2))A_{i-2}$ manieren,

en 3) op iA_i manieren. Dus

$$iA_i = \binom{n}{i-1} + A_{i-1} - (n-i+2)A_{i-2}.$$

Deze formule hebben we bewezen voor $1 \leq i \leq n+1$. Als $i > n$ dan is $A_i = 0$, dus voor $i = n+1$ levert deze formule $0 = 1 - A_n - A_{n-1}$. Dit klopt, want de code is 1-perfect. We vermenigvuldigen beide leden met z^{i-1} en sommeren over $i = 1,\ldots,n+2$

$$\sum_{i=1}^{n+2} iA_i z^{i-2} = \sum_{i=1}^{n+1} \left\{ \binom{n}{i-1} z^{i-1} - A_{i-1} z^{i-1} - nz^{i-1} A_{i-2} + (i-2) z^{i-1} A_{i-2} \right\}$$

dus

$$A'(z) = (1+z)^n - A(z) - nzA(z) + z^2 A'(z)$$

daar $A(0) = 1$ is de oplossing

$$(3.6.2) \qquad A(z) = \frac{1}{n+1}(1+z)^n + \frac{n}{n+1}(1+z)^{(n-1)/2}(1-z)^{(n+1)/2}.$$

We willen nu uit de weight enumerator van een code de weight enumerator van de duale code afleiden. Om het verband tussen beide op te sporen gebruiken we als hulpmiddel karakters.

Zij $(G,+)$ een groep en T de groep van de complexe getallen met modulus gelijk aan 1 en met vermenigvuldiging als operatie. Een *karakter* χ is een homomorfisme $\chi: G \to T$. Dus

$$\chi(g_1+g_2) = \chi(g_1) \cdot \chi(g_2)$$

en

$$\chi(-g_1) = (\chi(g_1))^{-1}.$$

(3.6.3) LEMMA. *Zij* 0 *het eenheidselement in* $(G,+)$. *Dan is* $\chi(0) = 1$.

Een karakter χ heet het *hoofdkarakter* als $\forall g \in G\ [\chi(g) = 1]$.

(3.6.4) LEMMA. *Als* χ *het hoofdkarakter is, dan is* $\sum_{g\in G} \chi(g) = |G|$.
Als χ *niet het hoofdkarakter is, dan is* $\sum_{g\in G} \chi(g) = 0$.

BEWIJS.

$$\chi(h) \sum_{g\in G} \chi(g) = \sum_{g\in G} \chi(h+g) = \sum_{k\in G} \chi(k)$$

dus

$$(\chi(h) - 1) \sum_{g\in G} \chi(g) = 0.$$

Als er een h is met $\chi(h) \neq 1$ dan $\sum_{g\in G} \chi(g) = 0$, anders $\sum_{g\in G} \chi(g) = \sum_{g\in G} 1 = |G|$. □

(3.6.5) STELLING. (MacWilliams identiteit). *Zij* V *een* (n,k)*-code over* GF(q), *zij* A(z) *de weight enumerator van* V, *en* B(z) *die van* $V^{\perp}$. *Dan geldt*

$$q^{-k}(1+(q-1)z)^n A\left(\frac{1-z}{1+(q-1)z}\right) = B(z).$$

(3.6.6) LEMMA. *Definieer* $g(\underline{u}) = \sum_{\underline{v}\in R} \chi(<\underline{u},\underline{v}>)\, z^{w(\underline{v})}$ *waarin* $w(\underline{v})$ *het gewicht van* $\underline{v}$ *is, en* χ *een willekeurig niet-hoofdkarakter; dan is* $B(z) = \frac{1}{|V|} \sum_{\underline{u}\in V} g(\underline{u})$.

BEWIJS. Zij R de n-dimensionale vectorruimte over GF(q).

$$\sum_{\underline{u}\in V} g(\underline{u}) = \sum_{\underline{u}\in V} \sum_{\underline{v}\in R} \chi(<\underline{u},\underline{v}>)\, z^{w(\underline{v})} = \sum_{\underline{v}\in R} z^{w(\underline{v})} \sum_{\underline{u}\in V} \chi(<\underline{u},\underline{v}>)$$

$$= \sum_{\underline{v}\in V^{\perp}} z^{w(\underline{v})} |V| = |V|\, B(z)$$

want de afbeelding $\underline{u} \mapsto \chi(<\underline{u},\underline{v}>)$ is een karakter op de additieve groep van de vectorruimte R en wel het hoofdkarakter als en slechts als $\underline{v} \in V^{\perp}$. □

BEWIJS VAN STELLING (3.6.5)

Zij g gedefinieerd als in het lemma, zij $\underline{u} = u_1u_2\ldots u_n$ en breid w uit tot GF(q) door

$$w(v) = \begin{cases} 0 & \text{als } v = 0 \\ 1 & \text{anders,} \end{cases} \qquad \text{voor } v \in GF(q).$$

Dan geldt

$$g(\underline{u}) = \sum_{\underline{v} \in R} \chi(<\underline{u},\underline{v}>)\, z^{w(\underline{v})} =$$

$$= \sum_{v_1 \ldots v_n \in R} z^{w(v_1)+\ldots+w(v_n)}\, \chi(u_1v_1+\ldots u_nv_n) =$$

$$= \sum_{v_1 \ldots v_n} z^{w(v_1)} \chi(u_1v_1) \ldots z^{w(v_n)} \chi(u_nv_n) =$$

$$= \prod_{i=1}^{n} \sum_{v \in GF(q)} z^{w(v)}\, \chi(u_iv).$$

Als $u_i = 0$ dan is de som gelijk aan $1 + (q-1)z$,
als $u_i \neq 0$ dan is de som gelijk aan $1 + z \sum_{\substack{\alpha \in GF(q) \\ \alpha \neq 0}} \chi(\alpha) = 1 - z$.

Dus

$$g(\underline{u}) = (1-z)^{w(\underline{u})} (1+(q-1)z)^{n-w(\underline{u})} = (1+(q-1)z)^n \left(\frac{1-z}{1+(q-1)z}\right)^{w(\underline{u})}$$

Nu is

$$B(z) = \frac{1}{|V|} \sum_{\underline{u} \in V} g(\underline{u}) = q^{-k}(1+(q-1)z)^n \sum_{\underline{u} \in V} \left(\frac{1-z}{1+(q-1)z}\right)^{w(\underline{u})}$$

$$= q^{-k}(1+(q-1)z)^n\, A\left(\frac{1-z}{1+(q-1)z}\right). \qquad \square$$

3.7. COMMENTAAR

Een van de baanbrekers in de theorie van groep codes was D. Slepian. Zijn artikelen (1956 en later) geven nu, door de snelle groei van het vak, weinig informatie meer maar ze hebben grote invloed gehad.

De lezer die meer wil weten over drempel decodering raadplege MASSEY (1963).

Een generalisatie van de weight enumerator en de identiteit van MacWilliams vinden we in hoofdstuk VII.

In VAN LINT (1971) wordt met behulp van (3.6.2) aangetoond dat het gemiddelde aantal fouten per blok in een Hamming code na het decoderen groter kan zijn dan ervoor! Het hangt dus van het kanaal af of het wel of niet zinvol is om een Hamming code te gebruiken.

3.8. OPGAVEN

(3.8.1) Beschouw een code over een alfabet van 3 symbolen met woordlengte n. Hoeveel woorden zijn er met Hamming-afstand maximaal 3 tot een gegeven codewoord?

(3.8.2) Beschouw de vectorruimte $\{0,1\}^6$ met Hamming-afstand (= blokken nullen en enen, blok lengte 6). Wat is het aantal punten in een bol met straal 1? Is het mogelijk 9 vectoren (woorden) te vinden zo dat voor ieder paar $\underline{x},\underline{y}$ geldt $\underline{x}\neq\underline{y} \Rightarrow d_H(\underline{x},\underline{y}) \geq 3$?

(3.8.3) Als een (n,k) code over GF(q) een generator G heeft waarin geen kolom met alleen nullen voorkomt, dan is de som van de gewichten van de codewoorden $n(q-1)q^{k-1}$. Bewijs dit.

(3.8.4) Als V een binaire (n,k)-code is, dan hebben alle woorden even gewicht, of de codewoorden van even gewicht vormen een (n,k-1) code. Bewijs dit.

(3.8.5) Zij C een code met generator matrix

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

Decodeer a) 1 1 0 1 0 1 1,
b) 0 1 1 0 1 1 1,
c) 0 1 1 1 0 0 0.

(3.8.6) De parity check matrix van een binaire code is

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

Decodeer a) 1 1 1 1 0 1 0 0 0,
b) 1 1 0 1 0 1 0 1 1,
c) 0 1 0 0 1 0 0 1 0.

(3.8.7) Zij p priem. Bestaat er een zelfduale (8,4)-code over GF(p)?

(3.8.8) Hoe gedraagt de rate van een (n,k)-Hamming code zich voor grote k?

(3.9.9) U speelt mee in de voetbaltoto en wilt zeker zijn van de 1^e of 2^e prijs. Hoeveel rijtjes moet U invullen om er zeker van te zijn dat ieder rijtje van 13 keer een 1, 2 of 3 in hoogstens één positie verschilt van een door U ingevuld rijtje?

(3.8.10) Beschouw de code van § 3.5. Decodeer met drempeldecodering het woord (1,1,1,0,0,0,0).

(3.8.11) Wat is de weight enumerator van de (8,4)-verlengde binaire Hamming code?

(3.8.12) Zij C een binaire code met weight enumerator A(z). Druk de weight enumerator van $\overline{C}$ uit in A(z).

(3.8.13) Zij C de $(2^k-1, 2^k-k-1)$-binaire Hamming code. Bepaal de weight enumerator van $\overline{C}^{\perp}$. Wat is het verband met § 2.2?

(3.8.14) Beschouw de code C van § 2.4. Bewijs dat uit de eigenschappen van S_5 volgt dat $\overline{C} = \overline{C}^{\perp}$ en dat daaruit volgt dat de gewichten van de woorden van $\overline{C}$ door 3 deelbaar zijn. Toon aan dat een lineaire combinatie van minder dan vier rijen van de generator een gewicht ≥ 5 heeft. Bepaal dan de weight enumerator van C.